OT Smart Home Security Analyzer (OTSHSA)

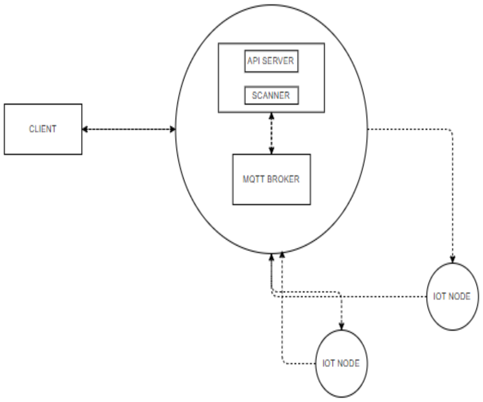
TABLE OF CONTENTS

Introduction - OT Smart Home Security Analyzer (OTSHSA)

OT Smart Home Security Analyzer (OTSHSA) is an IoT security controller/analyzer that detects the presence of new IoT nodes in the home environment and does Automatic Service Fingerprinting and Security Posture assessment with the help of a scanner module. OT Smart Home Security Analyzer acts as the internet gateway and Proxy Firewall for all connected OT devices. The scanner module periodically scans the existing nodes for any new vulnerabilities. As a remediation measure, devices with known high vulnerabilities are blocked and not allowed to connect to the internet. The rogue device detection capability of the OT Smart Home Security Analyzer prevents the node from connecting to the internet and communicating with other nodes with MAC filtering. Replay attacks and device cloning attack prevention are done with OT clients' identity verification. The framework provides real-time analysis of each node's security posture and allows users to restrict each node's communication with the network.

Application Architecture

OTSHSA application is designed with certain distributed architecture concepts and has components like IoT sensors and a server that can be accessed and managed by a web GUI interface. The server hosts a RESTful API interface for the web clients to communicate with it. The web client interface consumes the Restful APIs from the server, which allows users to leverage the features provided by the solution.



The lightweight **publish/subscribe** messaging protocol MQTT (MQ Telemetry Transport) based interface is used for secure bidirectional communication with the OTSHA server and IoT sensors. The use of MQTT offers optional support for encrypting messages using TLS and authentication of clients using modern authentication protocols, such as Oauth.

Technology Stack

Below is a list of various technologies used for different elements of this solution.

Server

- **i.** Python
- **ii.** Fastapi
- **iii.** MongoDB
- **iv.** MQTT

Client

- **v.** Typescript
- **vi.** Angular

Low-Level Design Description and Code Flow

The below section provides details of the low-level design of the solution along with code flow.

1. Server

- The server is written in Python, and it uses the fast API framework to work with a MongoDB instance. FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.6+ and is based on standard Python-type hints.

- The whole application runs within a reactor loop, which simulates an event loop on a single thread. It means that every operation is being executed on a single line, and is never blocked, in case of an idle function, subsequent instructions are executed and vice versa.

2. Database

- The database used is MongoDB, which is a NoSQL cross-platform document-oriented database program. MongoDB uses JSON-like documents with optional schemas.

- Multiple interfaces using the **pydantic Base model** class are developed; they are called in for different Read, Write, Update operations.

- The fast API has a class-based ODM (Object Document mapper), making it easier to define Object-Oriented programming concepts and use implemented design patterns.

- Each table has a class inherited from the **beanie. Document** which implements all the required attributes and methods for effective communication with MongoDB instance.

- Each **models.py** contains defined interfaces for reading/WRITE/UPDATE operations, providing a validation layer for interacting with the underlying database and requests.

- **Pydantic** and **typing** modules in Python 3.6+ versions are used to take advantage of the current features.

Three main classes describe the data stored by the database

2.1 Device

- Defined in **otshsa/discovery/models/device.py**
- Stores the data related to a device recognized by the user.
- Stores IP Address, MAC Address of a specific device.

2.2Scan

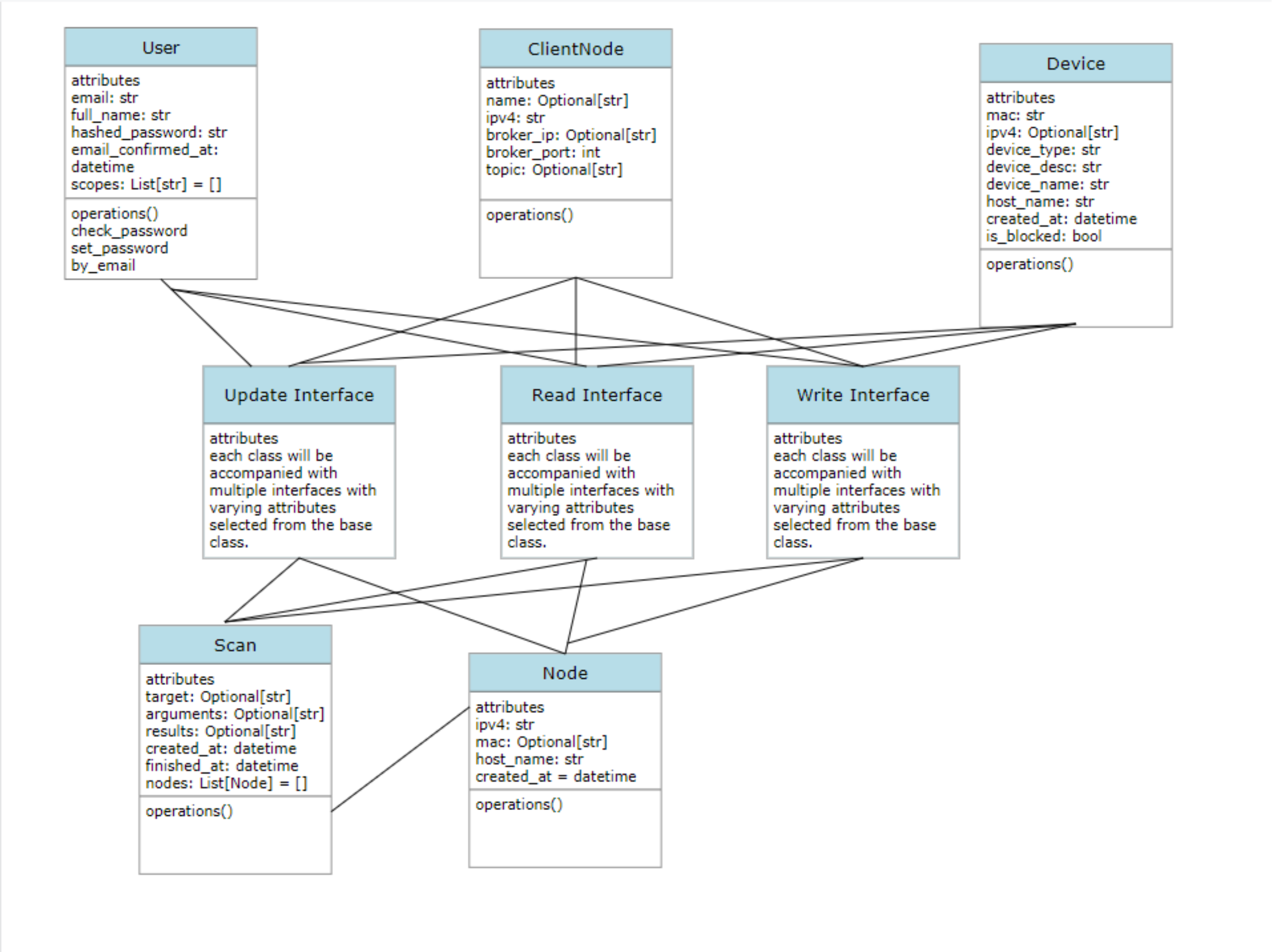
- Defined in **otshsa/discovery/models/scan.py**
- Stores the data related to a recent network scan.
- Stores IP Address, MAC Address of detected devices.

2.3ClientNode

- Defined in **otshsa/discovery/models/client.py**
- Stores the data related to a client IOT node.

- Stores IP Address, Broker IP Address to which the node is connected, a topic on which the node is publishing data to the broker.

The below UML diagram enlists database design elements.



3. API Routes

The server exposes APIs written as simple functions in each directory’s routers.py. Each method is attached to an everyday object which exposes them as each HTTP method on corresponding endpoints.

- Auth routes
  - Login - Logs in a user with username & password
  - Logout
  - Defined in **otshsa/auth/routers.py**
- Device routes

    - Device CRUD (Create/Read/Update/Delete)
    - Defined in **otshsa/discovery/routers.py**
- ClientNode routes
    - ClientNode CRUD
    - Defined in **otshsa/inventory/routers.py**
- Scan
    - Read Scan operation.
    - Defined in **otshsa/discovery/routers.py**
- Analysis
    - Device detection
        - The specific details related to the device are captured and used to detect the firmware information regarding the device.
        - Defined in **otshsa/discovery/routers.py**
    - Device risk analysis
        - Once the device is detected, with the help of a unique identifier for risk analysis will perform.
        - Defined in **otshsa/discovery/routers.py**
- Default credentials routes
    - CRUD operations
    - Defined in **otshsa/inventory/routers.py**

4. Scan

- The server scans the network using network mapping tools to identify neighboring devices in the network. It captures information regarding each device, like IP Address, MAC address, Hostname.
- The information is passed along to the scanner for practical risk analysis.

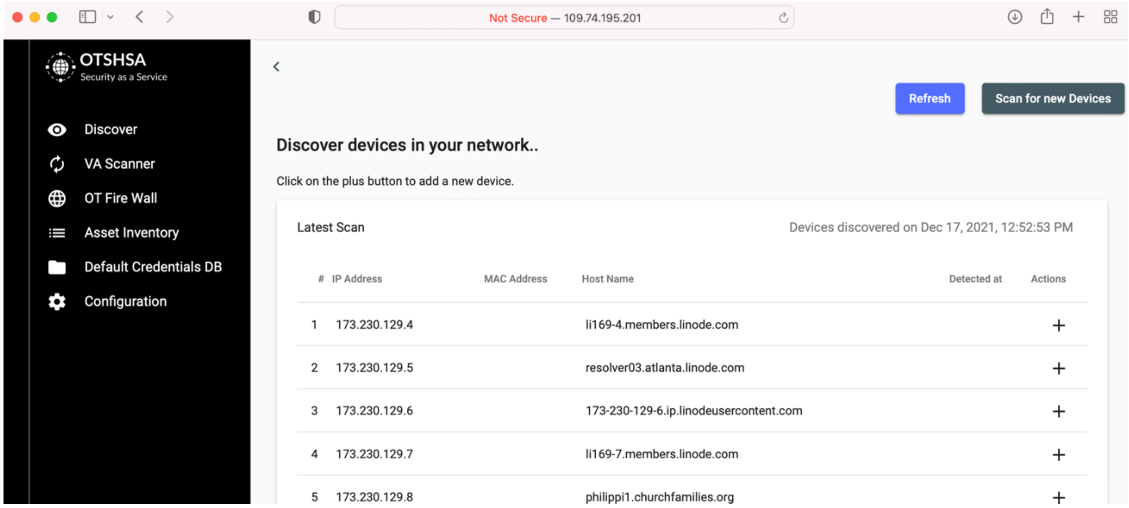
5. Client

The client is written in Typescript, using the Angular framework. Angular is a TypeScript-based free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations.

6.Screens

There are mainly 7 screens. Each screen is written as a **Component** in the application. Each components makes use of several sub-components and work together to comprise the application UI.

Screen 1 – Discover



Discover devices in your network..

Click on the plus button to add a new device.

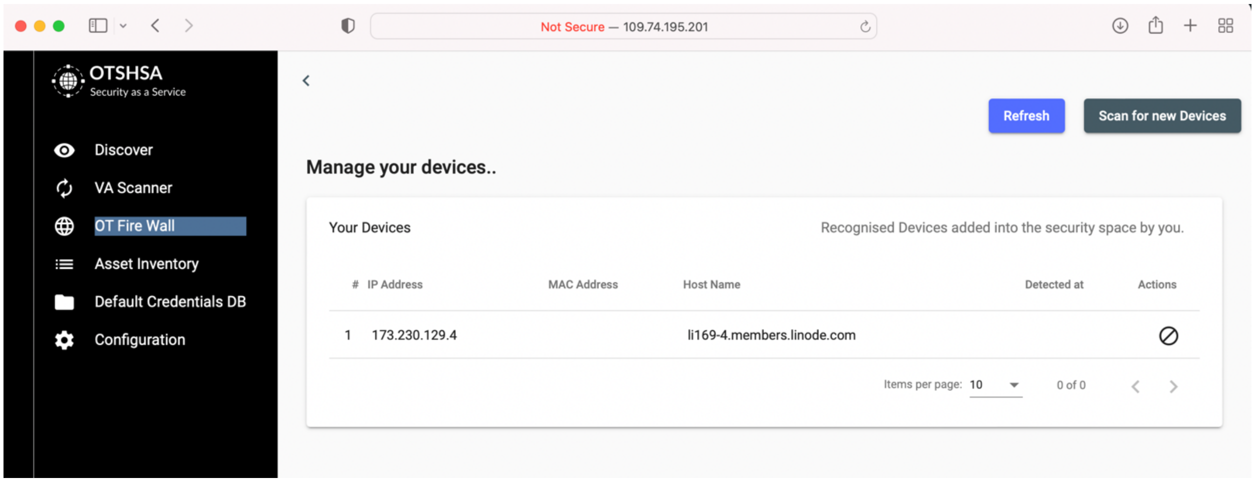
Latest Scan

Devices discovered on Dec 17, 2021, 12:52:53 PM

#	IP Address	MAC Address	Host Name	Detected at	Actions
1	173.230.129.4		li169-4.members.linode.com		+
2	173.230.129.5		resolver03.atlanta.linode.com		+
3	173.230.129.6		173-230-129-6.ip.linodeusercontent.com		+
4	173.230.129.7		li169-7.members.linode.com		+
5	173.230.129.8		philippi1.churchfamilies.org		+

- Lists out the devices connected to the local network.
- Allows user to add a device into our security environment by creating a new device on clicking '+' button.
- Defined in **web-ui/src/app/pages/discover/discover.component.ts**

Screen 2 - VA Scanner



Manage your devices..

Your Devices

Recognised Devices added into the security space by you.

#	IP Address	MAC Address	Host Name	Detected at	Actions
1	173.230.129.4		li169-4.members.linode.com		⊘

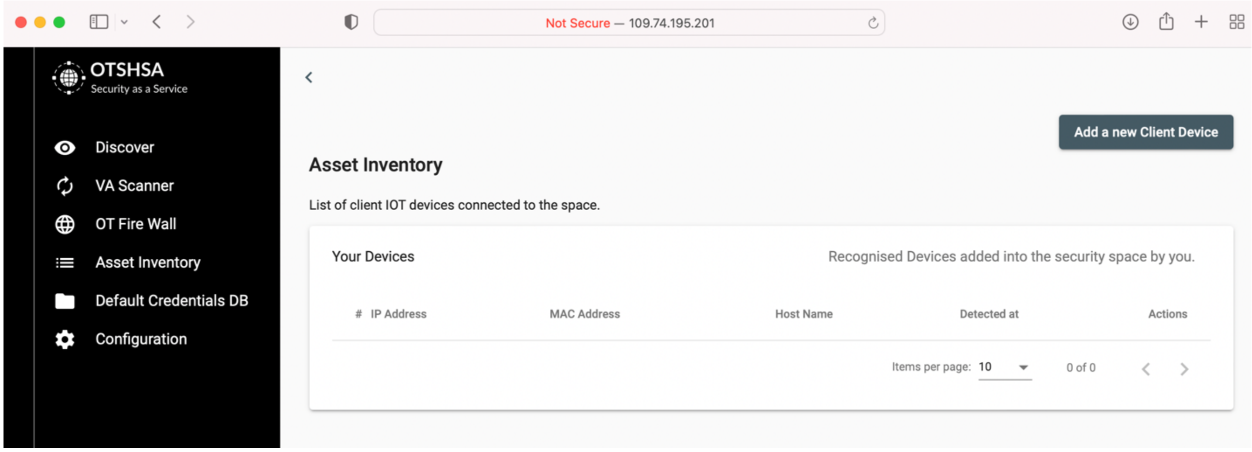
Items per page: 10 ▾   0 of 0   <   >

- Lists out the devices add to the security network.
- Allows user to navigate to **Device Details** screen, where specific details are displayed.
- Defined in **web-ui/src/app/pages/scanner/scanner.component.ts**

Screen 3 - Device Details

- Displays out the devices added to the security network.
- Allows user to detect firmware information of each device.
- Allows user to analyse risk information of each device.
- Defined in **web-ui/src/app/pages/device-details/device-details.component.ts**

Screen 4 - Asset Inventory



Asset Inventory

List of client IOT devices connected to the space.

Your Devices

Recognised Devices added into the security space by you.

#	IP Address	MAC Address	Host Name	Detected at	Actions

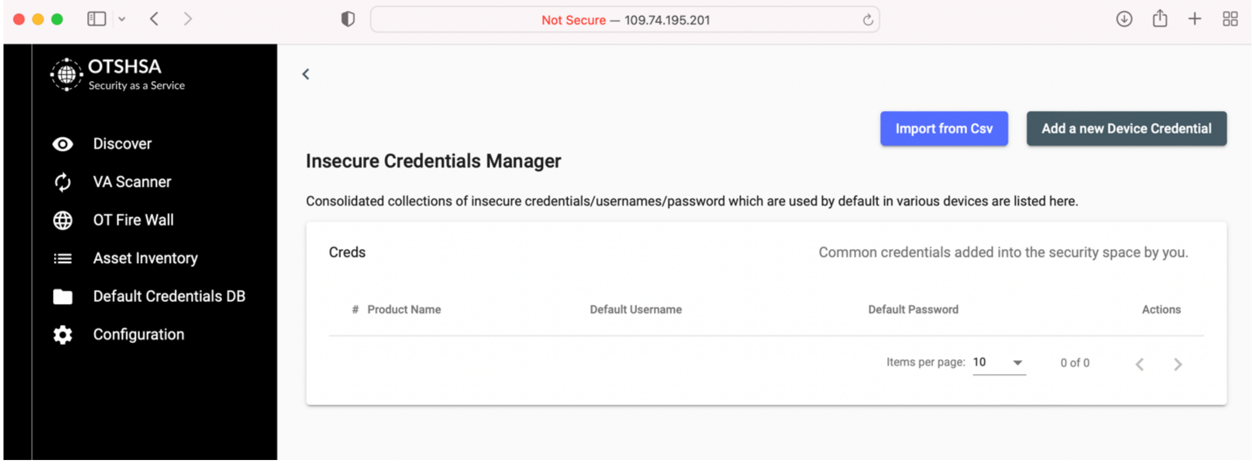
Items per page: 10 0 of 0

- Displays out the client IOT devices added to the security network.
- Allows user to navigate to Analytics tab.
- Defined in **web-ui/src/app/pages/inventory/inventory.component.ts**

Screen 5 - Analytics

- Displays out the client IOT device added to the security network.
- Real time communication with the device.
- Displays messages published by the device to the broker.
- MQTT client library for angular is initialised with the broker info for subscribing to published topics real time.
- Defined in **web-ui/src/app/pages/analytics/analytics.component.ts**

Screen 6 - Default Credentials DB



Insecure Credentials Manager

Consolidated collections of insecure credentials/usernames/password which are used by default in various devices are listed here.

Creds

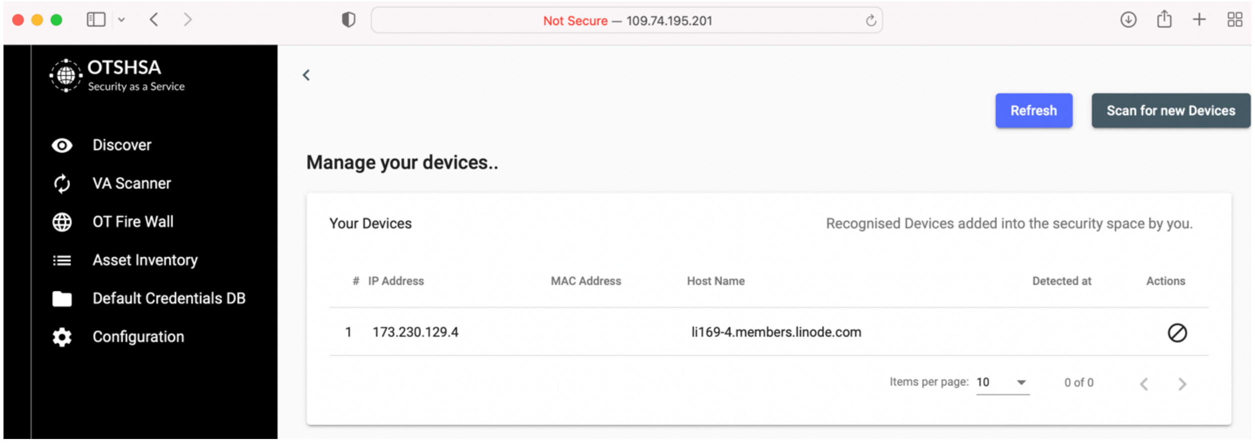
Common credentials added into the security space by you.

#	Product Name	Default Username	Default Password	Actions

Items per page: 10 0 of 0

- Stores default credentials of several products.
- Scanner uses this information for risk analysis.
- Can Import data from CSV file.
- Defined in **web-ui/src/app/pages/credentials/credentials.component.ts**

Screen 7 - OT Firewall



Manage your devices..

Your Devices

Recognised Devices added into the security space by you.

#	IP Address	MAC Address	Host Name	Detected at	Actions
1	173.230.129.4		li169-4.members.linode.com		⊘

Items per page: 10 0 of 0

- Lists out the devices added from the network.
- Provides capability to block and unblock devices using iptables.
- Defined in **web-ui/src/app/pages/firewall/firewall.component.ts**

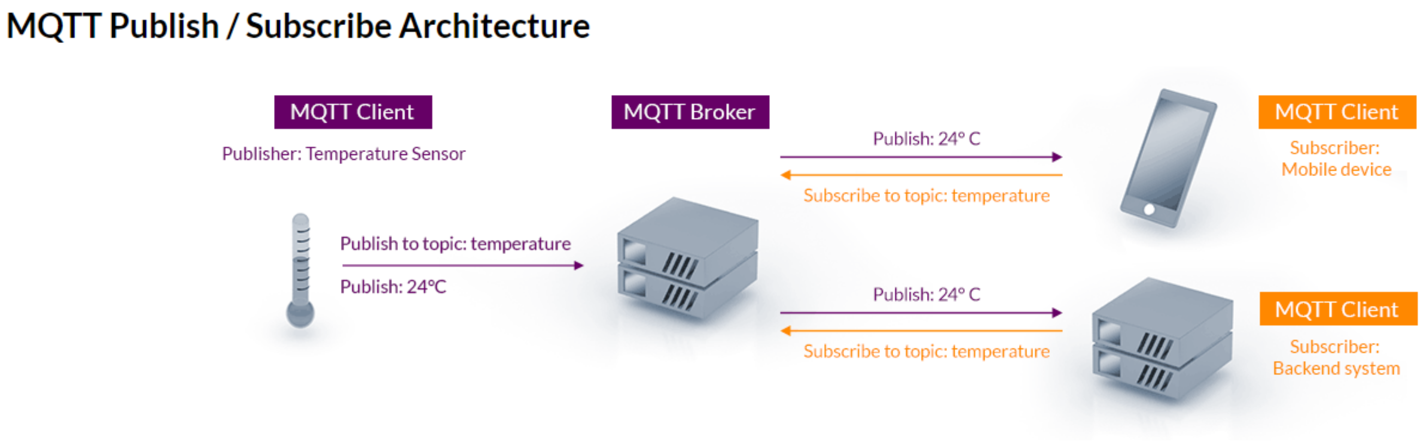
7.Access Control

The access control is implemented using Token authentication for users and by Firewall module for IOT devices. The firewall module provides capability for users to block/unblock devices from the internet communication.

8.MQTT Overview

The MQTT (1) protocol is used for communication between devices, client and broker. MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

- MQTT clients are very small, require minimal resources so can be used on small microcontrollers. MQTT message(!) headers are small to optimise network bandwidth.
- The MQTT protocol provides username and password fields in the CONNECT message for authentication. The client has the option to send a username and password when it connects to an MQTT broker(1).
- MQTT allows for messaging between device to cloud and cloud to device. This makes for easy broadcasting messages to groups of things.
- MQTT (1)can scale to connect with millions of IoT devices.
- Reliability of message delivery is important for many IoT use cases. This is why MQTT (1)has 3 defined quality of service levels: 0 - at most once, 1- at least once, 2 - exactly once
- Many IoT devices connect over unreliable cellular networks. MQTT’s support for persistent sessions reduces the time to reconnect the client with the broker.
- MQTT (1)makes it easy to encrypt messages using TLS and authenticate clients using modern authentication protocols, such as OAuth.


MQTT Publish / Subscribe Architecture
MQTT Client
Publisher: Temperature Sensor
MQTT Broker
Publish to topic: temperature
Publish: 24°C
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Mobile device
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Backend system

9.Simulating Clients

- Mock clients can be found in **otshsa/clients** directory.
- To run a mock client script, run it using python like, #**python clients/mock_temp_sensor.py**
- Then, go to **Asset Inventory** in **WebUI** to create a new client device by clicking on the button **Create New Client Device**.
- Enter details such as **broker IP** (it should be server IP), **Publishing topic** (topic to which client subscribe to), **Name** (Optional).
- Proceed to Analytics tab using the **arrow** icon in the right, to analyse/monitor real time communication with the client.

10.Docker Based Installation Steps(2):

Step 1: Update the apt package index and install packages to allow apt to use a repository over HTTPS:

```
#sudo apt update
#sudo apt install apt-transport-https ca-certificates curl software-properties-common curl gnupg lsb-release
```

Step 2:Add Docker's official GPG key:

```
#curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Step 3: Use the following command to set up the stable repository. To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below.

```
#echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Step 4: Update the apt package index, and install the latest version of Docker Engine and containerd, or go to the next step to install a specific version:

```
#sudo apt-get update
#sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Step 5: Run this command to download the current stable release of Docker Compose:

```
#sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Step 6: Apply executable permissions to the binary:

```
#sudo chmod +x /usr/local/bin/docker-compose
```

Step7: Finally unzip the OTSHA package directory shared and from the otsha directory build and run the container:

```
#unzip OTSHA.zip
#cd OTSHA
#docker-compose build
##docker-compose up
```

Access the web UI using the url http://127.0.0.1:4200 and simulate the client using previously mentioned steps.

Demo: A hosted demo of OTSHA instance can be accessed here http://109.74.195.201:4200/auth/login

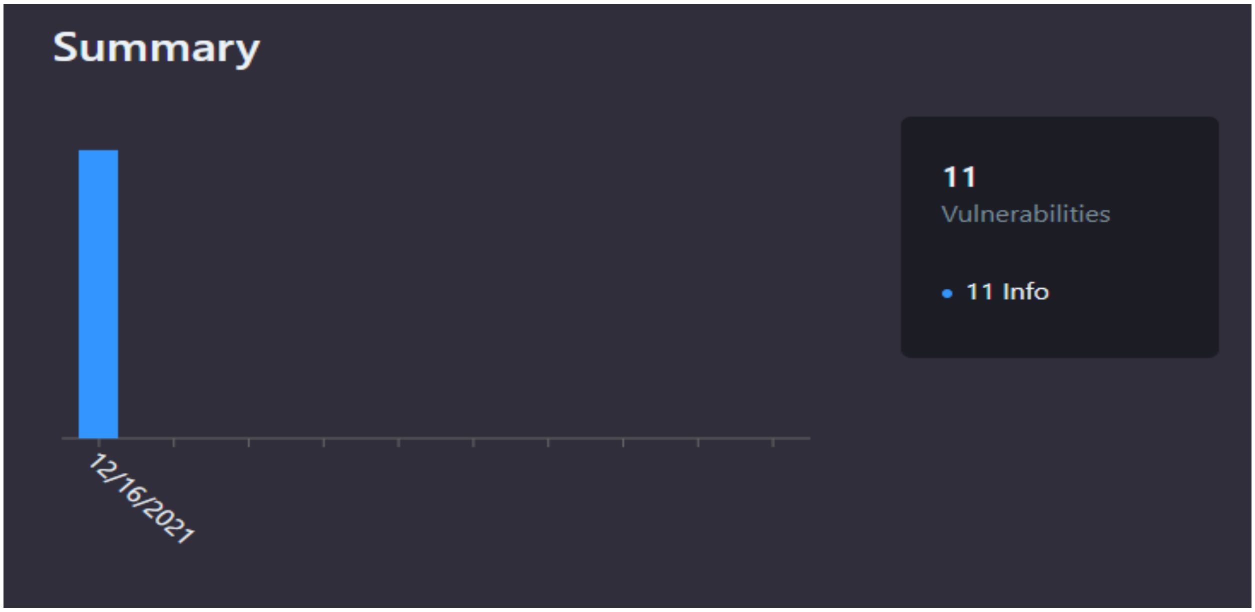
Username: admin@email.com

Password: abcd1234

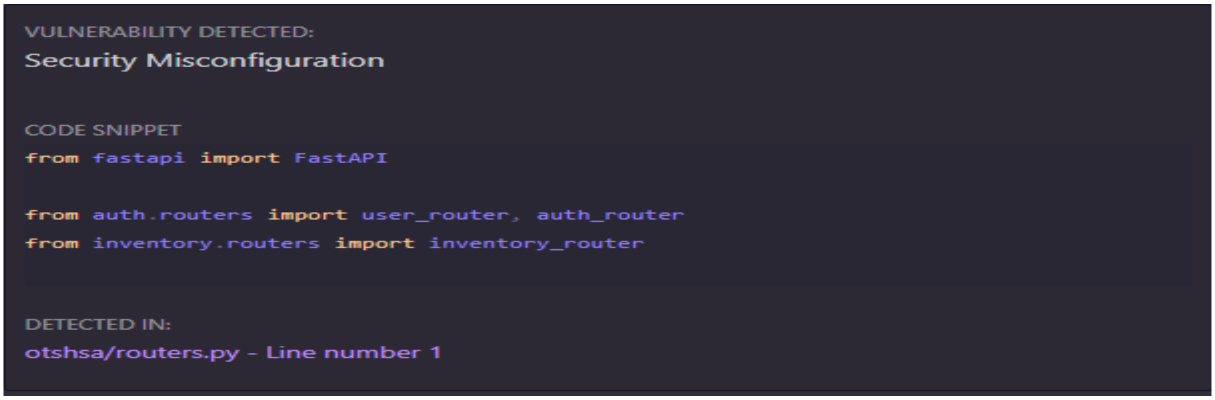
Note: For detailed installation steps please refee the README file in OTSHA.zip file.

11.Secure Code Review (SAST):

An in depth offline secure code review was conducted against the code base. Below are the list of vulnerabilities and necessary remediation steps are taken to improve the quality of the code. The vulnerabilities found as part of the security assessment is listed as follows,



Observation-1: Consider using FastAPI security middleware TrustedHostMiddleware to improve overall security. at line 1.



VULNERABILITY DETECTED:

Security Misconfiguration

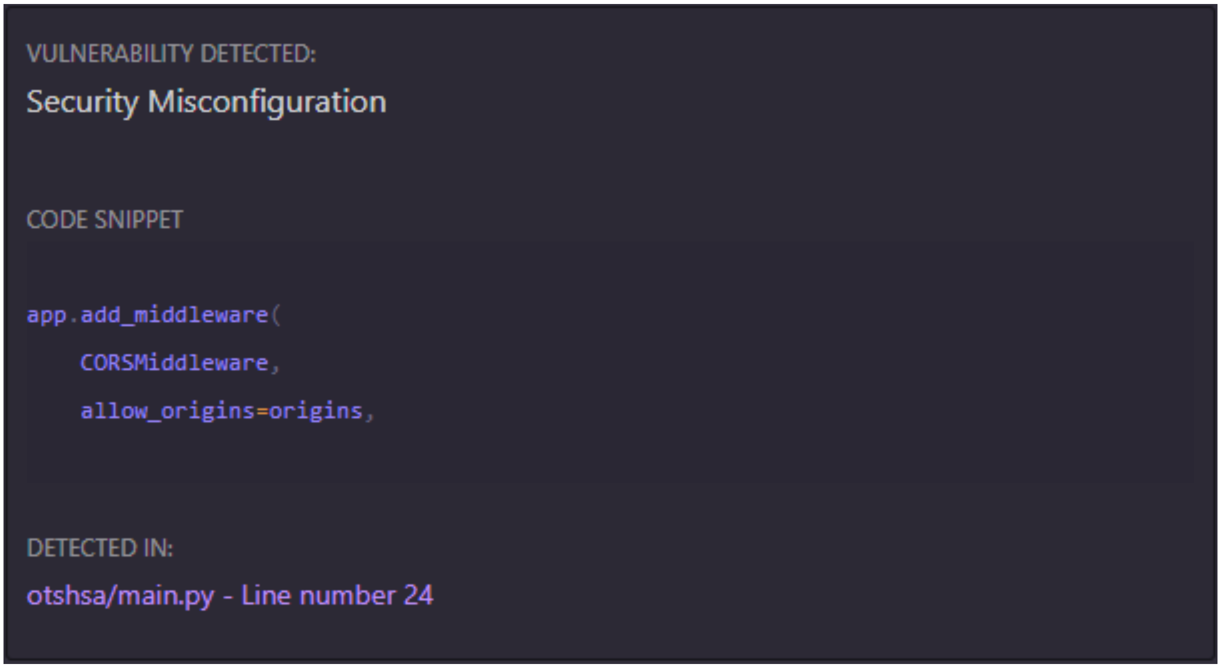
CODE SNIPPET

```
from fastapi import FastAPI

from auth.routers import user_router, auth_router
from inventory.routers import inventory_router
```

DETECTED IN:

otshsa/routers.py - Line number 1

Observation-2: Use of allowed credentials with CORS would decrease the overall API security. at line 24.



VULNERABILITY DETECTED:

Security Misconfiguration

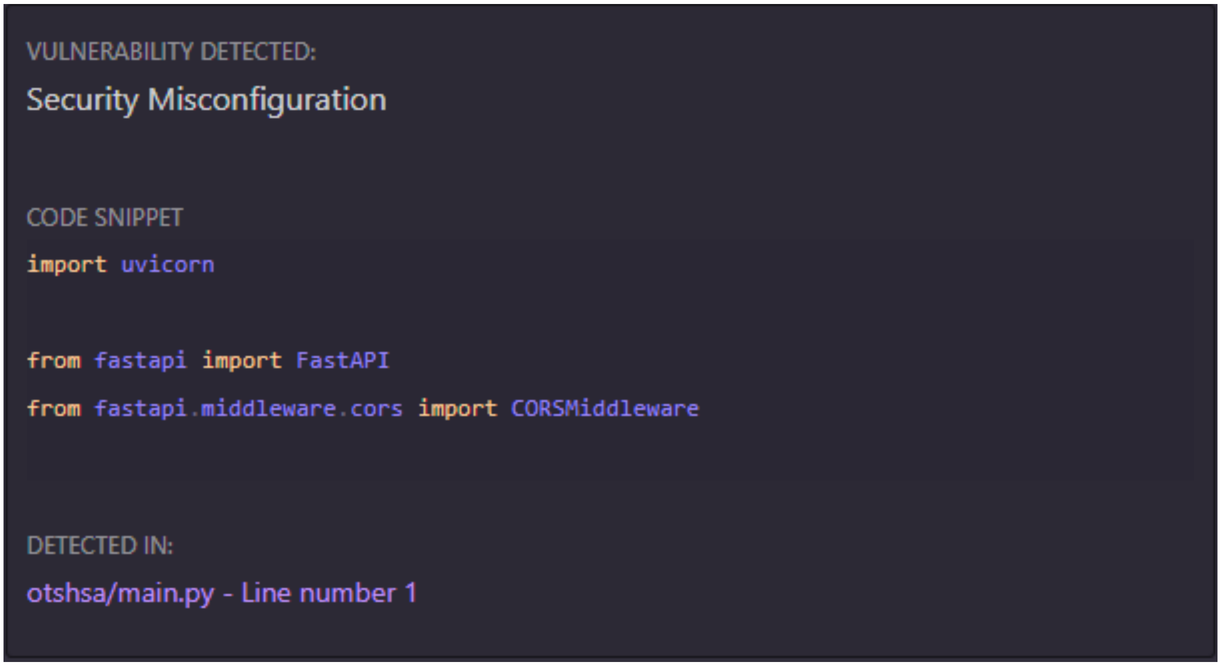
CODE SNIPPET

```
app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
```

DETECTED IN:

otshsa/main.py - Line number 24

Observation-3: Consider using FastAPI security middleware TrustedHostMiddleware to improve overall security. at line 1.



VULNERABILITY DETECTED:

Security Misconfiguration

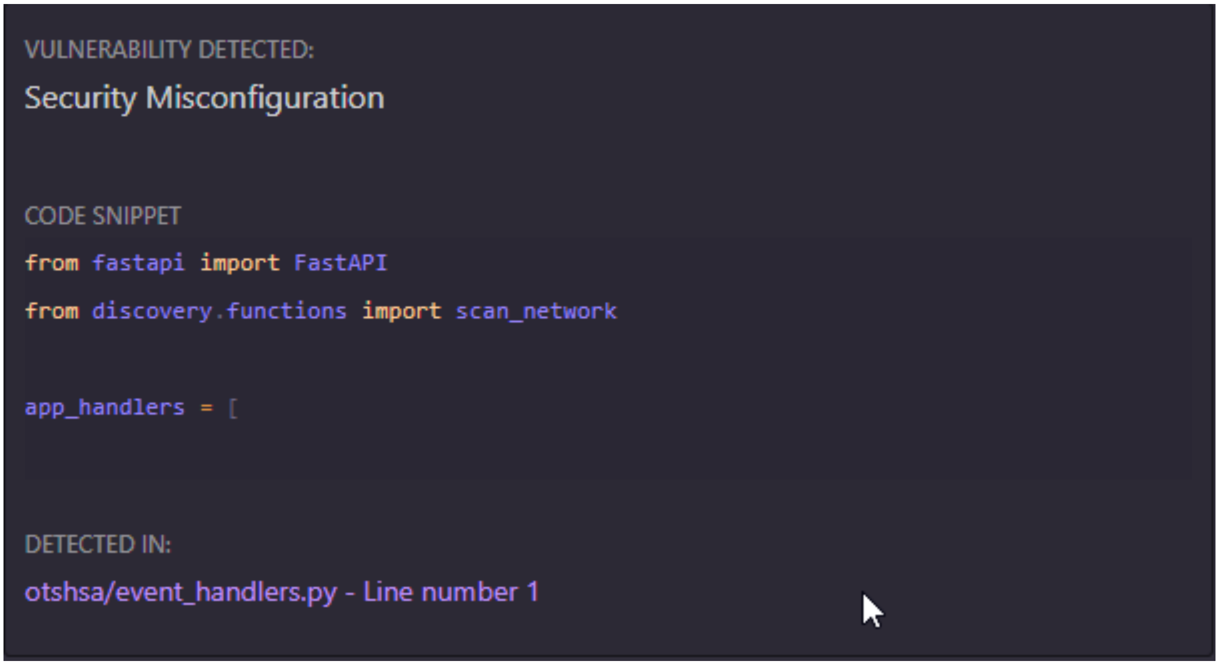
CODE SNIPPET

```
import uvicorn

from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
```

DETECTED IN:

otshsa/main.py - Line number 1

Observation-4: Consider using FastAPI security middleware TrustedHostMiddleware to improve overall security. at line 1.



VULNERABILITY DETECTED:

Security Misconfiguration

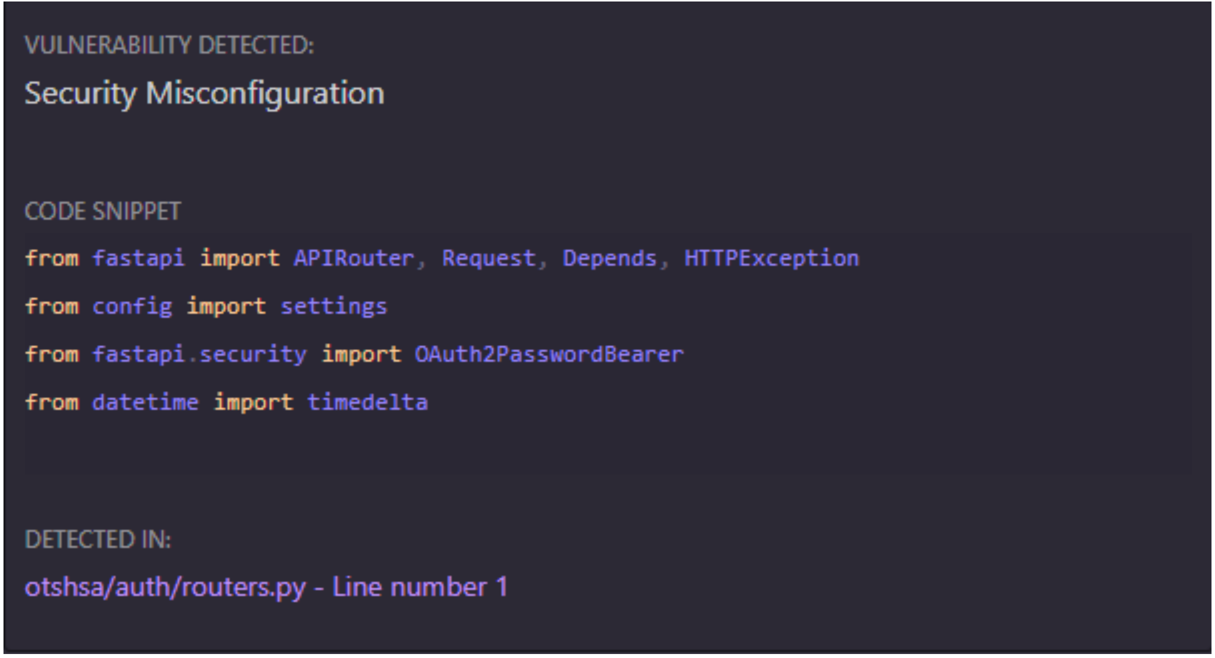
CODE SNIPPET

```
from fastapi import FastAPI
from discovery.functions import scan_network

app_handlers = [
```

DETECTED IN:

otshsa/event_handlers.py - Line number 1

Observation-5: Consider using FastAPI security middleware TrustedHostMiddleware to improve overall security. at line 1.



VULNERABILITY DETECTED:

Security Misconfiguration

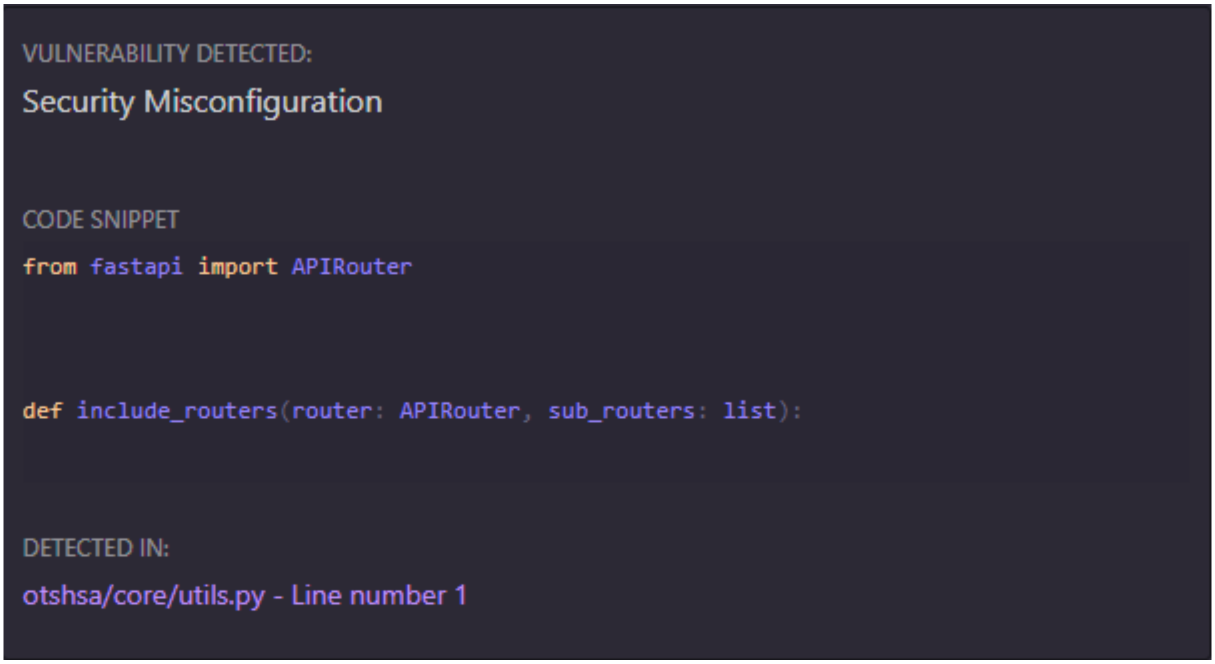
CODE SNIPPET

```
from fastapi import APIRouter, Request, Depends, HTTPException
from config import settings
from fastapi.security import OAuth2PasswordBearer
from datetime import timedelta
```

DETECTED IN:

otshsa/auth/routers.py - Line number 1

Observation-7: Consider using FastAPI security middleware TrustedHostMiddleware to improve overall security. at line 1.



VULNERABILITY DETECTED:

Security Misconfiguration

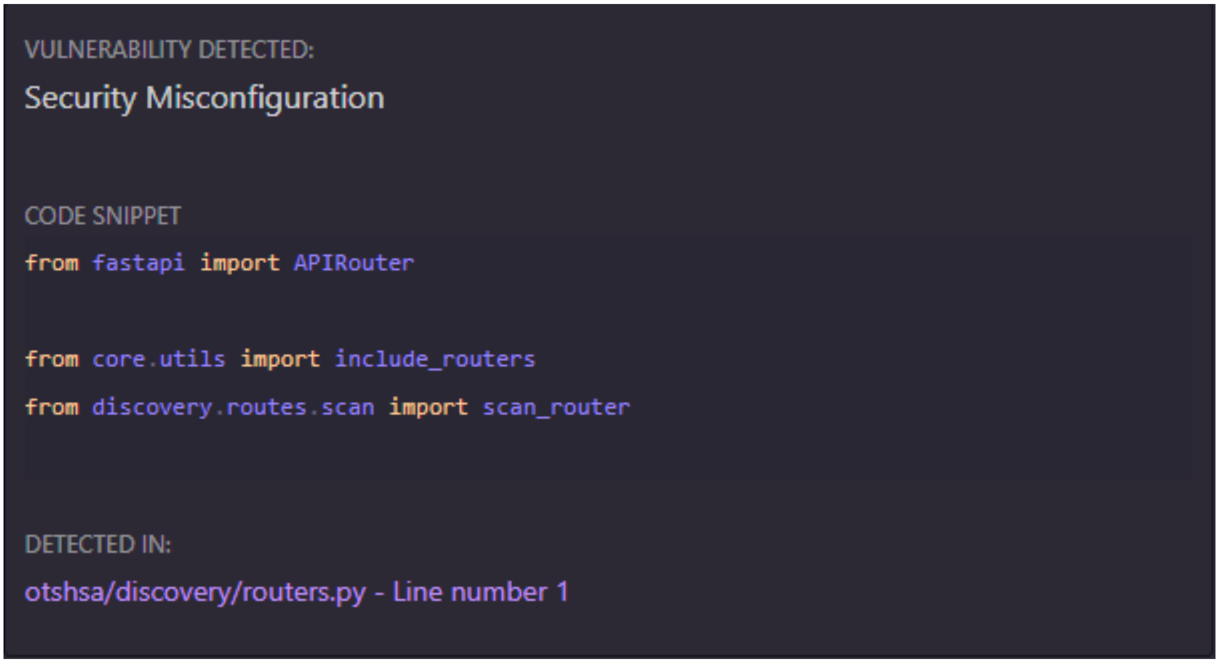
CODE SNIPPET

```
from fastapi import APIRouter


def include_routers(router: APIRouter, sub_routers: list):
```

DETECTED IN:

otshsa/core/utils.py - Line number 1

Observation-8: Consider using FastAPI security middleware TrustedHostMiddleware to improve overall security. at line 1.



VULNERABILITY DETECTED:

Security Misconfiguration

CODE SNIPPET

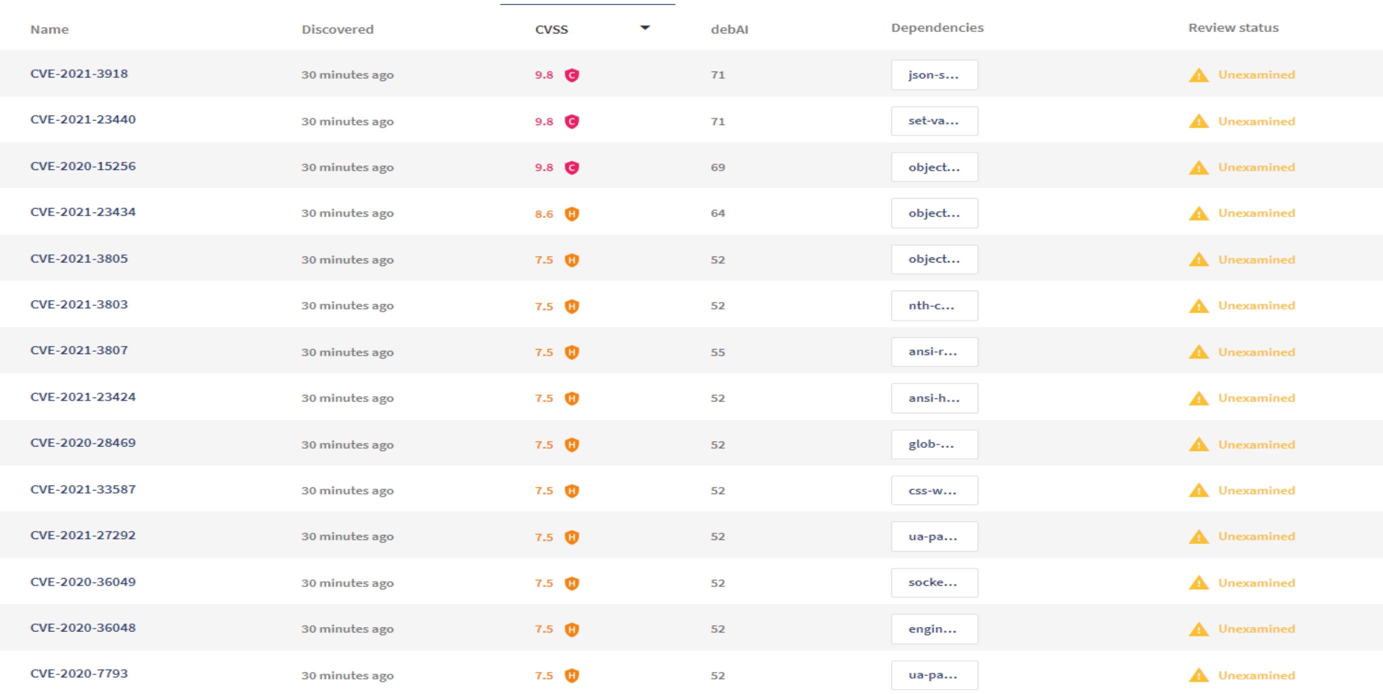
```
from fastapi import APIRouter

from core.utils import include_routers
from discovery.routes.scan import scan_router
```

DETECTED IN:

otshsa/discovery/routers.py - Line number 1

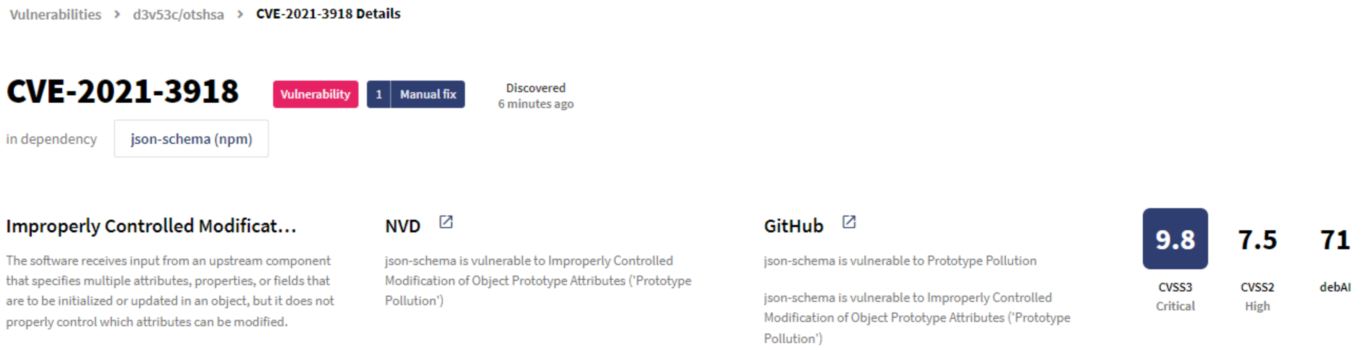
12.Software Composition Analysis (SCA):

Detailed Software Analysis of the developed code base was conducted, and possible remediation steps needed are taken as needed.



Name	Discovered	CVSS	debAI	Dependencies	Review status
CVE-2021-3918	30 minutes ago	9.8 C	71	json-s...	⚠ Unexamined
CVE-2021-23440	30 minutes ago	9.8 C	71	set-va...	⚠ Unexamined
CVE-2020-15256	30 minutes ago	9.8 C	69	object...	⚠ Unexamined
CVE-2021-23434	30 minutes ago	8.6 H	64	object...	⚠ Unexamined
CVE-2021-3805	30 minutes ago	7.5 H	52	object...	⚠ Unexamined
CVE-2021-3803	30 minutes ago	7.5 H	52	nth-c...	⚠ Unexamined
CVE-2021-3807	30 minutes ago	7.5 H	55	ansi-r...	⚠ Unexamined
CVE-2021-23424	30 minutes ago	7.5 H	52	ansi-h...	⚠ Unexamined
CVE-2020-28469	30 minutes ago	7.5 H	52	glob-...	⚠ Unexamined
CVE-2021-33587	30 minutes ago	7.5 H	52	css-w...	⚠ Unexamined
CVE-2021-27292	30 minutes ago	7.5 H	52	ua-pa...	⚠ Unexamined
CVE-2020-36049	30 minutes ago	7.5 H	52	socke...	⚠ Unexamined
CVE-2020-36048	30 minutes ago	7.5 H	52	engin...	⚠ Unexamined
CVE-2020-7793	30 minutes ago	7.5 H	52	ua-pa...	⚠ Unexamined

CVE-2021-3918 - json-schema (npm)



Vulnerabilities › d3v53c/otshsa › **CVE-2021-3918 Details**

CVE-2021-3918 Vulnerability 1 Manual fix Discovered 6 minutes ago

in dependency json-schema (npm)

Improperly Controlled Modificat...

The software receives input from an upstream component that specifies multiple attributes, properties, or fields that are to be initialized or updated in an object, but it does not properly control which attributes can be modified.

NVD

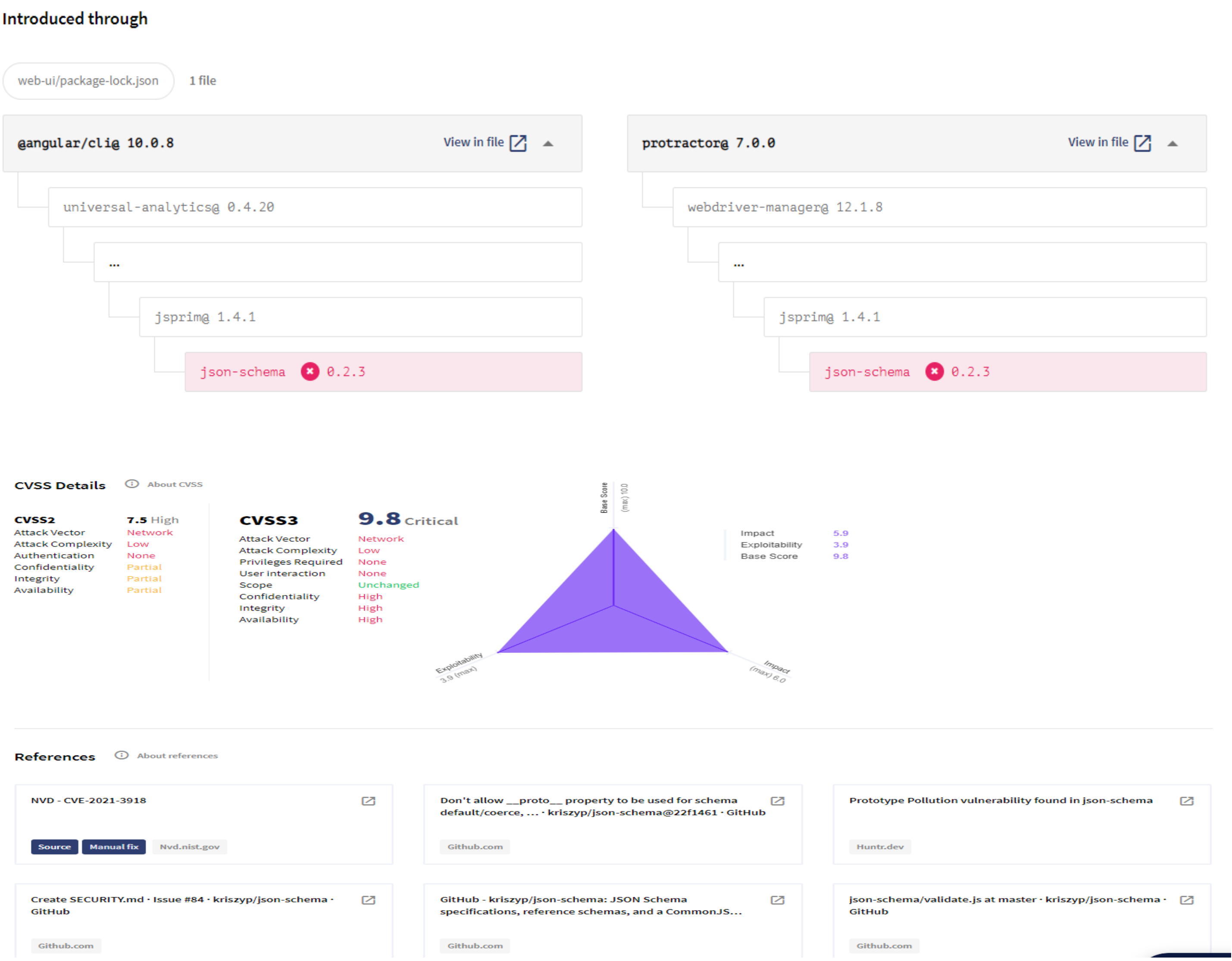
json-schema is vulnerable to Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution')

GitHub

json-schema is vulnerable to Prototype Pollution

json-schema is vulnerable to Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution')

9.8	7.5	71
CVSS3 Critical	CVSS2 High	debAI



Introduced through

web-ui/package-lock.json 1 file

@angular/cli@ 10.0.8 View in file
- universal-analytics@ 0.4.20
  - ...
    - jsprim@ 1.4.1
      - json-schema ✖ 0.2.3

protractor@ 7.0.0 View in file
- webdriver-manager@ 12.1.8
  - ...
    - jsprim@ 1.4.1
      - json-schema ✖ 0.2.3

CVSS Details ⓘ About CVSS

CVSS2	7.5 High
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	Partial
Integrity	Partial
Availability	Partial

CVSS3	9.8 Critical
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Impact	5.9
Exploitability	3.9
Base Score	9.8

References ⓘ About references

NVD - CVE-2021-3918
Source Manual fix Nvd.nist.gov

Don't allow __proto__ property to be used for schema default/coerce, ... · kriszyp/json-schema@22f1461 · GitHub
Github.com

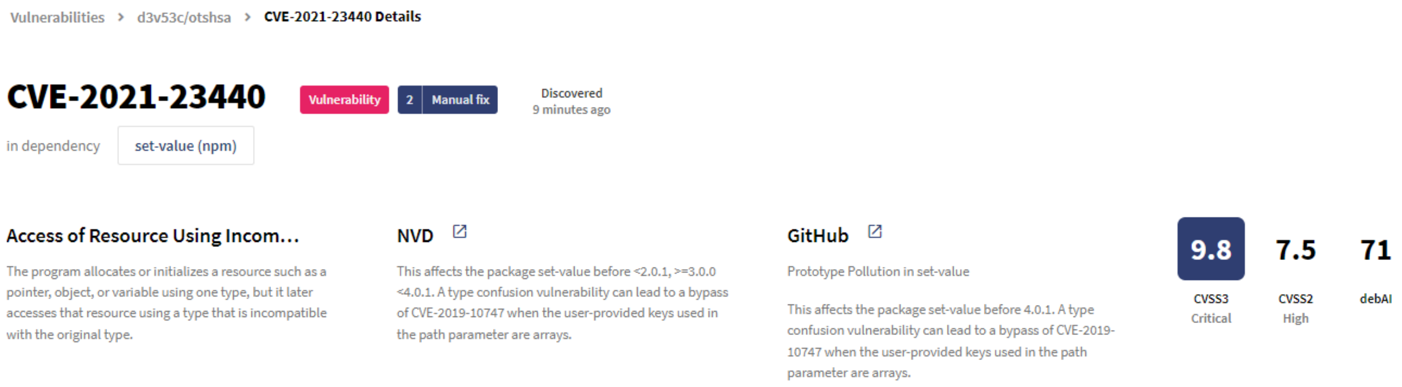
Prototype Pollution vulnerability found in json-schema
Huntr.dev

Create SECURITY.md · Issue #84 · kriszyp/json-schema · GitHub
Github.com

GitHub - kriszyp/json-schema: JSON Schema specifications, reference schemas, and a CommonJS...
Github.com

json-schema/validate.js at master · kriszyp/json-schema · GitHub
Github.com

CVE-2021-23440 - set-value (npm)



Vulnerabilities › d3v53c/otshsa › CVE-2021-23440 Details

CVE-2021-23440

Vulnerability 2 Manual fix

Discovered
9 minutes ago

in dependency set-value (npm)

Access of Resource Using Incom...

The program allocates or initializes a resource such as a pointer, object, or variable using one type, but it later accesses that resource using a type that is incompatible with the original type.

NVD

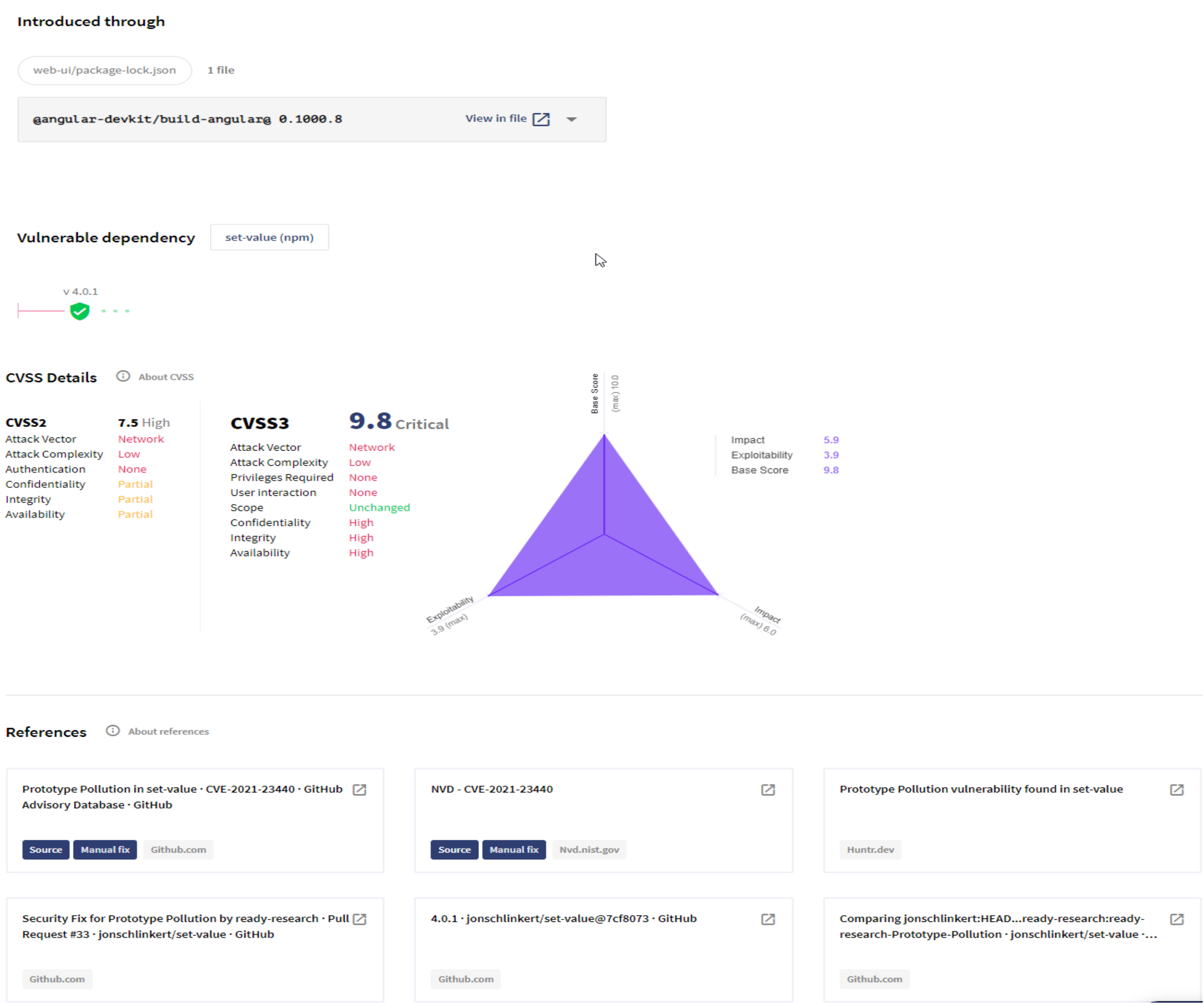
This affects the package set-value before <2.0.1, >=3.0.0 <4.0.1. A type confusion vulnerability can lead to a bypass of CVE-2019-10747 when the user-provided keys used in the path parameter are arrays.

GitHub

Prototype Pollution in set-value

This affects the package set-value before 4.0.1. A type confusion vulnerability can lead to a bypass of CVE-2019-10747 when the user-provided keys used in the path parameter are arrays.

9.8	7.5	71
CVSS3 Critical	CVSS2 High	debAI



Introduced through

web-ui/package-lock.json 1 file

```
@angular-devkit/build-angular@ 0.1000.8
```

View in file

Vulnerable dependency

set-value (npm)

v 4.0.1

CVSS Details

About CVSS

CVSS2	7.5 High
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	Partial
Integrity	Partial
Availability	Partial

CVSS3	9.8 Critical
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Impact	5.9
Exploitability	3.9
Base Score	9.8

References

About references

Prototype Pollution in set-value · CVE-2021-23440 · GitHub Advisory Database · GitHub

Source | Manual fix | Github.com

NVD - CVE-2021-23440

Source | Manual fix | Nvd.nist.gov

Prototype Pollution vulnerability found in set-value

Huntr.dev

Security Fix for Prototype Pollution by ready-research · Pull Request #33 · jonschlinkert/set-value · GitHub

Github.com

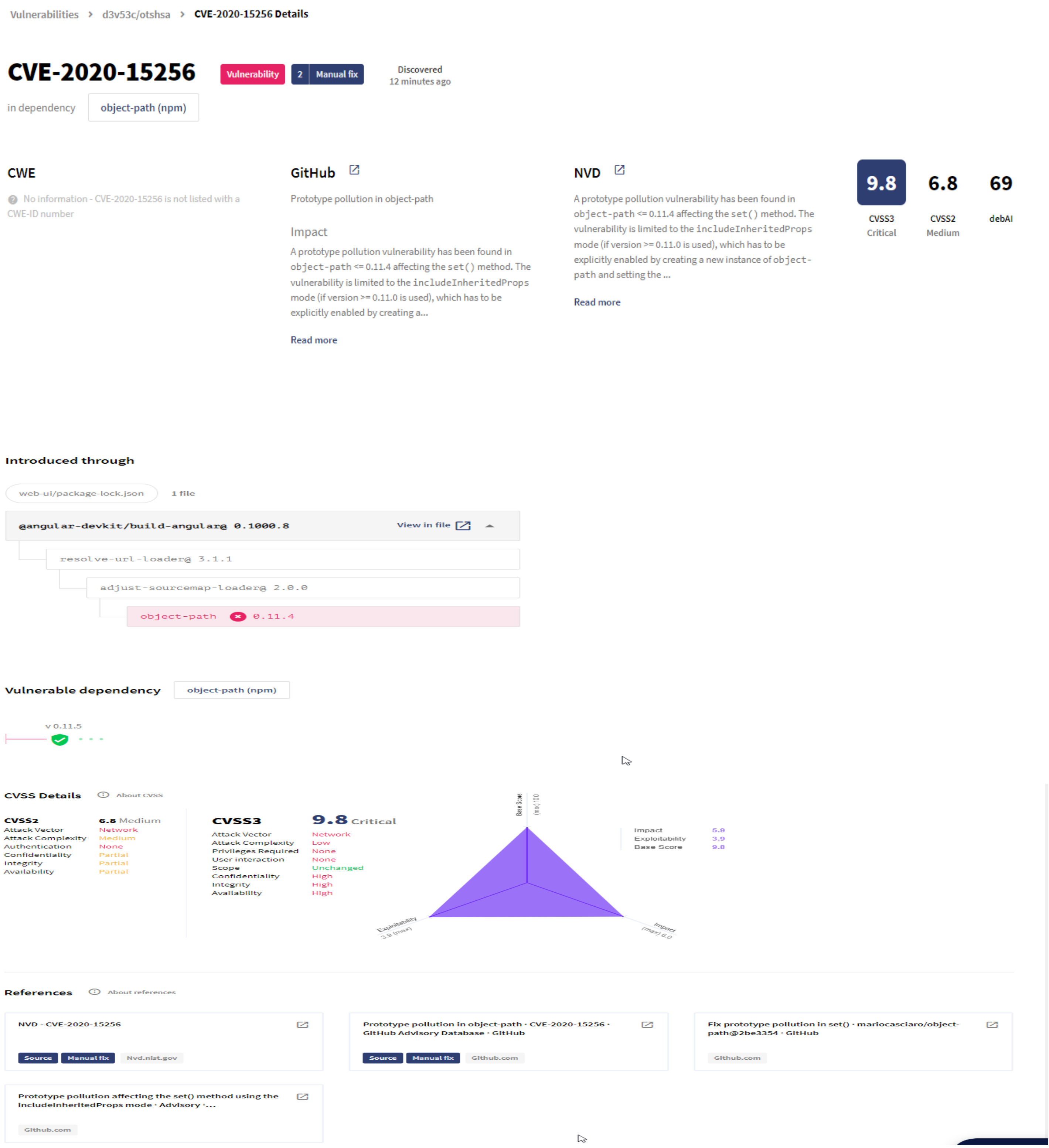
4.0.1 · jonschlinkert/set-value@7cf8073 · GitHub

Github.com

Comparing jonschlinkert:HEAD...ready-research:ready-research-Prototype-Pollution · jonschlinkert/set-value ·...

Github.com

CVE-2020-15256 - object-path (npm)



Vulnerabilities › d3v53c/otshsa › **CVE-2020-15256 Details**

CVE-2020-15256

Vulnerability | 2 Manual fix | Discovered 12 minutes ago

in dependency object-path (npm)

CWE

No information - CVE-2020-15256 is not listed with a CWE-ID number

GitHub

Prototype pollution in object-path

Impact

A prototype pollution vulnerability has been found in `object-path` <= 0.11.4 affecting the `set()` method. The vulnerability is limited to the `includeInheritedProps` mode (if version >= 0.11.0 is used), which has to be explicitly enabled by creating a...

Read more

NVD

A prototype pollution vulnerability has been found in `object-path` <= 0.11.4 affecting the `set()` method. The vulnerability is limited to the `includeInheritedProps` mode (if version >= 0.11.0 is used), which has to be explicitly enabled by creating a new instance of `object-path` and setting the ...

Read more

9.8	6.8	69
CVSS3 Critical	CVSS2 Medium	debAI

Introduced through

web-ui/package-lock.json 1 file

```
@angular-devkit/build-angular@ 0.1000.8          View in file
  resolve-url-loader@ 3.1.1
    adjust-sourcemap-loader@ 2.0.0
      object-path ✖ 0.11.4
```

Vulnerable dependency object-path (npm)

v 0.11.5 ✓

CVSS Details ⓘ About CVSS

CVSS2	6.8 Medium
Attack Vector	Network
Attack Complexity	Medium
Authentication	None
Confidentiality	Partial
Integrity	Partial
Availability	Partial

CVSS3	9.8 Critical
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Impact	5.9
Exploitability	3.9
Base Score	9.8

References ⓘ About references

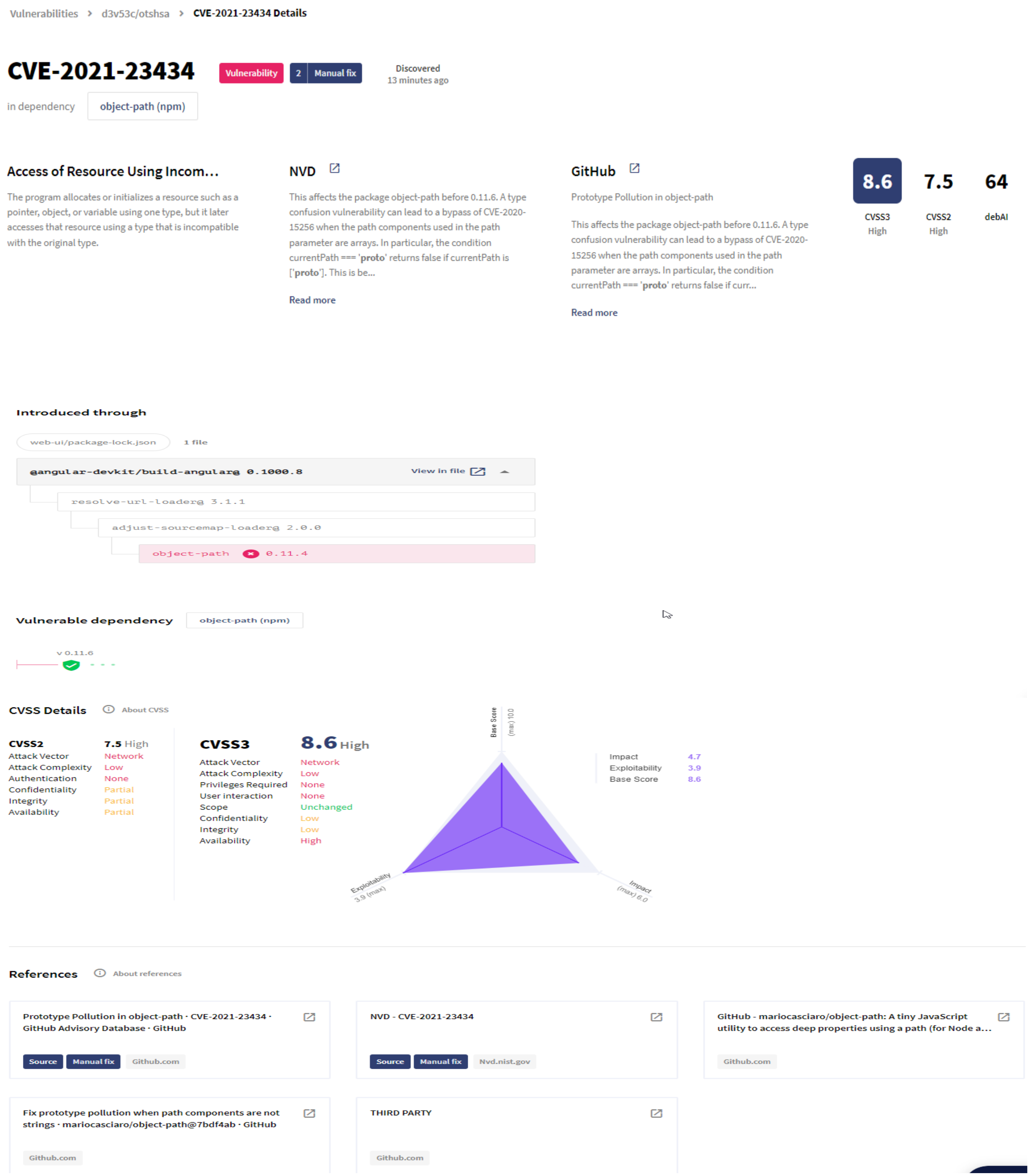
NVD - CVE-2020-15256
Source | Manual fix | Nvd.nist.gov

Prototype pollution in object-path · CVE-2020-15256 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

Fix prototype pollution in set() · mariocasciaro/object-path@2be3354 · GitHub
Github.com

Prototype pollution affecting the set() method using the includeInheritedProps mode · Advisory ·...
Github.com

CVE-2021-23434 - object-path (npm)



Vulnerabilities › d3v53c/otshsa › **CVE-2021-23434 Details**

CVE-2021-23434

Vulnerability | 2 | Manual fix

Discovered
13 minutes ago

in dependency | object-path (npm)

Access of Resource Using Incom...

The program allocates or initializes a resource such as a pointer, object, or variable using one type, but it later accesses that resource using a type that is incompatible with the original type.

NVD

This affects the package object-path before 0.11.6. A type confusion vulnerability can lead to a bypass of CVE-2020-15256 when the path components used in the path parameter are arrays. In particular, the condition currentPath === '**proto**' returns false if currentPath is ['**proto**']. This is be...

Read more

GitHub

Prototype Pollution in object-path

This affects the package object-path before 0.11.6. A type confusion vulnerability can lead to a bypass of CVE-2020-15256 when the path components used in the path parameter are arrays. In particular, the condition currentPath === '**proto**' returns false if currentPath is curr...

Read more

8.6	7.5	64
CVSS3 High	CVSS2 High	debAI

Introduced through

web-ui/package-lock.json   1 file

```
@angular-devkit/build-angular@ 0.1000.8                View in file
    resolve-url-loader@ 3.1.1
        adjust-sourcemap-loader@ 2.0.0
            object-path ✖ 0.11.4
```

Vulnerable dependency

object-path (npm)

v 0.11.6 ✓

CVSS Details

About CVSS

CVSS2	7.5 High
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	Partial
Integrity	Partial
Availability	Partial

CVSS3	8.6 High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	Low
Integrity	Low
Availability	High

Impact	4.7
Exploitability	3.9
Base Score	8.6

References

About references

Prototype Pollution in object-path · CVE-2021-23434 · GitHub Advisory Database · GitHub

Source | Manual fix | Github.com

NVD - CVE-2021-23434

Source | Manual fix | Nvd.nist.gov

GitHub - mariocasciaro/object-path: A tiny JavaScript utility to access deep properties using a path (for Node a...

Github.com

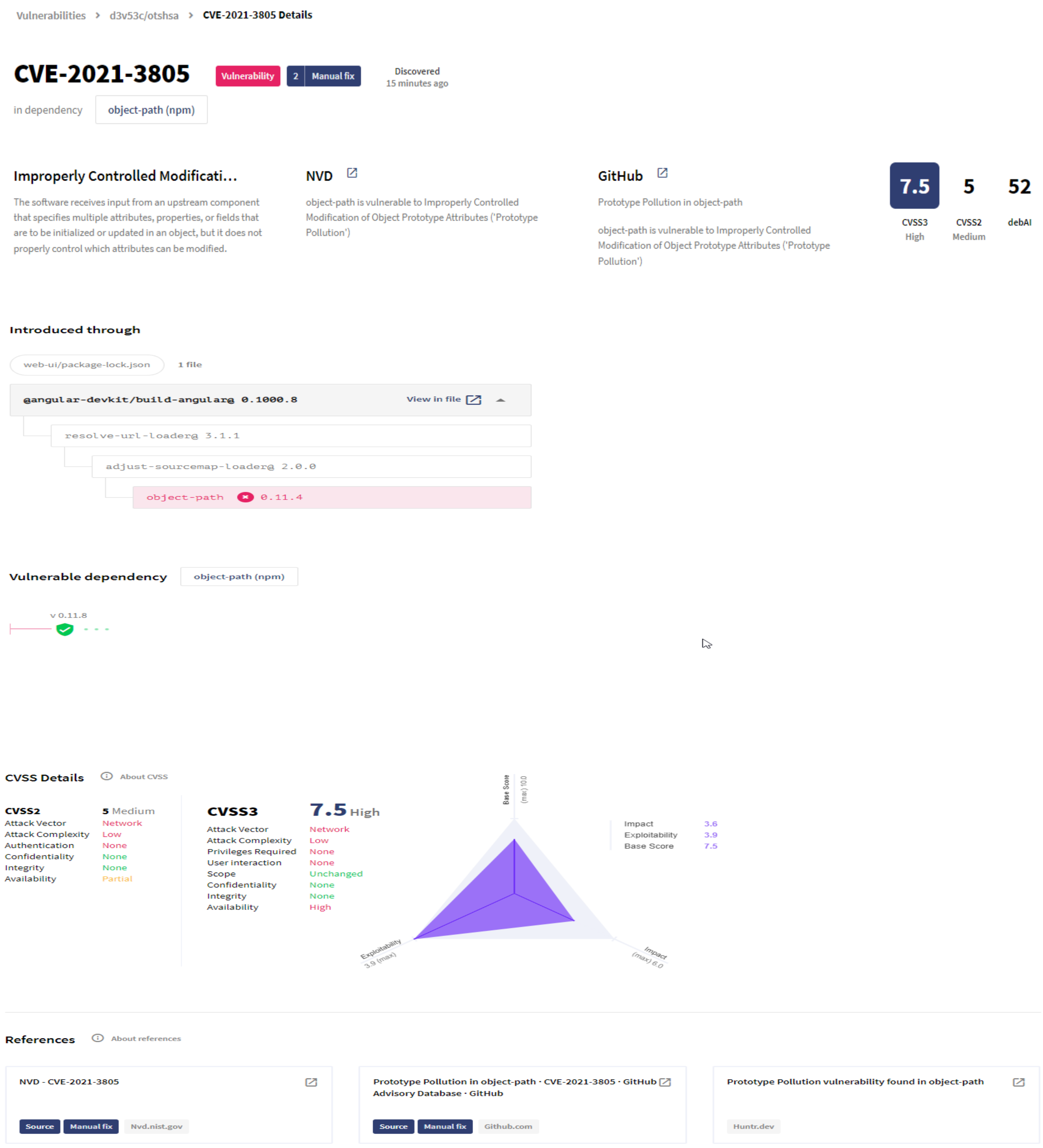
Fix prototype pollution when path components are not strings · mariocasciaro/object-path@7bdf4ab · GitHub

Github.com

THIRD PARTY

Github.com

Security Misconfiguration



Vulnerabilities › d3v53c/otshsa › **CVE-2021-3805 Details**

CVE-2021-3805

Vulnerability | 2 | Manual fix

Discovered
15 minutes ago

in dependency | object-path (npm)

Improperly Controlled Modificati...

The software receives input from an upstream component that specifies multiple attributes, properties, or fields that are to be initialized or updated in an object, but it does not properly control which attributes can be modified.

NVD

object-path is vulnerable to Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution')

GitHub

Prototype Pollution in object-path

object-path is vulnerable to Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution')

7.5	5	52
CVSS3 High	CVSS2 Medium	debAI

Introduced through

web-ui/package-lock.json 1 file

```
@angular-devkit/build-angular@ 0.1000.8                View in file
    resolve-url-loader@ 3.1.1
        adjust-sourcemap-loader@ 2.0.0
            object-path ✖ 0.11.4
```

Vulnerable dependency object-path (npm)

v 0.11.8

CVSS Details ⓘ About CVSS

CVSS2	5 Medium
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Partial

CVSS3	7.5 High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

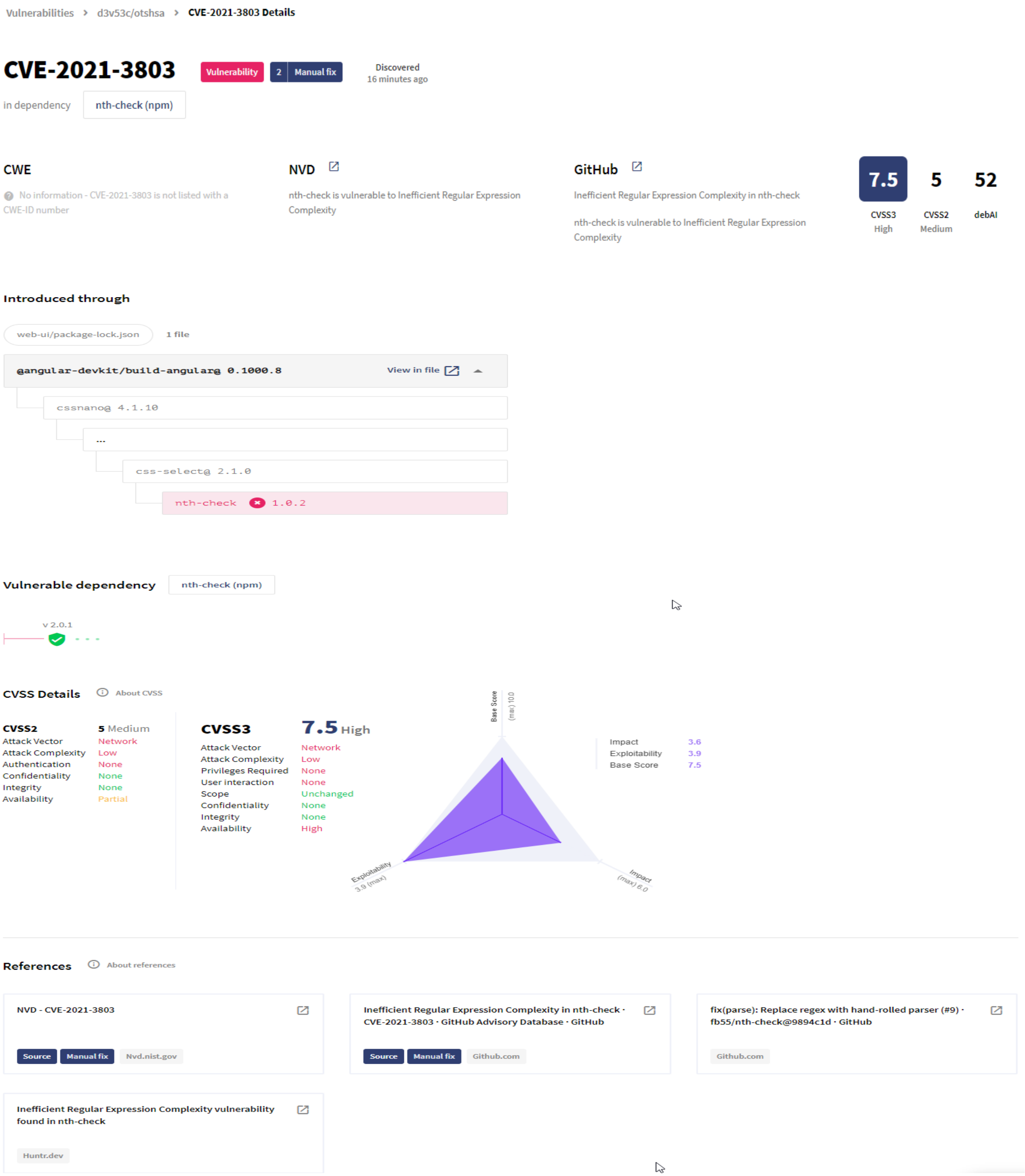
References ⓘ About references

NVD - CVE-2021-3805
Source | Manual fix | Nvd.nist.gov

Prototype Pollution in object-path · CVE-2021-3805 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

Prototype Pollution vulnerability found in object-path
Huntr.dev

CVE-2021-3803 - nth-check (npm)



Vulnerabilities › d3v53c/otshsa › **CVE-2021-3803 Details**

CVE-2021-3803

Vulnerability | 2 Manual fix

Discovered
16 minutes ago

in dependency nth-check (npm)

CWE

No information - CVE-2021-3803 is not listed with a CWE-ID number

NVD

nth-check is vulnerable to Inefficient Regular Expression Complexity

GitHub

Inefficient Regular Expression Complexity in nth-check

nth-check is vulnerable to Inefficient Regular Expression Complexity

7.5	5	52
CVSS3 High	CVSS2 Medium	debAI

Introduced through

web-ui/package-lock.json 1 file

```
@angular-devkit/build-angular@ 0.1000.8          View in file
  cssnano@ 4.1.10
    ...
      css-select@ 2.1.0
        nth-check ✖ 1.0.2
```

Vulnerable dependency nth-check (npm)

v 2.0.1 ✓

CVSS Details ⓘ About CVSS

CVSS2	5 Medium
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Partial

CVSS3	7.5 High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

References ⓘ About references

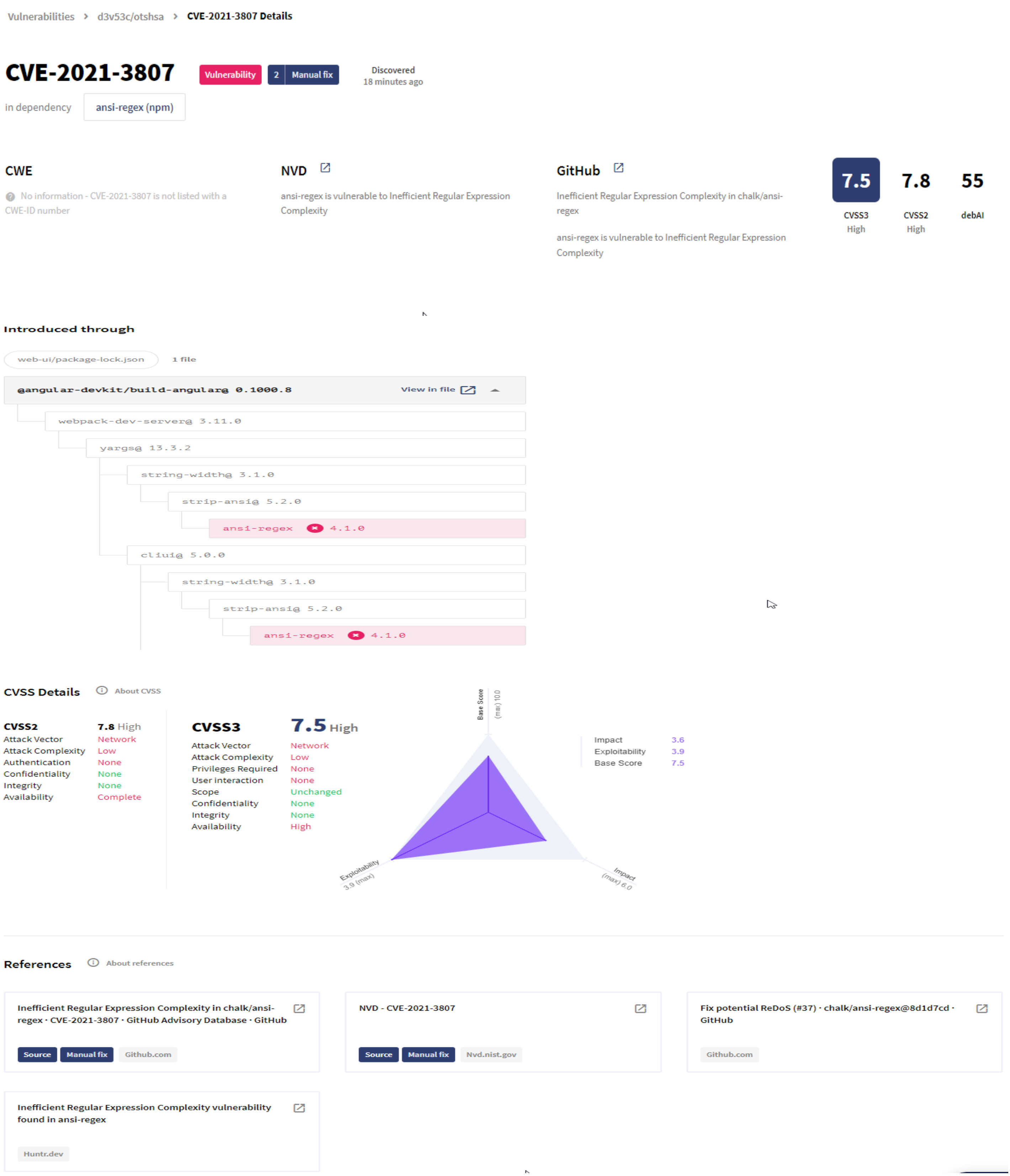
NVD - CVE-2021-3803
Source | Manual fix | Nvd.nist.gov

Inefficient Regular Expression Complexity in nth-check · CVE-2021-3803 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

fix(parse): Replace regex with hand-rolled parser (#9) · fb55/nth-check@9894c1d · GitHub
Github.com

Inefficient Regular Expression Complexity vulnerability found in nth-check
Huntr.dev

CVE-2021-3807 - ansi-regex (npm)



Vulnerabilities › d3v53c/otshsa › **CVE-2021-3807 Details**

CVE-2021-3807

Vulnerability | 2 Manual fix

Discovered
18 minutes ago

in dependency ansi-regex (npm)

CWE

No information - CVE-2021-3807 is not listed with a CWE-ID number

NVD

ansi-regex is vulnerable to Inefficient Regular Expression Complexity

GitHub

Inefficient Regular Expression Complexity in chalk/ansi-regex

ansi-regex is vulnerable to Inefficient Regular Expression Complexity

7.5	7.8	55
CVSS3 High	CVSS2 High	debAI

Introduced through

web-ui/package-lock.json   1 file

```
@angular-devkit/build-angular@ 0.1000.8                View in file
    webpack-dev-server@ 3.11.0
        yargs@ 13.3.2
            string-width@ 3.1.0
                strip-ansi@ 5.2.0
                    ansi-regex ✖ 4.1.0
            cliui@ 5.0.0
                string-width@ 3.1.0
                    strip-ansi@ 5.2.0
                        ansi-regex ✖ 4.1.0
```

CVSS Details   About CVSS

CVSS2	**7.8** High
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Complete

CVSS3	**7.5** High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

References   About references

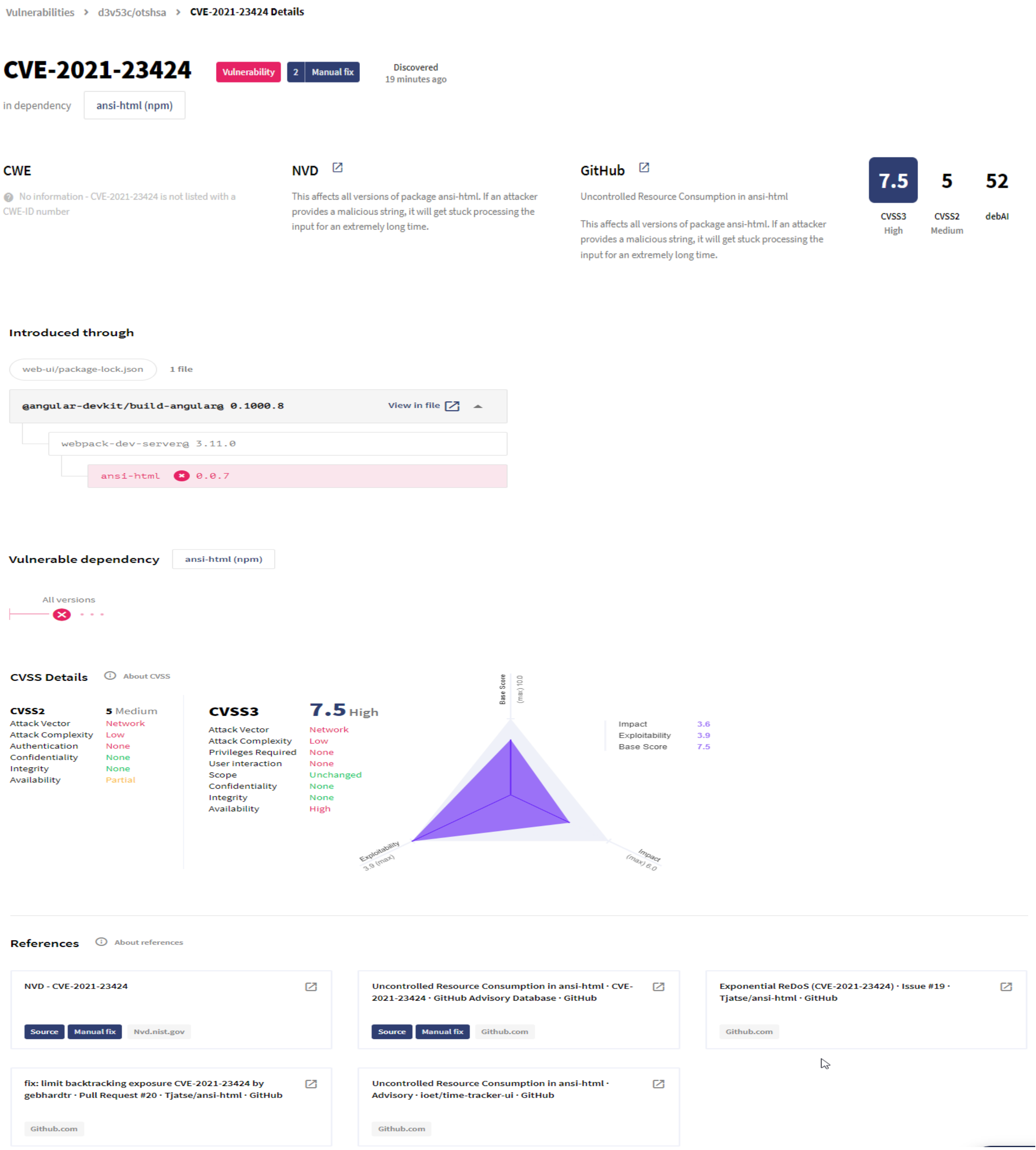
Inefficient Regular Expression Complexity in chalk/ansi-regex · CVE-2021-3807 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

NVD - CVE-2021-3807
Source | Manual fix | Nvd.nist.gov

Fix potential ReDoS (#37) · chalk/ansi-regex@8d1d7cd · GitHub
Github.com

Inefficient Regular Expression Complexity vulnerability found in ansi-regex
Huntr.dev

CVE-2021-23424 - ansi-html (npm)



Vulnerabilities › d3v53c/otshsa › **CVE-2021-23424 Details**

CVE-2021-23424

Vulnerability | 2 Manual fix | Discovered
19 minutes ago

in dependency ansi-html (npm)

CWE

No information - CVE-2021-23424 is not listed with a CWE-ID number

NVD

This affects all versions of package ansi-html. If an attacker provides a malicious string, it will get stuck processing the input for an extremely long time.

GitHub

Uncontrolled Resource Consumption in ansi-html

This affects all versions of package ansi-html. If an attacker provides a malicious string, it will get stuck processing the input for an extremely long time.

7.5	5	52
CVSS3 High	CVSS2 Medium	debAI

Introduced through

web-ui/package-lock.json 1 file

```
@angular-devkit/build-angular@ 0.1000.8          View in file
    webpack-dev-server@ 3.11.0
        ansi-html ✕ 0.0.7
```

Vulnerable dependency ansi-html (npm)

All versions ✕

CVSS Details About CVSS

CVSS2	**5** Medium
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Partial

CVSS3	**7.5** High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

References About references

NVD - CVE-2021-23424
Source | Manual fix | Nvd.nist.gov

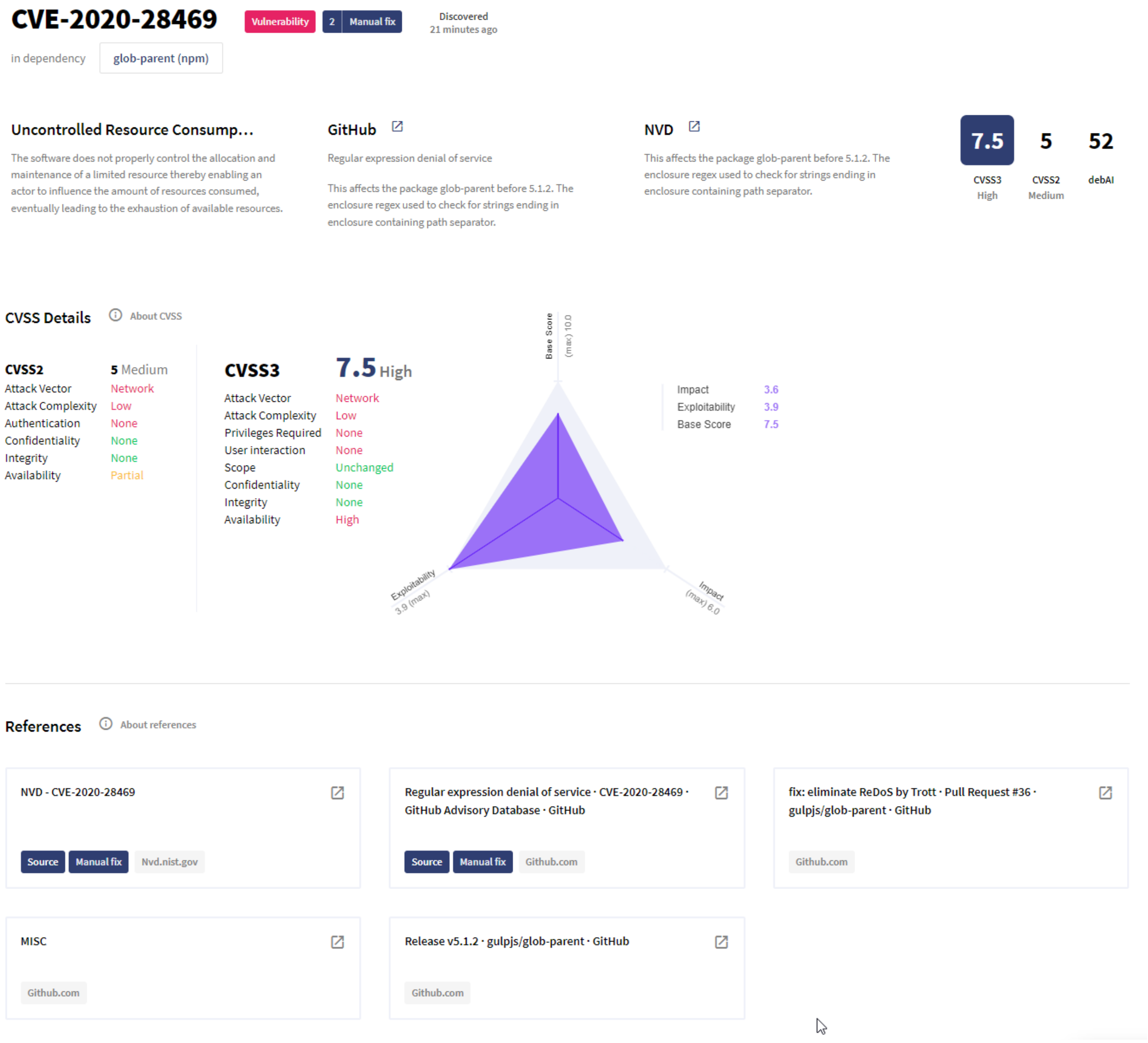
Uncontrolled Resource Consumption in ansi-html · CVE-2021-23424 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

Exponential ReDoS (CVE-2021-23424) · Issue #19 · Tjatse/ansi-html · GitHub
Github.com

fix: limit backtracking exposure CVE-2021-23424 by gebhardtr · Pull Request #20 · Tjatse/ansi-html · GitHub
Github.com

Uncontrolled Resource Consumption in ansi-html · Advisory · ioet/time-tracker-ui · GitHub
Github.com

CVE-2020-28469 - glob-parent (npm)



CVE-2020-28469

Vulnerability 2 Manual fix

Discovered
21 minutes ago

in dependency glob-parent (npm)

Uncontrolled Resource Consump...

The software does not properly control the allocation and maintenance of a limited resource thereby enabling an actor to influence the amount of resources consumed, eventually leading to the exhaustion of available resources.

GitHub

Regular expression denial of service

This affects the package glob-parent before 5.1.2. The enclosure regex used to check for strings ending in enclosure containing path separator.

NVD

This affects the package glob-parent before 5.1.2. The enclosure regex used to check for strings ending in enclosure containing path separator.

7.5	5	52
CVSS3 High	CVSS2 Medium	debAI

CVSS Details

About CVSS

CVSS2	5 Medium
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Partial

CVSS3	7.5 High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

References

About references

NVD - CVE-2020-28469
Source Manual fix Nvd.nist.gov

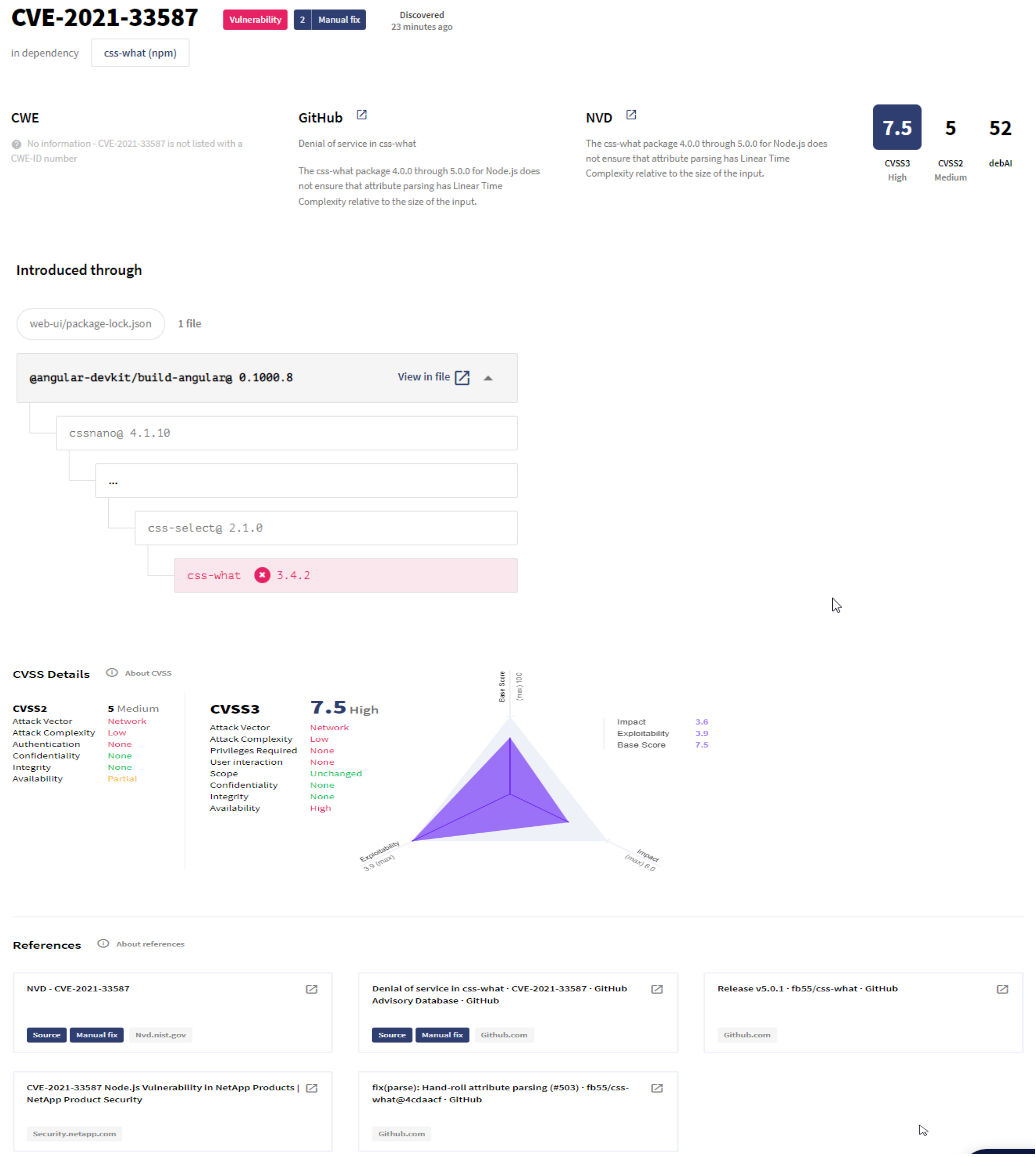
Regular expression denial of service · CVE-2020-28469 · GitHub Advisory Database · GitHub
Source Manual fix Github.com

fix: eliminate ReDoS by Trott · Pull Request #36 · gulpjs/glob-parent · GitHub
Github.com

MISC
Github.com

Release v5.1.2 · gulpjs/glob-parent · GitHub
Github.com

CVE-2020-28469 - glob-parent (npm)



CVE-2021-33587

Vulnerability | 2 | Manual fix

Discovered
23 minutes ago

in dependency css-what (npm)

CWE

No information - CVE-2021-33587 is not listed with a CWE-ID number

GitHub

Denial of service in css-what

The css-what package 4.0.0 through 5.0.0 for Node.js does not ensure that attribute parsing has Linear Time Complexity relative to the size of the input.

NVD

The css-what package 4.0.0 through 5.0.0 for Node.js does not ensure that attribute parsing has Linear Time Complexity relative to the size of the input.

7.5	5	52
CVSS3 High	CVSS2 Medium	debAI

Introduced through

web-ui/package-lock.json 1 file

```
@angular-devkit/build-angular@ 0.1000.8        View in file
  cssnano@ 4.1.10
    ...
      css-select@ 2.1.0
        css-what ✖ 3.4.2
```

CVSS Details ⓘ About CVSS

CVSS2	5 Medium
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Partial

CVSS3	7.5 High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

References ⓘ About references

NVD - CVE-2021-33587
Source | Manual fix | Nvd.nist.gov

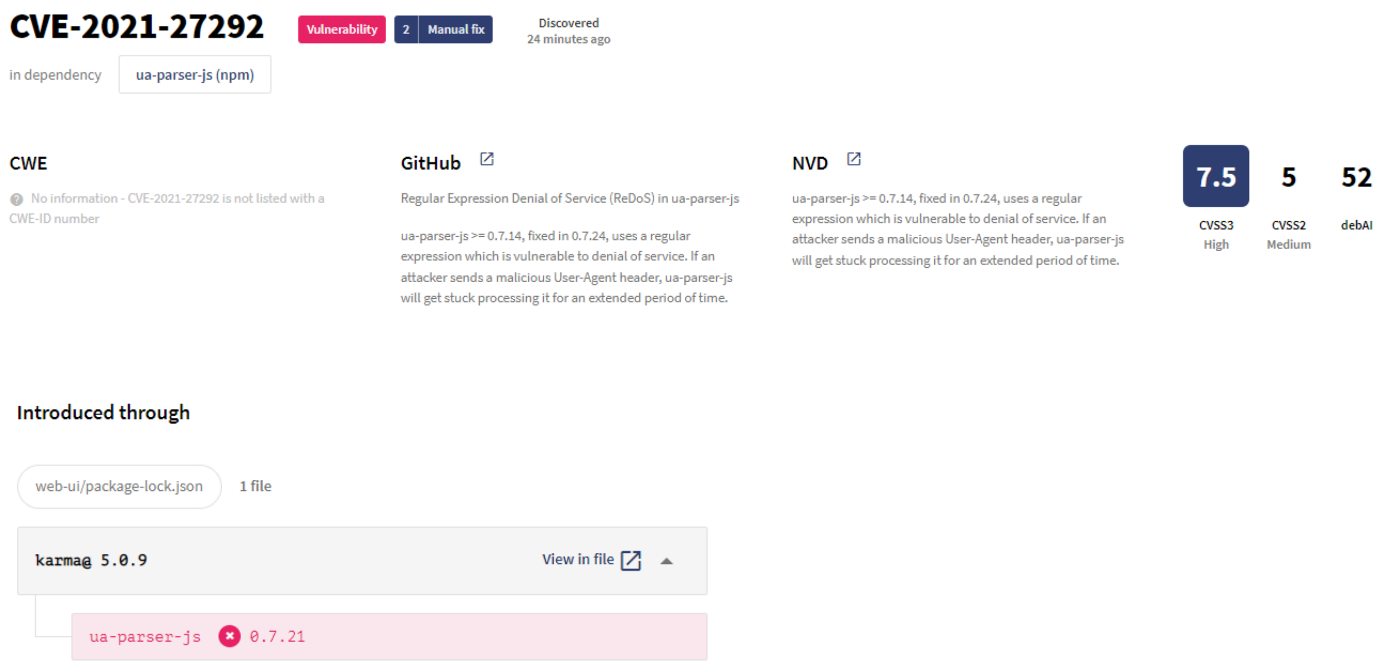
Denial of service in css-what · CVE-2021-33587 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

Release v5.0.1 · fb55/css-what · GitHub
Github.com

CVE-2021-33587 Node.js Vulnerability in NetApp Products | NetApp Product Security
Security.netapp.com

fix(parse): Hand-roll attribute parsing (#503) · fb55/css-what@4cdaacf · GitHub
Github.com

CVE-2020-28469 - glob-parent (npm)



CVE-2021-27292

Vulnerability | 2 Manual fix

Discovered
24 minutes ago

in dependency ua-parser-js (npm)

CWE

No information - CVE-2021-27292 is not listed with a CWE-ID number

GitHub

Regular Expression Denial of Service (ReDoS) in ua-parser-js

ua-parser-js >= 0.7.14, fixed in 0.7.24, uses a regular expression which is vulnerable to denial of service. If an attacker sends a malicious User-Agent header, ua-parser-js will get stuck processing it for an extended period of time.

NVD

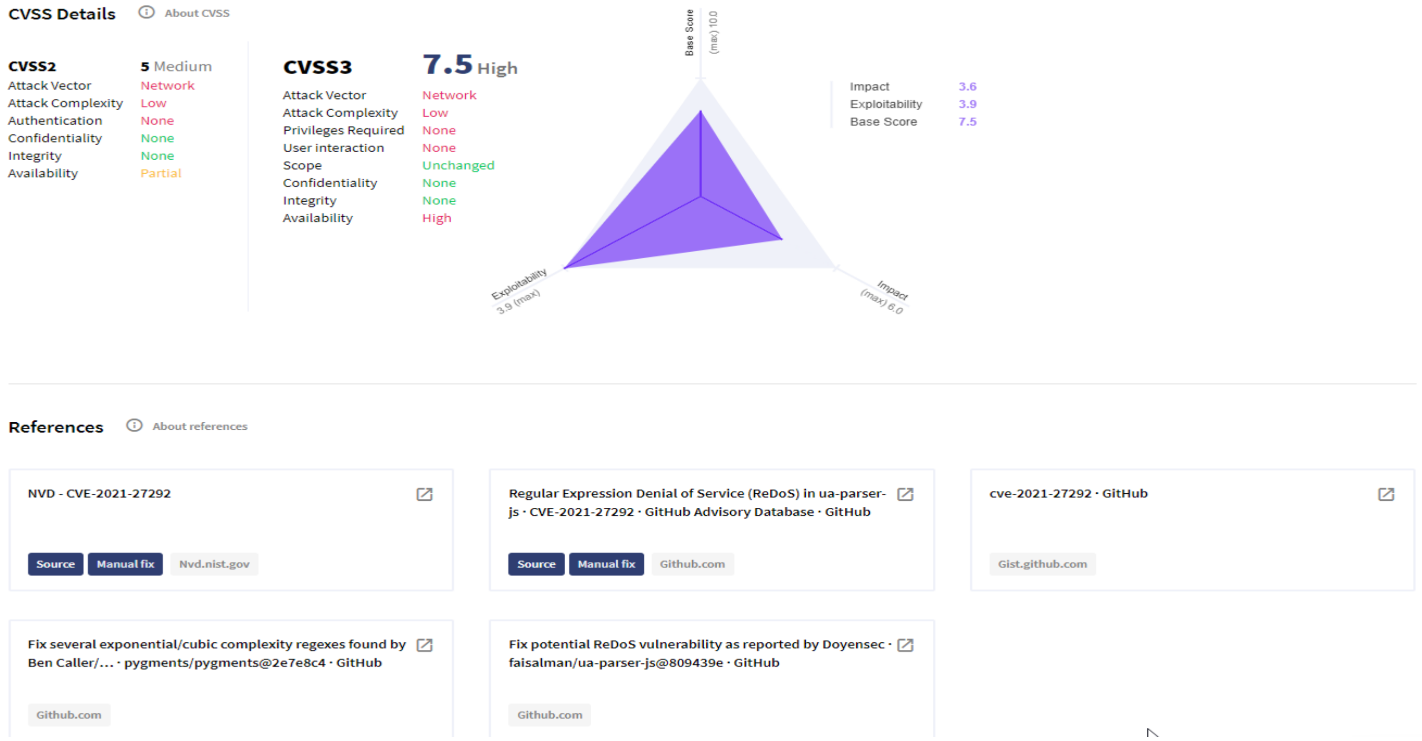
ua-parser-js >= 0.7.14, fixed in 0.7.24, uses a regular expression which is vulnerable to denial of service. If an attacker sends a malicious User-Agent header, ua-parser-js will get stuck processing it for an extended period of time.

7.5	5	52
CVSS3 High	CVSS2 Medium	debAI

Introduced through

web-ui/package-lock.json 1 file

```
karma@ 5.0.9                                  View in file
  ua-parser-js ✖ 0.7.21
```



CVSS Details ⓘ About CVSS

CVSS2	5 Medium
Attack Vector	Network
Attack Complexity	Low
Authentication	None
Confidentiality	None
Integrity	None
Availability	Partial

CVSS3	7.5 High
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

Impact	3.6
Exploitability	3.9
Base Score	7.5

References ⓘ About references

NVD - CVE-2021-27292
Source | Manual fix | Nvd.nist.gov

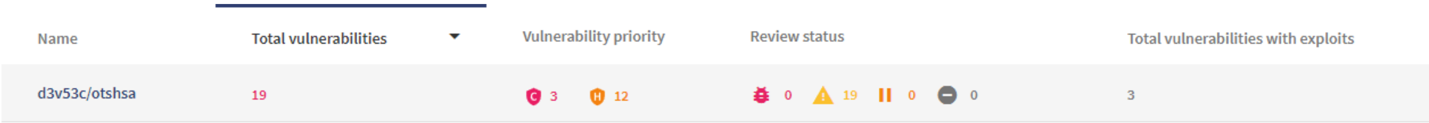
Regular Expression Denial of Service (ReDoS) in ua-parser-js · CVE-2021-27292 · GitHub Advisory Database · GitHub
Source | Manual fix | Github.com

cve-2021-27292 · GitHub
Gist.github.com

Fix several exponential/cubic complexity regexes found by Ben Caller/… · pygments/pygments@2e7e8c4 · GitHub
Github.com

Fix potential ReDoS vulnerability as reported by Doyensec · faisalman/ua-parser-js@809439e · GitHub
Github.com

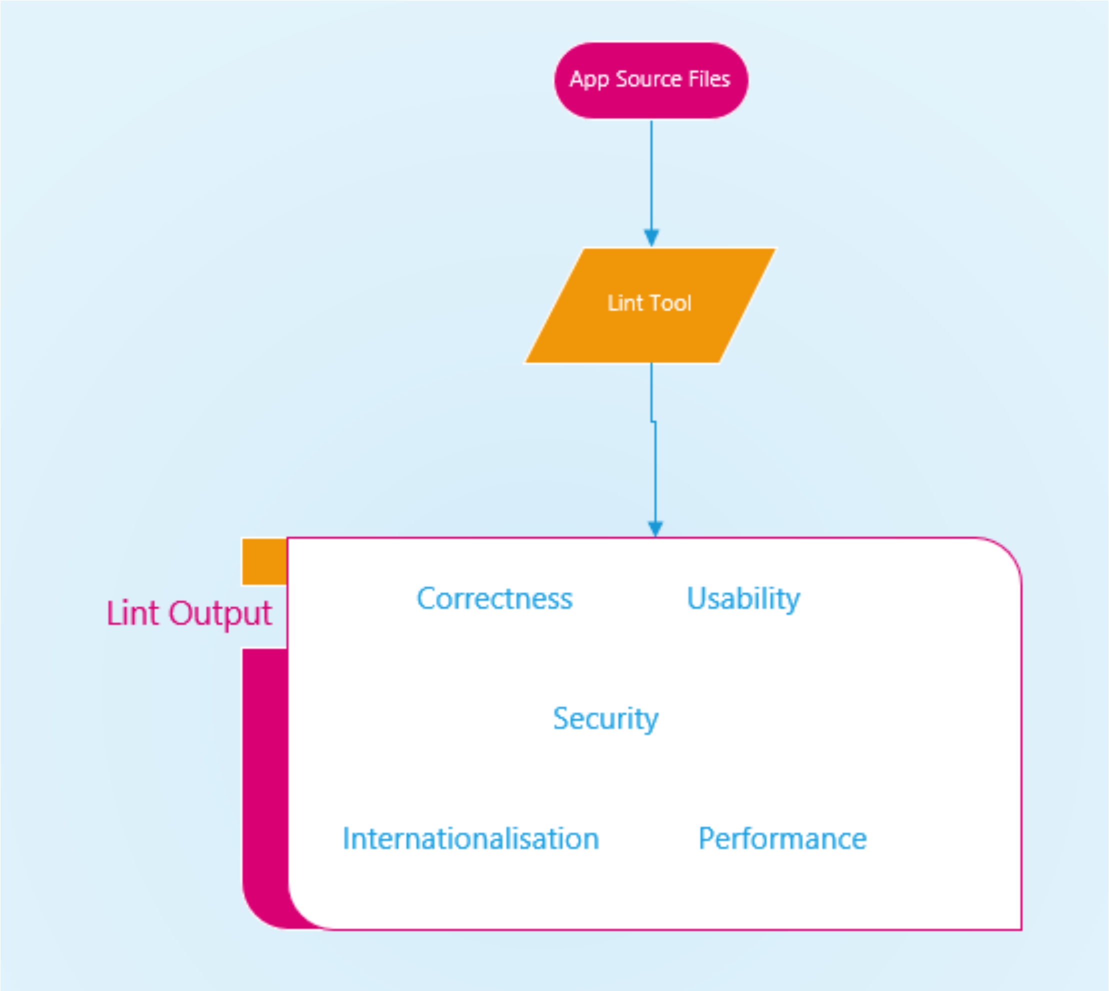
Summary



Name	Total vulnerabilities ▾	Vulnerability priority	Review status	Total vulnerabilities with exploits
d3v53c/otshsa	19	C 3 H 12	🐞 0 ⚠ 19 ⏸ 0 ⊖ 0	3

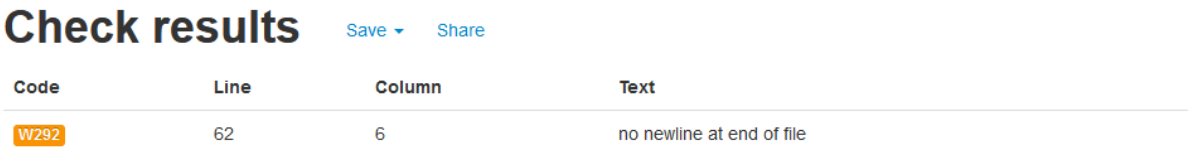
12. Code improvement with lint checks

In addition to ensuring the application meets functional requirements by building tests, it was essential to ensure that the code has no structural problems by running it through lints. Lints were used for identifying and correcting issues with the structural quality of the code. Critical improvements were made for correctness, security, performance, usability, and internationalization.

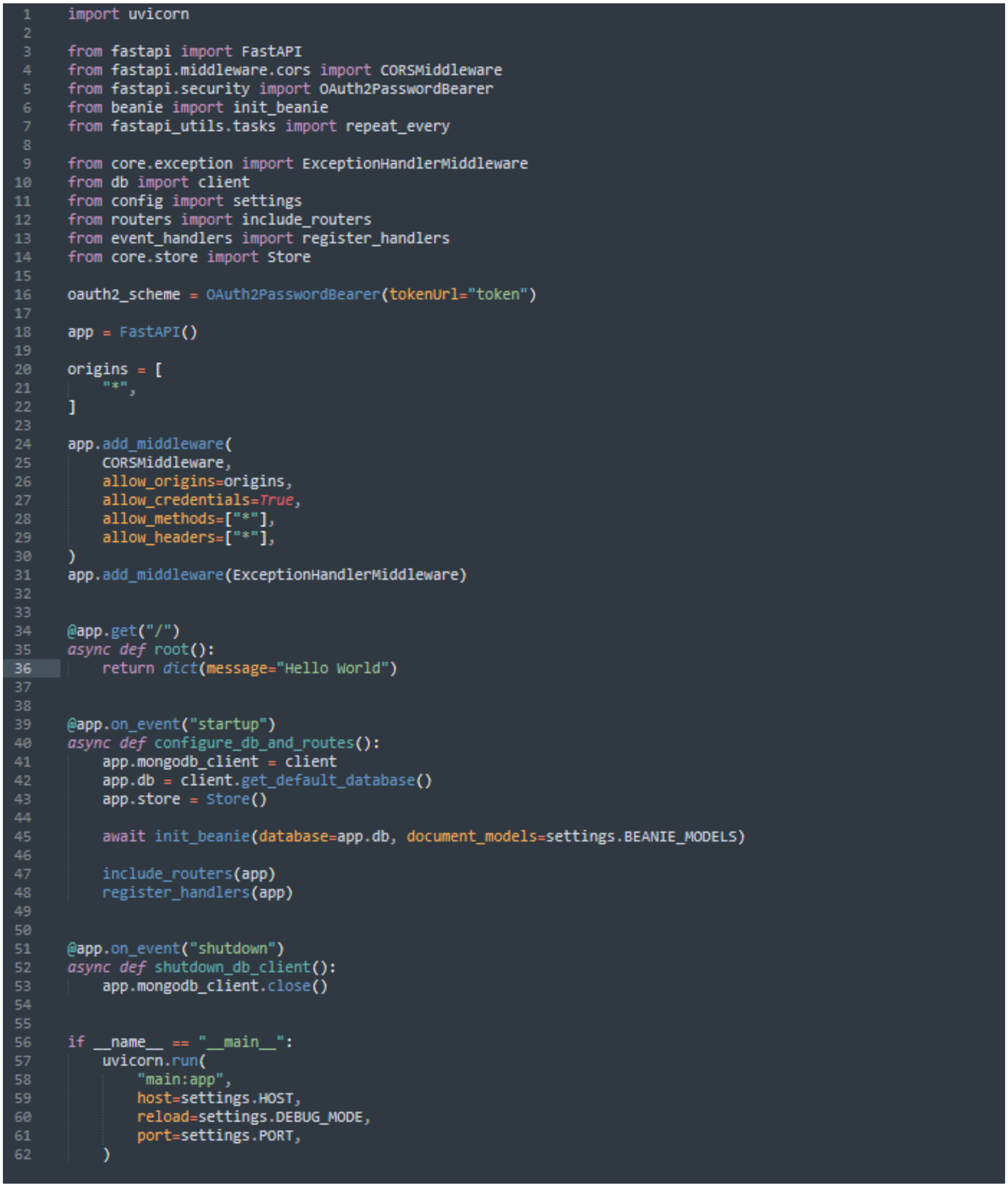


13.1. Main (main.py)

13.1.1 Lint Output before rectification.

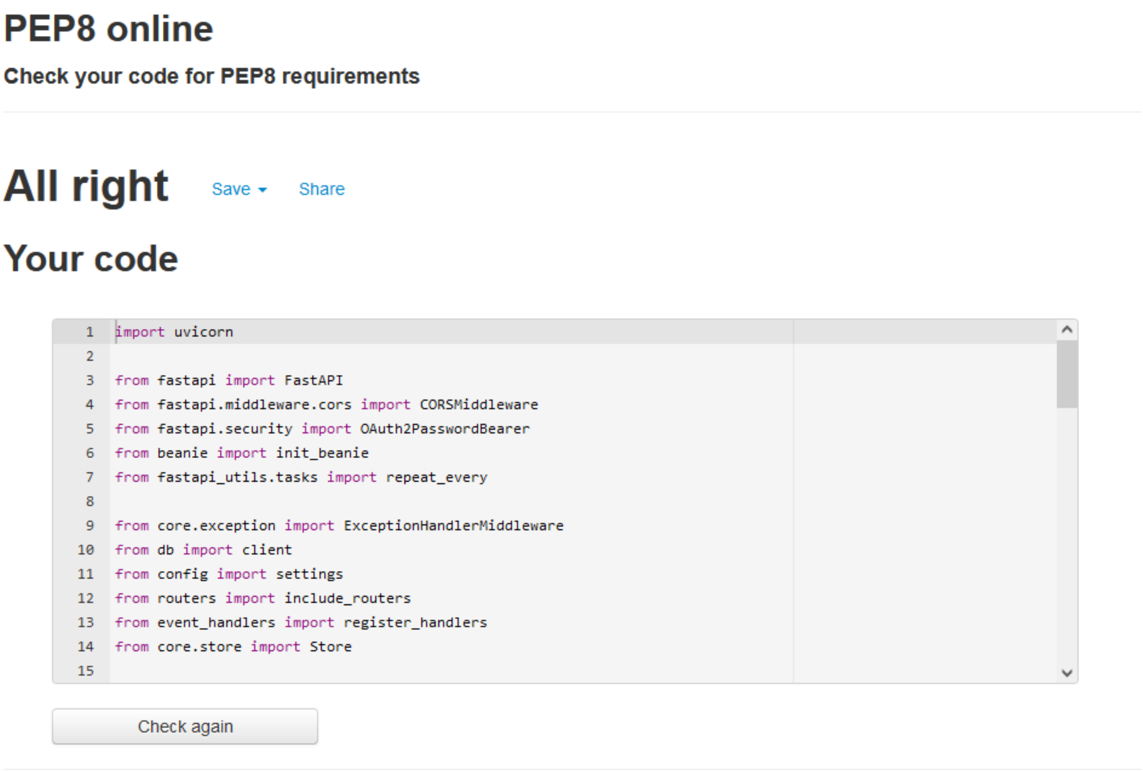


Code	Line	Column	Text
W292	62	6	no newline at end of file



```python
import uvicorn

from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from fastapi.security import OAuth2PasswordBearer
from beanie import init_beanie
from fastapi_utils.tasks import repeat_every

from core.exception import ExceptionHandlerMiddleware
from db import client
from config import settings
from routers import include_routers
from event_handlers import register_handlers
from core.store import Store

oauth2_scheme = OAuth2PasswordBearer(tokenUrl="token")

app = FastAPI()

origins = [
    "*",
]

app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
app.add_middleware(ExceptionHandlerMiddleware)


@app.get("/")
async def root():
    return dict(message="Hello World")


@app.on_event("startup")
async def configure_db_and_routes():
    app.mongodb_client = client
    app.db = client.get_default_database()
    app.store = Store()

    await init_beanie(database=app.db, document_models=settings.BEANIE_MODELS)

    include_routers(app)
    register_handlers(app)


@app.on_event("shutdown")
async def shutdown_db_client():
    app.mongodb_client.close()


if __name__ == "__main__":
    uvicorn.run(
        "main:app",
        host=settings.HOST,
        reload=settings.DEBUG_MODE,
        port=settings.PORT,
    )
```

13.1.2 Lint output post rectification.



PEP8 online

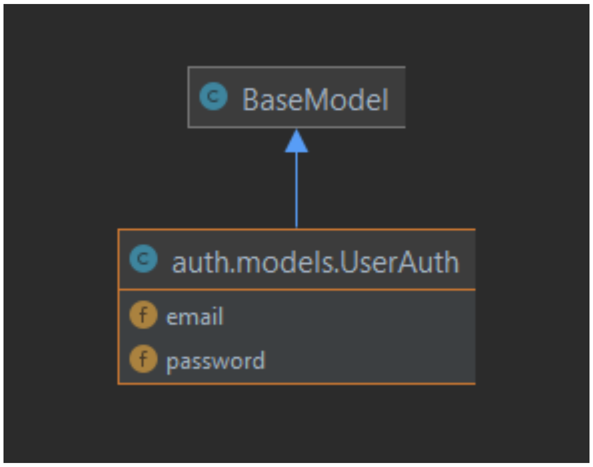
Check your code for PEP8 requirements

All right

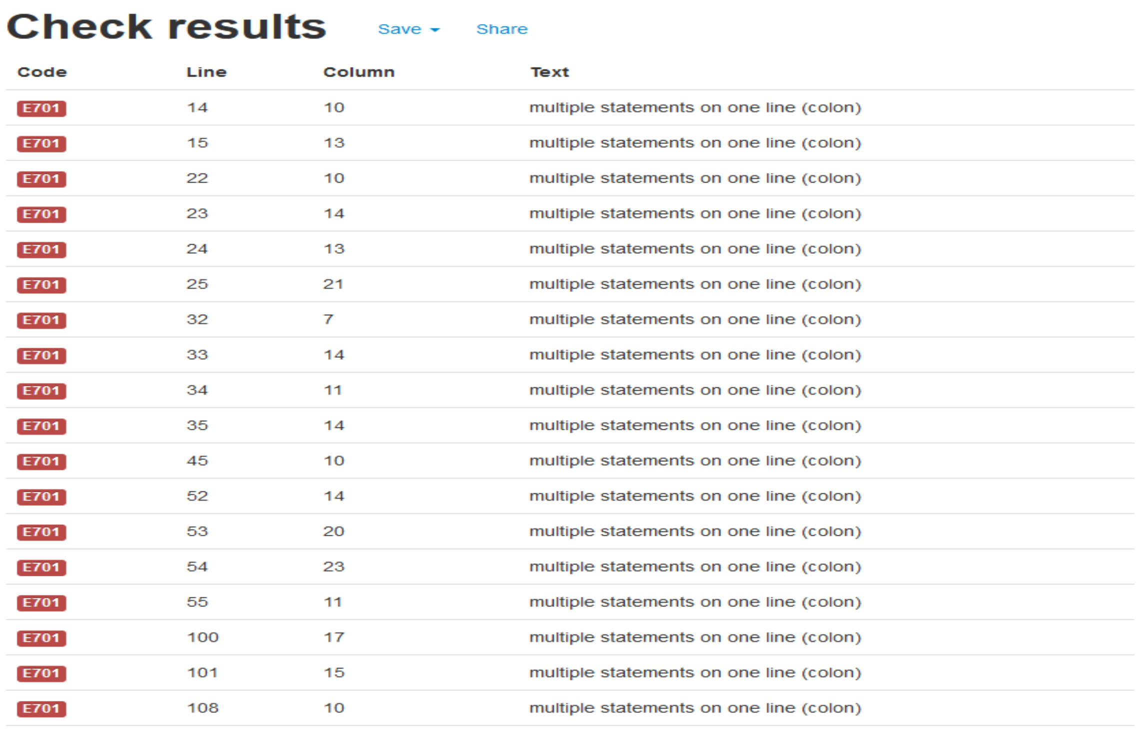
Your code

```
import uvicorn

from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from fastapi.security import OAuth2PasswordBearer
from beanie import init_beanie
from fastapi_utils.tasks import repeat_every

from core.exception import ExceptionHandlerMiddleware
from db import client
from config import settings
from routers import include_routers
from event_handlers import register_handlers
from core.store import Store

```

13.2. Models (models.py)



13.2.1 Lint Output before rectification.



Check results

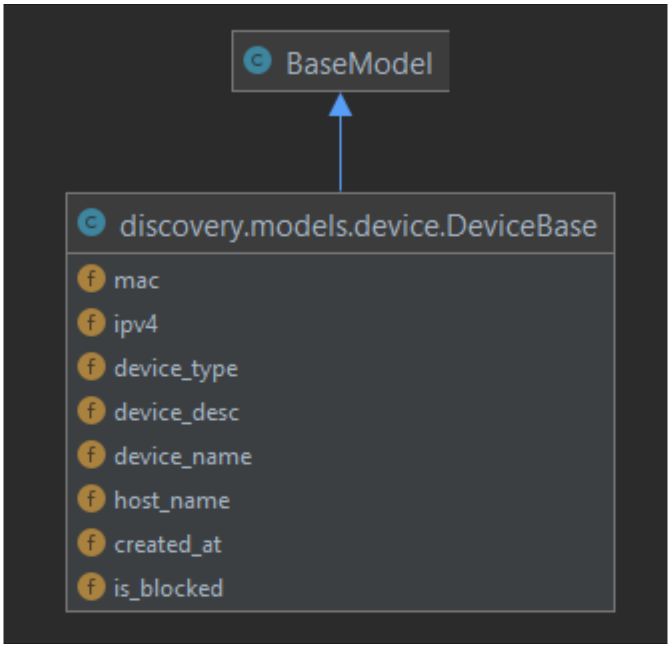
Save ▾ Share

Code	Line	Column	Text
E701	14	10	multiple statements on one line (colon)
E701	15	13	multiple statements on one line (colon)
E701	22	10	multiple statements on one line (colon)
E701	23	14	multiple statements on one line (colon)
E701	24	13	multiple statements on one line (colon)
E701	25	21	multiple statements on one line (colon)
E701	32	7	multiple statements on one line (colon)
E701	33	14	multiple statements on one line (colon)
E701	34	11	multiple statements on one line (colon)
E701	35	14	multiple statements on one line (colon)
E701	45	10	multiple statements on one line (colon)
E701	52	14	multiple statements on one line (colon)
E701	53	20	multiple statements on one line (colon)
E701	54	23	multiple statements on one line (colon)
E701	55	11	multiple statements on one line (colon)
E701	100	17	multiple statements on one line (colon)
E701	101	15	multiple statements on one line (colon)
E701	108	10	multiple statements on one line (colon)

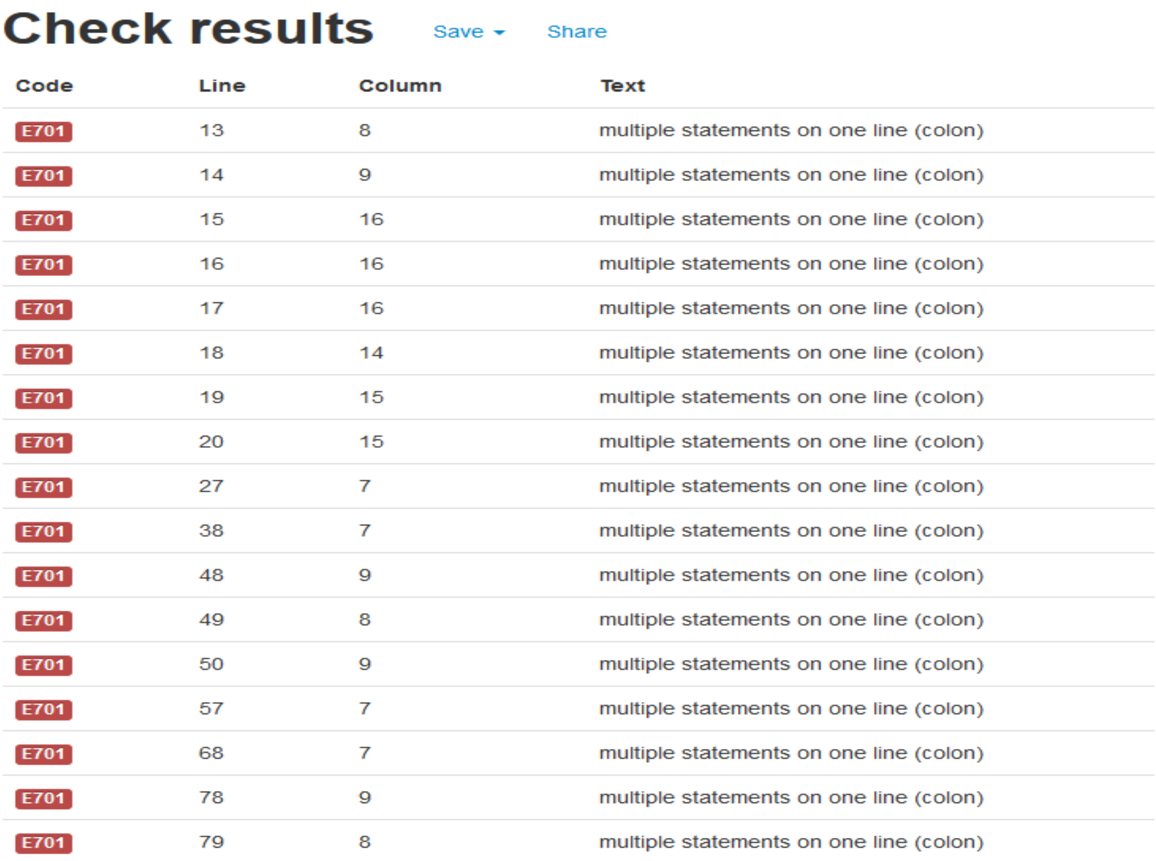
13.2.2 Lint output post rectification.

No rectification was performed on the code.

13.3. Device(device.py)



13.3.1 Lint Output before rectifications.

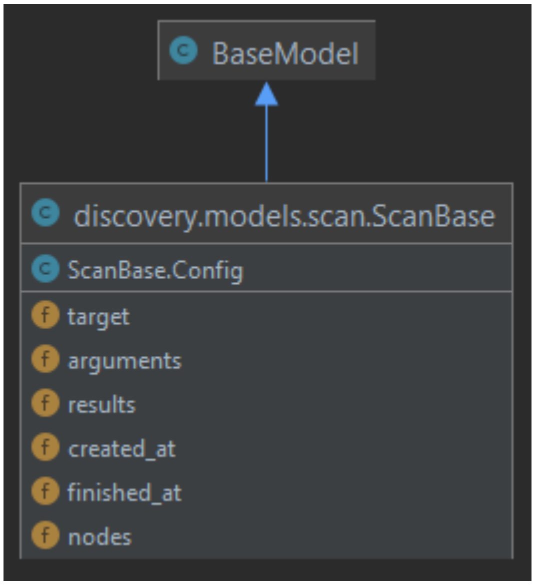


Check results

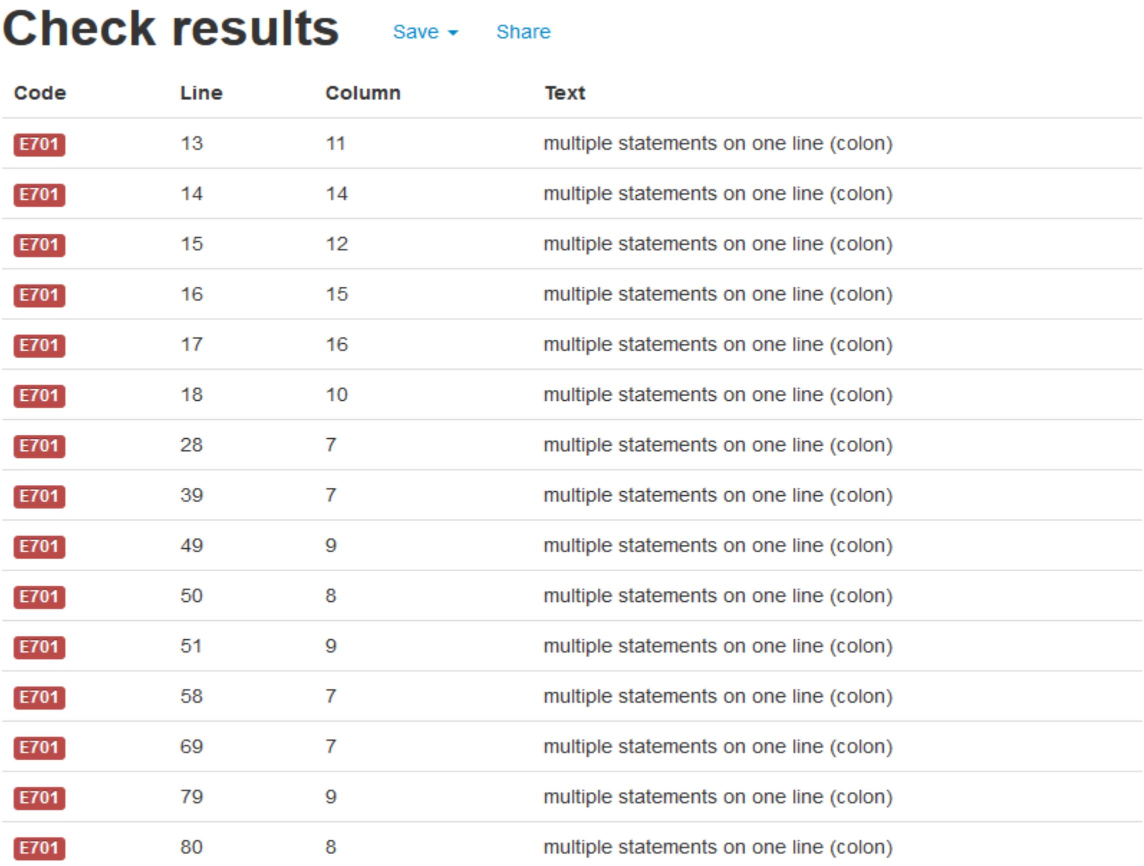
Code	Line	Column	Text
E701	13	8	multiple statements on one line (colon)
E701	14	9	multiple statements on one line (colon)
E701	15	16	multiple statements on one line (colon)
E701	16	16	multiple statements on one line (colon)
E701	17	16	multiple statements on one line (colon)
E701	18	14	multiple statements on one line (colon)
E701	19	15	multiple statements on one line (colon)
E701	20	15	multiple statements on one line (colon)
E701	27	7	multiple statements on one line (colon)
E701	38	7	multiple statements on one line (colon)
E701	48	9	multiple statements on one line (colon)
E701	49	8	multiple statements on one line (colon)
E701	50	9	multiple statements on one line (colon)
E701	57	7	multiple statements on one line (colon)
E701	68	7	multiple statements on one line (colon)
E701	78	9	multiple statements on one line (colon)
E701	79	8	multiple statements on one line (colon)

3.2 Lint output post rectifications.
No rectification was performed on the code.

13.4. Scan(scan.py)



13.4.1 Lint Output before rectifications.



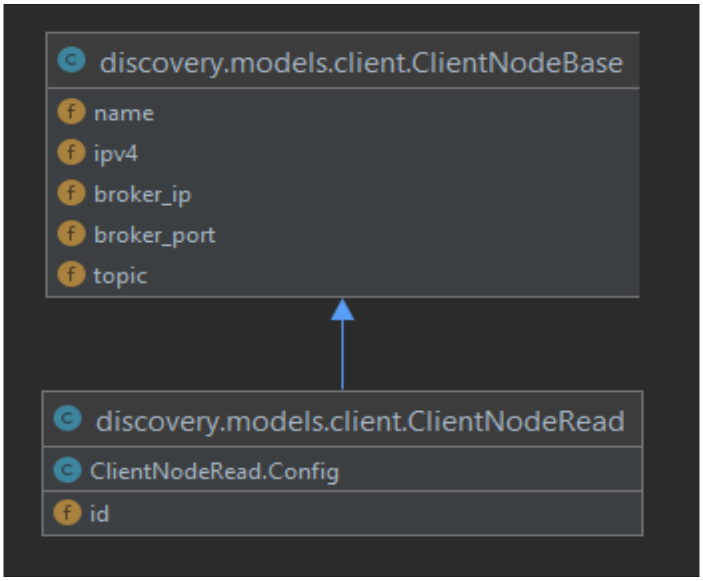
Check results

Code	Line	Column	Text
E701	13	11	multiple statements on one line (colon)
E701	14	14	multiple statements on one line (colon)
E701	15	12	multiple statements on one line (colon)
E701	16	15	multiple statements on one line (colon)
E701	17	16	multiple statements on one line (colon)
E701	18	10	multiple statements on one line (colon)
E701	28	7	multiple statements on one line (colon)
E701	39	7	multiple statements on one line (colon)
E701	49	9	multiple statements on one line (colon)
E701	50	8	multiple statements on one line (colon)
E701	51	9	multiple statements on one line (colon)
E701	58	7	multiple statements on one line (colon)
E701	69	7	multiple statements on one line (colon)
E701	79	9	multiple statements on one line (colon)
E701	80	8	multiple statements on one line (colon)

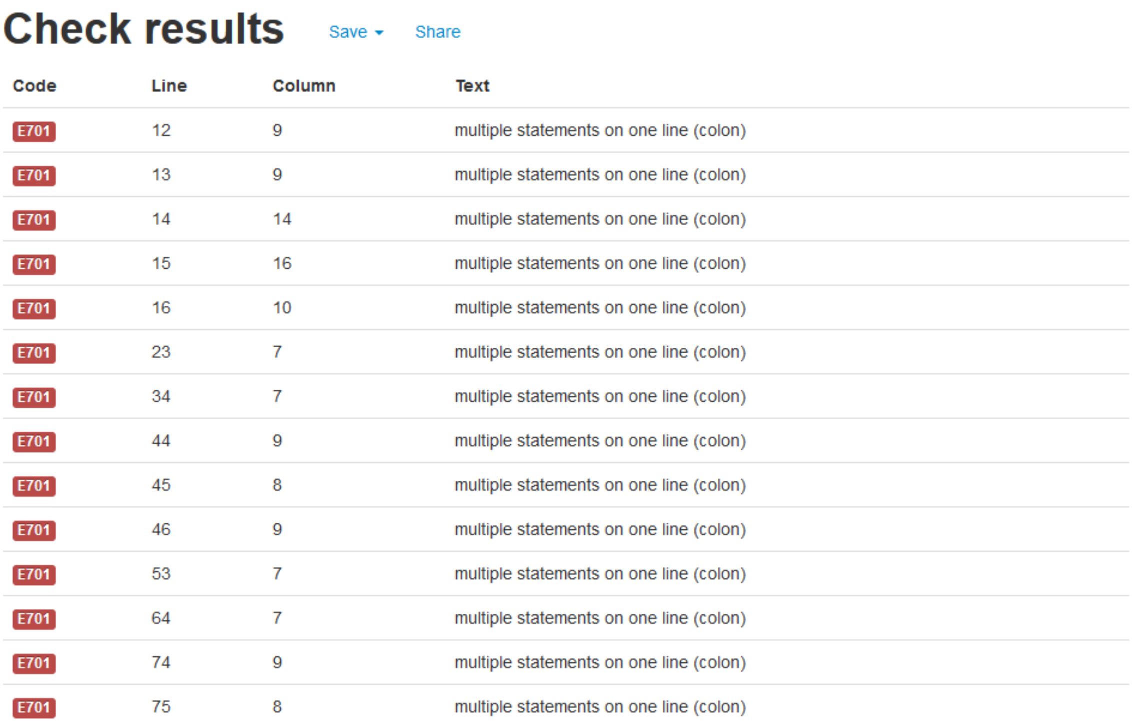
13.4.2 Lint output post rectifications.

No rectification was performed on the code.

13.5. Client Node(client.py)


discovery.models.client.ClientNodeBase
name
ipv4
broker_ip
broker_port
topic
discovery.models.client.ClientNodeRead
ClientNodeRead.Config
id

13.5.1 Lint Output before rectifications.



Check results

Code	Line	Column	Text
E701	12	9	multiple statements on one line (colon)
E701	13	9	multiple statements on one line (colon)
E701	14	14	multiple statements on one line (colon)
E701	15	16	multiple statements on one line (colon)
E701	16	10	multiple statements on one line (colon)
E701	23	7	multiple statements on one line (colon)
E701	34	7	multiple statements on one line (colon)
E701	44	9	multiple statements on one line (colon)
E701	45	8	multiple statements on one line (colon)
E701	46	9	multiple statements on one line (colon)
E701	53	7	multiple statements on one line (colon)
E701	64	7	multiple statements on one line (colon)
E701	74	9	multiple statements on one line (colon)
E701	75	8	multiple statements on one line (colon)

13.5.2 Lint output post rectifications.

Necessary rectification was performed on the code.

References:

1. Mqtt.org. 2021. *MQTT - The Standard for IoT Messaging*. [online] Available at: <https://mqtt.org> [Accessed 20 December 2021].

2. Docker Documentation. 2021. *Docker Documentation*. [online] Available at: <https://docs.docker.com/> [Accessed 20 December 2021].