Driving OWASP ZAP with Selenium

OWASP

The Open Web Application Security Project

About Me

OWASP
The Open Web Application Security Project

- Mark Torrens
  - Recently moved into Cyber Security
  - Based in London
  - Completing MSc Cyber Security @ University of York
  - Security Architect for Kainos

- Mateusz Kalinowski
  - Java research

kainos®

OWASP
The Open Web Application Security Project

- OWASP Zed Attack Proxy (ZAP)

“The OWASP Zed Attack Proxy (ZAP) is one of the world’s most popular free security tools and is actively maintained by hundreds of international volunteers. It can help you **automatically find security vulnerabilities** in your web applications while you are developing and testing your applications. Its also a great tool for experienced pentesters to use for **manual security testing**.”

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

OWASP
The Open Web Application Security Project

- Selenium 

"***Selenium automates browsers***. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) be automated as well."

https://www.seleniumhq.org/

OWASP
The Open Web Application Security Project

- Objective

  To use OWASP ZAP, to detect web application vulnerabilities in a CI/CD pipeline

- Problem

  Web applications have **Basic Authentication**, **User Logins** and **Form Validation** which stops ZAP in its tracks

OWASP
The Open Web Application Security Project

- Solution

Use Selenium scripts to drive ZAP

A project may already have Selenium scripts

ZAP does have Zest scripts but Selenium is more widely known and may already be being maintained on a project

OWASP
The Open Web Application Security Project

- ZAP's Passive and Active Scans

Passive scans record the requests and responses sent to a web app and creates alerts for detected vulnerabilities

Active scans actively modify the recorded requests and responses to determine further vulnerabilities

OWASP
The Open Web Application Security Project

- Pipeline Steps

1. Start ZAP
2. Run Selenium Scripts (Passive Scan)
3. Wait for Passive scan to complete
4. Start Active Scan
5. Wait for Active scan to complete
6. Retrieve alerts and report

OWASP
The Open Web Application Security Project

- Start ZAP

```
zap.sh \
-daemon \
-host some-host \
-port some-port \
-config api.addrs.addr.regex=true
-config api.disablekey=true

zap.sh - A start up script provided by ZAP
-daemon - Start in a headless configuration
-host - The ZAP host
-port - The ZAP port
-config api.addrs.addr.regex=true - Allow any source IP to connect
-config api.disablekey=true - Execute ZAP API endpoints without the need
for an API key
```

A Docker image called **owasp/zap2docker-bare** exists which can be used to start ZAP

OWASP
The Open Web Application Security Project

- Selenium Driver Settings

```
// Set Chrome Options
ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.addArguments("--ignore-certificate-errors");

// Set proxy
String proxyAddress = "ZAP-HOST:8888";
Proxy proxy = new Proxy();
proxy.setHttpProxy(proxyAddress)
    .setSslProxy(proxyAddress);

// Set Desired Capabilities
DesiredCapabilities capabilities = DesiredCapabilities.chrome();
capabilities.setCapability(CapabilityType.PROXY, proxy);
capabilities.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
capabilities.setCapability(CapabilityType.ACCEPT_INSECURE_CERTS,
          true);
capabilities.setCapability(ChromeOptions.CAPABILITY, chromeOptions);
```

OWASP
The Open Web Application Security Project

- Security Response Headers

If the target web application has security response headers in place, specifically **Strict-Transport-Security** the web driver should be configured as follows

```
chromeOptions.addArguments("--ignore-certificate-errors");

capabilities.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
capabilities.setCapability(CapabilityType.ACCEPT_INSECURE_CERTS, true);
```

OWASP
The Open Web Application Security Project

- Passive Scan

A passive scan is run when Selenium drives the Web Driver through the ZAP proxy

The passive scan creates the scan tree and allows ZAP to be knowledgeable enough about the web application to perform the active scan

OWASP
The Open Web Application Security Project

- Wait for Passive Scan

There will always be a short delay for ZAP to complete the passive scan, before alerts and reports are available

The status of a passive active scan is determined by running endpoint **JSON/pscan/view/recordsToScan**

The passive scan is complete when 0 is returned

OWASP

The Open Web Application Security Project

```
wait_for_passive_scan_to_complete() {

    STATUS_URL="http://$1:$2/"
    STATUS_URL+="JSON/pscan/view/recordsToScan/?"
    STATUS_URL+="zapapiformat=JSON&"
    STATUS_URL+="formMethod=GET&"

    SCAN_STATUS=0
    until [ $SCAN_STATUS -eq 0 ]; do
        sleep 10

        # Get Scan status
        SCAN_STATUS_RES=$(curl -s $STATUS_URL)

        # Parse scan status
        SCAN_STATUS=$(echo $SCAN_STATUS_RES | jq -r '.recordsToScan')

        # Display status
        echo Scan $SCAN_STATUS% complete

        done

    echo Passive Scan Complete
}

wait_for_passive_scan_to_complete $ZAP_HOST $ZAP_PORT
```

OWASP
The Open Web Application Security Project

- Start Active Scan

An active scan is started by running endpoint **JSON/ascan/action/scan**

If ZAP is reachable, this endpoint returns a new Scan ID

OWASP
The Open Web Application Security Project

```
start_active_scan() {

    SCAN_URL="http://$1:$2/"
    SCAN_URL+="JSON/ascan/action/scan/?"
    SCAN_URL+="zapapiformat=JSON&"
    SCAN_URL+="formMethod=GET&"
    SCAN_URL+="url=https://$3&"

    # Start Active ZAP Scan
    SCAN_ID_RES=$(curl -s $SCAN_URL)

    # Parse for scan ID
    SCAN_ID=$(echo $SCAN_ID_RES | jq -r '.scan')

    # Display scan ID
    echo Scan ID: $SCAN_ID
}

ZAP_HOST="localhost"
ZAP_PORT="8080"
TARGET="my-app.azurewebsites.net"

start_active_scan $ZAP_HOST $ZAP_PORT $TARGET
```

OWASP
The Open Web Application Security Project

- Wait for Active Scan

The status of an active scan is determined by running endpoint **JSON/ascan/view/status**

If the scan exists, a value between 0 and 100 is returned, representing the percentage of the scan which has completed

OWASP
The Open Web Application Security Project

```
wait_for_active_scan_to_complete() {

    STATUS_URL="http://$1:$2/"
    STATUS_URL+="JSON/ascan/view/status/?"
    STATUS_URL+="zapapiformat=JSON&"
    STATUS_URL+="apikey=&"
    STATUS_URL+="formMethod=GET&"
    STATUS_URL+="scanId=$SCAN_ID"

    SCAN_STATUS=0
    until [ $SCAN_STATUS -eq 100 ]; do
        sleep 10

        # Get Scan status
        SCAN_STATUS_RES=$(curl -s $STATUS_URL)

        # Parse scan status
        SCAN_STATUS=$(echo $SCAN_STATUS_RES | jq -r '.status')

        # Display status
        echo Scan $SCAN_STATUS% complete

        done

    echo Active Scan Complete
}

wait_for_active_scan_to_complete $ZAP_HOST $ZAP_PORT
```

OWASP
The Open Web Application Security Project

- Get Scan Results

Once the active scan is complete, the alerts in the form of a JSON file and an HTML report can be retrieved from ZAP

Alerts: **JSON/core/view/alerts**

Report: **OTHER/core/other/htmlreport**

OWASP
The Open Web Application Security Project

- Get Alerts

```
get_alerts() {

    ALERTS_URL="http://$1:$2/"
    ALERTS_URL+="JSON/core/view/alerts/?"
    ALERTS_URL+="zapapiformat=JSON&"
    ALERTS_URL+="formMethod=GET&"
    ALERTS_URL+="baseurl=https://$3&"

    curl -s $ALERTS_URL > alerts.json
}

get_alerts $ZAP_HOST $ZAP_PORT $TARGET
```

- Get Report

```
get_report() {

    REPORT_URL="http://$1:$2/"
    REPORT_URL+="OTHER/core/other/htmlreport/?"
    REPORT_URL+="formMethod=GET"

    curl -s $REPORT_URL > report.html
}

get_report $ZAP_HOST $ZAP_PORT
```

OWASP
The Open Web Application Security Project

- Bonus

If you are targeting a web application with **Strict-Transport-Security** and you are using a browser, you will need to add ZAP's Dynamic SSL Certificate to your browser.

To retrieve the ZAP's SSL certificate do either:

1. ZAP -> Preferences -> Options -> Dynamic SSL Certificate

2. HTTP GET **ZAP_HOST:ZAP_PORT://OTHER/core/other/rootcert**

To import the ZAP SSL Certificate into Firefox:

Settings -> Privacy & Security -> View Certificates -> Authorities -> Import

OWASP
The Open Web Application Security Project

Thank You