LIFE

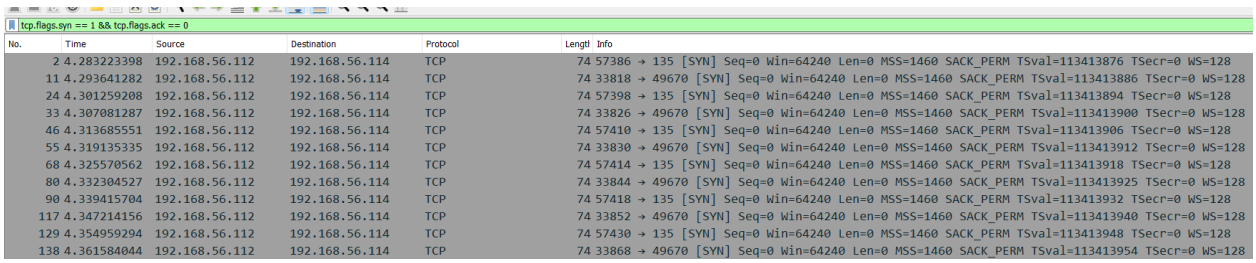
Надеемся, что это самый нормальный отчёт, за который не стыдно)

1) Определить тип уязвимости, которую пытались проэксплуатировать злоумышленники (CVE):

Так-как происходит атака, то атакующий должен физически как-то взаимодействовать с жертвой, а значит – отправлять какие-то данные, которые жертва будет обрабатывать.

Отфильтруем пакеты, в которых посылаются запросы с флагом синхронизации (Установки соединения).

tcp.flags.syn == 1 && tcp.flags.ack == 0



tcp.flags.syn == 1 && tcp.flags.ack == 0

No.	Time	Source	Destination	Protocol	Length	Info
2	4.283223398	192.168.56.112	192.168.56.114	TCP	74	57386 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413876 TSecr=0 WS=128
11	4.293641282	192.168.56.112	192.168.56.114	TCP	74	33818 → 49670 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413886 TSecr=0 WS=128
24	4.301259208	192.168.56.112	192.168.56.114	TCP	74	57398 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413894 TSecr=0 WS=128
33	4.307081287	192.168.56.112	192.168.56.114	TCP	74	33826 → 49670 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413900 TSecr=0 WS=128
46	4.313685551	192.168.56.112	192.168.56.114	TCP	74	57410 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413906 TSecr=0 WS=128
55	4.319135335	192.168.56.112	192.168.56.114	TCP	74	33830 → 49670 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413912 TSecr=0 WS=128
68	4.325570562	192.168.56.112	192.168.56.114	TCP	74	57414 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413918 TSecr=0 WS=128
80	4.332304527	192.168.56.112	192.168.56.114	TCP	74	33844 → 49670 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413925 TSecr=0 WS=128
90	4.339415704	192.168.56.112	192.168.56.114	TCP	74	57418 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413932 TSecr=0 WS=128
117	4.347214156	192.168.56.112	192.168.56.114	TCP	74	33852 → 49670 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413940 TSecr=0 WS=128
129	4.354959294	192.168.56.112	192.168.56.114	TCP	74	57430 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413948 TSecr=0 WS=128
138	4.361584044	192.168.56.112	192.168.56.114	TCP	74	33868 → 49670 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=113413954 TSecr=0 WS=128

Наблюдаем стабильные попытки подключения **192.168.56.112** к **192.168.56.114** к портам:

135, **445**, **49670** и **49666** (в самом конце трафика)

Исходя из вышесказанного:

Злоумышленник:

* IP - **192.168.56.112**

Жертва:

* IP - **192.168.56.114**

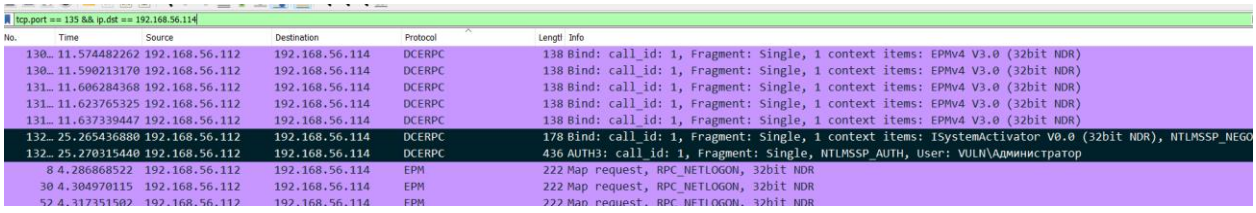
* Порты:

- **135** **DCE/RPC**
- **445** **SMB**
- **49670** **RPC_NETLOGON**
- **49666** **???**

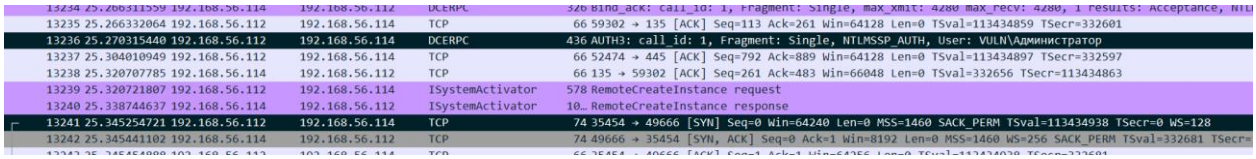
Подключение к **445** и **49666** порту произошло в конце трафика, а значит, **атаковать** вероятней всего могли по **135** и **49670** порту.

Начнём разбираться с событиями на **135** порту:

```
tcp.port == 135 && ip.dst == 192.168.56.114
```



Замечаем пакет аутентификации под учётной записью **Администратор** к нашей жертве. Посмотрим, что следовало дальше за этим пакетом:

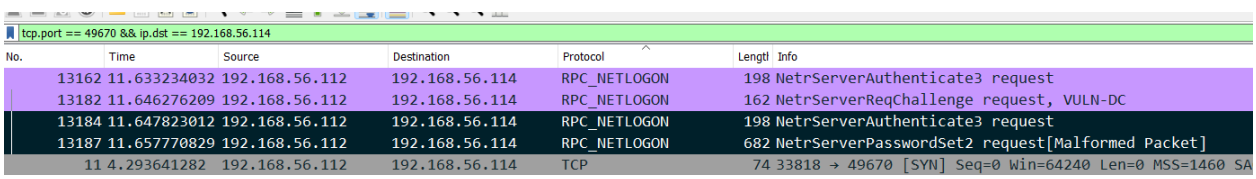


Заметим, что после него появился пакет по протоколу **ISystemActivator** и происходит первое и последнее подключение к порту **49666**.

ISystemActivator::RemoteCreateInstance отвечает за вызов и загрузку объектов, а также за авторизацию клиента. Возвращается идентификатор процесса и номер порта, на котором будет происходить взаимодействие. Отсюда следует, что у злоумышленника получилось зайти под учётной записью **Администратор** и всё дальнейшее взаимодействие с жертвой велось по **49666** порту.

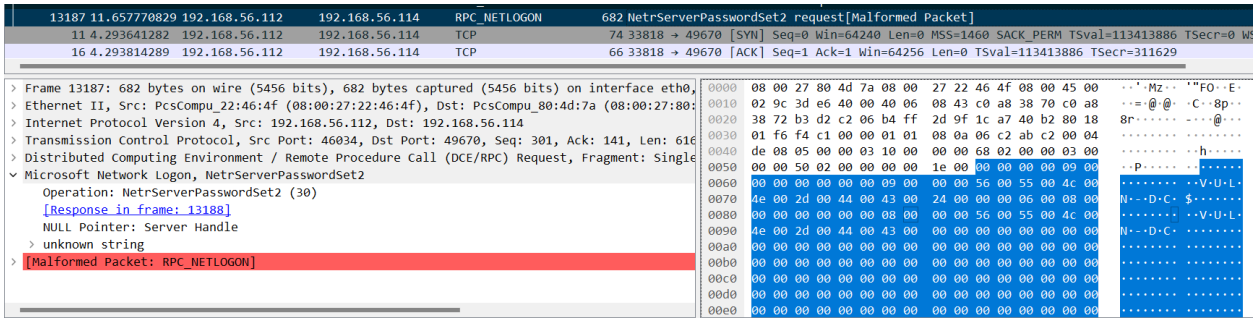
Атака прошла успешно, остаётся понять, за счёт чего злоумышленник получил доступ к жертве. Вернёмся к нашим портам и проанализируем передаваемые данные на **49670** порт.

```
tcp.port == 49670 && ip.dst == 192.168.56.114
```



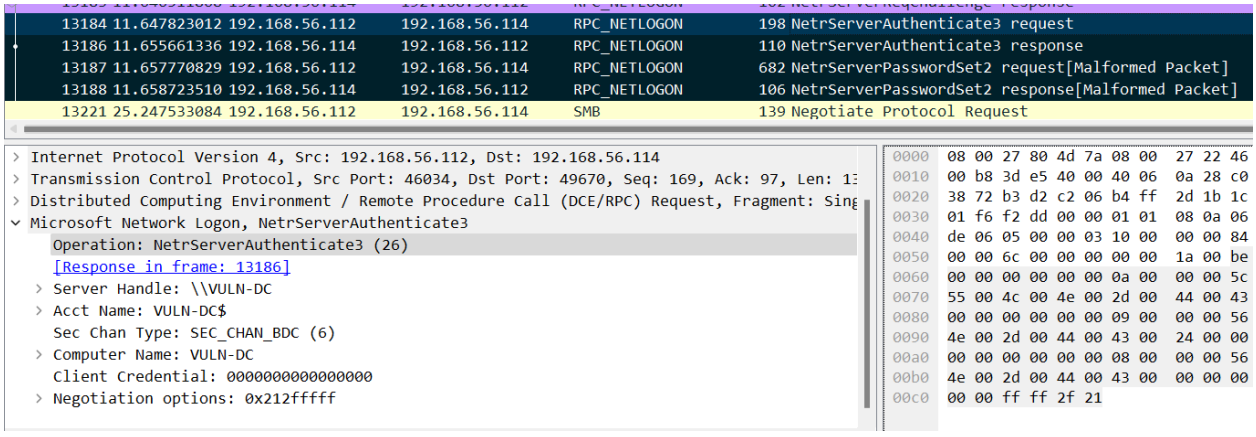
Замечаем помимо множества **попыток авторизации NeyrServerAuthenticate3** и провалов, пакет с методом **NetrServerPasswordSet2**. Данный метод отвечает за установку пароля в службе **Active Directory**.

Рассмотрим данный пакет подробней:

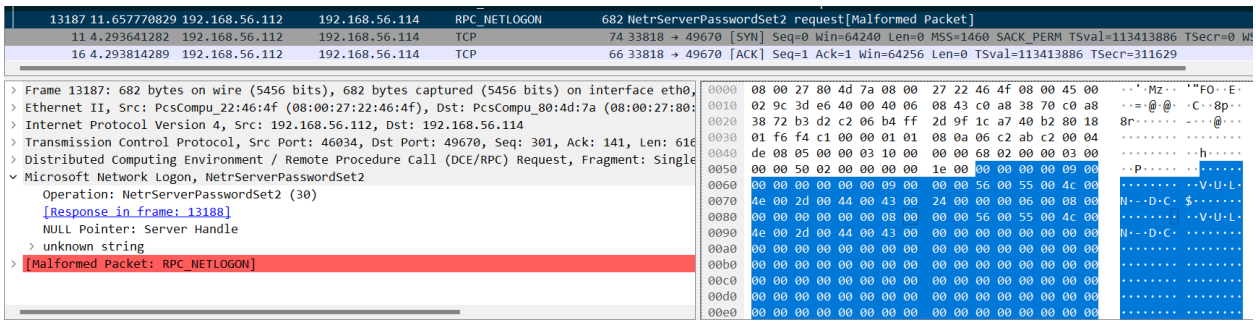


Видно предупреждение **Malformed Packet: RPC NETLOGON** которое может быть связано с уязвимостью **Zerologon**. Убедимся, она ли была использована:

Посмотрим пакет аутентификации:



Видим учётные данные клиента **0000000000000000**



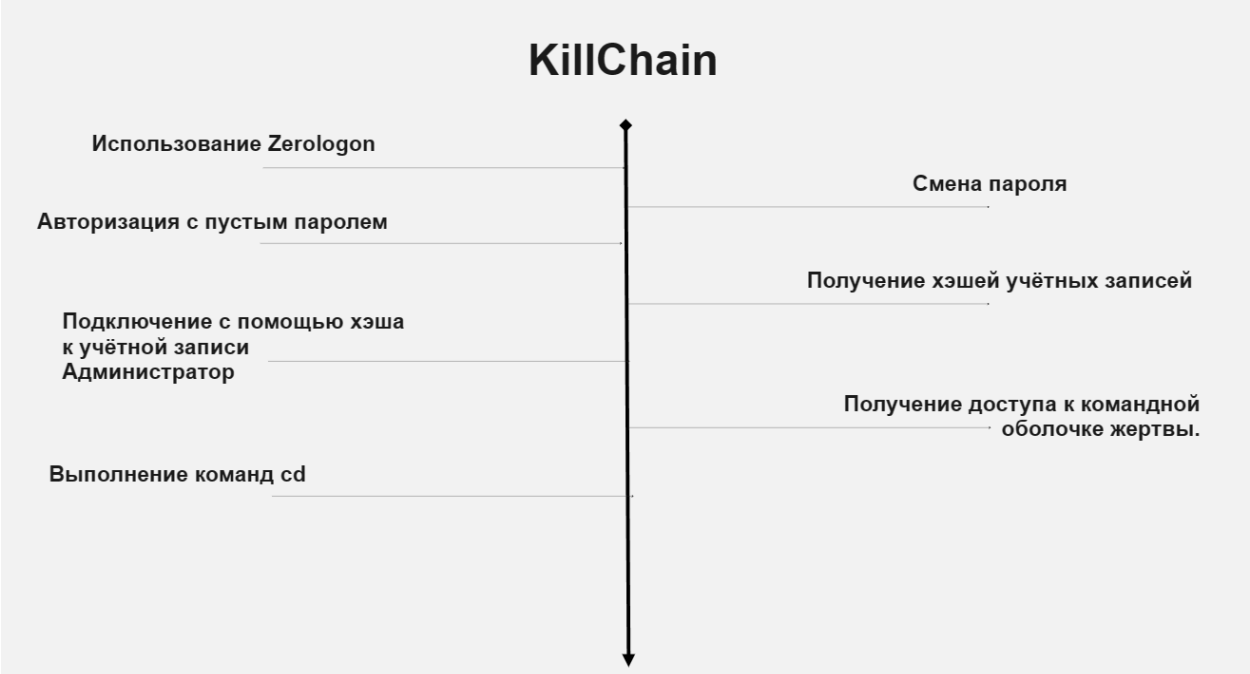
Видим, что методу **NetrServerPasswordSet2** передаются данные, в которых 516 байт – пустые, а остальные 8 – тоже.

Всё это характерно для уязвимости **Zerologon** (**CVE-2020-1472**). Она позволила получить доступ к учётной записи **Администратор**

Ответ: Zerologon (CVE-2020-1472)

Наглядный KillChain всей атаки:

(слева – направо и заново)



2) Определить предполагаемый хактул который пытались использовать злоумышленники:

Zerologon можно эксплуатировать с помощью **эксплоитов**, например (risksense/zerologon: Exploit for zerologon cve-2020-1472 (github.com)), либо с помощью готовых сборок решений, на подобии **impacket + mimikatz.** Также для подключения к жертве с помощью хэша могло использоваться **Evil-WinRM** (Hackplayers/evil-winrm: The ultimate WinRM shell for hacking/pentesting (github.com)) и подобные решения.

Ответ: **Exploit | Impacket и Evil-WinRM** (возможно другое, но подобное этому)

3) Описать контрмеры, позволяющие исключить возможность использования уязвимости в будущем

Microsoft пропатчили эту уязвимость - How to manage the changes in Netlogon secure channel connections associated with CVE-2020-1472 - Microsoft Support

И если верить описанию, то после февраля 2021 года протокол был переработан проблема с несовместимостью была устранена.

Создатели протокола рекомендуют:

- Обновить контроллеры домена
- Включить принудительный режим обращения к данной уязвимости
- Устранить несоответствующие устройства

Ответ: Обновить контроллер домена, устранить несоответствующие устройства и включить принудительный режим.