SECURITY ASSESSMENT

<<REPORT NAME>>

Submitted to: Development Department, Udajuicer
Security Analyst: Marvellous Onuma-Kalu

Date of Testing: 01-06-2024
Date of Report Delivery: 01-08-2024

Table of Contents

Contents

Security Engagement Summary

Engagement Overview

The Udajuicer’s legacy web application had experienced a cyber attack. This has been mitigated and the system was recovered and secured. However, the Development Team at Udajuicer has requested a vulnerability assessment of the current state of the organization’s legacy web-application.

The goal of the engagement is to understand what security risk the web-application is posing to the organization in its current state, identify any potential areas of concern, provide solutions for reducing risks and fix vulnerabilities.

Regular vulnerability assessments and prioritization would help Udajuicer allocate resources efficiently to address the most critical security concerns first.

The engagement will be completed by the Information Security Department. All testing activities were performed on the staging environment provided by the customer and completely isolated from the production data.

.

Scope

Vulnerability assessment can identify potential problems and weaknesses in an environment. The Udajuicer’s web-application hosts the company’s ecommerce solution, and is highly accessible from the internet. The website is used to handle customer requests making it a target for attackers.

This report summarizes what the Information Security Department believes are the most important issues to address in the application. A number of issues grouped by risk factors are identified. The risk ratings are to assist in prioritizing remediation efforts.

Risk Analysis

High

These issues identify conditions that could directly result in the compromise or unauthorized access of a network, system, application or sensitive Information. It suggests that there is a significant likelihood that the identified vulnerability could be exploited by malicious actors, which if exploited. could result in severe consequences such as data breaches, service disruptions, unauthorized access, financial losses, or damage to an organization's reputation. Examples of High-Risk issues include remote execution of commands, known buffer overflows, unauthorized access and disclosure of sensitive information.

Medium

The issues identify conditions that do not immediately or directly result in the compromise or unauthorized access of a network, system, application or sensible information, but do provide a capability or information that could, in combination with other capabilities or information, result in the compromise or unauthorized access of a network, application or information.

Examples of Medium-Risk issues include directory browsing, partial access to files on the system, disclosure of security mechanisms and unauthorized use of services.

Low

These issues identify conditions that do not immediately or directly result in the compromise or unauthorized access of a network, system, application or sensitive information, but do provide information that could be used in combination with other information to gain insights into how to compromise or gain unauthorized access.to a network, system, application or information.

Informational:

These issues, also known as information leakage, appear when a website unintentionally reveals sensitive information to its users.

Identified issues by risk factor:

Risk Level Number of Alerts

High 1

Medium 2

Low 1

Informational 3

Executive Risk Analysis

The website shows high, medium and low risk vulnerabilities. The web-application shows misconfiguration which could be used by an attacker to access data that is available in an unauthenticated manner.

Name	**Risk Level**	**Number of Instances**
Cloud Metadata Potentially Exposed	High	
Content Security Policy (CSP) Header Not Set	Medium	81
Cross-Domain Misconfiguration	Medium	100

Cross-Domain JavaScript Source File Inclusion	Low	146
Information Disclosure - Suspicious Comments	Informational	6
Modern Web Application	Informational	74
User Agent Fuzzer	Informational	24

Exploration of these flaws is inevitable. An attack can have a severe impact on the business. The Information Security Department strongly recommends remediating all issues detected to mitigate against the possible risk of a sensitive data compromise.

Executive Recommendation

- Issue a maintenance window to perform the necessary fixes. This should be done not to affect business operations.
- Send a message informing customers of the downtime. The appropriate personnel (public relations department) should do this.
- Create backups to restore the system in case of failure.
- Conduct follow-up scanning

Significant Vulnerability Summary

The Information Security Department has identified critical vulnerabilities that could lead to full compromise of the system, as well as several medium and low severity issues were found, which should be addressed promptly.

High Risk Vulnerabilities

- 90034 - Cloud Metadata Potentially Exposed

Medium Risk Vulnerabilities

- 10038 - Content Security Policy (CSP) Header Not Set
- 10098 - Cross-Domain Misconfiguration

Low Risk Vulnerabilities

- 10017 - Cross-Domain JavaScript Source File Inclusion

Informational Risk Vulnerabilities

- 10027 - Information Disclosure - Suspicious Comments
- 10109 - Modern Web Application
- 10104 - User Agent Fuzze

Others from Nmap Vulners

- CVE-2020-15778* 6.8 https://vulners.com/cve/CVE-2020-15778
- PRION:CVE-2020-12062* 5.0 https://vulners.com/prion/PRION:CVE-2020-12062
- PRION:CVE-2020-20012* 5.0 https://vulners.com/prion/PRION:CVE-2020-20012
- CVE-2020-12062 5.0 https://vulners.com/cve/CVE-2020-12062
- PRION:CVE-2021-28041 4.6 https://vulners.com/prion/PRION:CVE-2021-28041
- CVE-2021-28041 4.6 https://vulners.com/cve/CVE-2021-28041
- PRION:CVE-2020-15778 4.4 https://vulners.com/prion/PRION:CVE-2020-15778
- CVE-2021-41617 4.4 https://vulners.com/cve/CVE-2021-41617
- PRION:CVE-2020-14145 4.3 https://vulners.com/prion/PRION:CVE-2020-14145
- CVE-2020-14145 4.3 https://vulners.com/cve/CVE-2020-14145
- CVE-2016-20012 4.3 https://vulners.com/cve/CVE-2016-20012
- PRION:CVE-2021-41617 3.5 https://vulners.com/prion/PRION:CVE-2021-41617
- PRION:CVE-2021-36368 2.6 https://vulners.com/prion/PRION:CVE-2021-36368
- CVE-2021-36368 2.6 https://vulners.com/cve/CVE-2021-36368

Significant Vulnerability Detail

Cloud Metadata Potentially Exposed
Risk level : High

Description:

The cloud metadata attack attempts to abuse a misconfigured NGINX server in order to access the instance metadata maintained by cloud service providers such as AWS, Azure, and GCP. All of these providers provide metadata via an internal unroutable IP address “169.254.169.254”-this can be exposed by incorrectly configured NGINX servers and accessed by using IP address in the Host header field.

Information:

Based on the successful response status code cloud metadata may have been returned in the response. Check the response data to see if any cloud metadata has been returned.

Response:

```
<!doctype html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>OWASP Juice Shop</title>
 <meta name="description" content="Probably the most modern and sophisticated insecure web application">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
 <link rel="stylesheet" type="text/css"
href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css" />
 <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
 <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
 <script>
   window.addEventListener("load", function(){
     window.cookieconsent.initialise({
       "palette": {
         "popup": { "background": "#546e7a", "text": "#ffffff" },
         "button": { "background": "#558b2f", "text": "#ffffff" }
       },
       "theme": "classic",
       "position": "bottom-right",
       "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking
experience.", "dismiss": "Me want it!", "link": "But me wait!", "href":
"https://www.youtube.com/watch?v=9PnbKL3wuH4" }
     })});
 </script>
<link rel="stylesheet" href="styles.css"></head>
<body class="mat-app-background bluegrey-lightgreen-theme">
 <app-root></app-root>
<script src="runtime-es2015.js" type="module"></script><script src="runtime-es5.js" nomodule
defer></script><script src="polyfills-es5.js" nomodule defer></script><script src="polyfills-es2015.js"
type="module"></script><script src="vendor-es2015.js" type="module"></script><script src="vendor-es5.js"
nomodule defer></script><script src="main-es2015.js" type="module"></script><script src="main-es5.js"
nomodule defer></script></body>
</html>
```

The meta data returned include information that would allow an attacker to completely compromise the system.

Impact:

Exposing cloud metadata can have serious security implications, as it may lead to unauthorized access, data breaches, and compromise of sensitive information. Cloud metadata typically includes details about the underlying infrastructure and configuration settings, and if improperly exposed, it could provide valuable information to attackers. Some potential impacts of cloud metadata exposure include: Unauthorized Access and Control, Sensitive Information Leakage, Data Breaches, Elevated Risk of Insider Threats, compromised Identity and Access Management (IAM), Service Disruption and Data Loss, Increased Attack Surface.

Solution:

- Do not trust any user data in NGINX configs. In this case it is probably the use of the $host variable which is set from the ‘Host’ header and can be controlled by an attacker.
- Limit access to metadata to only authorized users and applications.
- Encrypt sensitive data, such as authentication credentials, stored in metadata to prevent unauthorized access even if the metadata is exposed.
- Conduct regular audits of cloud configurations and metadata settings

Reference: https://nginx.com/blog/trust-no-one-perils-of-trusting-user-input/

Content Security Policy (CSP) Header Not Set
Risk level - Medium

Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Impact:

CSP serves as a crucial security mechanism to mitigate certain types of web based attacks. Not setting a Content Security Policy CSP header can have several security implications including: Increased risk of

Cross-Site Scripting (XSS) attacks, Data Injection and Man-in-the-Middle (MitM) attacks, increased risk of Clickjacking, and more.

Solution:

- Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

- Regularly review and update your CSP policy based on changes to your application and emerging security threats.

References:

https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
http://www.w3.org/TR/CSP/
http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html
http://www.html5rocks.com/en/tutorials/security/content-security-policy/
http://caniuse.com/#feat=contentsecuritypolicy
http://content-security-policy.com/

Cross-Domain Misconfiguration
Risk level - Medium

Description: Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

Information: The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

Solution:

-Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).

- Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

Reference:

https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy

---

10017 - Cross-Domain JavaScript Source File Inclusion
Risk Level - Low

Description: The page includes one or more script files from a third-party domain.

Solution: Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

10027 - Information Disclosure - Suspicious Comments
Risk level - Informational

Description: The response appears to contain suspicious comments which may help an attacker.

Note: Matches made within script blocks or files are against the entire content not only comments.

Solution: Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

URL: http://192.168.0.20:3000/polyfills-es5.js

10109 - Modern Web Application
Risk Level - Informational

Information: The application appears to be a modern web application. If you need to explore it automatically, then the Ajax Spider may well be more effective than the standard one.

Solution: This is an informational alert so no changes are required

---

10104 - User Agent Fuzzer
Risk level- Informational

Information: Check for difference in response based on fuzzed User Agent (eg mobile sites, access as a Search Engine Crawler). Compare the response status code and the hashcode of the response body with the original response.

Reference: https://Owasp.org/wstg

Vulnerabilities discovered with NMAP-Vulners / tcp/22:

Description: Common Vulnerabilities and Exposures

ID: CVE-2020-15778

Information: scp in OpenSSH through 8.3p1 allows command injection in the scp.c toremote function, as demonstrated by backtick characters in the destination argument. NOTE: the vendor reportedly has stated that they intentionally omit validation of "anomalous argument transfers" because that could "stand a great chance of breaking existing workflows."

ID: CVE-2020-12062

Information: The scp client in OpenSSH 8.2 incorrectly sends duplicate responses to the server upon a utimes system call failure, which allows a malicious unprivileged user on the remote server to overwrite arbitrary files in the client's download directory by creating a crafted subdirectory anywhere on the remote server. The victim must use the command scp -rp to download a file hierarchy containing, anywhere inside, this crafted subdirectory. NOTE: the vendor points out that "this attack can achieve no more than a hostile peer is already able to achieve within the scp protocol" and "utimes does not fail under normal circumstances."

ID: CVE-2016-20012

Information: OpenSSH through 8.7 allows remote attackers, who have a suspicion that a certain combination of username and public key is known to an SSH server, to test whether this suspicion is correct. This occurs because a challenge is sent only when that combination could be valid for a login session. NOTE: the vendor does not recognize user enumeration as a vulnerability for this product.

ID: CVE-2021-28041

Description: ssh-agent in OpenSSH before 8.5 has a double free that may be relevant in a few less-common scenarios, such as unconstrained agent-socket access on a legacy operating system, or the forwarding of an agent to an attacker-controlled host.

ID: CVE-2021-41617

Description: sshd in OpenSSH 6.2 through 8.x before 8.8, when certain non-default configurations are used, allows privilege escalation because supplemental groups are not initialized as expected. Helper programs for AuthorizedKeysCommand and AuthorizedPrincipalsCommand may run with privileges associated with group memberships of the sshd process, if the configuration specifies running the command as a different user.

ID: CVE-2020-14145

Description: The client side in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached by the client). NOTE: Some reports state that 8.5 and 8.6 are also affected.

ID: CVE-2021-36836

Description: An issue was discovered in OpenSSH before 8.9. If a client is using public-key authentication with agent forwarding but without -oLogLevel=verbose, and an attacker has silently modified the server to support the None authentication option, then the user cannot determine whether FIDO authentication is going to confirm that the user wishes to connect to that server, or that the user wishes to allow that server to connect to a different server on the user’s behalf. Note: that vendor’s position is “this is not an authentication bypass, since nothing is being bypassed.

Methodology

The findings and recommendations outlined in this report are based on vulnerability scans performed against the Udajuicer’s web application.

Assessment Toolset Selection

1. OWASP ZAP is an open-source web application security scanner.

2. Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing.

3. Vulners is an Nmap NSE script, using some well-known services to provide info on vulnerabilities.

4. NVD (National Vulnerability Database) is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, product names, and impact metrics.

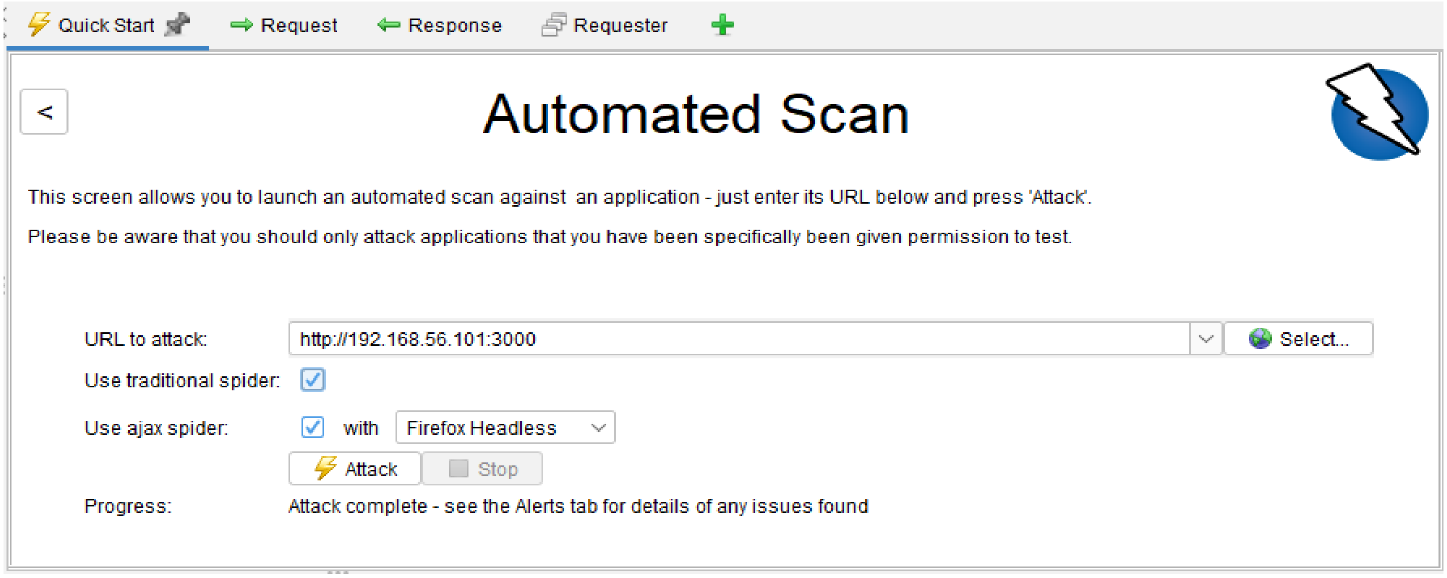
Assessment Methodology Detail

1. OWASP ZAP

This web application vulnerability scanning tool helps developers and security professionals detect and find vulnerabilities in web applications including compromised authentication, exposure of sensitive data, security misconfigurations, SQL injection, cross-site scripting (XSS), insecure deserialization, and components with known vulnerabilities.

Setup:

> Start ZAP and click the Quick Start tab of the Workspace Window.
> Click the large Automated Scan button.
> In the Attack URL text box, enter the full URL of the web application you want to attack.
> Click the Attack
> Select either **Use traditional spider**, **Use ajax spider**, or both



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://192.168.56.101:3000 Select...

Use traditional spider: ☑

Use ajax spider: ☑ with Firefox Headless

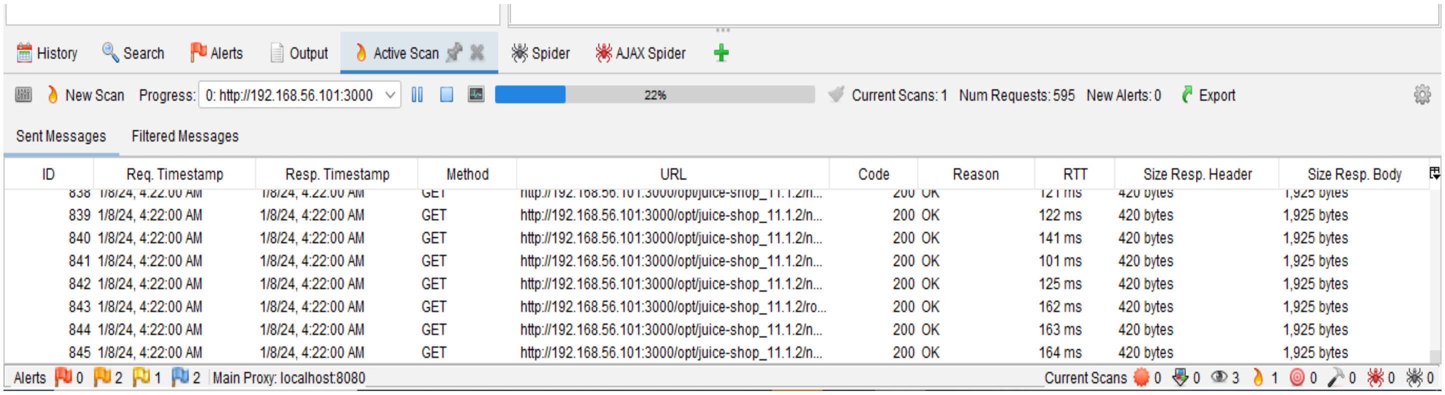
Attack Stop

Progress: Attack complete - see the Alerts tab for details of any issues found

ZAP will proceed to crawl the web application with its spider and passively scan each page it finds. Then ZAP will use the active scanner to attack all of the discovered pages, functionality, and parameters.

Progress bar

In the screenshot below, a number of scans are occurring, including a web page *spider* that *crawls* the pages.



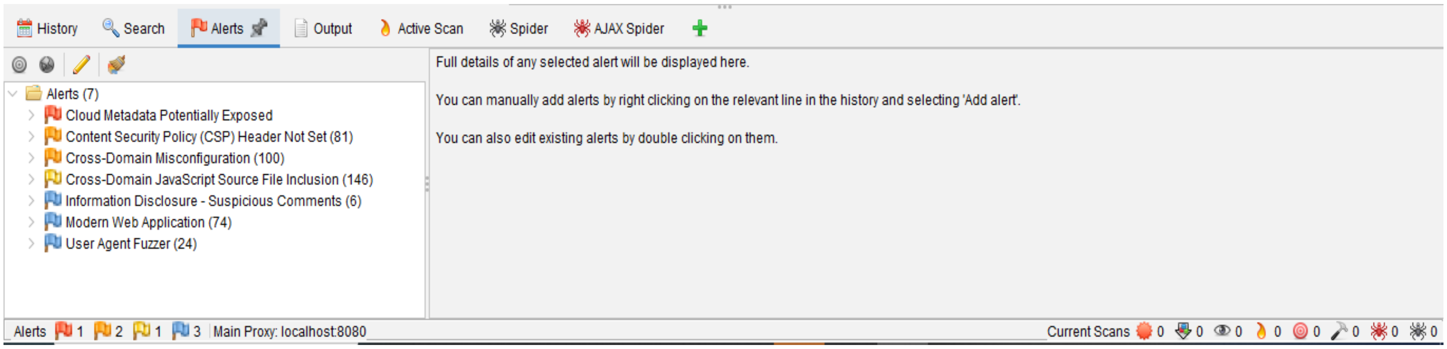
Progress: 0: http://192.168.56.101:3000 Current Scans: 1 Num Requests: 595 New Alerts: 0

ID	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
838	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	121 ms	420 bytes	1,925 bytes
839	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	122 ms	420 bytes	1,925 bytes
840	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	141 ms	420 bytes	1,925 bytes
841	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	101 ms	420 bytes	1,925 bytes
842	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	125 ms	420 bytes	1,925 bytes
843	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/ro...	200	OK	162 ms	420 bytes	1,925 bytes
844	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	163 ms	420 bytes	1,925 bytes
845	1/8/24, 4:22:00 AM	1/8/24, 4:22:00 AM	GET	http://192.168.56.101:3000/opt/juice-shop_11.1.2/n...	200	OK	164 ms	420 bytes	1,925 bytes

Alerts 0 2 1 2 Main Proxy: localhost:8080 Current Scans 0 0 3 1 0 0 0 0

Completed scan result

The results are displayed in the screenshot below.



- Alerts (7)
  - Cloud Metadata Potentially Exposed
  - Content Security Policy (CSP) Header Not Set (81)
  - Cross-Domain Misconfiguration (100)
  - Cross-Domain JavaScript Source File Inclusion (146)
  - Information Disclosure - Suspicious Comments (6)
  - Modern Web Application (74)
  - User Agent Fuzzer (24)

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts 1 2 1 3 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

This scanner found 7 vulnerabilities, 1 - high, 2 - medium, 1 - low and 3 - informational

Nmap Vulners NSE script

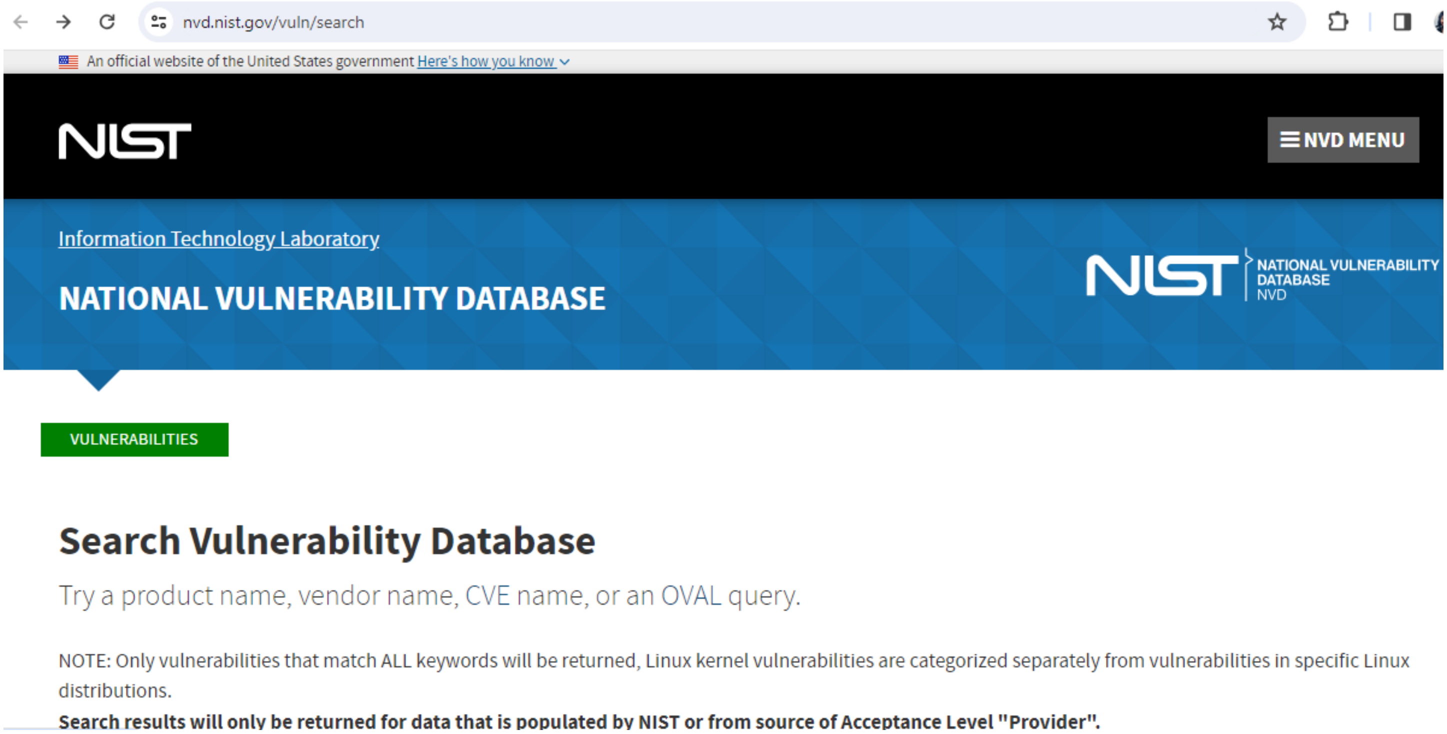
Nmap-vulners queries the Vulners exploit database every time we use the NSE script.

Input:

output:

NVD database

We identify 7 vulnerabilities with their corresponding CVEs. To obtain a description of the CVE, we will search the NVD (National Vulnerability Database). The screenshot below shows the NVD search page.


nvd.nist.gov/vuln/search
An official website of the United States government Here's how you know
NIST
NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NIST NATIONAL VULNERABILITY DATABASE NVD
VULNERABILITIES

Search Vulnerability Database

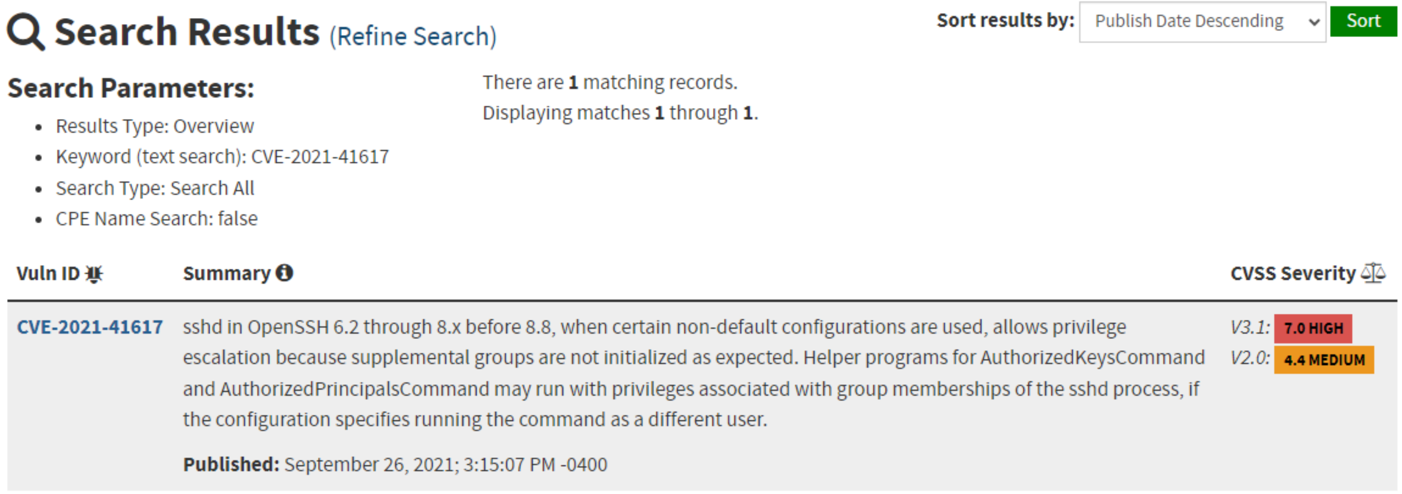
Try a product name, vendor name, CVE name, or an OVAL query.

NOTE: Only vulnerabilities that match ALL keywords will be returned, Linux kernel vulnerabilities are categorized separately from vulnerabilities in specific Linux distributions.

Search results will only be returned for data that is populated by NIST or from source of Acceptance Level "Provider".

CVE-2021-41617

The screenshot below shows the result for CVE-2021-41617 search



Search Results (Refine Search)

Sort results by: Publish Date Descending Sort

Search Parameters:

- Results Type: Overview
- Keyword (text search): CVE-2021-41617
- Search Type: Search All
- CPE Name Search: false

There are **1** matching records.
Displaying matches **1** through **1**.

Vuln ID	Summary	CVSS Severity
CVE-2021-41617	sshd in OpenSSH 6.2 through 8.x before 8.8, when certain non-default configurations are used, allows privilege escalation because supplemental groups are not initialized as expected. Helper programs for AuthorizedKeysCommand and AuthorizedPrincipalsCommand may run with privileges associated with group memberships of the sshd process, if the configuration specifies running the command as a different user. **Published:** September 26, 2021; 3:15:07 PM -0400	*V3.1:* 7.0 HIGH *V2.0:* 4.4 MEDIUM

All vulnerabilities have related references, definitions and severity which complete full information of any known bulletins. Visit https://vulners.com/ for detailed information.

Conclusion

The Information Security Department completed the vulnerability testing of the web application. This testing was based on the current technologies and known threats as of the date of this document. A detailed analysis and description of all the security issues discovered during the vulnerability exercise were provided in this report.

Note: Technology and risk continue to evolve with time. This implies that, the vulnerabilities associated with the operation of systems described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities, will also change.