*Web Application Pen-Testing*

*AY 2022/2023*

*Week 4.1 Practical*

*OWASP Top 10 - 2021*

*A03:2021-Injection*

*Part 1*

*Authenticated Vulnerability Scan using ZAP*

*SQL Injection*

*Table of Contents*

[*1.* *Setup* 3](#_Toc118806889)

[*1.1.* *Start and Login to Kali Linux VM with NAT enabled* 3](#_Toc118806890)

[*1.2.* *Install sqlmap* 4](#_Toc118806891)

[*1.3.* *Start OWASP Broken Web Apps (owaspbwa) VM* 5](#_Toc118806892)

[*1.4.* *Set Kali Linux VM to Host-only enabled* 6](#_Toc118806893)

[*2.* *Authenticated Vulnerability Scan* 7](#_Toc118806894)

[*2.1.* *Let’s get Spidering* 7](#_Toc118806896)

[*2.2.* *Configuring Website Specific “Context” for ZAP* 9](#_Toc118806897)

[*2.3.* *Spidering After Configuring the Website Specific Context for ZAP* 13](#_Toc118806898)

[*2.4.* *Authenticated Vulnerability Scan After Configuring the Website Specific Context for ZAP* 14](#_Toc118806899)

[*3.* *Exploitation* 15](#_Toc118806900)

[*3.1.* *Manual Exploitation* 15](#_Toc118806902)

[*3.2.* *Automated Exploitation using sqlmap* 17](#_Toc118806903)

[*4.* *Homework for Week 4.1* 21](#_Toc118806904)

# *Setup*

## *Start and Login to Kali Linux VM with NAT enabled*

*This practical requires Kali Linux VM to be connected to the Internet. Therefore, make sure the Virtual Machine Settings 🡪 Network Adapter 🡪 NAT*

|  |  |
| --- | --- |
| *Graphical user interface, application  Description automatically generated* | ***Login*** *into this Kali Linux VM*  *Graphical user interface, application  Description automatically generated* |

|  |  |
| --- | --- |
| *Tools with solid fill* | *In case your Kali Linux is* ***not responding*** *to changing to NAT (i.e., still not connected to the Internet). You can restart Kali Linux’s Ethernet Interface (eth0) by typing the following* ***2 commands one after the other*** *into the Kali Linux’s Terminal Emulator and press Enter:* |

*sudo ifdown eth0*

*Text

Description automatically generated*

*sudo ifup eth0*

*Text

Description automatically generated*

## *Install sqlmap*

*sqlmap goal is to detect and take advantage of SQL injection vulnerabilities in web applications. Once it detects one or more SQL injections on the target host, the user can choose among a variety of options to perform an extensive back-end database management system fingerprint, retrieve DBMS session user and database, enumerate users, password hashes, privileges, databases, dump entire or user’s specific DBMS tables/columns, run his own SQL statement, read specific files on the file system and more. [Source: https://www.kali.org/tools/sqlmap/]*

*Type the following command into the Kali Linux’s Terminal Emulator and press Enter:*

*sudo apt install sqlmap*

*Text

Description automatically generated*

## *Start OWASP Broken Web Apps (owaspbwa) VM*

*Make sure the Virtual Machine Settings 🡪 Network Adapter 🡪 Host-only*

*Graphical user interface, text

Description automatically generated*

1. *Type in the owaspbwa VM IP Address below:* *OWASPBWA\_IP*
2. *XXX.XXX.XX.XXX*
3. ***NO NEED*** *to login into this VM, just starting this VM is sufficient, as shown below.*

*Text

Description automatically generated*

## *Set Kali Linux VM to Host-only enabled*

*Make sure the Virtual Machine Settings 🡪 Network Adapter 🡪 Host-only*

|  |  |
| --- | --- |
| *Graphical user interface, text  Description automatically generated* | ***Login*** *into this Kali Linux VM*  *Graphical user interface, application  Description automatically generated* |

# *Authenticated Vulnerability Scan*

*OWASP ZAP can also be used to scan a wide range of vulnerabilities in web applications. Thus far, we have been using ZAP’s “Automated Scan” just for “Spidering” purpose. We used to stop the “Attack” once the spidering is done. In this exercise we would let the “Attack” continue into “Active Scan”, so that ZAP can help us scan for vulnerabilities in the target website.*

*However, there could be many situations where a website has a login feature that stops ZAP (or any other vulnerability scanner) to proceed further. This is because ZAP does not have the correct login credentials to enter the website. Therefore, we need to prepare ZAP for Authenticated Vulnerability Scan, whereby ZAP can submit to the website the correct login credentials whenever it is prompted to do so. This would enable ZAP to continue with its Spidering and Active Scanning without any hindrance. During a real-world grey-box web app pen-testing, it is common for the client or the webapp owner to provide such login credentials to the pen-tester.*



## *Let’s get Spidering*

*In this exercise we will focus on the “Damn Vulnerable Web Application” (DVWA) available inside the owaspbwa VM. Click on the link “Damn Vulnerable Web Application”.*

*Text

Description automatically generated*

*You will be presented with a “Login” page. You can notice that there is nothing you can do here without entering the correct Login credentials. Let us try Spidering this website using ZAP and see where ZAP gets stuck. Copy the URL: “https://OWASPBWAP\_IP/dvwa/” from the address bar.*

*Graphical user interface, website

Description automatically generated*

#### Start OWASP ZAP inside Kali Linux

*Graphical user interface, text, application

Description automatically generated*

#### Click “Automated Scan”

*Graphical user interface, application, Word

Description automatically generated*

*Paste the copied URL: http://OWASPBWAP\_IP/dvwa/ in the “URL to attack” text field. Check “Use traditional spider” and click “Attack”. Note: Error Pop Up may appear, click “OK” and click “Attack” again.*

*Graphical user interface, text, application, Word

Description automatically generated*

*In the below pane click on “Spider” tab and once the progress bar reaches 100%, click on “Stop”. Remember for now we are just doing spidering. On the left pane expand “Sites”, “http:// OWASPBWAP\_IP”, and “dvwa” folder. You can notice that ZAP was not able to spider the entire website because it did not have the correct Login credentials to pass through the Login page.*

*Graphical user interface, text, application

Description automatically generated*

## *Configuring Website Specific “Context” for ZAP*

*In the output above, among many pages that ZAP spidered and displayed to us, we can notice 1 interesting detail:*

*POST:login.php()(Login,password,username)*

*This is the page that is not allowing our ZAP to fully Spider the website, since ZAP does not have the correct Login and Password. To overcome such roadblocks, we need to prepare ZAP with the correct Login and Password. This preparation is known as configuring website specific “Context” for ZAP to do its job as intended without any hindrances. Follow the steps below:*

*In order to create New context, right click on the Login Authentication URL and click on Include In Context -> New Context.*

*Graphical user interface, text, application

Description automatically generated*

*As you create New Context, you will see another screen pop up from where you can also change the Context Name. We can change the name to “DVWA\_Login”. You can change the name according to your choice.*

*Graphical user interface, text, application

Description automatically generated*

*Click on “Include in Context” 🡪 Select the provided Regex and click “Modify”. Modify the existing Regex to http://OWASPBWAP\_IP/dvwa/\*.\* this indicates to ZAP that all pages under “dvwa” are in the context. Click “Modify”.*

*Graphical user interface, text, application

Description automatically generated*

*Now, click on Authentication sub menu and from the Authentication drop down select Form- Based Authentication .*

*Graphical user interface, text, application, email

Description automatically generated*

*Then, select Login Form Target URL by clicking on “Select” Button.*

*Graphical user interface, text, application

Description automatically generated*

*Graphical user interface, text

Description automatically generated*

*This will automatically discover the parameters we need. NOTE: from the Password Parameter drop down, select “password”.*

*Graphical user interface, text, application

Description automatically generated*

*Once you are done you need to define the username and password parameter. For that, click on “Users” sub-menu. Then, click on “Add” Button, to add new User. In our case, we will enter User Name as “dvwa\_User”. Our Username is “user” and Password is “user”, which we use for our DVWA Login. Assume that these credentials were given to you by your client before you start the pen-testing. Click “Add”.*

*Graphical user interface, text, application

Description automatically generated*

*Select the just added Users entry, and click on “Enable All”, and then click “OK”.*

*Graphical user interface, application

Description automatically generated*

*We need to set logged out response to let ZAP know when the application is Logged in and when it is Logged out. For that, in the spidering results - ZAP’s left pane, select GET:login.php, and on the right pane click “Response” tab. Slightly scroll down the HTML code to find the “<title>” of the Response page. This is the page that is displayed for every failed login attempt, informing the user to type in the correct username and password. Highlight the <title>Damn Vulnerable Web App (DVWA) - Login</title> and right click, select “Flag as Context” and select “DVWA\_Login: Authentication Logged-out indicator”.*

*Graphical user interface, text

Description automatically generated*

*You will notice that the “Regex pattern used to identify Logged Out messages:” is auto populated with the \Q<title>Damn Vulnerable Web App (DVWA) - Login</title>\E. Click “OK”*

*Graphical user interface, text, application

Description automatically generated*

*Finally! We are almost ready for Authentication Scan using ZAP. But, before that we need to make sure Forced User Mode is enabled. For that, click on the icon shown in the figure below.*

*Graphical user interface, text, application

Description automatically generated*

## *Spidering After Configuring the Website Specific Context for ZAP*

*Having configured the DVWA specific “Context” for ZAP, let us spider DVWA website and see how ZAP is now able to spider the entire website without any hindrance from the DVWA’s Login page. In the previous spidering results - ZAP’s left pane, right click on the folder “dvwa” and select “Attack” and “Spider” as shown in the steps indicated below.*

*Graphical user interface, application

Description automatically generated*

*Under the “User:” click on the dropdown menu and select “dvwa\_user” that we previously configured in the context.*

*Graphical user interface, text, application

Description automatically generated*

*Click “Start Scan”*

*Graphical user interface, text, application, email

Description automatically generated*

*Once the Spidering is done you can see that ZAP has spidered lot more than what it had done right at the beginning.*

*Graphical user interface, application

Description automatically generated*

## *Authenticated Vulnerability Scan After Configuring the Website Specific Context for ZAP*

*Now let us do an authenticated vulnerability scan on a particular spidered page that we think might be susceptible to SQL Injection. Under the folder “sqli”, right click on the GET:/(Submit,id) 🡪 Attack 🡪 Active Scan… Follow the steps below:*

*Graphical user interface, text, application, email

Description automatically generated*

*Graphical user interface, text, application, email

Description automatically generated*

*Let the Active Scan run for 5 mins to completion.*

*Graphical user interface, text

Description automatically generated*

*Click on the “Alerts” tab and see that ZAP has successfully identified SQL Injection vulnerability among many others that are mostly applicable to the GET:/(Submit,id) page that we selected above. You can do the same “Active Scan” in this manner on other spidered pages as well individually. As a pen-tester it is recommended to run these scans on targeted individual pages rather than the entire website, as it would take a long time and causes a huge burden on the webserver.*

*Graphical user interface, text, application, email

Description automatically generated*

# *Exploitation*



## *Manual Exploitation*

*Let us manually exploit the vulnerability discovered above, using the Authenticated Vulnerability Scan. Login to DVWA using the username: user and Password: user*

*Graphical user interface, text, application, website

Description automatically generated*

*Make sure that the DVWA Security is “low”, otherwise it would be difficult to perform exploitations for beginner pen-testers.*

*A screenshot of a computer

Description automatically generated*

*Let us now visit the page “sqli” that ZAP detected as vulnerable to SQL Injection Attack and type in the malicious SQL command recommended by ZAP: ZAP' OR 1=1 –- [Note: don’t forget the space after --]. Click “Submit”*

*Graphical user interface, text, website

Description automatically generated*

*You can notice that we are successful in dumping all the users’ First Name and Surname.*

*Graphical user interface, website

Description automatically generated*

## *Automated Exploitation using sqlmap*

Help on sqlmap: t*ype the following command into the Kali Linux’s Terminal Emulator and press Enter:*

Sqlmap -h

or for more detailed help :

sqlmap -hh

*Now visit the page “sqli” that ZAP detected as vulnerable to SQL Injection Attack and type in 1 and submit, so that your ZAP can record this transaction in its History.*

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

The above request should have been recorded by your ZAP under its “History” tab, as shown below. Copy the URL highlighted below and also the Cookie details we will use them to construct our sqlmap command.

Graphical user interface, text, application

Description automatically generated

sqlmap command to list the available databases is as below. Replace the URL and the Cookie details based on your own ZAP’s History record.

sudo sqlmap -u "http://192.168.37.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=frk0080d9gk1f5or3vdopo57q4" --random-agent --keep-alive --timeout 100 --dbs

Text

Description automatically generated with medium confidence

--dbs Enumerate DBMS databases

--cookie=COOKIE HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")

--keep-alive Use persistent HTTP(s) connections

--timeout=TIMEOUT Seconds to wait before timeout connection (default 30)

--random-agent Use randomly selected HTTP User-Agent header value

The output will be as shown below where we can see 2 databases:

Text

Description automatically generated

sqlmap command to list the tables inside a particular database: dvwa in this case is as below. Replace the URL and the Cookie details based on your own ZAP’s History record.

sudo sqlmap -u "http://192.168.37.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=frk0080d9gk1f5or3vdopo57q4" --random-agent --keep-alive --timeout 100 -D dvwa --tables

Text

Description automatically generated

--tables Enumerate DBMS database tables

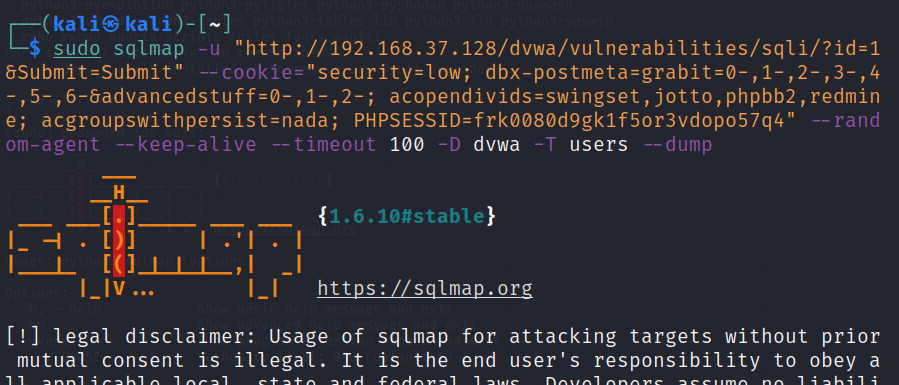
The output will be as shown below where we can see 2 tables under the dvwa database:

Text

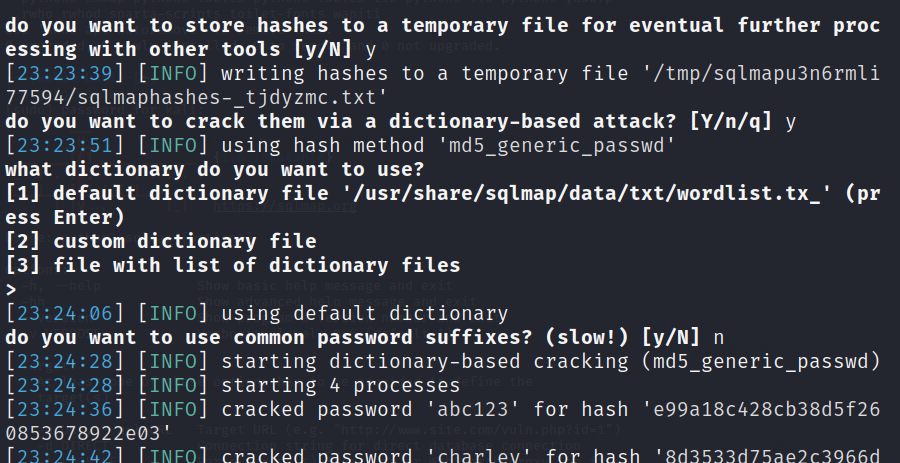
Description automatically generated

sqlmap command to dump the contents of a table: users in this case is as below. Replace the URL and the Cookie details based on your own ZAP’s History record.

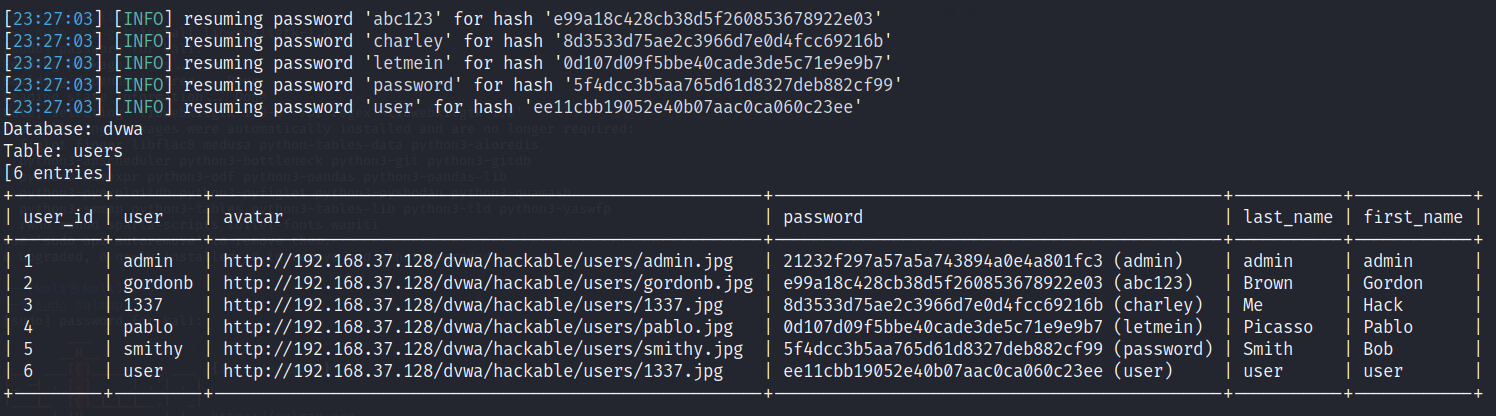
sudo sqlmap -u "http://192.168.37.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; PHPSESSID=frk0080d9gk1f5or3vdopo57q4" --random-agent --keep-alive --timeout 100 -D dvwa -T users –dump



sqlmap will not only dump the table contents but will also offer to crack hashed passwords. You can choose the following options when asked by sqlmap.



The final dumped output will be as below:



# *Homework for Week 4.1*

Please provide 5 ~ 10 screenshots or you may also record a video upload to YouTube and provide the links inside a document as evidence of completion for each of the tasks listed below. You may use this document to attach your screenshots and submit it to POLITEMall by Sunday 20th Nov 23:59.

URL: <https://portswigger.net/web-security/all-labs>

Attempt any 8 labs out of 16 labs under the category of **SQL injection**

Graphical user interface

Description automatically generated