**DOCUMENTATION FOR OWASP ZAP BASELINE SCAN**

***Dependencies:-***

**1. Docker Install**

* <https://docs.docker.com/docker-for-windows/install/>
* Refer to above page for download link and installation instructions/issues.

**2. Java with JDBC driver**

* <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
* <https://www.microsoft.com/en-in/download/details.aspx?id=11774>
* Above 2 links for Java and JDBC driver respectively.

**3. SQL Server**

* <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

**4. OWASP ZAP (Docker Container)**

Within Docker Terminal, the following commands must be executed.

***Pull docker container:-***

For the stable release:

docker pull owasp/zap2docker-stable

For the latest weekly release:

docker pull owasp/zap2docker-weekly

***Run OWASP ZAP on docker:-***

*Without File Parameters:-*

docker run -t owasp/zap2docker-weekly zap-baseline.py -t https://www.example.com

*With File Parameters:-*

docker run -v $(pwd):/zap/wrk/:rw -t owasp/zap2docker-weekly zap-baseline.py \

-t https://www.example.com -g gen.conf -r testreport.html

***Configuration File:-***

You can configure how the baseline scan runs with a configuration file. A default configuration file can be created using the '-g' parameter.

# zap-baseline rule configuration file

# Change WARN to IGNORE to ignore rule or FAIL to fail if rule matches

# Only the rule identifiers are used - the names are just for info

# You can add your own messages to each rule by appending them after a tab on each line.

10010 WARN (Cookie No HttpOnly Flag)

10011 WARN (Cookie Without Secure Flag)

10012 WARN (Password Autocomplete in Browser)

10015 WARN (Incomplete or No Cache-control and Pragma HTTP Header Set)

10016 WARN (Web Browser XSS Protection Not Enabled)

10017 WARN (Cross-Domain JavaScript Source File Inclusion)

10019 WARN (Content-Type Header Missing)

10020 WARN (X-Frame-Options Header Scanner)

10021 WARN (X-Content-Type-Options Header Missing)

10023 WARN (Information Disclosure - Debug Error Messages)

10024 WARN (Information Disclosure - Sensitive Informations in URL)

10025 WARN (Information Disclosure - Sensitive Information in HTTP Referrer Header)

10026 WARN (HTTP Parameter Override)

10027 WARN (Information Disclosure - Suspicious Comments)

10032 WARN (Viewstate Scanner)

10040 WARN (Secure Pages Include Mixed Content)

10105 WARN (Weak Authentication Method)

10202 WARN (Absence of Anti-CSRF Tokens)

2 WARN (Private IP Disclosure)

3 WARN (Session ID in URL Rewrite)

50001 WARN (Script Passive Scan Rules)

90001 WARN (Insecure JSF ViewState)

90011 WARN (Charset Mismatch)

90022 WARN (Application Error Disclosure)

90030 WARN (WSDL File Passive Scanner)

90033 WARN (Loosely Scoped Cookie)

***Usage:-***

Usage: zap-baseline.py -t <target> [options]

-t target target URL including the protocol, eg https://www.example.com

Options:

-c config\_file config file to use to INFO, IGNORE or FAIL warnings

-u config\_url URL of config file to use to INFO, IGNORE or FAIL warnings

-g gen\_file generate default config file (all rules set to WARN)

-m mins the number of minutes to spider for (default 1)

-r report\_html file to write the full ZAP HTML report

-w report\_md file to write the full ZAP Wiki (Markdown) report

-x report\_xml file to write the full ZAP XML report

-a include the alpha passive scan rules as well

-d show debug messages

-P specify listen port

-D delay in seconds to wait for passive scanning

-i default rules not in the config file to INFO

-j use the Ajax spider in addition to the traditional one

-l level minimum level to show: PASS, IGNORE, INFO, WARN or FAIL, use with -s to hide example URLs

-n context\_file context file which will be loaded prior to spidering the target

-p progress\_file progress file which specifies issues that are being addressed

-s short output format - dont show PASSes or example URLs

-z zap\_options ZAP command line options e.g. -z "-config aaa=bbb -config ccc=ddd"

For more details see https://github.com/zaproxy/zaproxy/wiki/ZAP-Baseline-Scan

### *Example Output:-*

docker run -t owasp/zap2docker-stable zap-baseline.py -t https://www.example.com

May 17, 2016 1:24:32 PM java.util.prefs.FileSystemPreferences$1 run

INFO: Created user preferences directory.

Total of 3 URLs

PASS: Cookie No HttpOnly Flag [10010]

PASS: Cookie Without Secure Flag [10011]

PASS: Password Autocomplete in Browser [10012]

PASS: Cross-Domain JavaScript Source File Inclusion [10017]

PASS: Content-Type Header Missing [10019]

PASS: Information Disclosure - Debug Error Messages [10023]

PASS: Information Disclosure - Sensitive Informations in URL [10024]

PASS: Information Disclosure - Sensitive Information in HTTP Referrer Header [10025]

PASS: HTTP Parameter Override [10026]

PASS: Information Disclosure - Suspicious Comments [10027]

PASS: Viewstate Scanner [10032]

PASS: Secure Pages Include Mixed Content [10040]

PASS: Weak Authentication Method [10105]

PASS: Private IP Disclosure [2]

PASS: Session ID in URL Rewrite [3]

PASS: Absence of Anti-CSRF Tokens [40014]

PASS: Script passive scan rules [50001]

PASS: Insecure JSF ViewState [90001]

PASS: Charset Mismatch [90011]

PASS: Application Error Disclosure [90022]

PASS: WSDL File Passive Scanner [90030]

PASS: Loosely Scoped Cookie [90033]

WARN: Incomplete or No Cache-control and Pragma HTTP Header Set [10015] x 3

https://www.example.com

https://www.example.com/robots.txt

https://www.example.com/sitemap.xml

WARN: Web Browser XSS Protection Not Enabled [10016] x 3

https://www.example.com

https://www.example.com/robots.txt

https://www.example.com/sitemap.xml

WARN: X-Frame-Options Header Not Set [10020] x 3

https://www.example.com

https://www.example.com/robots.txt

https://www.example.com/sitemap.xml

WARN: X-Content-Type-Options Header Missing [10021] x 3

https://www.example.com

https://www.example.com/robots.txt

https://www.example.com/sitemap.xml

FAIL: 0 WARN: 4 IGNORE: 0 PASS: 22

Refer to following links for complete documentation:-

1. <https://github.com/zaproxy/zaproxy/wiki/Docker>
2. <https://github.com/zaproxy/zaproxy/wiki/ZAP-Baseline-Scan>

**5. XML report parser**

Run the following command to generate XML report of baseline scan:

docker run -v $(pwd):/zap/wrk/:rw -t owasp/zap2docker-stable zap-baseline.py -t https://www.example.org -x XML\_report.xml –c config.conf

# -c to use specified configuration (can be excluded)

# -x to generate xml report with specified name in working directory

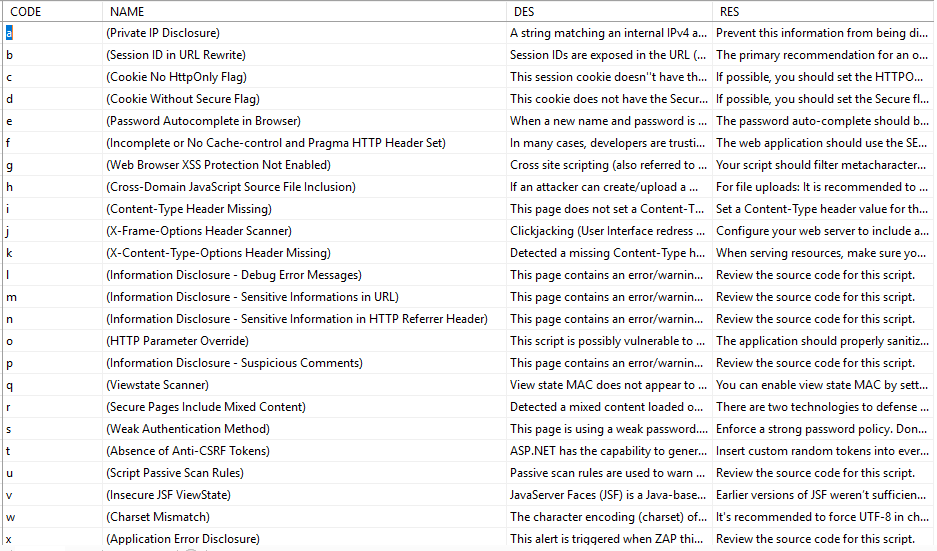
Scripts to populate SQL Database after parsing XML report have been written in both Python and Java and are in the same folder as this document. They each have separate dependencies.

Note – Scripts must be modified in order to provide appropriate ID no, Server Name, Username, Password, Port Number, Database name etc.

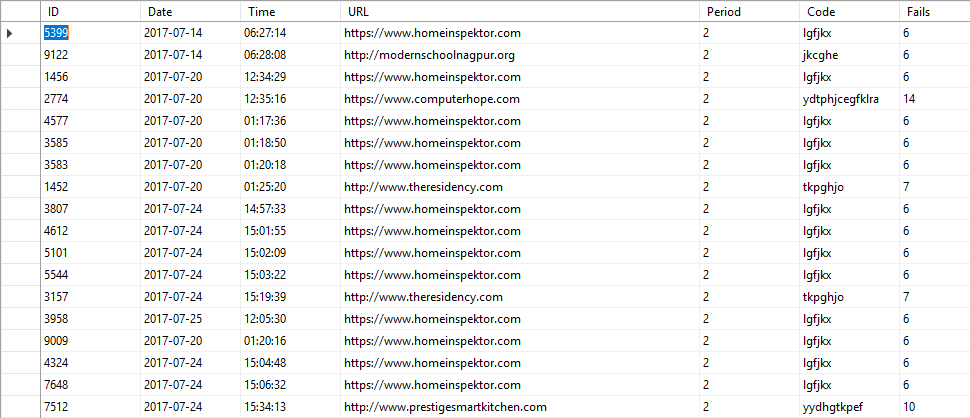
**6. SQL Server Database**

Microsoft SQL Server is DBMS used. The tables required to be populated are shown as follows:-

*Config Table:-*



*Result Table:-*



* The ID field is used to uniquely identify users.

(In these tables they are simply random 4 digit numbers)

* Period is the amount of time used to scan the website.
* The Code field in both tables are related to each other.

The character string code in the Result table is used to indicate the vulnerabilities detected during the scan.

* In the Config table, each entry is uniquely identified by a letter.
* If the Code entry in the Config table is a substring of the entry in the Result table, the vulnerability corresponding to the letter (in the Config table) was detected during the scan.
* DES and RES columns in the Config table indicate Description and Resolution for the vulnerability.
* Fails Column in the Result table indicated number of vulnerabilities detected (out of 25 total cases). It is essentially the string length of Code column in Result table.

**REFERENCE LINKS**

1. <https://github.com/zaproxy/zaproxy/wiki/Docker>
2. <https://github.com/zaproxy/zaproxy/wiki/ZAP-Baseline-Scan>
3. <https://www.youtube.com/watch?v=pYFtLA2yTR8> - Basics
4. <https://www.youtube.com/watch?v=3vVnMh6AUkk> - Automation

## Please modify the paths of locations referenced by the Scan\_Script.bat to appropriately to reference the correct files in your respective system.

##Also enter appropriate SQL database details in Java or Python program (i.e. Server name, Username etc.) if you are using those programs.

##The Java and Python programs each have their own dependencies; they both take command line argument which indicates the path of XML file.