

Security Research Project

Jacob Phillips

May 2025

Table of Contents

1.0 - Executive Summary.....	4
2.0 - Scope	5
2.1 - Eligibility.....	5
3.0 - Technical Briefing.....	6
3.1 - Methodology Overview	6
3.2 - Tools Used	6
3.3 - Internal Testing Approach.....	7
3.4 - Existing Test Coverage.....	7
4.0 – Findings.....	8
4.1 – Nmap Scan Results	8
4.1.1 – Nmap localhost scans	8
4.1.1.1 – Version 3.4.3 scan results.....	8
4.1.1.2 – Version 3.4.5 scan results.....	10
4.1.2 – Nmap container scans	11
4.1.2.1 – Dev-Server container.....	11
4.1.2.1.1 – Version 3.4.3 scan results.....	11
4.1.2.1.2 – Version 3.4.5 scan results.....	12
4.1.2.1.3 – Analysis	12
4.1.2.2 – Redis container.....	13
4.1.2.2.1 – Version 3.4.3 scan results.....	13
4.1.2.2.2 – Version 3.4.5 scan results.....	14
4.1.2.2.3 – Analysis	14
4.1.2.3 – Angular-Build container	15
4.1.2.3.1 – Version 3.4.3 scan results.....	15
4.1.2.3.2 – Version 3.4.5 scan results.....	15
4.1.2.3.3 - Analysis	15

4.1.2.4 – Cloud-Datastore container	16
4.1.2.4.1 – Version 3.4.3 scan results.....	16
4.1.2.4.2 – Version 3.4.5 scan results.....	16
4.1.2.4.3 - Analysis	17
4.1.2.5 – Webpack-Compiler container.....	17
4.1.2.5.1 – Version 3.4.3 scan results.....	17
4.1.2.5.2 – Version 3.4.5 scan results.....	18
4.1.2.5.3 - Analysis	18
4.1.2.6 – Firebase-Emulator container	18
4.1.2.6.1 – Version 3.4.3 scan results.....	18
4.1.2.6.2 – Version 3.4.5 scan results.....	19
4.1.2.6.3 - Analysis	19
5.0 – Nessus Essentials.....	20
5.1 – Nessus scan results (localhost).....	20
5.2 – Nessus can results (containers)	21
6.0 – Syft.....	22
7.0 – Grype	23
8.0 – Semgrep	37
9.0 – Analysis	45
9.1 – Network and Service Exposure.....	45
9.2 – Dependency-Level Vulnerabilities and Software Composition Risks	46
9.3 – Static Code Analysis and Application Security	46

1.0 - Executive Summary

This report presents a comparative security analysis of an open-source web application using the LASTAS (Learning About Security Through Antiquated Software) methodology. The objective was to assess changes between two versions (v3.4.3 and v3.4.5) to identify whether updates introduced or remediated any vulnerabilities.

Multiple layers of analysis were conducted using industry-recognized tools across the network, application, and source-code levels. The results highlight several security improvements, including the removal of undocumented or unnecessary exposed services in the latest version, and reinforce the importance of layered scanning approaches for containerized applications.

2.0 - Scope

This research was conducted as part of the Learning About Security Through Antiquated Software (LASTAS) methodology, which aims to teach and develop security skills through the assessment of outdated software versions. The selected project is an open-source, web-based educational platform designed to make quality education freely accessible to all learners.

The scope of this research included the following:

- Security assessment of two versions of the application: version 3.4.3 and version 3.4.5.
- Analysis focused on identifying changes in security posture between versions using both dynamic and static techniques.
- Use of professional-grade tools including Nmap, Nessus Essentials, Syft, Gripe, and Semgrep to identify vulnerabilities at multiple levels (network, container, dependency, and source code).

2.1 - Eligibility

The selected software meets the following eligibility criteria:

- Released under an OSI-approved license (Apache License 2.0).
- Actively developed and large-scale; not a toy, practice, or homework repository.
- Purpose aligns with ethical and academic standards:
 - No copyright violations.
 - Does not facilitate unethical behaviour.
 - Not an offensive, exploitative, or malicious tool.
- I am not affiliated with this project in a development or contributor capacity.

3.0 - Technical Briefing

3.1 - Methodology Overview

The LASTAS methodology involves analysing outdated versions of open-source projects to understand how security vulnerabilities may arise or be resolved over time. This report follows LASTAS by:

- Cloning the project's GitHub repository and switching between tags v3.4.3 and v3.4.5
- Comparing these versions using static and dynamic analysis tools
- Identifying and assessing exposed ports, running services, and vulnerable dependencies

3.2 - Tools Used

Each tool reflected in Table 1 served a specific purpose to ensure coverage across potential attack surfaces including network exposure, dependency vulnerabilities, and insecure coding practices.

Table 1 - Security tools used to conduct the assessment

Tool	Purpose
Nmap	Port scanning and service enumeration (both localhost and Docker internal IPs)
Nessus Essentials	Vulnerability scanning of running services and containers
Syft	Software Bill of Materials (SBOM) generation
Grype	Software Composition Analysis (SCA) for CVEs in dependencies
Semgrep	Static code analysis using OWASP Top 10 security rules

3.3 - Internal Testing Approach

Nmap scanning was conducted in two layers:

- First, by scanning the services and ports exposed to the host machine (localhost).
- Second, by directly scanning the Docker containers using their internal IP addresses (e.g., 172.18.0.x).

This two-layered approach was used to identify differences between what was externally accessible versus what was only visible within the containerised network.

Once these nmap scans were completed, Nessus Essentials was used to perform a full vulnerability assessment against both the localhost and the identified container IPs. This helped uncover known CVEs, misconfigurations, and exposed services on both layers.

Next, Syft was implemented to generate a Software Bill of Materials (SBOM) for the application. That SBOM was then analysed using Gype to conduct Software Composition Analysis (SCA), which identified potential vulnerabilities in third-party dependencies.

Finally, Semgrep was deployed to perform a Static Application Security Test (SAST) against the codebase using the OWASP Top 10 ruleset. This helped flag insecure code patterns, hardcoded secrets, and other potential security issues at the source-code level.

3.4 - Existing Test Coverage

Reviewing the project's GitHub Actions workflows displays that the project already includes a wide variety of tests including:

- Backend unit tests
- Frontend unit tests
- End-to-end unit tests

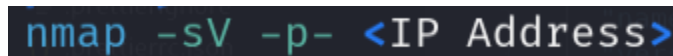
However, none of these involve security focused scanning using workflows such as Dependabot, CodeQL, or other static or dynamic vulnerability scanners. Therefore, the testing reflected in the report did not overlap with the project's existing CI/CD pipelines.

4.0 – Findings

Each section below presents and compares the findings from testing version 3.4.3 and version 3.4.5, based on the tools described in the Methodology section.

4.1 – Nmap Scan Results

The command used for nmap scanning is reflected in Figure 1 below.



```
nmap -sV -p- <IP Address>
```

Figure 1 - Nmap scan used for LASTAS method

The '-sV' option is for service version detection, where nmap attempts to determine the version of the services running on any ports it finds to be open. The '-p-' option tells nmap to scan all 65,535 TCP ports, not just the default top 1,000 most common ports. Together, this is used to perform a comprehensive port and service version scan using nmap.

4.1.1 – Nmap localhost scans

4.1.1.1 – Version 3.4.3 scan results

As reflected in Figure 2, nmap detected various ports and services to be open, mostly around HTTP. These ports and services are broken down in Table 2 below.


```

[phillj22@kali]~$ nmap -sV -p- 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:16 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000050s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
4000/tcp   open  http         Node.js Express framework
8000/tcp   open  http         CherryPy wsgiserver
8080/tcp   open  tcpwrapped
8181/tcp   open  http         CherryPy wsgiserver
9099/tcp   open  http         Node.js Express framework
42065/tcp  open  unknown
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port42065-TCP:V=7.95%I=7%O=5/15%Time=682630A0%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,65,"HTTP/1.0\x20400\x20Bad\x20Request\r\nContent-Type:\x20te
SF:xt/html;\x20charset=UTF-8\r\n\r\nWebSockets\x20request\x20was\x20expect
SF:red\r\n")%r(HTTPOptions,65,"HTTP/1.0\x20400\x20Bad\x20Request\r\nCon

```

Figure 2 - Nmap localhost scan for version v3.4.3

Table 2 - Nmap localhost results for v3.4.3

Port	Protocol	State	Service	Version
4000	TCP	Open	HTTP	Node.js Express Framework
8000	TCP	Open	HTTP	CherryPy wsgiserver
8080	TCP	Open	TCPWRAPPED	(None)
8181	TCP	Open	HTTP	CherryPy wsgiserver
9099	TCP	Open	HTTP	Node.js Express Framework
42065	TCP	Open	UNKNOWN	Unrecognised

The nmap scan of localhost as denoted by Figure 2 and Table 2, reported a number of open ports which are primarily used for web services.

Ports 4000, 8000, 8181, and 9099 were revealed to be running either Node.js Express or CherryPy servers, which are used for web app development.

Port 8080 with the service as TCPWRAPPED suggests that nmap could not identify the service due to TCP wrappers. This is most likely also an HTTP-based service, but it is obfuscated.

Port 42065 shows that it is open but the service also could not be identified. This is probably either a specific arbitrarily chosen port for development purposes, a specific port for a service the app uses, or possibly a listener. However, this could be risky as it could expose the application to backdoors and is not documented from my search of the repository.

4.1.1.2 – Version 3.4.5 scan results

As reflected in Figure 3 and Table 3, nmap reported back with the same ports as reflected in Table 2 with the exception of port 42065 no longer being open.

```
(phillj22@kali)-[~]
$ nmap -sV -p- 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:20 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000060s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
4000/tcp  open  http         Node.js Express framework
8000/tcp  open  http         CherryPy wsgiserver
8080/tcp  open  tcpwrapped
8181/tcp  open  http         CherryPy wsgiserver
9099/tcp  open  http         Node.js Express framework

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.82 seconds
```

Figure 3 - Nmap localhost scan v3.4.5

Table 3 - Nmap localhost results v3.4.5

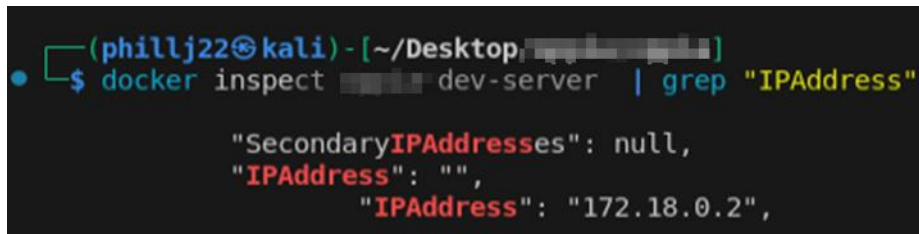
Port	Protocol	State	Service	Version
4000	TCP	Open	HTTP	Node.js Express Framework
8000	TCP	Open	HTTP	CherryPy wsgiserver
8080	TCP	Open	TCPWRAPPED	(None)

8181	TCP	Open	HTTP	CherryPy wsgiserver
9099	TCP	Open	HTTP	Node.js Express Framework

There are no significant concerns from the open ports detected on the local machine, as all identified services relate to web communication expected in a development environment. The disappearance of port 42065 in version 3.4.5 is a positive sign as it could be indicative of a security improvement or removal of an undocumented service

4.1.2 – Nmap container scans

Each Docker container is also assigned a private IP address, in which these were also tested using nmap. Each IP address was found using the command displayed below in Figure 4.



```
(phillj22@kali) - [~/Desktop, [redacted]]
$ docker inspect [redacted] dev-server | grep "IPAddress"
  "SecondaryIPAddresses": null,
  "IPAddress": "",
  "IPAddress": "172.18.0.2",
```

Figure 4 - Inspecting a Docker container for its IP address

4.1.2.1 – Dev-Server container

The IP address assigned to the container was 172.18.0.2 for both versions of the application.

4.1.2.1.1 – Version 3.4.3 scan results

```

(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:30 EDT
Nmap scan report for 172.18.0.2
Host is up (0.0000070s latency).
Not shown: 65528 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
8000/tcp  open  http    CherryPy wsgiserver
8181/tcp  open  http    CherryPy wsgiserver
15243/tcp open  http    Unicorn
17376/tcp open  http    Unicorn
18441/tcp open  http    Unicorn
19165/tcp open  http    Unicorn
19353/tcp open  http    Unicorn
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.85 seconds

```

Figure 5 - Nmap scan of the dev-server container v3.4.3

4.1.2.1.2 – Version 3.4.5 scan results

```

(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:26 EDT
Nmap scan report for 172.18.0.2
Host is up (0.0000080s latency).
Not shown: 65528 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
8000/tcp  open  http    CherryPy wsgiserver
8181/tcp  open  http    CherryPy wsgiserver
36503/tcp open  http    Unicorn
38651/tcp open  http    Unicorn
44939/tcp open  http    Unicorn
45767/tcp open  http    Unicorn
46445/tcp open  http    Unicorn
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.80 seconds

```

Figure 6 - Nmap scan of the dev-server container v3.4.5

4.1.2.1.3 – Analysis

Table 4 - Dev-server nmap scan comparison table

Present in v3.4.3?	Present in v3.4.5?	Port	Protocol	State	Service	Version
Yes	Yes	8000	TCP	Open	HTTP	CherryPy wsgiserver

Yes	Yes	8181	TCP	Open	HTTP	CherryPy wsgiserver
Yes	No	15243	TCP	Open	HTTP	Gunicorn
Yes	No	17376	TCP	Open	HTTP	Gunicorn
Yes	No	18441	TCP	Open	HTTP	Gunicorn
Yes	No	19165	TCP	Open	HTTP	Gunicorn
Yes	No	19353	TCP	Open	HTTP	Gunicorn
No	Yes	36503	TCP	Open	HTTP	Gunicorn
No	Yes	38651	TCP	Open	HTTP	Gunicorn
No	Yes	44939	TCP	Open	HTTP	Gunicorn
No	Yes	45767	TCP	Open	HTTP	Gunicorn
No	Yes	46445	TCP	Open	HTTP	Gunicorn

The two main web ports (8000 and 8181) are present in both versions. The handful of Gunicorn ports present change port numbers in both versions, but this seems to be the process manager just grabbing fresh numbers on each run. Because this is only accessible within the Docker network, it isn't too much of a concern. However, it could lead to confusion or possible accidental exposure.

4.1.2.2 – Redis container

The IP address assigned to the container was 172.18.0.6 in v3.4.3 and 172.18.0.3 in v3.4.5.

4.1.2.2.1 – Version 3.4.3 scan results

```

(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:34 EDT
Nmap scan report for 172.18.0.6
Host is up (0.0000070s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
6379/tcp  open  redis    Redis key-value store 6.2.4
MAC Address:  (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.42 seconds

```

Figure 7 - Nmap scan of redis container v3.4.3

4.1.2.2.2 – Version 3.4.5 scan results

```

(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.3
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:26 EDT
Nmap scan report for 172.18.0.3
Host is up (0.0000070s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
6379/tcp  open  redis    Redis key-value store 6.2.4
MAC Address:  (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.32 seconds

```

Figure 8 - Nmap scan of redis container v3.4.5

4.1.2.2.3 – Analysis

Table 5 - Redis nmap scan comparison table

Present in v3.4.3?	Present in v3.4.5?	Port	Protocol	State	Service	Version
Yes	Yes	6379	TCP	Open	redis	Redis key-value store 6.2.4

Both versions are running Redis version 6.2.4, which from the official end-of-life (EOL) schedule, went EOL 28th February 2025.

4.1.2.3 – Angular-Build container

The IP address assigned to the container was 172.18.0.4 in both versions of the application.

4.1.2.3.1 – Version 3.4.3 scan results

```
(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:32 EDT
Nmap scan report for 172.18.0.4
Host is up (0.0000070s latency).
All 65535 scanned ports on 172.18.0.4 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.37 seconds
```

Figure 9 - Nmap scan of angular-build container v3.4.3

4.1.2.3.2 – Version 3.4.5 scan results

```
(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:27 EDT
Nmap scan report for 172.18.0.4
Host is up (0.0000080s latency).
All 65535 scanned ports on 172.18.0.4 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.41 seconds
```

Figure 10 - Nmap scan of angular-build container v3.4.5

4.1.2.3.3 - Analysis

Table 6 - Angular-build nmap scan comparison table

Present in v3.4.3?	Present in v3.4.5?	Port	Protocol	State	Service	Version
Yes	Yes	N/A	N/A	Closed	N/A	N/A

All scanned ports are closed in both versions, showing that the build only container never listens on the network which is the ideal behaviour for a compile time image.

4.1.2.4 – Cloud-Datastore container

The IP address assigned to the container was 172.18.0.5 in v3.4.3 and 172.18.0.7 in v3.4.5.

4.1.2.4.1 – Version 3.4.3 scan results

```
(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:33 EDT
Nmap scan report for 172.18.0.5
Host is up (0.0000070s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
8089/tcp  open  rtsp
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF:Port8089-TCP:V=7.95%I=7%D=5/15%Time=6826418C%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,29,"HTTP/1.0\x20200\x200K\r\ncontent-length:\x203\r\n\r\nOk\r\n
SF:")%r(HTTPOptions,26,"HTTP/1.0\x20200\x200K\r\ncontent-length:\x200\r\n
SF:\r\n")%r(RTSPRequest,26,"RTSP/1.0\x20200\x200K\r\ncontent-length:\x200
SF:\r\n\r\n")%r(FourOhFourRequest,38,"HTTP/1.0\x20404\x20Not\x20Found\r\n
SF:content-length:\x2010\r\n\r\nNot\x20Found\n");
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.95 seconds
```

Figure 11 - Nmap scan of cloud-datastore container v3.4.3

4.1.2.4.2 – Version 3.4.5 scan results

```
(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:28 EDT
Nmap scan report for 172.18.0.7
Host is up (0.0000080s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
8089/tcp  open  rtsp
46023/tcp open  unknown
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
-----
NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)
SF:Port8089-TCP:V=7.95%I=7%D=5/16%Time=6827ADF8%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,29,"HTTP/1.0\x20200\x200K\r\ncontent-length:\x203\r\n\r\nOk\r\n
SF:")%r(HTTPOptions,26,"HTTP/1.0\x20200\x200K\r\ncontent-length:\x200\r\n
SF:\r\n")%r(RTSPRequest,26,"RTSP/1.0\x20200\x200K\r\ncontent-length:\x200
```

Figure 12 - Nmap scan of cloud-datastore container v3.4.5

4.1.2.4.3 - Analysis

Table 7 - Cloud-datastore nmap scan comparison table

Present in v3.4.3?	Present in v3.4.5?	Port	Protocol	State	Service	Version
Yes	Yes	8089	TCP	Open	RTSP	(None)
Yes	Yes	Unrecognised	Unrecognised	Unrecognised	Unrecognised	Unrecognised
No	Yes	15243	TCP	Open	Unrecognised	Gunicorn

The datastore emulator's usual port 8089 is open in both versions which is expected. Nmap also labels some traffic as "Unrecognised" but this is just most probably due to the emulator. Version 3.4.5 adds port 15243 running Gunicorn which will need to be verified that this helper service is required and to make sure that it is kept internal only.

4.1.2.5 – Webpack-Compiler container

The IP address assigned to the container was 172.18.0.3 in v3.4.3 and 172.18.0.5 in v3.4.5.

4.1.2.5.1 – Version 3.4.3 scan results

```
(phillj22@kali)~$ nmap -sV -p- 172.18.0.3
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:31 EDT
Nmap scan report for 172.18.0.3
Host is up (0.0000070s latency).
All 65535 scanned ports on 172.18.0.3 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds
```

Figure 13 - Nmap scan of webpack-compiler container v3.4.3

4.1.2.5.2 – Version 3.4.5 scan results

```
(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:27 EDT
Nmap scan report for 172.18.0.5
Host is up (0.0000080s latency).
All 65535 scanned ports on 172.18.0.5 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: [REDACTED] (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.36 seconds
```

Figure 14 - Nmap scan of webpack-compiler container v3.4.5

4.1.2.5.3 - Analysis

Table 8 - webpack-compiler nmap scan comparison table

Present in v3.4.3?	Present in v3.4.5?	Port	Protocol	State	Service	Version
Yes	Yes	N/A	N/A	Closed	N/A	N/A

Like the Angular build container, every port remains closed across versions, confirming no runtime services are exposed and is ideal for a compiler image.

4.1.2.6 – Firebase-Emulator container

The IP address assigned to the container was 172.18.0.8 in both versions of the application.

4.1.2.6.1 – Version 3.4.3 scan results

```

(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-15 15:35 EDT
Nmap scan report for 172.18.0.8
Host is up (0.0000080s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
4000/tcp  open  http    Node.js Express framework
9099/tcp  open  http    Node.js Express framework
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.66 seconds

```

Figure 15 - Nmap scan of firebase-emulator container v3.4.3

4.1.2.6.2 – Version 3.4.5 scan results

```

(phillj22@kali)-[~]
$ nmap -sV -p- 172.18.0.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 17:29 EDT
Nmap scan report for 172.18.0.8
Host is up (0.0000080s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
4000/tcp  open  http    Node.js Express framework
9099/tcp  open  http    Node.js Express framework
MAC Address: (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.47 seconds

```

Figure 16 - Nmap scan of firebase-emulator container v3.4.5

4.1.2.6.3 - Analysis

Present in v3.4.3?	Present in v3.4.5?	Port	Protocol	State	Service	Version
Yes	Yes	4000	TCP	Open	HTTP	Node.js Express Framework
Yes	Yes	9099	TCP	Open	HTTP	Node.js Express Framework

Ports 4000 and 9099 are still open in both versions for the Firebase emulator. This is normal for local development, and because of this nothing new or risky is introduced here.

5.0 – Nessus Essentials

The Nessus scans were conducted against localhost (127.0.0.1) and the six Docker containers covered in section 4.1 – Nmap scan results.

5.1 – Nessus scan results (localhost)

Table 9 - Nessus scan results for v3.4.3 localhost

Severity	CVSS	Name	Count
High	7.7	Node.js Multiple Vulnerabilities	1
High	7.4	OpenJDK Multiple Vulnerabilities	1
Medium	6.5	Node.js Module node-tar < 6.2.1 DoS	70
Medium	6.5	SSL Certificate Cannot be Trusted	1
Medium	6.1	JQuery 1.2 < 3.5.0 Multiple XSS	1
Medium	4.3	Web Application Potentially Vulnerable to Clickjacking	1

Table 10 - Nessus scan results for v3.4.5 localhost

Severity	CVSS	Name	Count
High	7.5	Ruby RACK < 2.2.1.4 / 3.0.16 / 3.1.14 DoS vulnerability	1
High	7.4	OpenJDK Multiple Vulnerabilities	1
Medium	6.5	Node.js Module node-tar < 6.2.1 DoS	70
Medium	6.5	SSL Certificate Cannot be Trusted	1
Medium	6.2	Node.js 20.x / 22.x / 23.x / 24.x Multiple Vulnerabilities	1
Medium	6.1	JQuery 1.2 < 3.5.0 Multiple XSS	1
Medium	4.3	Web Application Potentially Vulnerable to Clickjacking	1

The vulnerabilities that are struck through are false positives as they are reporting back to configurations on the local environment (Kali Linux) and are not associated with the application in any way.

Both versions are still weighed down by lots of Medium level warnings, especially the 70 identical node tar DoS findings. The fix is straightforward as it just requires upgrading the modules to version 6.2.1 or newer and they will disappear.

This also goes for the JQuery cross site scripting (XSS) vulnerability, which can be resolved with updating JQuery to atleast version 3.5.0.

Something easy to overlook is the clickjacking warning, which shows up in both scans at Medium severity. Because the application doesn't send an X Frame Options header, an attacker could frame the site and trick a user into clicking on hidden buttons. Implementing an X Frame Options: DENY header will resolve this issue.

5.2 – Nessus scan results (containers)

All of the Docker containers reported back with only two vulnerabilities, with the exception of the Redis container, in which it reported back with three.

Table 11 - Nessus scan summary for Docker containers

Severity	CVSS	Name	Count
High	9.8	Redis Server Unprotected by Password Authentication	1
Medium	6.5	IP Forwarding Enabled	1
Low	2.1	ICMP Timestamp Request Remote Date Disclosure	1

The only high vulnerability reported on both versions of the Docker containers is that the Redis server running is not protected by password authentication, suggesting the 'requirepass' directive in the redis.conf configuration file.

The IP Forwarding Enabled isn't an issue as this is required for the containers to talk to the outside world and is normal for these setups.

The ICMP Timestamp Low vulnerability also is not an issue as this is the local development stack and not the public-facing production suite – however, it may be worth bringing it up in a pull request as it could be used by an attacker to measure clock skew.

6.0 – Syft

The latest version (v3.4.5) reported back with 100 less packages than the older version (v3.4.3). The latest version and the older version both reported back with 51 executables. However, the latest version reported back with 5 less file digests and file metadata locations.

```
(phillj22@kali) - [~/Desktop]
$ syft dir: . -o json > sbom.json
✓ Indexed file system
✓ Cataloged contents
  ✓ Packages [2,929 packages]
  ✓ Executables [51 executables]
  ✓ File digests [80 files]
  ✓ File metadata [80 locations]
[0000] WARN no explicit name and version provided for directory source, deriving artifact ID from the given path (which is not ideal)
```

Figure 17 - Syft results for v3.4.3

Table 12 - Syft results table for v3.4.3

Packages	Executables	File digests	File metadata
2929	51	80	80

```
(phillj22@kali) - [~/Desktop]
$ syft dir: . -o json > sbom.json
✓ Indexed file system
✓ Cataloged contents
  ✓ Packages [2,892 packages]
  ✓ File digests [75 files]
  ✓ File metadata [75 locations]
  ✓ Executables [51 executables]
[0001] WARN no explicit name and version provided for directory source, deriving artifact ID from the given path (which is not ideal)
A newer version of syft is available for download: 1.25.0 (installed version is 1.24.0)
```

Figure 18 - Syft results for v3.4.5

Table 13 - Syft results table for v3.4.5

Packages	Executables	File digests	File metadata
2892	51	75	75

7.0 – Grype

Overall, the latest version (3.4.5) has three less high-risk vulnerabilities present on the dependency level. These dependency-level vulnerabilities are reflected in Table 16.

```
(phillj22@kali)-[~/Desktop, ██████████]
$ grype sbom:sbom.json
✓ Vulnerability DB [updated]
✓ Scanned for vulnerabilities [152 vulnerability matches]
└─ by severity: 12 critical, 61 high, 57 medium, 22 low, 0 negligible
└─ by status: 138 fixed, 14 not-fixed, 0 ignored
```

Figure 19 - Grype results for v3.4.3

Table 14 - Grype results table for v3.4.3

Critical	High	Medium	Low
12	61	57	22

```
(phillj22@kali)-[~/Desktop, ██████████]
$ grype sbom:sbom.json
✓ Vulnerability DB [updated]
✓ Scanned for vulnerabilities [150 vulnerability matches]
└─ by severity: 12 critical, 59 high, 57 medium, 22 low, 0 negligible
└─ by status: 136 fixed, 14 not-fixed, 0 ignored
```

Figure 20 - Grype results for v3.4.5

Table 15 - Grype results for table v3.4.5

Critical	High	Medium	Low
12	59	57	22

Table 16 - Gripe dependency vulnerability results

NAME	INSTALLED	FIXED-IN	TYPE	VULNERABILITY	SEVERITY	EPSS%	RISK
pyarrow	7.0.0	14.0.1	python	GHSA-5wvp-7f3h-6wmm	Critical	99.4	82.1
json5	0.5.1	1.0.2	npm	GHSA-9c47-m6qq-7p4h	High	97.12	29.2
json5	1.0.1	1.0.2	npm	GHSA-9c47-m6qq-7p4h	High	97.12	29.2
json5	2.2.1	2.2.2	npm	GHSA-9c47-m6qq-7p4h	High	97.12	29.2
loader-utils	0.2.17	1.4.1	npm	GHSA-76p3-8jx3-jpfq	Critical	94.32	14.8
loader-utils	2.0.0	2.0.3	npm	GHSA-76p3-8jx3-jpfq	Critical	94.32	14.8
tough-cookie	2.5.0	4.1.3	npm	GHSA-72xf-g2v4-qvf3	Medium	89.36	3
tough-cookie	3.0.1	4.1.3	npm	GHSA-72xf-g2v4-qvf3	Medium	89.36	3
loader-utils	2.0.0	2.0.4	npm	GHSA-hhq3-ff78-jv3g	High	84.6	1.9
lodash.set	4.3.2		npm	GHSA-p6mc-m468-83gw	High	84.31	1.8

protobufjs	6.11.2	6.11.4	npm	GHSA-h755-8qp9-cq85	Critical	81.14	1.6
protobufjs	6.11.3	6.11.4	npm	GHSA-h755-8qp9-cq85	Critical	81.14	1.6
protobufjs	7.1.2	7.2.5	npm	GHSA-h755-8qp9-cq85	Critical	81.14	1.6
terser	5.5.1	5.14.2	npm	GHSA-4wf5-vphf-c2xc	High	82.67	1.5
json-ptr	2.2.0	3.0.0	npm	GHSA-8gwj-8hxc-285w	Medium	81.55	0.9
degenerator	2.2.0	3.0.1	npm	GHSA-9j49-mfvp-vmhm	High	75.79	0.8
pac-resolver	4.2.0	5.0.0	npm	GHSA-9j49-mfvp-vmhm	High	75.79	0.8
ua-parser-js	0.7.32	0.7.33	npm	GHSA-fhg7-m89q-25r3	High	75.51	0.7
ua-parser-js	1.0.32	1.0.33	npm	GHSA-fhg7-m89q-25r3	High	75.51	0.7
certifi	2023.7.22	2024.7.4	python	GHSA-248v-346w-9cwc	Low	82.4	0.6
es5-ext	0.10.62	0.10.63	npm	GHSA-4gmj-3p3h-gm8h	Low	82.04	0.6

loader-utils	2.0.0	2.0.4	npm	GHSA-3rfm-jhwj-7488	High	70.29	0.5
angular	1.8.3	BLANK	npm	GHSA-4w4v-5hc9-xrr2	High	68.87	0.5
get-func-name	2.0.0	2.0.1	npm	GHSA-4q6p-r6v2-jvc5	High	68.75	0.5
angular	1.8.3		npm	GHSA-prc3-vjfx-vhm9	Medium	71.26	0.4
elliptic	6.5.4	6.5.7	npm	GHSA-49q7-c7j4-3p7m	Low	74.74	0.4
got	9.6.0	11.8.5	npm	GHSA-pfrx-2q88-qc97	Medium	71.28	0.4
jpeg-js	0.3.7	0.4.0	npm	GHSA-w7q9-p3jq-fmhm	Medium	70.38	0.4
braces	2.3.2	3.0.3	npm	GHSA-grv7-fg5c-xmjg	High	63.39	0.4
braces	3.0.2	3.0.3	npm	GHSA-grv7-fg5c-xmjg	High	63.39	0.4
engine.io	6.2.1	6.4.2	npm	GHSA-q9mw-68c2-j6m5	Medium	68.56	0.4
pillow	10.1.0	10.2.0	python	GHSA-3f63-hfp8-52jq	Critical	60.17	0.3
marked	0.7.0	4.0.10	npm	GHSA-rrrm-qjm4-v8hf	High	63.17	0.3

marked	0.7.0	4.0.10	npm	GHSA-5v2h-r2cx-5xgj	High	62.88	0.3
minimatch	3.0.4	3.0.5	npm	GHSA-f8q6-p94x-37v3	High	62.8	0.3
ip	1.1.8		npm	GHSA-2p57-rm9w-gvfp	High	61.96	0.3
ip	2.0.0		npm	GHSA-2p57-rm9w-gvfp	High	61.96	0.3
request	2.88.2		npm	GHSA-p8p7-x288-28g6	Medium	68.39	0.3
body-parser	1.20.1	1.20.3	npm	GHSA-qwcr-r2fm-qrc7	High	59.96	0.3
tar-fs	2.1.1	2.1.2	npm	GHSA-pq67-2www-3xjx	High	59.59	0.3
node-forge	0.10.0	1.0.0	npm	GHSA-8fr3-hfg3-gpgp	Medium	66.05	0.3
ejs	3.1.8	3.1.10	npm	GHSA-ghr5-ch3p-vcr6	Medium	66.86	0.2
protobufjs	6.11.2	6.11.3	npm	GHSA-g954-5hwp-pp24	High	55.22	0.2
browserify-sign	4.2.1	4.2.2	npm	GHSA-x9w5-v3q2-3rhw	High	55.18	0.2

decode-uri-component	0.2.0	0.2.1	npm	GHSA-w573-4hg7-7wgq	High	54.73	0.2
ws	6.2.2	6.2.3	npm	GHSA-3h5v-q93c-6h6q	High	54.37	0.2
ws	7.5.9	7.5.10	npm	GHSA-3h5v-q93c-6h6q	High	54.37	0.2
ws	8.2.3	8.17.1	npm	GHSA-3h5v-q93c-6h6q	High	54.37	0.2
ws	8.5.0	8.17.1	npm	GHSA-3h5v-q93c-6h6q	High	54.37	0.2
ajv	5.5.2	6.12.3	npm	GHSA-v88g-cgmw-v5xw	Medium	62.56	0.2
angular	1.8.3		npm	GHSA-qwqh-hm9m-p5hr	Medium	62.72	0.2
angular	1.8.3		npm	GHSA-m2h2-264f-f486	Medium	62.52	0.2
@grpc/grpc-js	1.6.12	1.8.22	npm	GHSA-7v5v-9h63-cj86	Medium	60	0.2
@grpc/grpc-js	1.7.3	1.8.22	npm	GHSA-7v5v-9h63-cj86	Medium	60	0.2
micromatch	3.1.10	4.0.8	npm	GHSA-952p-6rrq-rcjv	Medium	57.73	0.2
micromatch	4.0.5	4.0.8	npm		Medium	57.73	0.2

				GHSA-952p-6rrq-rcjv			
codemirror	5.17.0	5.58.2	npm	GHSA-4gw3-8f77-f72c	Medium	55.9	0.2
semver	5.7.1	5.7.2	npm	GHSA-c2qf-rxjj-qgqw	High	45.83	0.2
semver	6.3.0	6.3.1	npm	GHSA-c2qf-rxjj-qgqw	High	45.83	0.2
semver	7.3.4	7.5.2	npm	GHSA-c2qf-rxjj-qgqw	High	45.83	0.2
semver	7.3.8	7.5.2	npm	GHSA-c2qf-rxjj-qgqw	High	45.83	0.2
waitress	2.1.2	3.0.1	python	GHSA-3f84-rpwh-47g6	High	42.27	0.1
xml2js	0.4.23	0.5.0	npm	GHSA-776f-qx25-q3cc	Medium	48.45	0.1
http-proxy-middleware	0.19.1	2.0.7	npm	GHSA-c7qv-q95q-8v27	High	38.7	0.1
angular	1.8.3		npm	GHSA-2qqx-w9hr-q5gx	Medium	46.88	0.1
angular	1.8.3		npm	GHSA-2vrf-hf26-jrp5	Medium	46.88	0.1
webpack-dev-middleware	3.7.2	5.3.4	npm	GHSA-wr3j-pwj9-hqq6	High	37.98	0.1

webpack-dev-middleware	3.7.3	5.3.4	npm	GHSA-wr3j-pwj9-hqq6	High	37.98	0.1
tar	4.4.19	6.2.1	npm	GHSA-f5x3-32g6-xq36	Medium	43.1	0.1
tar	6.1.12	6.2.1	npm	GHSA-f5x3-32g6-xq36	Medium	43.1	0.1
pymongo	3.13.0	4.6.3	python	GHSA-m87m-mmvp-v9qm	Medium	46.92	0.1
jose	2.0.6	2.0.7	npm	GHSA-hhhv-q57g-882q	Medium	43.74	0.1
socket.io-parser	4.2.1	4.2.3	npm	GHSA-cqmj-92xf-r6r9	Medium	38.17	< 0.1
cross-spawn	4.0.2	6.0.6	npm	GHSA-3xgq-45jj-v275	High	34.19	< 0.1
cross-spawn	5.1.0	6.0.6	npm	GHSA-3xgq-45jj-v275	High	34.19	< 0.1
cross-spawn	6.0.5	6.0.6	npm	GHSA-3xgq-45jj-v275	High	34.19	< 0.1
cross-spawn	7.0.3	7.0.5	npm	GHSA-3xgq-45jj-v275	High	34.19	< 0.1
ip	1.1.8	1.1.9	npm	GHSA-78xj-cgh5-2h22	Low	52.71	< 0.1
ip	2.0.0	2.0.1	npm	GHSA-78xj-cgh5-2h22	Low	52.71	< 0.1
node-forge	0.10.0	1.3.0	npm		High	31.71	

				GHSA-x4jg-mjrx-434g			< 0.1
http-cache-semantics	4.1.0	4.1.1	npm	GHSA-rc47-6667-2j5j	High	31.56	< 0.1
nth-check	1.0.2	2.0.1	npm	GHSA-rp65-9cf3-cjxr	High	31.5	< 0.1
node-forge	0.10.0	1.3.0	npm	GHSA-cfm4-qjh2-4765	High	29.89	< 0.1
path-to-regexp	0.1.7	0.1.10	npm	GHSA-9wv6-86v2-598j	High	29.74	< 0.1
path-to-regexp	1.8.0	1.9.0	npm	GHSA-9wv6-86v2-598j	High	29.74	< 0.1
jpeg-js	0.3.7	0.4.4	npm	GHSA-xvf7-4v9q-58w6	High	29.13	< 0.1
follow-redirects	1.15.2	1.15.6	npm	GHSA-cxjh-pqwp-8mfp	Medium	32.5	< 0.1
follow-redirects	1.15.3	1.15.6	npm	GHSA-cxjh-pqwp-8mfp	Medium	32.5	< 0.1
jsonwebtoken	8.5.1	9.0.0	npm	GHSA-8cf7-32gw-wr33	High	26.67	< 0.1
pillow	10.1.0	10.3.0	python	GHSA-44wm-f244-xhp3	High	27.91	< 0.1
axios	0.27.2	0.28.0	npm	GHSA-wf5p-g6vw-rhxx	Medium	31.59	< 0.1

basic-auth-connect	1.0.0	1.1.0	npm	GHSA-7p89-p6hx-q4fw	High	26.6	< 0.1
webob	1.8.7	1.8.8	python	GHSA-mg3v-6m49-jhp3	Medium	31.91	< 0.1
@babel/traverse	7.20.5	7.23.2	npm	GHSA-67hx-6x53-jw92	Critical	21.34	< 0.1
babel-traverse	6.26.0		npm	GHSA-67hx-6x53-jw92	Critical	21.34	< 0.1
socket.io	4.5.4		npm	GHSA-m9gf-397r-hwpg	Low	34.19	< 0.1
express	4.18.2	4.19.2	npm	GHSA-rv95-896h-c2vc	Medium	30.52	< 0.1
postcss	5.2.18	8.4.31	npm	GHSA-7fh5-64p2-3v2j	Medium	31.02	< 0.1
postcss	6.0.23	8.4.31	npm	GHSA-7fh5-64p2-3v2j	Medium	31.02	< 0.1
postcss	7.0.39	8.4.31	npm	GHSA-7fh5-64p2-3v2j	Medium	31.02	< 0.1
postcss	8.2.15	8.4.31	npm	GHSA-7fh5-64p2-3v2j	Medium	31.02	< 0.1
postcss	8.4.19	8.4.31	npm	GHSA-7fh5-64p2-3v2j	Medium	31.02	< 0.1
angular	1.8.3		npm	GHSA-m9gf-397r-hwpg	Low	34.19	< 0.1

elliptic	6.5.4	6.5.7	npm	GHSA-977x-g7h5-7qgw	Low	32.69	< 0.1
gunicorn	20.1.0	22.0.0	python	GHSA-w3h3-4rj7-4ph4	High	13.87	< 0.1
follow-redirects	1.15.2	1.15.4	npm	GHSA-jchw-25xp-jwwc	Medium	18.44	< 0.1
follow-redirects	1.15.3	1.15.4	npm	GHSA-jchw-25xp-jwwc	Medium	18.44	< 0.1
node-forge	0.10.0	1.3.0	npm	GHSA-2r2c-g63r-vccr	Medium	19.29	< 0.1
path-to-regexp	0.1.7	0.1.12	npm	GHSA-rhx6-c78j-4q9w	High	11.97	< 0.1
elliptic	6.5.4	6.5.7	npm	GHSA-f7q4-pwc6-w24p	Low	21.19	< 0.1
bootstrap	4.6.2	5.0.0	npm	GHSA-vc8w-jr9v-vj7f	Medium	14.71	< 0.1
elliptic	6.5.4	6.5.6	npm	GHSA-434g-2637-qmqr	Low	19.38	< 0.1
postcss	5.2.18	7.0.36	npm	GHSA-566m-qj78-rww5	Medium	15.27	< 0.1
postcss	6.0.23	7.0.36	npm	GHSA-566m-qj78-rww5	Medium	15.27	< 0.1

jsonwebtoken	8.5.1	9.0.0	npm	GHSA-hjrf-2m68-5959	Medium	12.33	< 0.1
firebase	8.10.1	10.9.0	npm	GHSA-3wf4-68gx-mp8	Medium	8.38	< 0.1
serialize-javascript	6.0.0	6.0.2	npm	GHSA-76p7-773f-r4q5	Medium	8.16	< 0.1
waitress	2.1.2	3.0.1	python	GHSA-9298-4cf8-g4wj	Critical	2.94	< 0.1
axios	0.27.2	0.30.0	npm	GHSA-jr5f-v2jv-69x6	High	3.91	< 0.1
urllib3	1.26.18	1.26.19	python	GHSA-34jh-p97f-mpxf	Medium	8.18	< 0.1
rollup	2.38.4	2.79.2	npm	GHSA-gcx4-mw62-g8wm	High	3.93	< 0.1
elliptic	6.5.4	6.6.0	npm	GHSA-fc9h-whq2-v747	Low	9.38	< 0.1
firebase-tools	9.6.0	13.6.0	npm	GHSA-rcm2-22f3-pqv3	Low	15.43	< 0.1
gunicorn	20.1.0	23.0.0	python	GHSA-hc5x-x2vx-497g	High	2.89	< 0.1
jsonwebtoken	8.5.1	9.0.0	npm	GHSA-qwph-4952-7xr6	Medium	4.17	< 0.1
send	0.18.0	0.19.0	npm	GHSA-m6fv-jmcg-4jfg	Low	6.01	< 0.1
express	4.18.2	4.20.0	npm	GHSA-qw6h-vgh9-j6wx	Low	5.82	< 0.1

@babel/helpers	7.20.6	7.26.10	npm	GHSA-968p-4wvh-cqc8	Medium	3.45	< 0.1
@babel/runtime	7.12.5	7.26.10	npm	GHSA-968p-4wvh-cqc8	Medium	3.45	< 0.1
@babel/runtime	7.20.6	7.26.10	npm	GHSA-968p-4wvh-cqc8	Medium	3.45	< 0.1
serve-static	1.15.0	1.16.0	npm	GHSA-cm22-4g7w-348p	Low	5.63	< 0.1
word-wrap	1.2.3	1.2.4	npm	GHSA-j8xg-fqg3-53r7	Medium	3.84	< 0.1
angular	1.8.3		npm	GHSA-mqm9-c95h-x2p6	Low	5.45	< 0.1
cookie	0.3.1	0.7.0	npm	GHSA-pxg6-pf52-xh8x	Low	7.83	< 0.1
cookie	0.4.2	0.7.0	npm	GHSA-pxg6-pf52-xh8x	Low	7.83	< 0.1
cookie	0.5.0	0.7.0	npm	GHSA-pxg6-pf52-xh8x	Low	7.83	< 0.1
requests	2.31.0	2.32.0	python	GHSA-9wx4-h78v-vm56	Medium	2.86	< 0.1
angular	1.8.3		npm	GHSA-mqm9-c95h-x2p6	Low	5.45	< 0.1

@google-cloud/firestore	4.15.1	6.1.0	npm	GHSA-4g6q-77j7-vvjc	Medium	2.61	< 0.1
nanoid	3.3.1	3.3.8	npm	GHSA-mwcw-c2x4-8c55	Medium	0.71	< 0.1
nanoid	3.3.4	3.3.8	npm	GHSA-mwcw-c2x4-8c55	Medium	0.71	< 0.1
elliptic	6.5.4	6.6.1	npm	GHSA-vjh7-7g9h-fjfh	Critical	N/A	N/A
d3-color	1.4.1	3.1.0	npm	GHSA-36jr-mh4h-2g58	High	N/A	N/A
d3-color	2.0.0	3.1.0	npm	GHSA-36jr-mh4h-2g58	High	N/A	N/A
node-forge	0.10.0	1.0.0	npm	GHSA-5rrq-pxf6-6jx5	Low	N/A	N/A
node-forge	0.10.0	1.0.0	npm	GHSA-gf8q-jrpm-jvxq	Low	N/A	N/A

While both versions contain the same number of critical vulnerabilities, version 3.4.5 shows a slight improvement by reducing the number of high-severity issues by three. This dependency-level breakdown highlights reoccurring vulnerable packages such as json5, loader-utils, and protobufjs, which are present in this table above, in multiple versions.

8.0 – Semgrep

The latest version (3.4.5) was found to have 5 less findings when compared to the older version. These findings are broken down below.

```
Scan Summary
✓ Scan completed successfully.
• Findings: 23 (23 blocking)
• Rules run: 1047
• Targets scanned: 5532
• Parsed lines: ~99.9%
• Scan skipped:
  ◦ Files larger than files 1.0 MB: 5
  ◦ Files matching .semgrepignore patterns: 358
• Scan was limited to files tracked by git
• For a detailed list of skipped files and lines, run semgrep with the --verbose flag
Ran 1047 rules on 5532 files: 23 findings.
```

Figure 21 - Semgrep scan summary for v3.4.3

```
Scan Summary
✓ Scan completed successfully.
• Findings: 18 (18 blocking)
• Rules run: 1047
• Targets scanned: 5463
• Parsed lines: ~99.9%
• Scan skipped:
  ◦ Files larger than files 1.0 MB: 5
  ◦ Files matching .semgrepignore patterns: 383
• Scan was limited to files tracked by git
• For a detailed list of skipped files and lines, run semgrep with the --verbose flag
Ran 1047 rules on 5463 files: 18 findings.
```

Figure 22 - Semgrep scan summary for v3.4.5

Vulnerability Location:	Intentionally withheld
Status:	Error
Semgrep Keyword:	python.lang.security.deserialization.avoid-pyyaml-load.avoid-pyyaml-load

Link:	https://sg.run/we9Y
Description:	Detected a possible YAML deserialization vulnerability. <code>`yaml.unsafe_load`</code> , <code>`yaml.Loader`</code> , <code>`yaml.CLoader`</code> , and <code>`yaml.UnsafeLoader`</code> are all known to be unsafe methods of deserializing YAML. An attacker with control over the YAML input could create special YAML input that allows the attacker to run arbitrary Python code. This would allow the attacker to steal files, download and install malware, or otherwise take over the machine. Use <code>`yaml.safe_load`</code> or <code>`yaml.SafeLoader`</code> instead.

Vulnerability Location:	Intentionally withheld
Status:	Error
Semgrep Keyword:	python.lang.security.audit.subprocess-shell-true.subprocess-shell-true
Link:	https://sg.run/J92w
Description:	Found 'subprocess' function '\$FUNC' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to

	execute commands. Use 'shell=False' instead.
--	--

Vulnerability Location:	Intentionally withheld
Status:	Error
Semgrep Keyword:	python.lang.security.audit.subprocess-shell-true.subprocess-shell-true
Link:	https://sg.run/J92w
Description:	Found 'subprocess' function '\$FUNC' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to execute commands. Use 'shell=False' instead.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	python.lang.security.insecure-hash-algorithms-md5.insecure-hash-algorithm-md5
Link:	https://sg.run/vYrY
Description:	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a

	cryptographic signature. Use SHA256 or SHA3 instead.
--	--

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	python.lang.security.insecure-hash-algorithms-md5.insecure-hash-algorithm-md5
Link:	https://sg.run/vYrY
Description:	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	python.lang.security.insecure-hash-algorithms.insecure-hash-algorithm-sha1
Link:	https://sg.run/ydYx
Description:	Detected SHA1 hash algorithm which is considered insecure. SHA1 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	typescript.angular.security.audit.angular-domsanitizer.angular-bypassecuritytrust
Link:	https://sg.run/KWxP
Description:	Detected the use of `\$_TRUST`. This can introduce a Cross-Site-Scripting (XSS) vulnerability if this comes from user-provided input. If you have to use `\$_TRUST`, ensure it does not come from user-input or use the appropriate prevention mechanism e.g. input validation or sanitization depending on the context.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	typescript.angular.security.audit.angular-domsanitizer.angular-bypassecuritytrust
Link:	https://sg.run/KWxP
Description:	Detected the use of `\$_TRUST`. This can introduce a Cross-Site-Scripting (XSS) vulnerability if this comes from user-provided input. If you have to use `\$_TRUST`, ensure it does not come from user-input or use the appropriate prevention mechanism e.g. input

	validation or sanitization depending on the context.
--	--

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	javascript.browser.security.wildcard-postmessage-configuration.wildcard-postmessage-configuration
Link:	https://sg.run/PJ4p
Description:	The target origin of the window.postMessage() API is set to "*". This could allow for information disclosure due to the possibility of any origin allowed to receive the message.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	html.security.plaintext-http-link.plaintext-http-link
Link:	https://sg.run/RA5q
Description:	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	html.security.plaintext-http-link.plaintext-http-link
Link:	https://sg.run/RA5q
Description:	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.

Vulnerability Location:	Intentionally withheld
Status:	Warning
Semgrep Keyword:	python.lang.security.insecure-hash-algorithms-md5.insecure-hash-algorithm-md5
Link:	https://sg.run/vYrY
Description:	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.

Vulnerability Location:	Intentionally withheld
Status:	Warning

Semgrep Keyword:	python.lang.security.insecure-hash-algorithms-md5.insecure-hash-algorithm-md5
Link:	https://sg.run/vYrY
Description:	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.

Vulnerability Location:	Intentionally withheld
Status:	Error
Semgrep Keyword:	yaml.github-actions.security.run-shell-injection.run-shell-injection
Link:	https://sg.run/pkzk
Description:	Using variable interpolation `\${{...}}` with `github` context data in a `run:` step could allow an attacker to inject their own code into the runner. This would allow them to steal secrets and code. `github` context data can have arbitrary user input and should be treated as untrusted. Instead, use an intermediate environment variable with `env:` to store the data and use the environment variable in the `run:` script. Be sure to use double-quotes the environment variable, like this: "\$ENVVAR".

Vulnerability Location:	Intentionally withheld
Status:	Error
Semgrep Keyword:	yaml.github-actions.security.run-shell-injection.run-shell-injection
Link:	https://sg.run/pkzk
Description:	Using variable interpolation `\${{...}}` with `github` context data in a `run:` step could allow an attacker to inject their own code into the runner. This would allow them to steal secrets and code. `github` context data can have arbitrary user input and should be treated as untrusted. Instead, use an intermediate environment variable with `env:` to store the data and use the environment variable in the `run:` script. Be sure to use double-quotes the environment variable, like this: "\$ENVVAR".

9.0 – Analysis

This security assessment of the application's versions 3.4.3 and 3.4.5 provide some pretty comprehensive insights into the current state of the application's security, especially those involving exposed services, third-party dependencies, and other potential weaknesses in the code itself.

9.1 – Network and Service Exposure

The nmap scans of the localhost and containerized environments show multiple open HTTP related ports across both versions, which is normal for a web app running different microservices and therefore shouldn't be a concern necessarily.

Obviously, these would be running HTTPS (443) in production to encrypt web traffic, but for development this is normal. The only port that could be a concern is the undocumented port (42065) as it could be used as potential backdoor.

9.2 – Dependency-Level Vulnerabilities and Software Composition

Risks

The Redis instance running version 6.2.4 across both versions is pretty significant as this version is End-of-Life (EOL) as of February 28th, 2025. This could introduce security risks as it is no longer receiving frequent patches or security updates. This is important as Redis could be a pretty large target for attackers, and because of this should perhaps raise a PR or report this security issue following their vulnerability disclosure process. This also goes in hand to what Nessus reported, as the Redis instance is not protected by password authentication, and could also suggest implementing the 'requirepass' directive in the redis.conf configuration file.

The Software Bill of Materials (SBOM) generated using Syft, followed by Gype scans, reflects a huge amount of dependency-level vulnerabilities across third-party libraries. These outdated packages have CVEs against them, which could allow remote code execution (RCE), privilege escalation, and data leakage.

The clickjacking vulnerability reflected in the Nessus report indicates that the application is missing some important HTTP security headers like X-Frame-Options.

9.3 – Static Code Analysis and Application Security

Semgrep scans flagged issues related to unsafe functions, inadequate input validation, and weak encryption methods being used. These code-level vulnerabilities are directly aligned with the OWASP Top 10 risks and represent valid threat vectors that could lead to attackers using injection attacks to gain unauthorized access or data breaches.