

Usage of components with known security vulnerabilities

Part 1 – Selection of open-source project

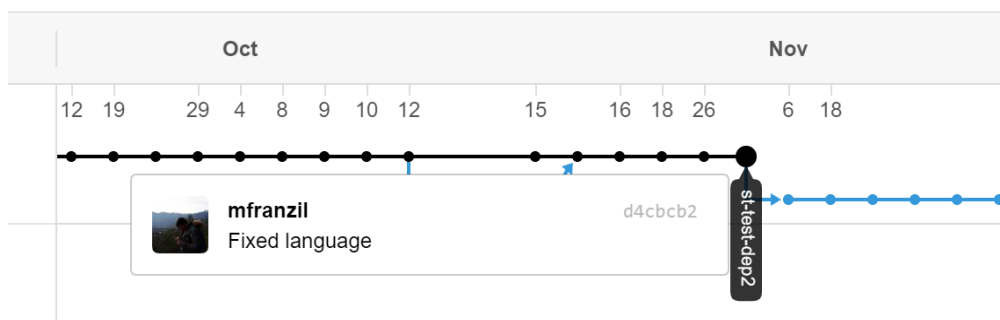
Firstly, I chose a suitable open-source project for this assignment. After having examined various public and well-known public projects, I instead chose to focus on a project I did by myself (with some third party collaboration). This project is a Python webapp functioning as a Telegram bot. The bot itself is tasked with sending daily messages in my student dormitory (NEST)'s kitchen chat, informing people on their daily cleaning turns. This bot has been refined, refactored and changed over time implementing several functionalities. Now, it sits under the InnovationTeamNEST umbrella repo:

<https://github.com/InnovationTeamNest/cooking-club-nest-bot>

Since 2018, all of the changes to the bot were made by me, including library updates. In fact, Dependabot has been active for a while on this repo, prompting me with emails in case vulnerabilities were discovered.

Part 2 – Project fork

As the current snapshot, the project offered no known vulnerabilities. I therefore chose to backtrack to commit d4cbcb2, one of the first featuring the requirements.txt file (needed by the dependency tracker)



but old enough to warrant a plethora of vulnerable libraries.

This commit was forked into its own repo (available under mfranzil/cooking-club-nest-bot). In this new repo, the dependency tracker and Dependabot were enabled.

Part 3 – Dependency graph

The next step was to visit the Dependency graph under the Insights tab.

The screenshot shows the GitHub repository page for mfranzil/cooking-club-nest-bot. The repository is forked from InnovationTeamNest/cooking-club-nest-bot. The Insights tab is selected, and the Dependency graph is displayed. The graph shows a warning: "We found potential security vulnerabilities in your dependencies. Dependencies defined in these manifest files have known security vulnerabilities and should be updated: requirements.txt 6 vulnerabilities found". A button "View Dependabot alerts" is visible. The page also shows a sidebar with navigation links: Pulse, Contributors, Traffic, Commits, Code frequency, Dependency graph (selected), Network, and Forks.

As the dependency graph would take pages to be fully visualized in this report, the following is a screenshot highlighting only the vulnerable packages found by Dependabot.

Dependencies defined in requirements.txt 37		
> pyca / cryptography	Known security vulnerability in 2.3.1	
httplib2 / httplib2	Known security vulnerability in 0.11.3	
▼ pallets / Jinja2	Known security vulnerability in 2.10	
> python-babel / babel		>= 0.8
> psf / requests	Known security vulnerability in 2.19.1	
urllib3 / urllib3	Known security vulnerability in 1.23	
▼ pallets / werkzeug	Known security vulnerability in 0.14.1	
> nedbat / coveragepy coverage		
> pytest-dev / pytest		
> sphinx-doc / sphinx		
> tox-dev / tox		

Part 4 – Dependabot

Visiting the Security > Dependabot tab yields the following view, once alerts are enabled:

6 Open ✓ 0 Closed		Sort ▼
🚩 cryptography	1 hour ago by GitHub requirements.txt	moderate severity
🚩 httplib2	1 hour ago by GitHub requirements.txt	low severity
🚩 werkzeug	1 hour ago by GitHub requirements.txt	high severity
🚩 urllib3	1 hour ago by GitHub requirements.txt	high severity
🚩 Jinja2	1 hour ago by GitHub requirements.txt	high severity
🚩 requests	1 hour ago by GitHub requirements.txt	moderate severity

Clicking a vulnerability opens a detailed page view, highlighting the issues, the suggested remedy, the details of the security breach and more.

Usually, the implementation of such security fixes is as simple as changing a line in the requirements file (such as requirements.txt in Python or pom.xml in Maven-based Java projects). Indeed, all vulnerabilities found in

the project were very minor, usually requiring the aforementioned “line fix”. Usually, library makers – although sometimes lazy – separate logically their updates in major, minor, and patch versions. Patch versions usually feature the biggest amount of security fixes, but usually do not come with API changes or significant alterations in the library’s behavior. This usually means that major and minor versions are the most difficult to update to, with diligent library makers often providing migration guides to support developers in their transition.

Either way, a healthy round of tests is absolutely needed no matter the update. While 99.9% of the times a library version change (speaking about patches) should not affect the code, something may always go wrong. This consequently results in more work to do the bigger the project.

The following pages contain screenshots of selected vulnerabilities for cooking-club-nest-bot. All of these require simple fixes in the requirements file.

werkzeug

Create Dependabot security updateDismiss

OpenGitHub opened this alert 2 hours ago

1 werkzeug vulnerability found in requirements.txt 2 hours ago

Remediation

Upgrade **werkzeug** to version **0.15.3** or later. For example:

```
werkzeug>=0.15.3
```

Always verify the validity and compatibility of suggestions with your codebase.

Details

CVE-2019-14806

high severity

Vulnerable versions: < 0.15.3
Patched version: 0.15.3

Pallets Werkzeug before 0.15.3, when used with Docker, has insufficient debugger PIN randomness because Docker containers share the same machine id.

cryptography

Create Dependabot security updateDismiss

OpenGitHub opened this alert 1 hour ago

1 cryptography vulnerability found in requirements.txt 1 hour ago

Remediation

Upgrade **cryptography** to version **3.2** or later. For example:

```
cryptography>=3.2
```

Always verify the validity and compatibility of suggestions with your codebase.

Details

GHSA-hggm-jpg3-v476

moderate severity

Vulnerable versions: < 3.2
Patched version: 3.2

Impact

RSA decryption was vulnerable to Bleichenbacher timing vulnerabilities, which would impact people using RSA decryption in online scenarios.

Patches

This is fixed in cryptography 3.2. [pyca/cryptography@58494b4](#) is the resolving commit.

requests

[Create Dependabot security update](#)[Dismiss](#)[Open](#) GitHub opened this alert 2 hours ago

1 requests vulnerability found in requirements.txt 2 hours ago

Remediation

Upgrade **requests** to version **2.20.0** or later. For example:

```
requests>=2.20.0
```

Always verify the validity and compatibility of suggestions with your codebase.

Details

CVE-2018-18074

moderate severity

Vulnerable versions: <= 2.19.1

Patched version: 2.20.0

The Requests package through 2.19.1 before 2018-09-14 for Python sends an HTTP Authorization header to an http URI upon receiving a same-hostname https-to-http redirect, which makes it easier for remote attackers to discover credentials by sniffing the network.

urllib3

[Create Dependabot security update](#)[Dismiss](#)[Open](#) GitHub opened this alert 2 hours ago

1 urllib3 vulnerability found in requirements.txt 2 hours ago

Remediation

Upgrade **urllib3** to version **1.24.2** or later. For example:

```
urllib3>=1.24.2
```

Always verify the validity and compatibility of suggestions with your codebase.

Details

CVE-2019-11324


high severity

Vulnerable versions: < 1.24.2

Patched version: 1.24.2

The urllib3 library before 1.24.2 for Python mishandles certain cases where the desired set of CA certificates is different from the OS store of CA certificates, which results in SSL connections succeeding in situations where a verification failure is the correct outcome. This is related to use of the ssl_context, ca_certs, or ca_certs_dir argument.

To verify that not only simple project like mine are affected by security breaches, I opted to fork another project, Jenkins. I backtracked several months, forked the project and visited Dependabot. I received 21 security alerts. Yet, most of them required little work just like the alerts on cooking-club-nest-bot. The following is a screenshot from the aforementioned fork.

 mfranzil / jenkins


forked from jenkinsci/jenkins


Watch 0


Star 0


Fork 6.6k


<> Code


 Pull requests

 Actions

 Projects

 Security 21

 Insights

 Settings

Overview

Security policy

Security advisories 0

Dependabot alerts 21


Code scanning alerts

Dependabot alerts

Dismiss all

21 Open 0 Closed


Sort

 ini

1 hour ago by GitHub

war/yarn.lock


low severity

 yargs-parser

1 hour ago by GitHub

war/yarn.lock

low severity

 dot-prop

high severity