**16-09-22**

**Dear Interviewer, Greetings!!**

Thank you for the opportunity to perform the technical assessment. It was really a great refreshment and able to learn few open source tool and techniques. Am indeed thankful for that and looking forward to hear from you soon..

Please find the below findings..

Sincerely,

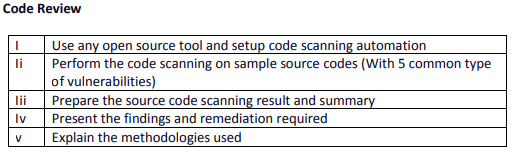
[Rangarajan]

**Assignment Details: 1**

Technical Assignment Prepare the testing environment and choose 2 of the following assignments below that best match to your past experiences for one of the following Github repositories

• <https://github.com/scalessec/Toast-Swift>

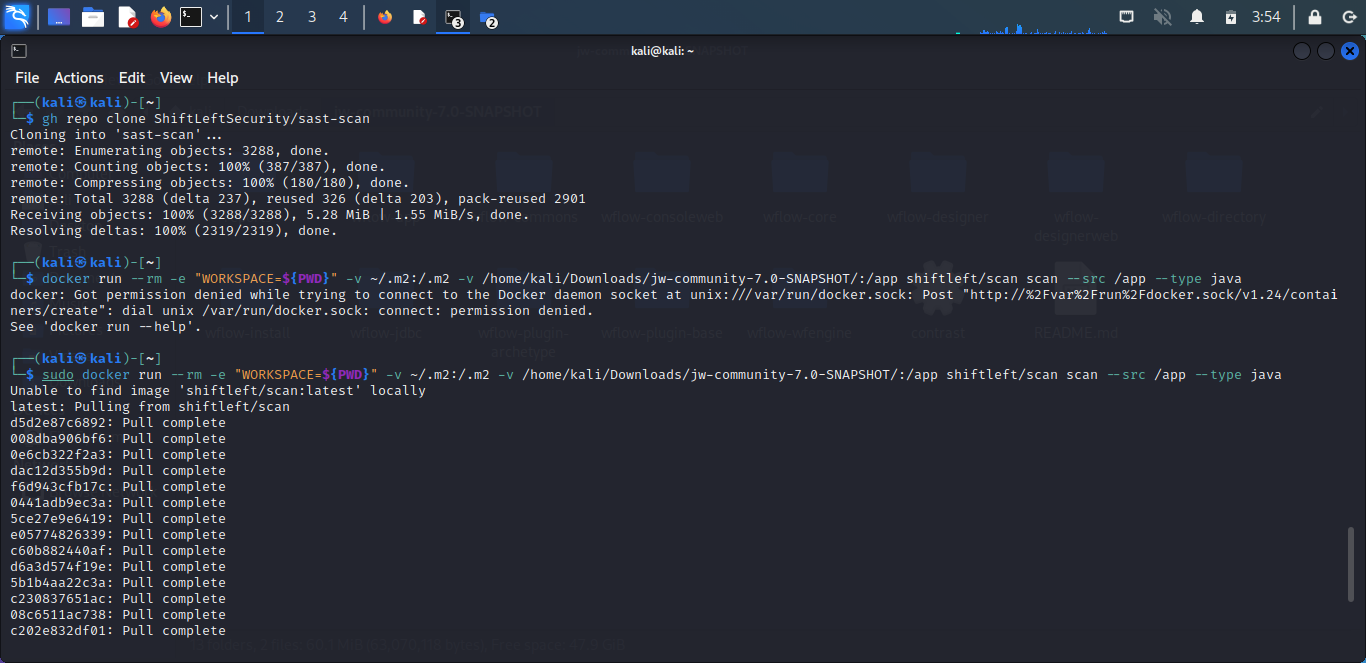
• <https://github.com/jogetworkflow/jw-community>

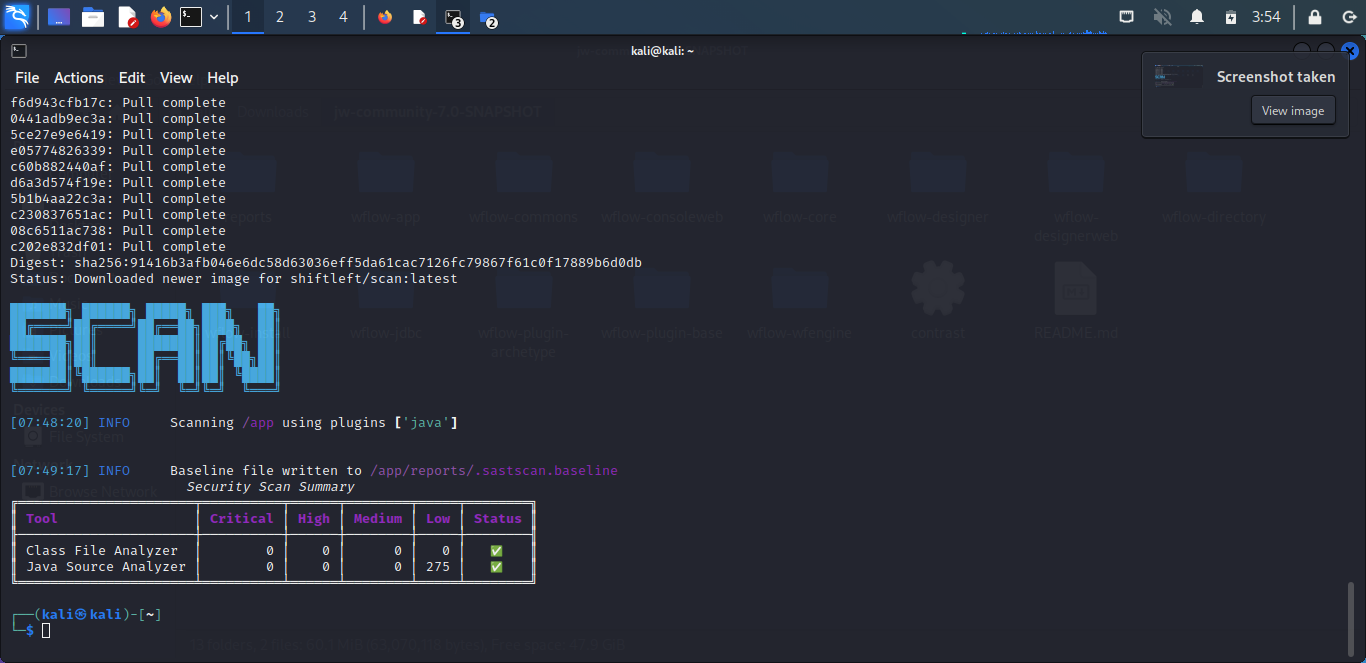


1. **Tool Name : ShiftLeftSecurity Scan**

ShiftLeft Scan lets you protect custom code with static analysis (SAST), secure open-source libraries (SCA), and employ hard-coded secrets detection and OSS license violation checks.

**Platform: Kali VM.**





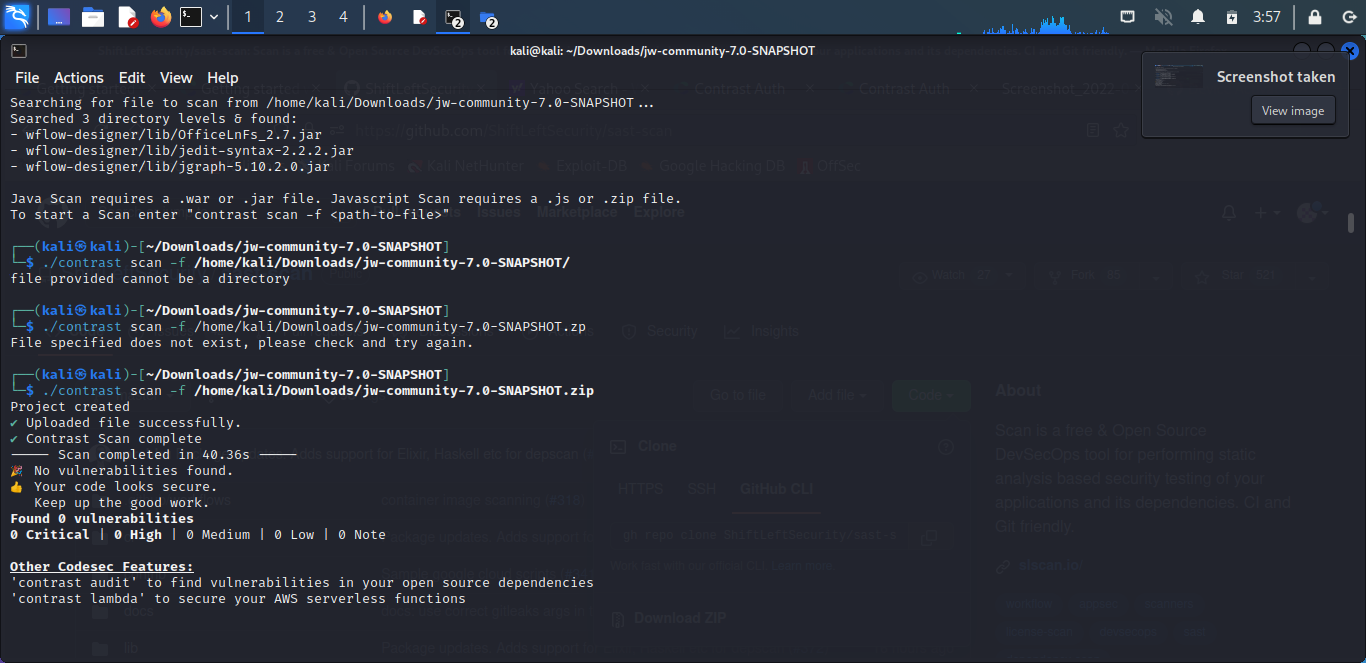
**Summary of the finding s:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool** | **Critical** | **High** | **Medium** | **Low** |
| **Class File Analyzer** | **0** | **0** | **0** | **0** |
| **Java Source Analyzer** | **0** | **0** | **0** | **275** |

1. **Tool Name: Contrast OSS**

Contrast OSS works by installing an intelligent agent that equips the application with smart sensors to analyze code in real time from within the application

**Plafform: Kali VM**



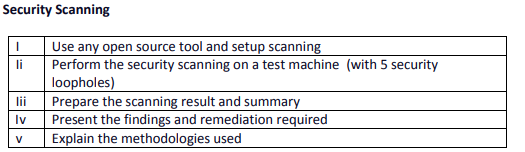
**Summary of the finding s:**

|  |  |  |  |
| --- | --- | --- | --- |
| Critical | High | Medium | Low |
| 0 | 0 | 0 | 0 |

**Assignment Details: 2**

Technical Assignment Prepare the testing environment and choose 2 of the following assignments below that best match to your past experiences for one of the following Github repositories

• <https://github.com/scalessec/Toast-Swift>

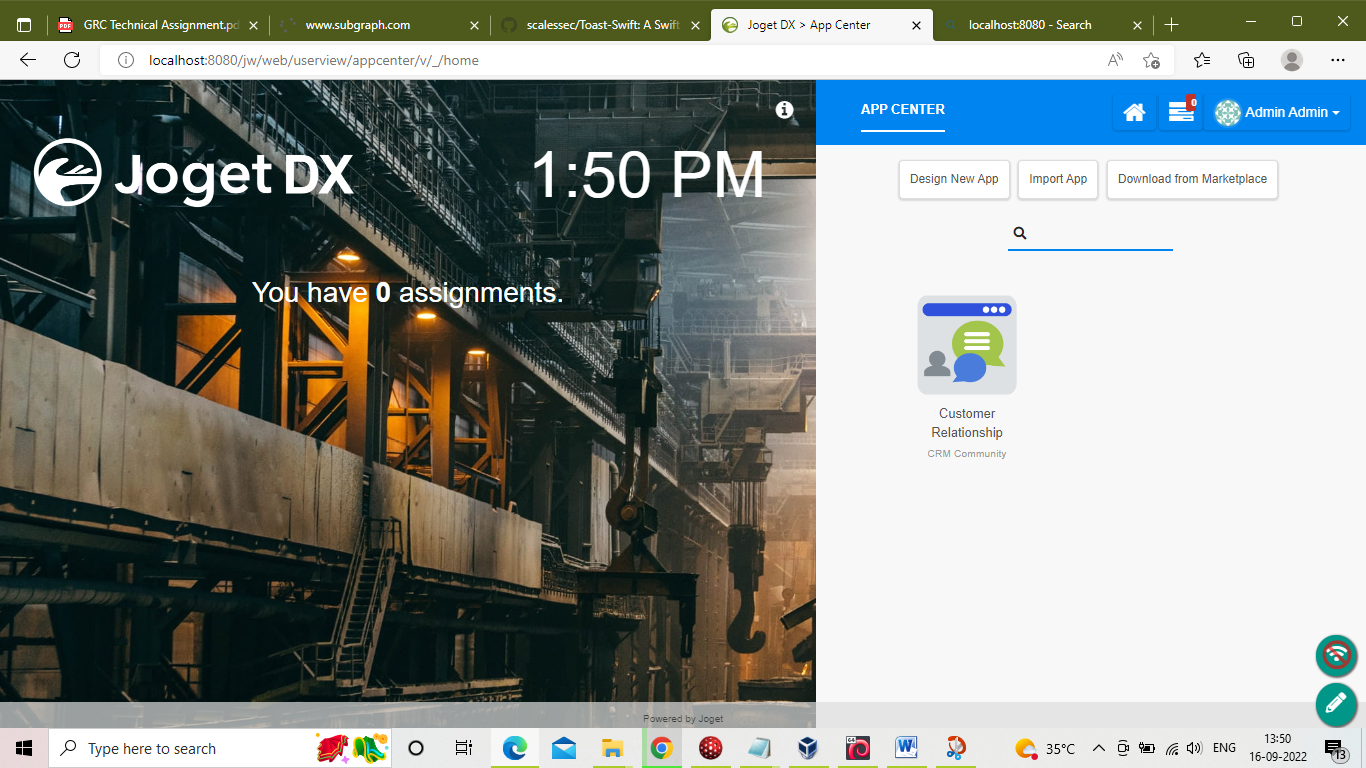
• <https://github.com/jogetworkflow/jw-community> 

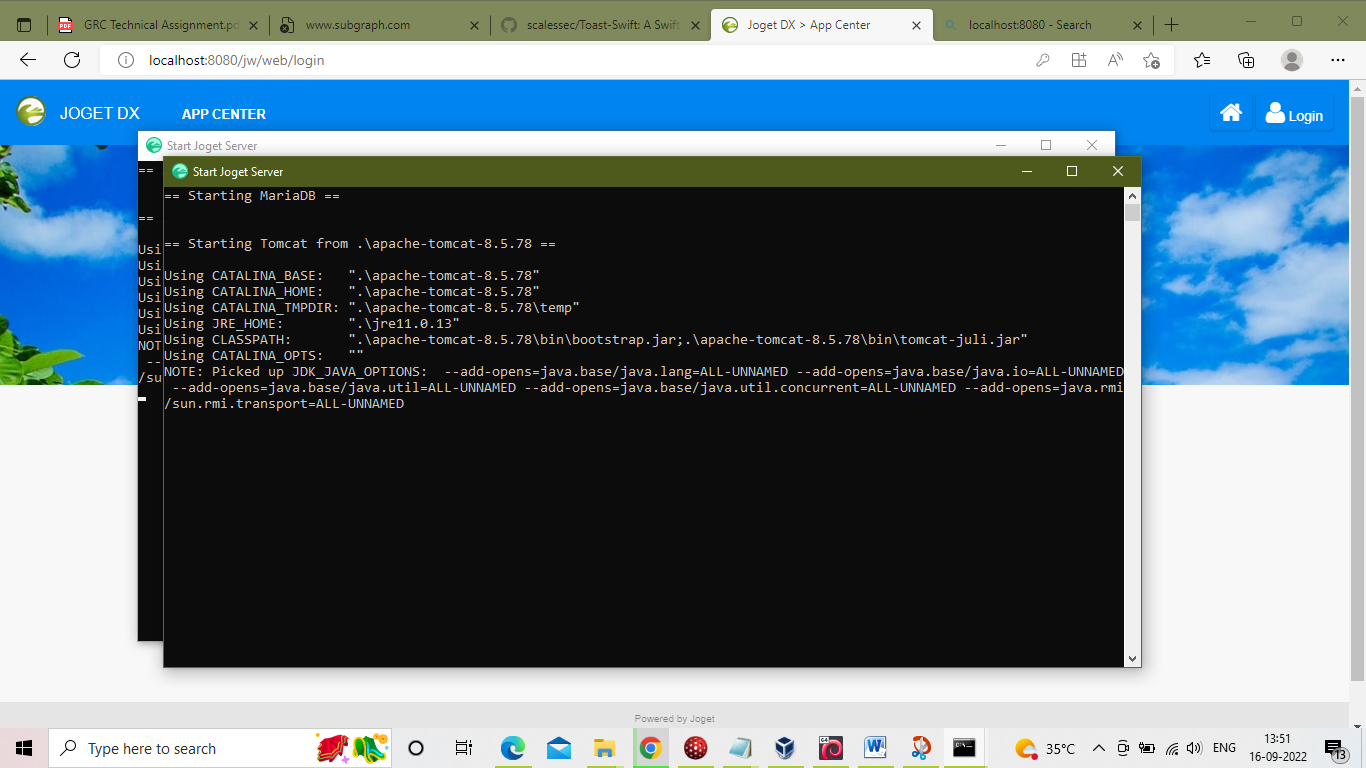
**Tool Name : Vega**

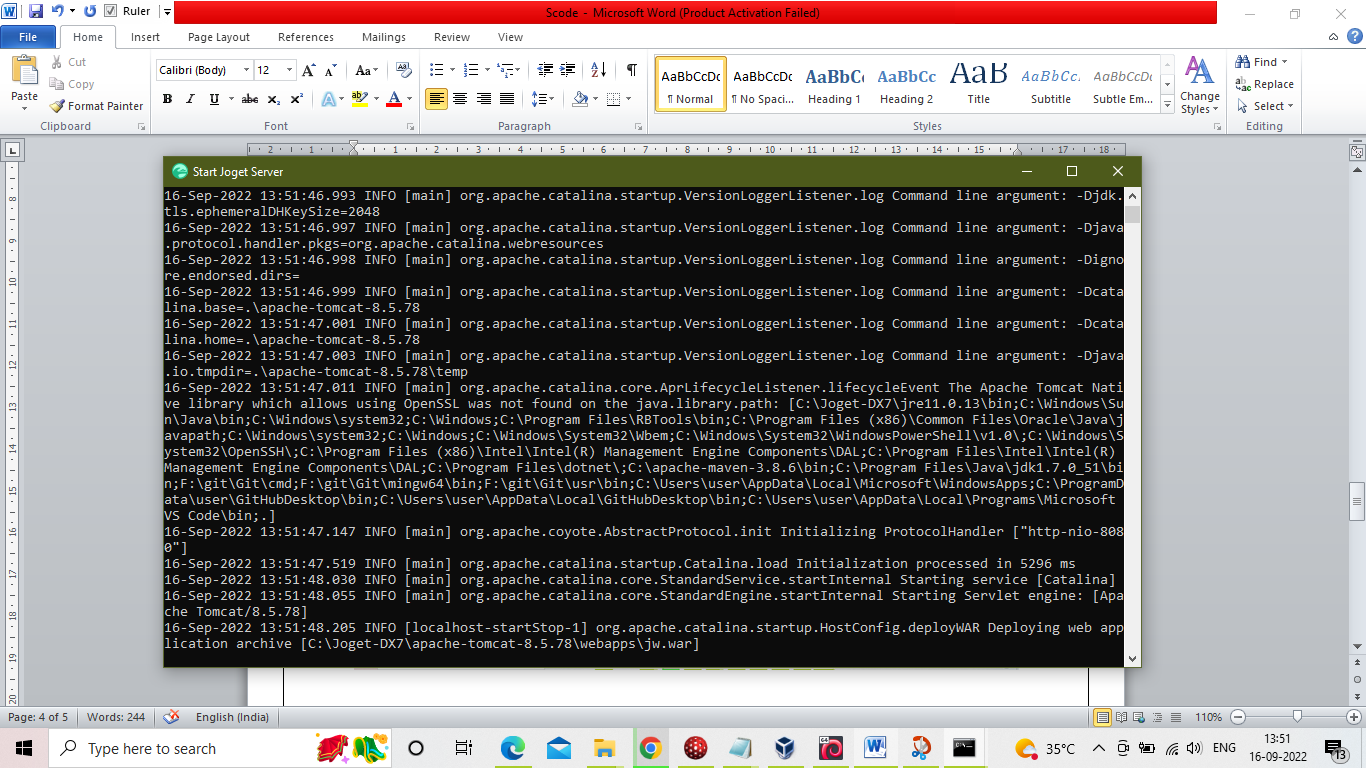
Vega, the Open Source Web Application Security Platform.

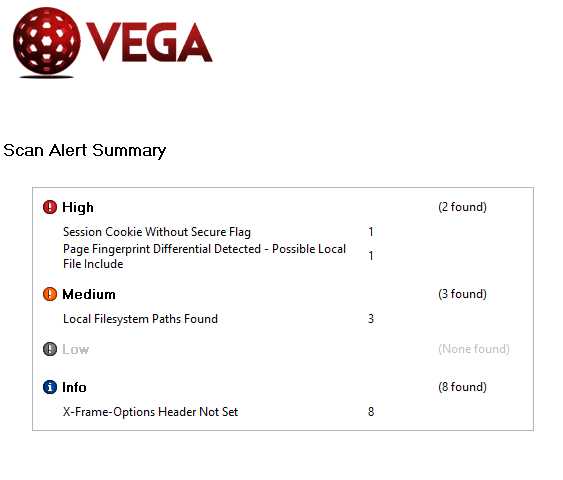
Platform : Windows 10 OS & Joget setup 7.0.32

Web Application:









**Summary of the findings:**

|  |  |  |  |
| --- | --- | --- | --- |
| Critical | High | Medium | Low |
| 0 | 2 | 3 | 0 |

**Details of findings:**

**High : 1**

|  |  |
| --- | --- |
| **Classification** | [Error Message](about:blank#description) |
| **Resource** | **/jw/web/userview/appcenter/v/\_/home** |
| **Parameter** | **\_action** |
| **Method** | **GET** |
| **Risk** | High |

**Request**

[GET /jw/web/userview/appcenter/v/\_/home?\_action=/./](about:blank)

**Resource Content**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">    <head>  <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  <meta charset="utf-8" />  <meta name="viewport" content="width=device-width, initial-scale=1">  <meta name="msapplication-tap-highlight" content="no"/>  <meta name="theme-color" content="#0084F0"/>  <link rel="apple-touch-icon" hr... |

**Impact**

* It has detected a different response fingerprint in relation to a local file include injection attempt.
* This may indicate a local file include vulnerability, though this is not confirmed.
* If this is due to a local file include vulnerability, exploitation of local file include vulnerabilities can allow attackers to gain unauthorized access to files, which may also aid in other attacks.
* Differing responses may also indicate the presence of a file enumeration vulnerability, which instead of allowing the attacker to gain access to file contents, may allow them to determine if files exist on the system.

**Remediation**

* To prevent this type of vulnerability, the developer should canonicalize the path of any filesystem resource that has a path composed of externally-supplied input and then perform an authorization check prior to access.
* The realpath() library call will return the canonical path of the resource. It is implemented in PHP, Perl, and Python.
* For Ruby frameworks, File.expand\_path can be used.
* GetFullPath() can be used on ASP.NET applications.
* getCanonicalPath() can be used in Java code.
* Additional protection against unauthorized access to filesystem resources can be obtained by using chroot() or similar mechanisms to limit filesystem access to the web application and http server process, although this can be difficult to manage.

**References**

Some additional links with relevant information published by third-parties:

* [Directory Traversal (Wikipedia)](http://en.wikipedia.org/wiki/Directory_traversal)
* [Path Traversal (OWASP)](https://www.owasp.org/index.php/Path_Traversal)
* [Avoiding Path Traversal (OWASP)](https://www.owasp.org/index.php/File_System#Path_traversal)

**High : 2**

|  |  |
| --- | --- |
| **Classification** | [Information](about:blank#description) |
| **Resource** | **/jw/** |
| **Risk** | High |

**Request**

[GET /jw/](about:blank)

**Resource Content**

|  |
| --- |
| JSESSIONID=E6DE9713FE07767DE2F649E3C4156332; Path=/jw; HttpOnly |

**Impact**

* Cookies can be exposed to network eavesdroppers.
* Session cookies are authentication credentials; attackers who obtain them can get unauthorized access to affected web applications.

**Remediation**

* When creating the cookie in the code, set the secure flag to true.

**References**

Some additional links with relevant information published by third-parties:

* [Secure Flag](https://www.owasp.org/index.php/SecureFlag)
* [HttpOnly OWASP Reference](https://www.owasp.org/index.php/HttpOnly)

**MEDIUM : 1**

|  |  |
| --- | --- |
| **Classification** | [Information](about:blank#description) |
| **Resource** | **/jw/web/userview/appcenter/v/\_/home** |
| **Risk** | Medium |

**Request**

[GET /jw/web/userview/appcenter/v/\_/home](about:blank)

**Resource Content**

|  |
| --- |
| /lib/material-design-iconic-font/fonts/Material-Design-Iconic-Font.woff |

**Impact**

* It has detected what may be absolute filesystem paths in scanned content.
* Disclosure of these paths reveals information about the filesystem layout.
* This information can be sensitive, its disclosure can increase the chances of success for other attacks.

**Remediation**

* Absolute paths are often found in error output.
* Both the system administrators and developers should be made aware, as the problem may be due to an application error or server misconfiguration.
* Error output containing sensitive information such as absolute system paths should not be sent to remote clients on production servers.
* This output should be sent to another output stream, such as an error log.

**References**

Some additional links with relevant information published by third-parties:

* [Information Leakage (OWASP)](https://www.owasp.org/index.php/Information_Leakage)

**mEDIUM: 2**

|  |  |
| --- | --- |
| **Classification** | [Information](about:blank#description) |
| **Resource** | **/jw/nosuchpage123** |
| **Risk** | Medium |

**Request**

[GET /jw/nosuchpage123](about:blank)

**Resource Content**

|  |
| --- |
| /home/style.css |

**Impact**

* It has detected what may be absolute filesystem paths in scanned content.
* Disclosure of these paths reveals information about the filesystem layout.
* This information can be sensitive, its disclosure can increase the chances of success for other attacks.

**Remediation**

* Absolute paths are often found in error output.
* Both the system administrators and developers should be made aware, as the problem may be due to an application error or server misconfiguration.
* Error output containing sensitive information such as absolute system paths should not be sent to remote clients on production servers.
* This output should be sent to another output stream, such as an error log.

**References**

Some additional links with relevant information published by third-parties:

* [Information Leakage (OWASP)](https://www.owasp.org/index.php/Information_Leakage)

**mEDIUM: 3**

|  |  |
| --- | --- |
| **Classification** | [Information](about:blank#description) |
| **Resource** | **/** |
| **Risk** | Medium |

**Request**

[GET /](about:blank)

**Resource Content**

|  |
| --- |
| /apache/tomcat/tree/ |

**Impact**

* It has detected what may be absolute filesystem paths in scanned content.
* Disclosure of these paths reveals information about the filesystem layout.
* This information can be sensitive, its disclosure can increase the chances of success for other attacks.

**Remediation**

* Absolute paths are often found in error output.
* Both the system administrators and developers should be made aware, as the problem may be due to an application error or server misconfiguration.
* Error output containing sensitive information such as absolute system paths should not be sent to remote clients on production servers.
* This output should be sent to another output stream, such as an error log.

**References**

Some additional links with relevant information published by third-parties:

* [Information Leakage (OWASP](https://www.owasp.org/index.php/Information_Leakage)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Document\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*