UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

PROJECT REPORT

AUTOMATED SOFTWARE VULNERABILITY TESTING

GROUP MEMBERS

MUHAMAMD HAMZA KHAN

FAHAD AHMED

AREEB ULLAH KHAN

PROFESSOR

FABIO PALOMBA

COORDINATOR

EMANUELE IANNONE

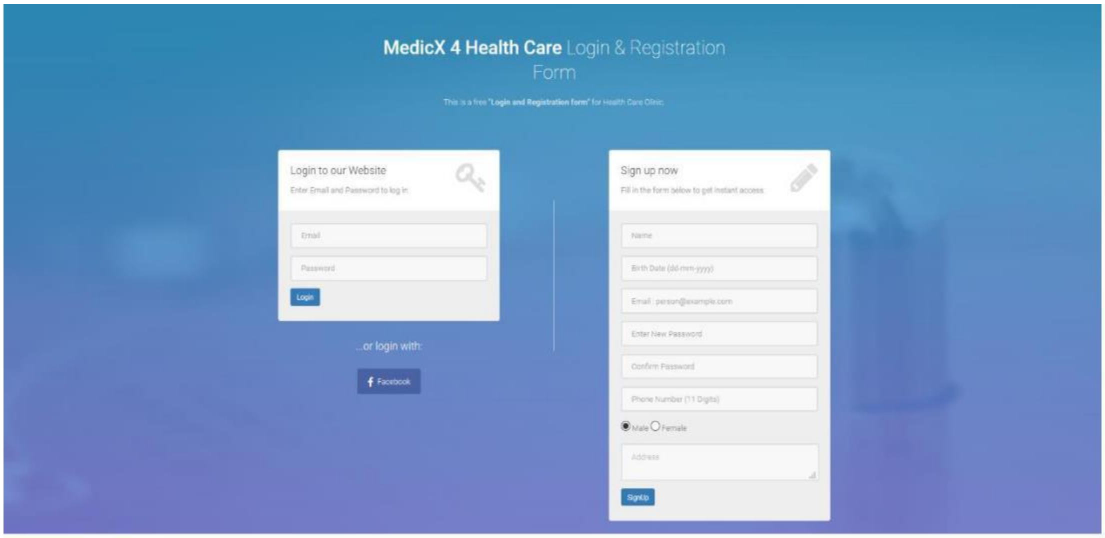
GIAMMARIA GIORDANO

ABSTRACT

Software Testing is the main part segment of software engineering life cycle. Each product in world must be tested to measure the quality, reliability, maintainability before it is delivered to end users. The primary life cycle covers the main object that includes acquisition, supply, development operation and maintenance. We use the tool OWASP to find weakness and vulnerabilities of application by customize mode. We have examined authentication in adequacy that permits automated attacks such as credential stuffing where the attacker has a list of username and passwords, session ID bypass authentication page with hypothesize credentials. The testing approach which follows Black Box and gray box testing techniques which examined vulnerability with process of planning, definition assessment and remediation.

INTRODUCTION OF PROJECT

The case study we select for inspection of project is an application of Clinic Management System, that was downloaded from GitHub repository. The project consists of three panel administrators, doctor, and patient's panel. Doctor and patients panel have almost same attributes as this have but with different use cases, for example doctor and patients see their profiles, current and pending appointment treatment history, invoices notification and feedback etc. The third is administrator panel that view, edit, update, or delete any functionality, resolve issues, or access to any sort of information.



MedicX 4 Health Care Login & Registration Form

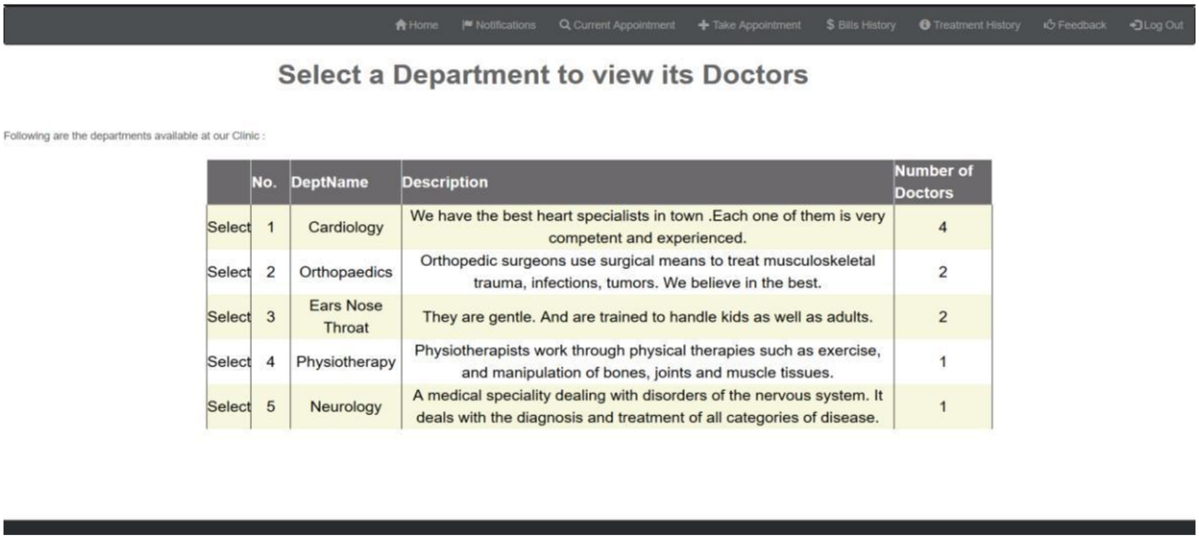
This is a free "Login and Registration form" for Health Care Clinic.

Login to our Website

Enter Email and Password to log in:

Sign up now

Fill in the form below to get instant access:



Select a Department to view its Doctors

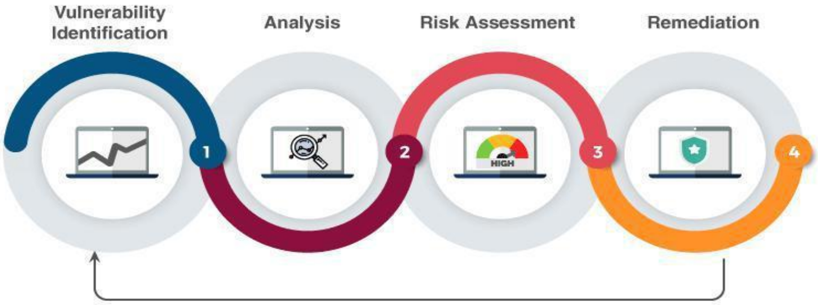
Following are the departments available at our Clinic :

	No.	DeptName	Description	Number of Doctors
Select	1	Cardiology	We have the best heart specialists in town .Each one of them is very competent and experienced.	4
Select	2	Orthopaedics	Orthopedic surgeons use surgical means to treat musculoskeletal trauma, infections, tumors. We believe in the best.	2
Select	3	Ears Nose Throat	They are gentle. And are trained to handle kids as well as adults.	2
Select	4	Physiotherapy	Physiotherapists work through physical therapies such as exercise, and manipulation of bones, joints and muscle tissues.	1
Select	5	Neurology	A medical speciality dealing with disorders of the nervous system. It deals with the diagnosis and treatment of all categories of disease.	1

Methodologies / plan that use to work on project?

In the initial phase of testing, we perform black box testing technique that mainly focus on input and output of software application, based on requirement and specification. Grey box testing is also the main part in our project testing technique to test a software product or application with partial knowledge of internal structure of the application. The purpose of grey box testing is to search and identify the defects due to improper code structure or improper use of applications.

For inspection of vulnerability of application, we create plan that is consist of vulnerability identification, in which we examine threats, weakness of application that cause hacker/ intruders to gain unauthorized access in system. The diagram below shows the process to inspect vulnerability.



What type of vulnerability we face while testing?

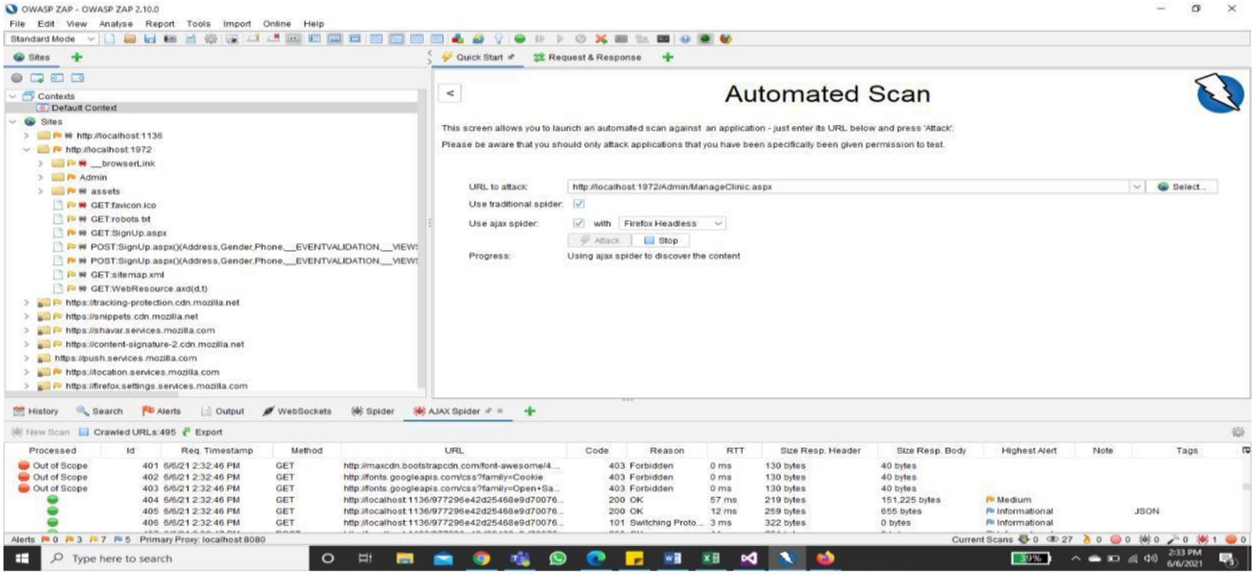
We faced Broken authentication threat while testing the vulnerability in our case study. Some points discussed below to find out the main cause of broken authentication.

- Authentication is broken when attackers can compromise passwords, keys or session, and other details to assume user identities.
- Credential stuffing: The use of list of known passwords, is a common attack, application session timeouts had not set properly.
- Passwords are not properly hashed and salted.

Cross - Site scripting

Cross- site scripting is the 2nd vulnerability we tested in our case study, these attacks are a type of injection, in which malicious scripts are injected. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script to a different end user.

There are some screens shot of testing techniques that are applied and found vulnerability in an application of our case study.


OWASP ZAP - OWASP ZAP 2.10.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request & Response
History Search Alerts Output WebSockets Spider AJAX Spider
Type here to search

Automated Scan

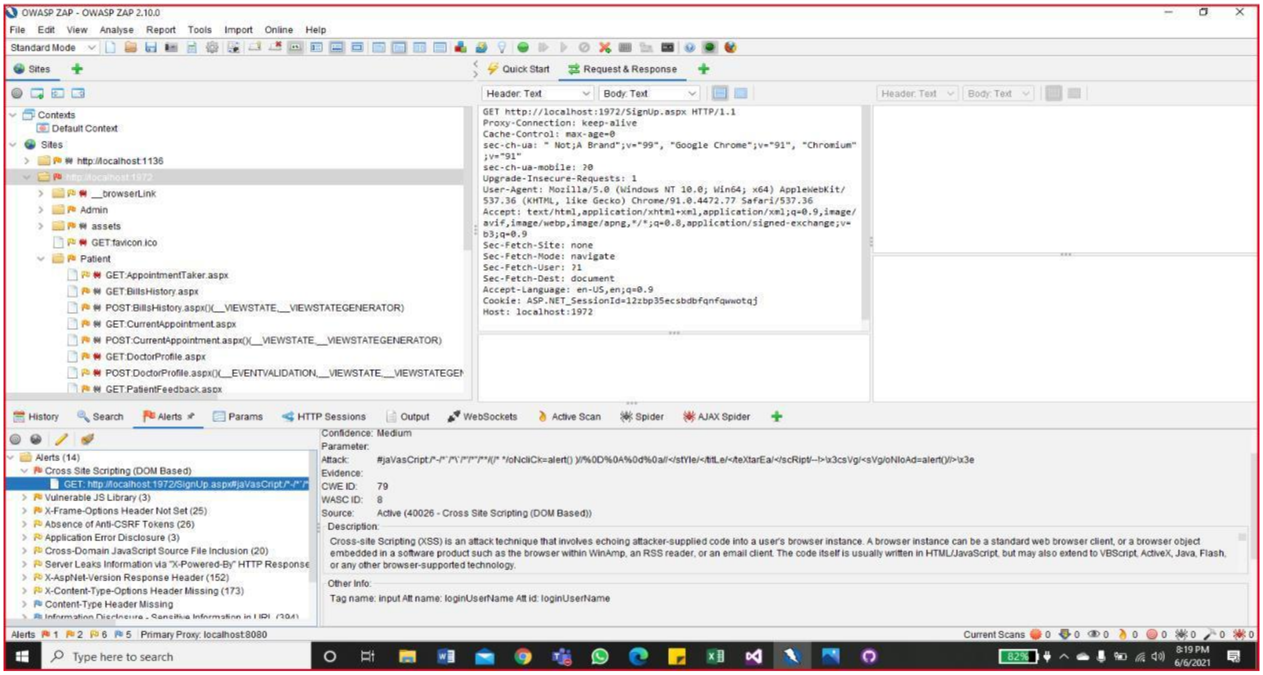
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://localhost:1972/Admin/ManageClinic.aspx
Use traditional spider: ☑
Use ajax spider: ☑ with Firefox Headless
Attack Stop
Progress: Using ajax spider to discover the content

New Scan Crawled URLs:495 Export

Processed	Id	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	Note	Tags
Out of Scope	401	6/6/21 2:32:46 PM	GET	http://maxcdn.bootstrapcdn.com/font-awesome/4...	403	Forbidden	0 ms	130 bytes	40 bytes			
Out of Scope	402	6/6/21 2:32:46 PM	GET	http://fonts.googleapis.com/css?family=Cookie	403	Forbidden	0 ms	130 bytes	40 bytes			
Out of Scope	403	6/6/21 2:32:46 PM	GET	http://fonts.googleapis.com/css?family=Open+Sa...	403	Forbidden	0 ms	130 bytes	40 bytes			
●	404	6/6/21 2:32:46 PM	GET	http://localhost:1136/977296e42d25468e9d70076...	200	OK	57 ms	219 bytes	151,225 bytes	Medium		
●	405	6/6/21 2:32:46 PM	GET	http://localhost:1136/977296e42d25468e9d70076...	200	OK	12 ms	259 bytes	655 bytes	Informational		JSON
●	406	6/6/21 2:32:46 PM	GET	http://localhost:1136/977296e42d25468e9d70076...	101	Switching Proto...	3 ms	322 bytes	0 bytes	Informational		

Alerts 0 3 7 5 Primary Proxy: localhost:8080

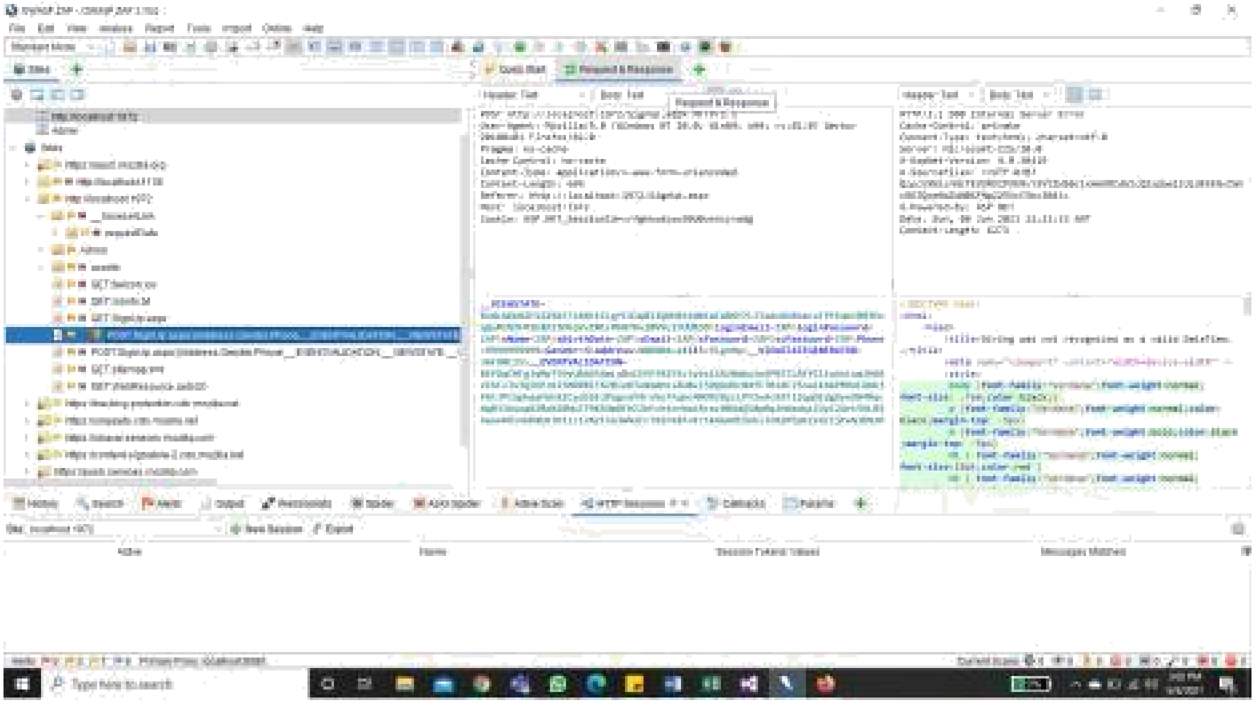

OWASP ZAP - OWASP ZAP 2.10.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request & Response
Header: Text
Body: Text
History Search Alerts Params HTTP Sessions Output WebSockets Active Scan Spider AJAX Spider
Type here to search

```
GET http://localhost:1972/SignUp.aspx HTTP/1.1
Proxy-Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: " Not;A Brand";v="99", "Google Chrome";v="91", "Chromium"
;v="91"
sec-ch-ua-mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/91.0.4472.77 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=
b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Language: en-US,en;q=0.9
Cookie: ASP.NET_SessionId=12zbp35ecsbdbfqnfqwwotqj
Host: localhost:1972
```

Alerts (14)
- Cross Site Scripting (DOM Based)
  - GET: http://localhost:1972/SignUp.aspx#jaVasCript:/*-/*`/*\`/*'/*"/**/(/*
- Vulnerable JS Library (3)
- X-Frame-Options Header Not Set (25)
- Absence of Anti-CSRF Tokens (26)
- Application Error Disclosure (3)
- Cross-Domain JavaScript Source File Inclusion (20)
- Server Leaks Information via "X-Powered-By" HTTP Response
- X-AspNet-Version Response Header (152)
- X-Content-Type-Options Header Missing (173)
- Content-Type Header Missing
- Information Disclosure - Sensitive Information in URL (394)

Confidence: Medium
Parameter:
Attack: #jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert())//%0D%0A%0d%0a//</stYle/</titLe/</teXtarEa/</scRipt/--!>\x3csVg/<sVg/oNloAd=alert()//>\x3e
Evidence:
CWE ID: 79
WASC ID: 8
Source: Active (40026 - Cross Site Scripting (DOM Based))
Description:
Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.
Other Info:
Tag name: input Att name: loginUserName Att id: loginUserName

Alerts 1 2 6 5 Primary Proxy: localhost:8080



```
POST http://localhost:1972/SignUp.aspx HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: [illegible]
Referer: http://localhost:1972/SignUp.aspx
Host: localhost:1972
Cookie: ASP.NET_SessionId=[illegible]
```

```
HTTP/1.1 500 Internal Server Error
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: [illegible]
X-Powered-By: ASP.NET
Date: Sun, 06 Jun 2021 [illegible] GMT
Content-Length: [illegible]
```

```
<!DOCTYPE html>
<html>
    <head>
        <title>String was not recognized as a valid DateTime.</title>
```

What tool we used?

OPEN WEB APPLICATION SECURITY PROJECT

The Open Web Application Security Project (OWASP) is a non-profit foundation dedicated to improving the security of software. OWASP operates under an ‘open community’ model, where anyone can participate in and contribute to projects, events, online chats, and more. A guiding principle of OWASP is that all materials and information are free and easily accessed on their website, for everyone. OWASP offers everything from tools, videos, forums, projects, to events. In short, OWASP is a repository of all thing’s web-application-security, backed by the extensive knowledge and experience of its open community contributors.

The **Zed Attack Proxy (ZAP)** is an easy-to-use integrated penetration testing tool for finding vulnerabilities in web applications.

It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing. ZAP provides automated scanners as well as a set of tools that allow you to find security vulnerabilities manually.

How do we react on report those alerts?

Broken authentication is an umbrella term for several vulnerabilities that attackers exploit to impersonate legitimate users online. Broadly, broken authentication refers to weaknesses in two areas:

- Credential management
- Session management

There are two main factor of credential management that are prevented to minimize these attacks.

- Weak passwords
- Weak cryptography

Weak passwords: The consumer creates a weak password like '12345' or 'pass123'. The hacker can use various password cracking techniques like rainbow tables and dictionaries to gain access to the system.

Weak cryptography: Using weak encryption techniques like base64 and weak hashing algorithms like SHA1 and MD5 make credentials vulnerable.

- Session management

This happens when the web application produces a session cookie, which value is easily guessable. An attacker may be able to counterfeit session cookie by guessing its value (for example after a brute force attempt) and thus easily perform a session hijacking attack.

- Session timeout

Setting a reasonable session timeout can help ensure that attackers who steal session IDs have a limited timeframe to use them in.

A session timeout can depend on the security profile of the application. 5-10 minutes may be reasonable, but the organization should decide based on risk.

- Network encryption

Encrypting traffic using SSLJTLS is another way to protect user information and is becoming more common. Hacker could intercept web traffic and steal session data, but if traffic is encrypted, this is no longer possible, or at least very difficult. Encryption is a good first defense against session hijacking.

- Cross-site scripting

In general, effectively preventing XSS vulnerabilities is likely to involve a combination of the following measures:

- Filter input on arrival

At the point where user input is received, filter as strictly as possible based on what is expected or valid input.

- Encode data on output.

Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.

CONCLUSION

Consider security at all stages of development cycle by its customizing tools. Authentication security testing tool that scans through your web application to identity any security vulnerabilities as possible, ZAP generates the scan report in the form of Alerts that are marked with color coded flags which Is helpful for identifying the weakness of authentication by it customize priorities. It is providing best approach to find application’s bug by its selective testing type and then user can easily resolve bugs.