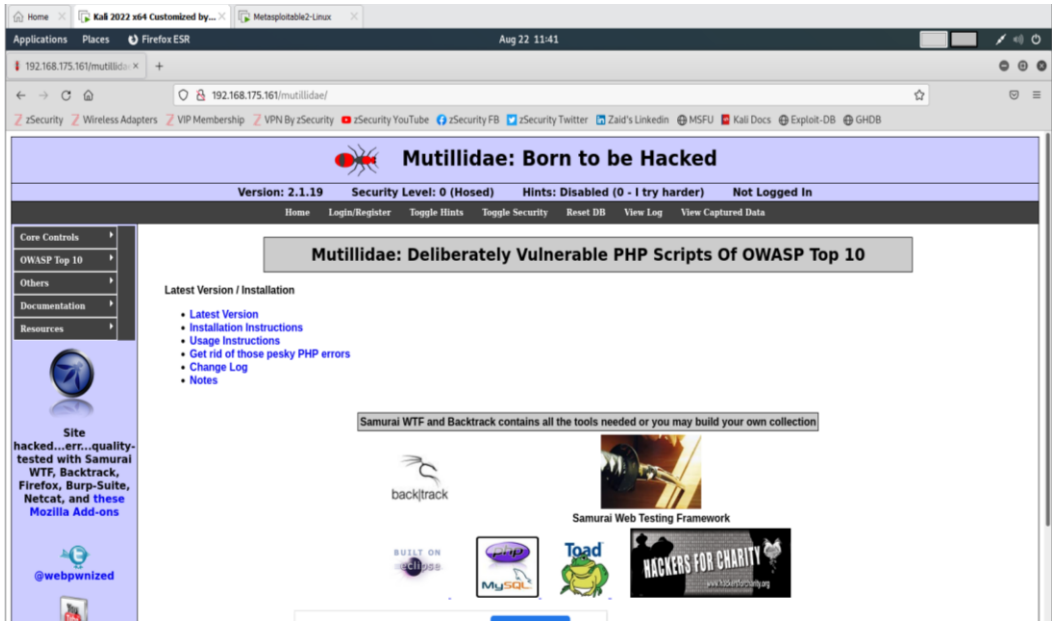
PROJECT 8 : Performing Web Application penetration using automated target attack using ZAP

Mutillidae is an open-source insecure web application that is designed for penetration testers to practice web app-specific vulnerability exploitation.

Performing Web Application penetration using automated target attack using ZAP

➔ *Target* - **http://192.168.175.161/mutillidae/**

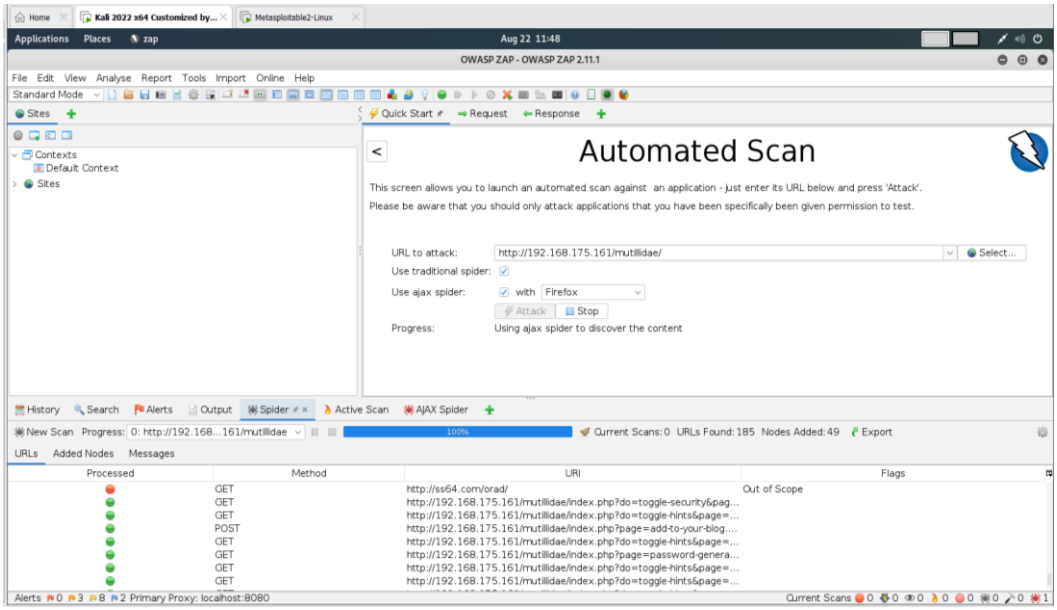
<u>Step 1:</u> A visit to target homepage



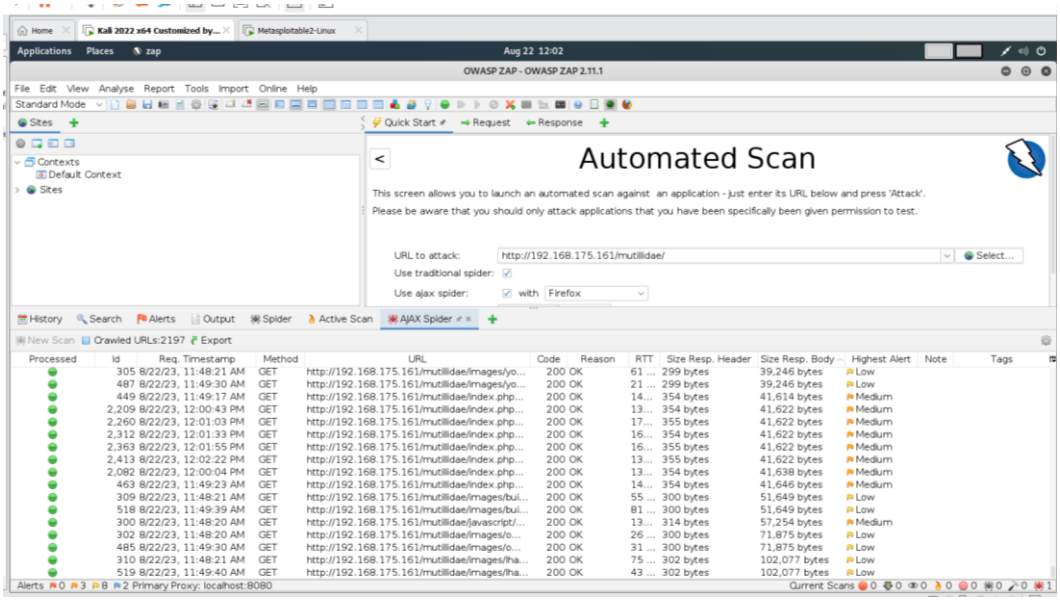
Target homepage

<u>Step 2:</u> Website scan in OWASP ZAP with spider, ajax spider, active scan and some alert

Spider: This package contains an open-source intelligence (OSINT) automation tool. Its goal is to automate the process of gathering intelligence about a given target, which may be an IP address, domain name, hostname, network subnet, ASN, e-mail address or person's name.

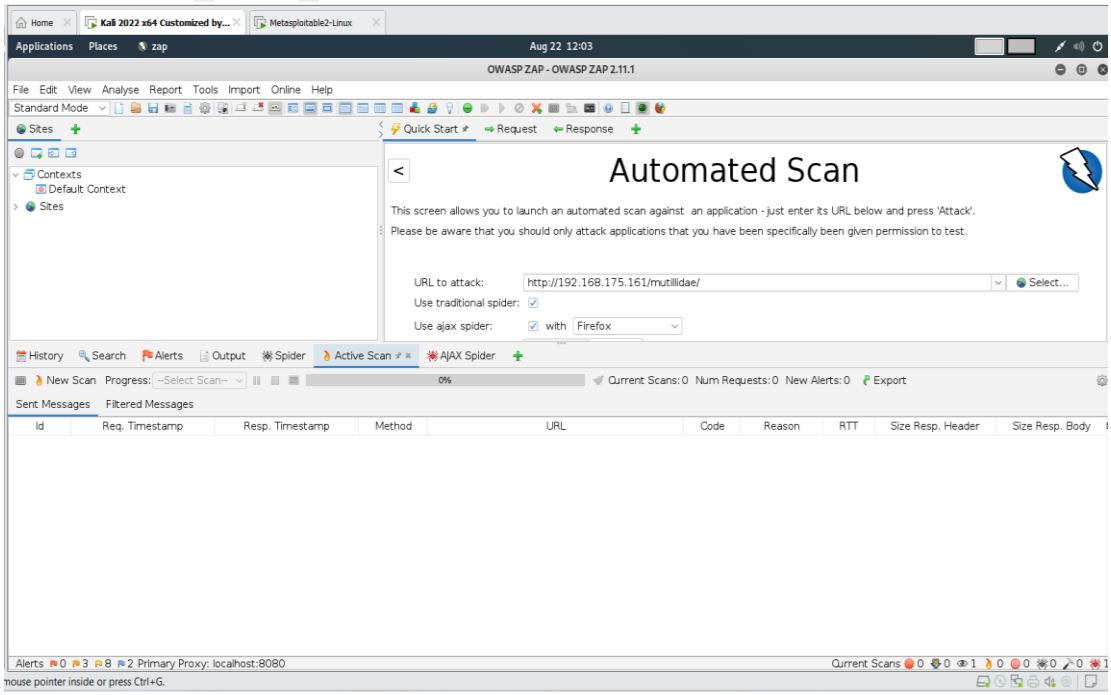


Ajax spider: The AJAX Spider is an add-on for a crawler called Crawljax. The add-on sets up a local proxy in ZAP to talk to Crawljax. The AJAX Spider allows you to crawl web applications written in AJAX in far more depth than the native Spider. Use the AJAX Spider if you may have web applications written in AJAX.



Active scan: Active scanning is an attack on those targets. You should NOT use it on web applications that you do not own.

It should be noted that active scanning can only find certain types of vulnerabilities. Logical vulnerabilities, such as broken access control, will not be found by any active or automated vulnerability scanning.

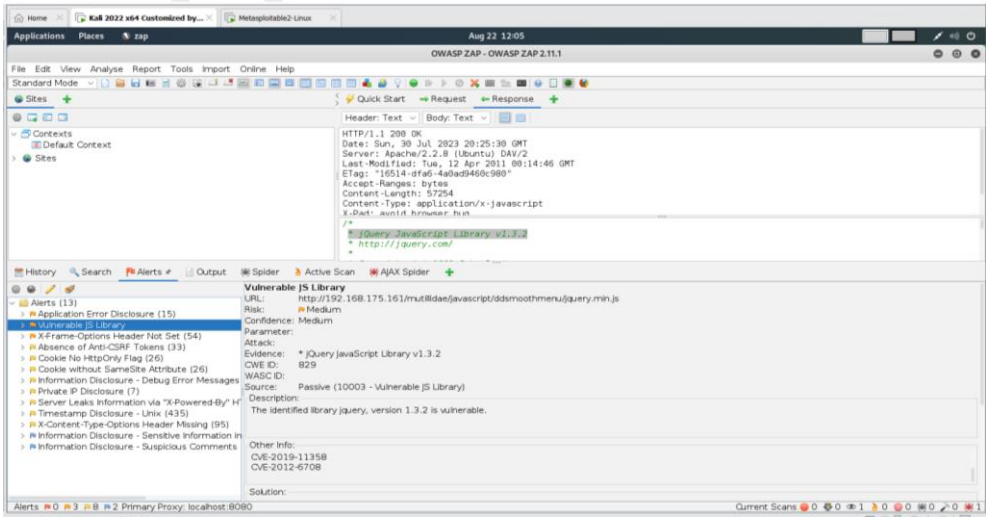


Alert: An alert is a potential vulnerability and is associated with a specific request.
A request can have more than one alert.

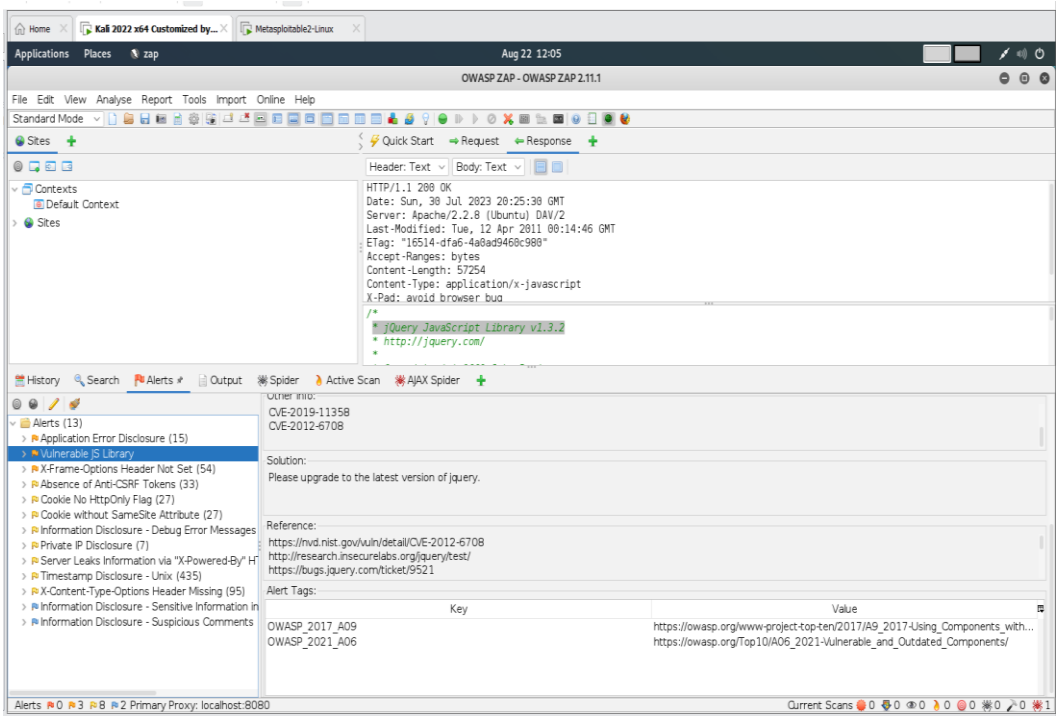
Alerts are shown in the UI with a flag indicating the risk:

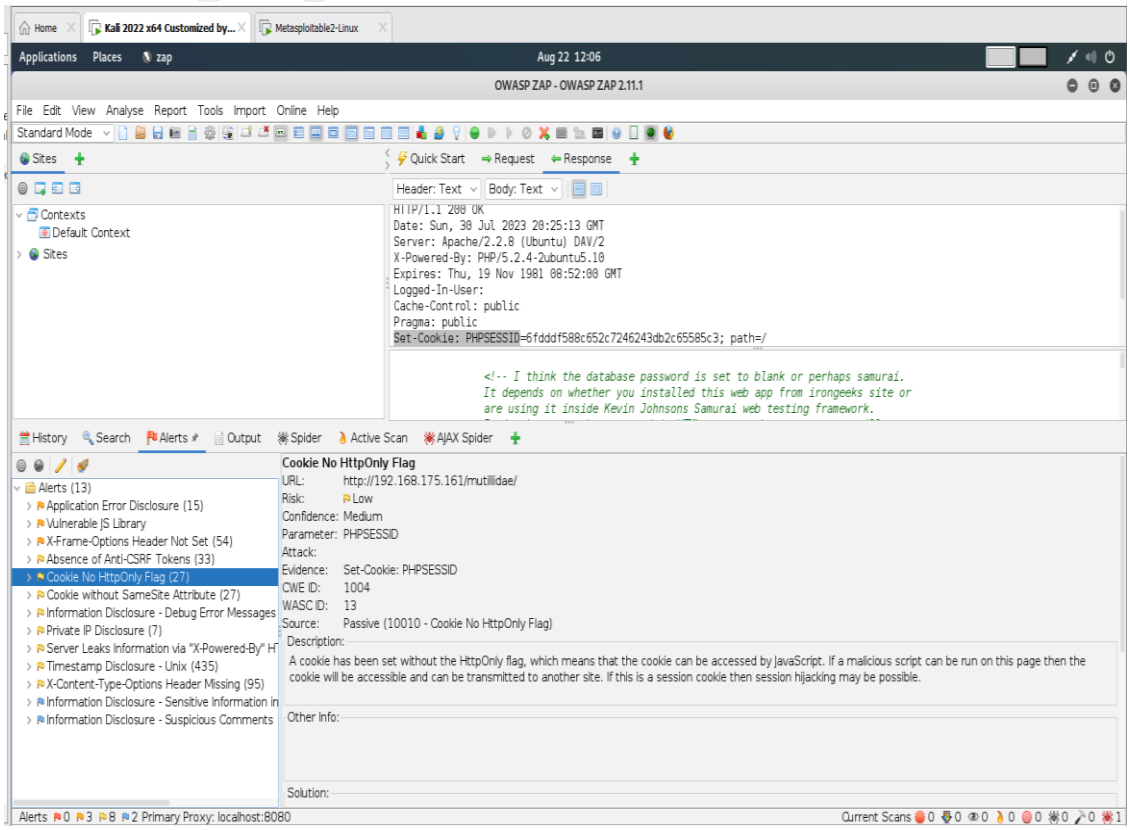
- High
- Medium
- Low
- Informational
- False Positive

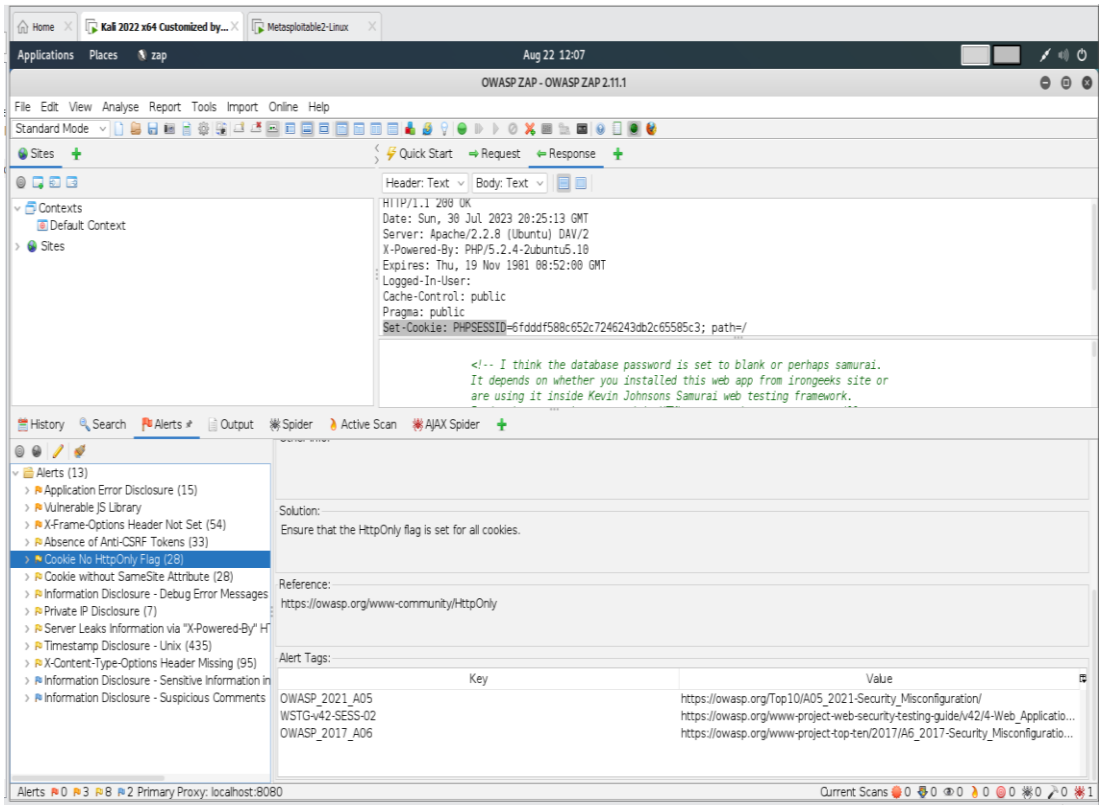
Alerts are flagged in the History tab with a flag which indicates the highest risk alert.
All alerts are listed in the Alerts tab and a count of the total number of alerts by risk is shown in the footer.

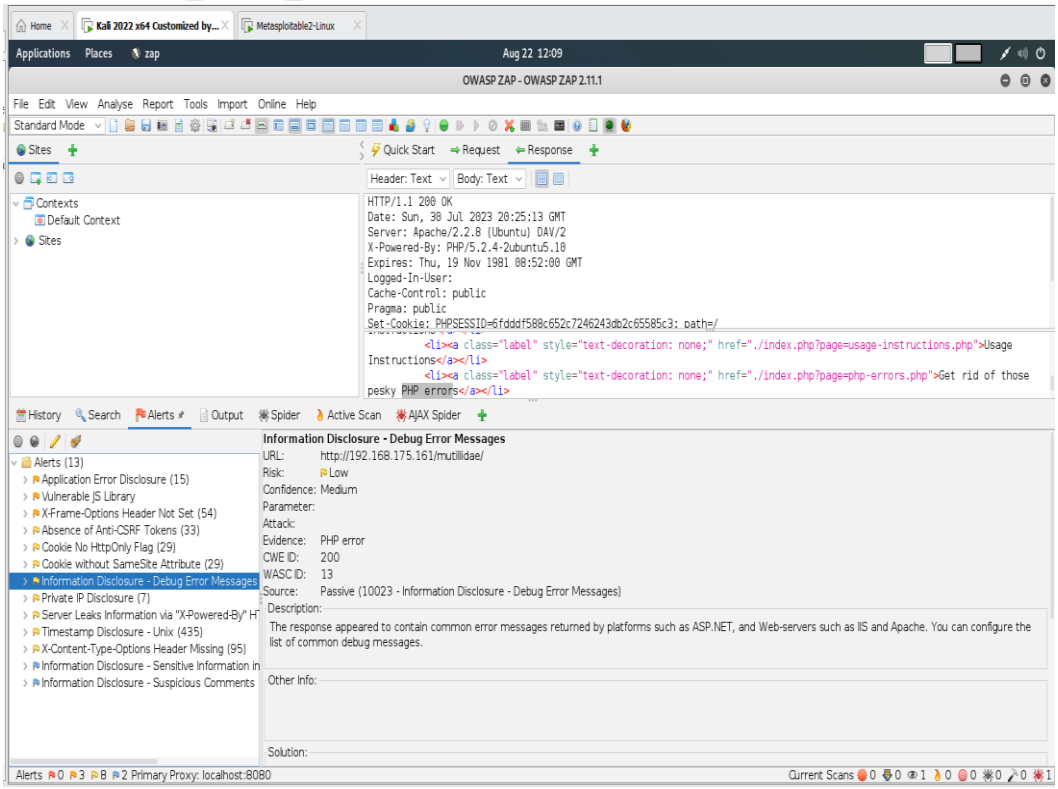


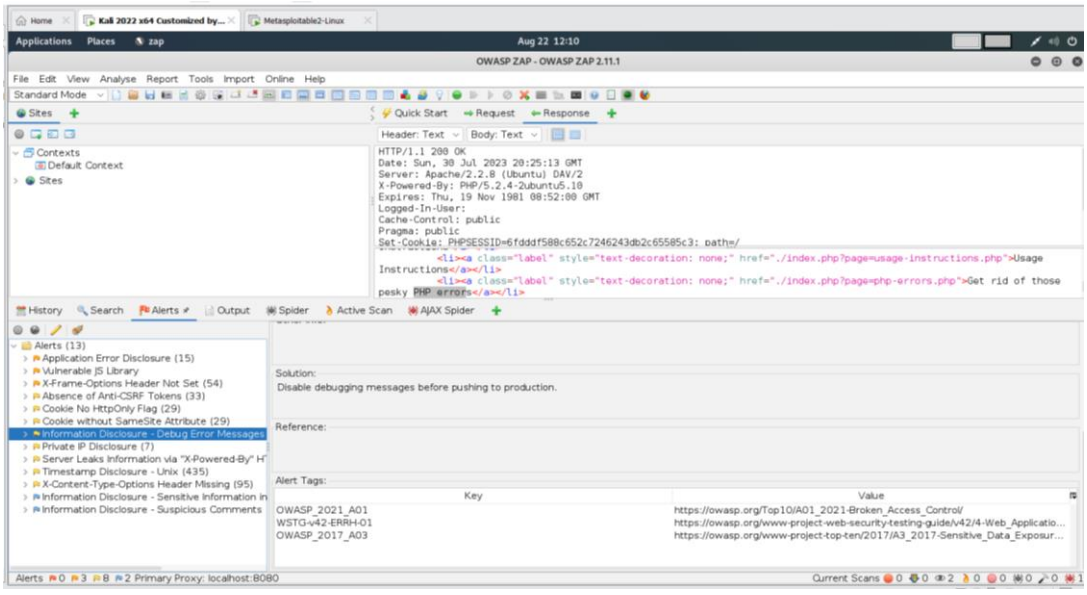
Alerts can be raised by various ZAP components, including but not limited to: active scanning, passive scanning, scripts, by addons (extensions), or manually using the Add Alert dialog (which also allows you to update or change alert details/information).




Home
Kali 2022 x64 Customized by...
Metasploitable2-Linux
Applications
Places
zap
Aug 22 12:06
OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Date: Sun, 30 Jul 2023 20:25:13 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Set-Cookie: PHPSESSID=6fdddf588c652c7246243db2c65585c3; path=/
<!-- I think the database password is set to blank or perhaps samurai.
It depends on whether you installed this web app from irongeeks site or
are using it inside Kevin Johnsons Samurai web testing framework.
History
Search
Alerts
Output
Spider
Active Scan
AJAX Spider
Alerts (13)
Application Error Disclosure (15)
Vulnerable JS Library
X-Frame-Options Header Not Set (54)
Absence of Anti-CSRF Tokens (33)
Cookie No HttpOnly Flag (27)
Cookie without SameSite Attribute (27)
Information Disclosure - Debug Error Messages
Private IP Disclosure (7)
Server Leaks Information via "X-Powered-By" H
Timestamp Disclosure - Unix (435)
X-Content-Type-Options Header Missing (95)
Information Disclosure - Sensitive Information in
Information Disclosure - Suspicious Comments
Cookie No HttpOnly Flag
URL: http://192.168.175.161/mutillidae/
Risk: Low
Confidence: Medium
Parameter: PHPSESSID
Attack:
Evidence: Set-Cookie: PHPSESSID
CWE ID: 1004
WASC ID: 13
Source: Passive (10010 - Cookie No HttpOnly Flag)
Description:
A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Other Info:
Solution:
Alerts 0 3 8 2 Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0 1


Home
Kali 2022 x64 Customized by...
Metasploitable2-Linux
Applications
Places
zap
Aug 22 12:07
OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Date: Sun, 30 Jul 2023 20:25:13 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Set-Cookie: PHPSESSID=6fdddf588c652c7246243db2c65585c3; path=/
<!-- I think the database password is set to blank or perhaps samurai.
It depends on whether you installed this web app from irongeeks site or
are using it inside Kevin Johnsons Samurai web testing framework.
History
Search
Alerts
Output
Spider
Active Scan
AJAX Spider
Alerts (13)
Application Error Disclosure (15)
Vulnerable JS Library
X-Frame-Options Header Not Set (54)
Absence of Anti-CSRF Tokens (33)
Cookie No HttpOnly Flag (28)
Cookie without SameSite Attribute (28)
Information Disclosure - Debug Error Messages
Private IP Disclosure (7)
Server Leaks Information via "X-Powered-By" H
Timestamp Disclosure - Unix (435)
X-Content-Type-Options Header Missing (95)
Information Disclosure - Sensitive Information in
Information Disclosure - Suspicious Comments
Solution:
Ensure that the HttpOnly flag is set for all cookies.
Reference:
https://owasp.org/www-community/HttpOnly
Alert Tags:
Key
Value
OWASP_2021_A05
https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
WSTG-v42-SESS-02
https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Applicatio...
OWASP_2017_A06
https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguratio...
Alerts 0 3 8 2 Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0 1


Home
Kali 2022 x64 Customized by...
Metasploitable2-Linux
Applications Places zap
Aug 22 12:09
OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Date: Sun, 30 Jul 2023 20:25:13 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Set-Cookie: PHPSESSID=6fdddf588c652c7246243db2c65585c3; path=/
Usage Instructions
Get rid of those pesky PHP errors
History Search Alerts Output Spider Active Scan AJAX Spider
Alerts (13)
Application Error Disclosure (15)
Vulnerable JS Library
X-Frame-Options Header Not Set (54)
Absence of Anti-CSRF Tokens (33)
Cookie No HttpOnly Flag (29)
Cookie without SameSite Attribute (29)
Information Disclosure - Debug Error Messages
Private IP Disclosure (7)
Server Leaks Information via "X-Powered-By" H
Timestamp Disclosure - Unix (435)
X-Content-Type-Options Header Missing (95)
Information Disclosure - Sensitive Information in
Information Disclosure - Suspicious Comments
Information Disclosure - Debug Error Messages
URL: http://192.168.175.161/mutillidae/
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: PHP error
CWE ID: 200
WASC ID: 13
Source: Passive (10023 - Information Disclosure - Debug Error Messages)
Description:
The response appeared to contain common error messages returned by platforms such as ASP.NET, and Web-servers such as IIS and Apache. You can configure the list of common debug messages.
Other Info:
Solution:
Alerts 0 3 8 2 Primary Proxy: localhost:8080
Current Scans 0 0 1 0 0 0 0 1


Home
Kali 2022 x64 Customized by...
Metasploitable2-Linux
Applications Places zap
Aug 22 12:10
OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Date: Sun, 30 Jul 2023 20:25:13 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Set-Cookie: PHPSESSID=6fdddf588c652c7246243db2c65585c3; path=/
Usage Instructions
Get rid of those pesky PHP errors
History Search Alerts Output Spider Active Scan AJAX Spider
Alerts (13)
Application Error Disclosure (15)
Vulnerable JS Library
X-Frame-Options Header Not Set (54)
Absence of Anti-CSRF Tokens (33)
Cookie No HttpOnly Flag (29)
Cookie without SameSite Attribute (29)
Information Disclosure - Debug Error Messages
Private IP Disclosure (7)
Server Leaks Information via "X-Powered-By" H
Timestamp Disclosure - Unix (435)
X-Content-Type-Options Header Missing (95)
Information Disclosure - Sensitive Information in
Information Disclosure - Suspicious Comments
Solution:
Disable debugging messages before pushing to production.
Reference:
Alert Tags:
Key
Value
OWASP_2021_A01
https://owasp.org/Top10/A01_2021-Broken_Access_Control/
WSTG-v42-ERRH-01
https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Applicatio...
OWASP_2017_A03
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposur...
Alerts 0 3 8 2 Primary Proxy: localhost:8080
Current Scans 0 0 2 0 0 0 0 1

Solution of alert:

OWASP_2017_A09

Sol:

There should be a patch management process in place to:

*Remove unused dependencies, unnecessary features, components, files, and documentation.

* Continuously inventory the versions of both client-side and server-side components (e.g. frameworks, libraries) and their dependencies using tools like versions, DependencyCheck, retire.js, etc. Continuously monitor sources like CVE and NVD for vulnerabilities in the components. Use software composition analysis tools to automate the process. Subscribe to email alerts for security vulnerabilities related to components you use.

* Only obtain components from official sources over secure links. Prefer signed packages to reduce the chance of including a modified, malicious component.

* Monitor for libraries and components that are unmaintained or do not create security patches for older versions. If patching is not possible, consider deploying a virtual patch to monitor, detect, or protect against the discovered issue.

Every organization must ensure that there is an ongoing plan for monitoring, triaging, and applying updates or configuration changes for the lifetime of the application or portfolio.

WSTG-v42-SESS-02

By design cookies do not have the capabilities to guarantee the integrity and confidentiality of the information stored in them. Those limitations make it impossible for a server to have confidence about how a given cookie’s attributes were set at creation. In order to give the servers such features in a backwards-compatible way, the industry has introduced the concept of Cookie Name Prefixes to facilitate passing such details embedded as part of the cookie name.

OWASP_2021_a01

Sol: Access control is only effective in trusted server-side code or server-less API, where the attacker cannot modify the access control check or metadata.

● Except for public resources, deny by default.

● Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.

● Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.

● Unique application business limit requirements should be enforced by domain models.

● Disable web server directory listing and ensure file metadata (e.g., .git) and backup files are not present within web roots.

● Log access control failures, alert admins when appropriate (e.g., repeated failures).

● Rate limit API and controller access to minimize the harm from automated attack tooling.

Stateful session identifiers should be invalidated on the server after logout. Stateless JWT tokens should rather be short-lived so that the window of opportunity for an attacker is minimized. For longer lived JWTs it's highly recommended to follow the OAuth standards to revoke access.

WSTG-v42-EERH-01

Sol:

● Manage exceptions in a centralized manner to avoid duplicated try/catch blocks in the code. Ensure that all unexpected behavior is correctly handled inside the application.

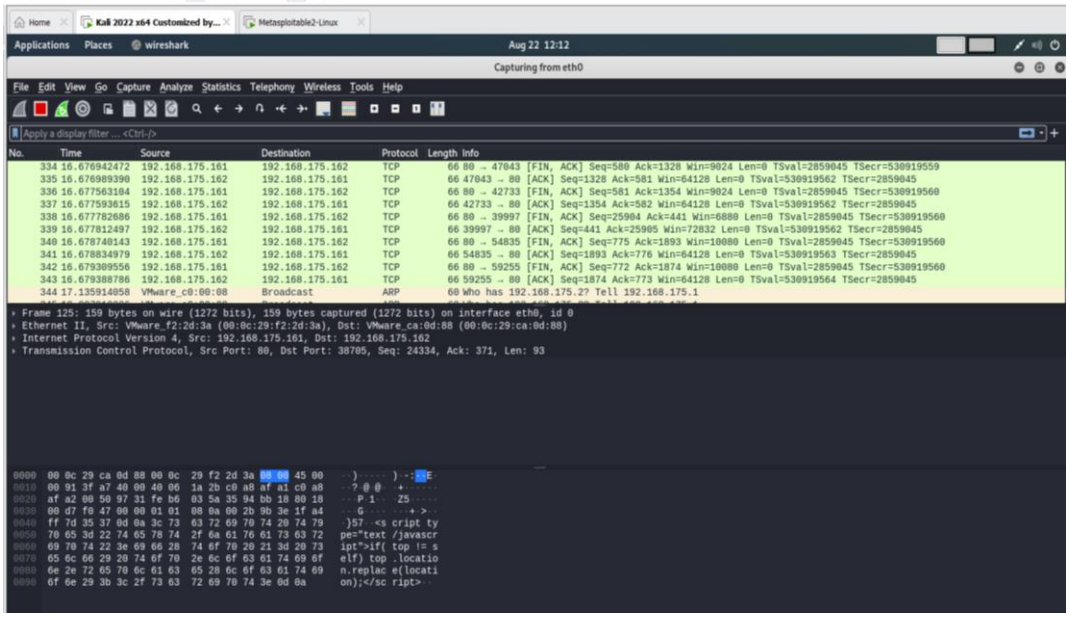
● Ensure that error messages displayed to users do not leak critical data, but are still verbose enough to enable the proper user response.

● Ensure that exceptions are logged in a way that gives enough information for support, QA, forensics or incident response teams to understand the problem.

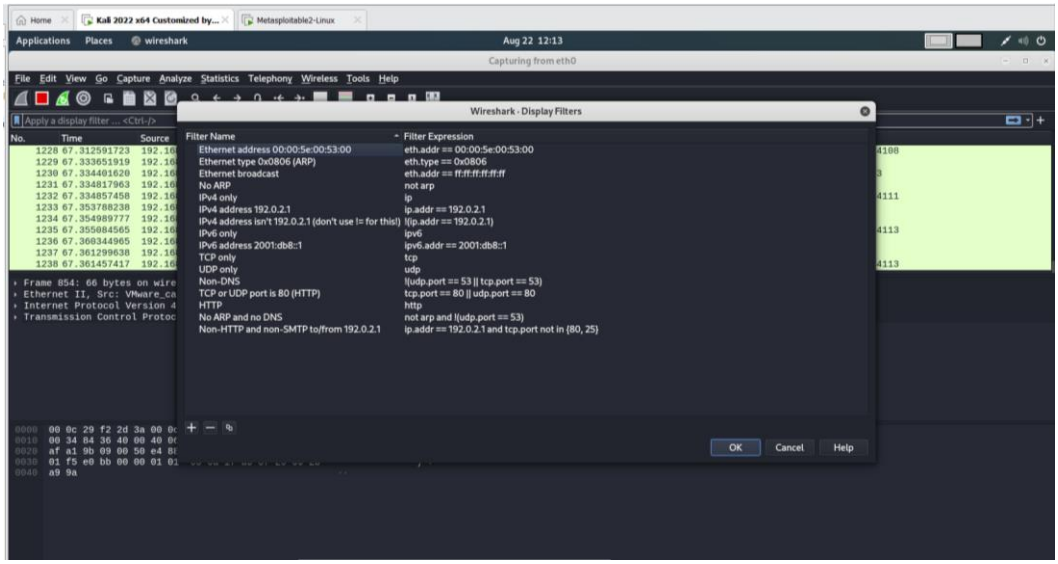
● Carefully test and verify error handling code

<u>Step 3:</u> Packet scanning in Wireshark

Wireshark is a popular network protocol analyser that allows you to capture and inspect the data traveling back and forth on a network. Packet scanning in Wireshark involves capturing and analysing network packets to gain insights into the traffic, identify potential issues, and troubleshoot network problems. It's a valuable tool for network administrators, security professionals, and developers.



Random packets Analyse: "Random packet analysis" could refer to the process of analysing packets that have been captured from a network and appear to have random or unpredictable content. This type of analysis might involve examining the structure, headers, and payloads of these packets to understand their nature and purpose, even if their content appears to lack a discernible pattern.



The use of this tool in our project is as shown in the above screenshot's.

Thank you!

Project 8 team Members:

- P.Aishwarya
- Saranya.P
- Joel Biju
- Prajwal Adhav
- Nivin kv