Repository Design Prospectus for

The International Space Station and NASA

The B Team (Group 2)

Table of Contents

1. Introduction

Data sharing between the International Space Station (ISS) and NASA's Ground Control (GC) is a critical aspect of operations for mission success. This proposal outlines the secure design of a data-sharing repository and will discuss:

- Technical requirements
- Security concerns, mitigations
- Testing requirements
- Design schedule

The proposal is meant to provide a blueprint for implementation.

2. Data-sharing Repository

The proposed repository (Figure 1) is meant to achieve the following operations (Figure 2):

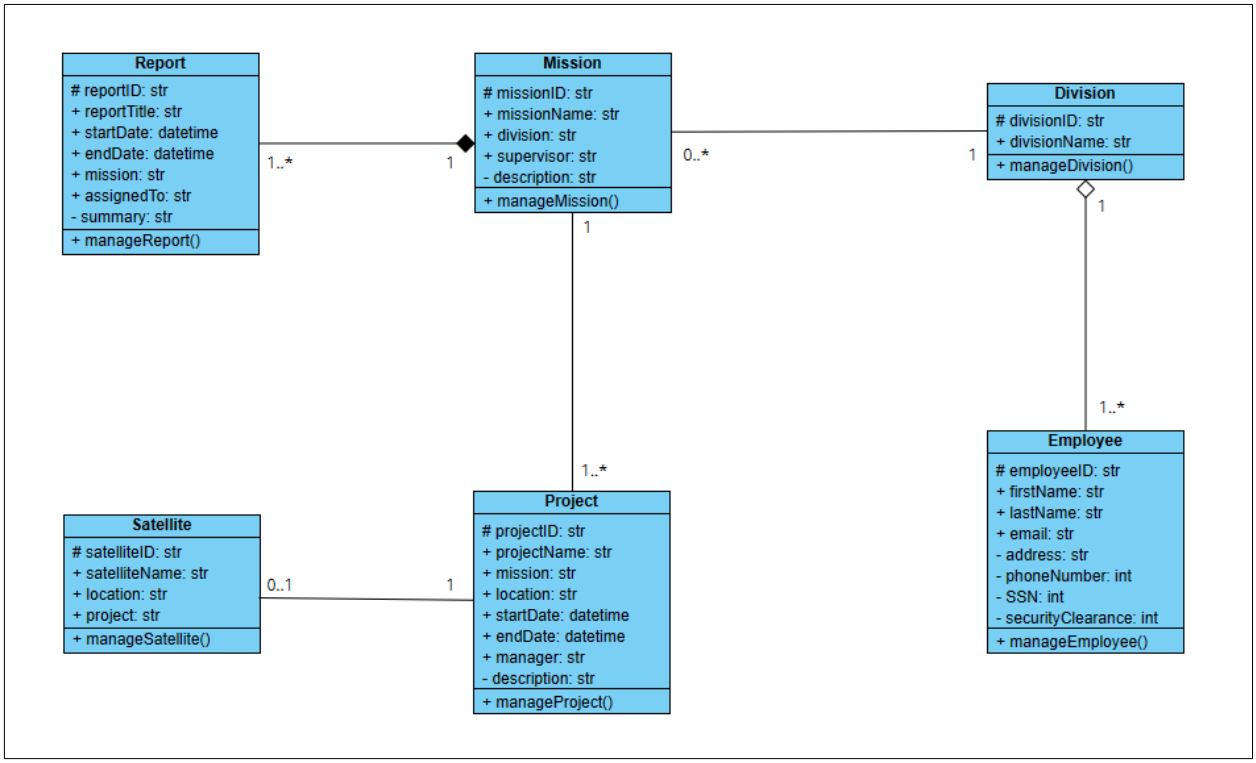

Figure 1: Repository Class Diagram

- Manage and store ISS satellite information

- Manage and store ISS mission information
- Generate and store ISS mission reports
- Share ISS reports with GC
- Deny access to unauthorized ISS partner agencies

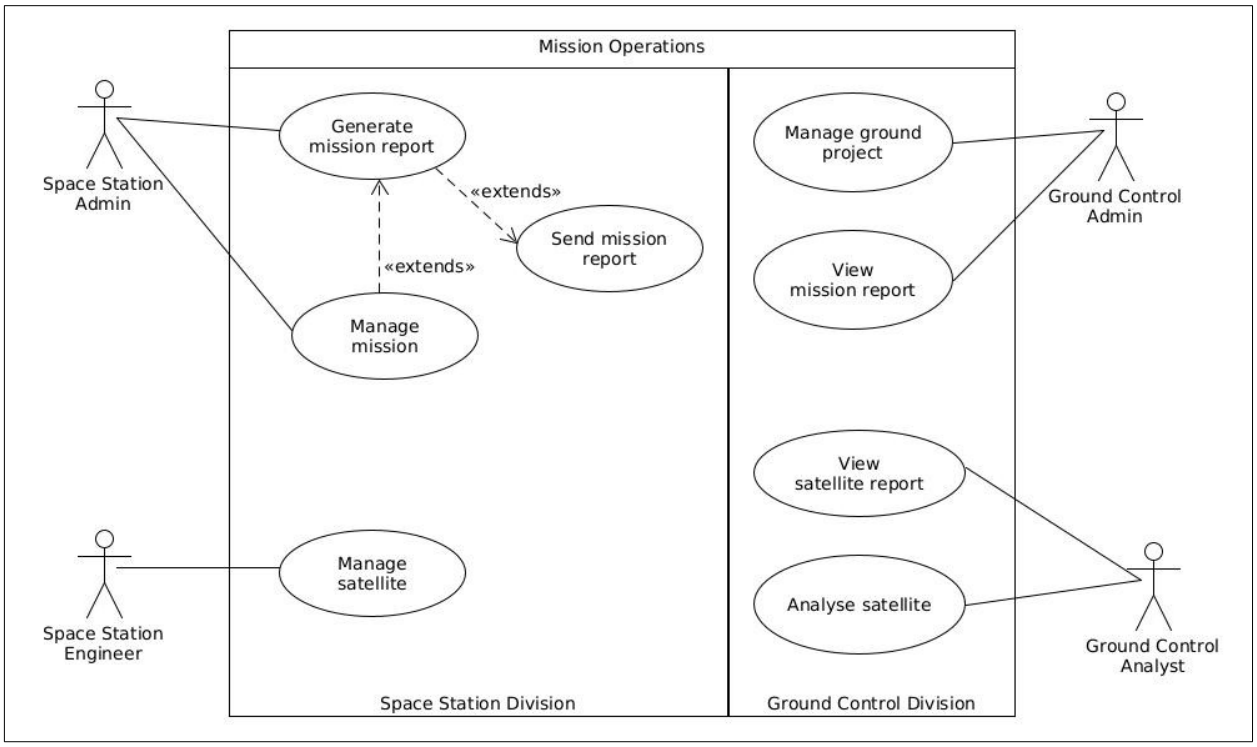

Figure 2: Repository Use Case Diagram

2.1 Security Concerns

Major ISS (IISTF, 2007) security concerns include

- Hardware/software design flaws
- Catastrophic system failures

while GC (NIST, 2018) concerns focus on data

- Confidentiality (GDPR, 2018a)
- Integrity (GDPR, 2018b)
- Availability (GDPR, 2018c)

These concerns inform the sections below.

3. Repository Requirements

3.1 Technical Requirements

Technical requirements have been chosen with secure agile developement in mind (Hof & Pohl, 2015), and include:

- Linux Ubuntu-Kernel OS
  - *SELinux* – enhanced security updates (Red Hat, 2019)
  - Virtual File System – complete access control (Gooch, 2005)
- Python libraries
  - sqlite3 (SQLite, 2022; Stribny, 2020; Synk, 2023a)
    - SQL database engine
    - Secure at version 5.1.5 or higher
  - logging (Brownlee, 2022; Python Software Foundation, 2023)
    - Event logging system
    - Thread-safe
- Local SOAP API (SmartBear, 2020)
  - Language/platform independent
  - Built-in error handling

3.2 Program Testing

The following functional testing methods (Jain, 2022) will be performed:

- Unit testing - each function/method/feature

- Integration/end-to-end testing (horizontal)

with appropriate python libraries applied (Table 1).

Table 1: Python Testing Libraries

Library	**Function**	**Test Type**	**Security Rating (Synk, 2023b)**	**Reference**
pylint	Static code analyser	Unit	97/100	PyPi, 2023
pytest	Python testing framework	Integration	97/100	Krekel, 2015
pandas	Data analysis/manipulation tool	Unit	93/100	NumFOCUS, 2023

3.3 Assumptions and Limitations

As the repository will be a monolithic prototype, runtime and data storage assumptions have been made:

- Data download requirements: 8GB RAM, 2MB/min
- CPU: quad-cores

The following limitations should be noted due to limited resources:

- Libraries are open-source
- Limited CPU/storage
- Command line testing only
    - Local ping for request/reply

4. Security Requirements

4.1 Vulnerabilities

Relevant vulnerabilities have been sourced from the CAPEC database (Table 2; Mitre, 2021). A possible repository breach sequence is diagrammed in *Figure 3*.

Table 2: Repository Vulnerabilities

Vulnerability	Consequence	Likelihood	Severity
Cookie tampering	Read/modify data, privilege elevation	High	High
Cross-site scripting	Unauthorised command execution	High	Very high
Denial-of-service	Unreliable execution	High	Medium
Logic defeat	Unauthorised command execution	High	High
XML injection	Read data, privilege elevation	High	n/a

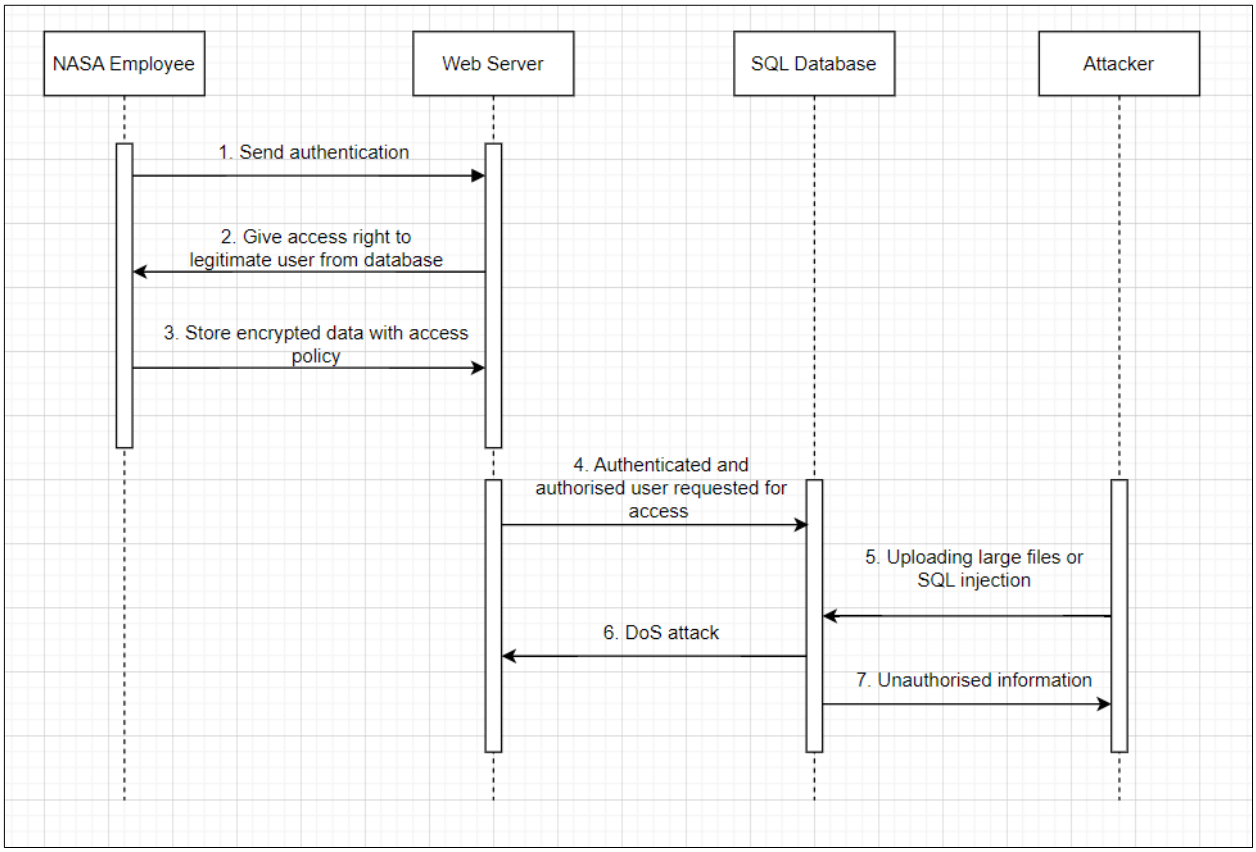

Figure 3: Repository Breach Sequence Diagram

4.2 Vulnerability Mitigations

Repository security has been conceived around three themes central to GDPR compliance (GDPR, 2018d; Pinto & Stuttard, 2011):

- Access control

- Two-factor-Authorization
- group-based restriction of application permissions

- Session management
  - HTTP cookies
  - Logging implementation

- Data Protection
  - Input sanitation
  - Customized error messages
  - Boundary Validation (Figure 4)
  - Encryption
    - Database – SQLite encryption extension (SQLite, n.d.)
    - HTTP cookies -- Datascript (Avi Networks, 2019)
  - Source code fortification (Table 3; Mitre, 2021)

Table 3: Attack-Specific Mitigations

Vulnerability	**Mitigation**
Cookie tampering	Validate input, generate MAC
Cross-site scripting	No client-side scripting/XHR proxy, enforce encoding
Denial-of-service	Configure scale limitations
Logic defeat	Validate input, create ‘allowlist’, avoid ‘GET’ request
XML injection	Validate input, customize error messages

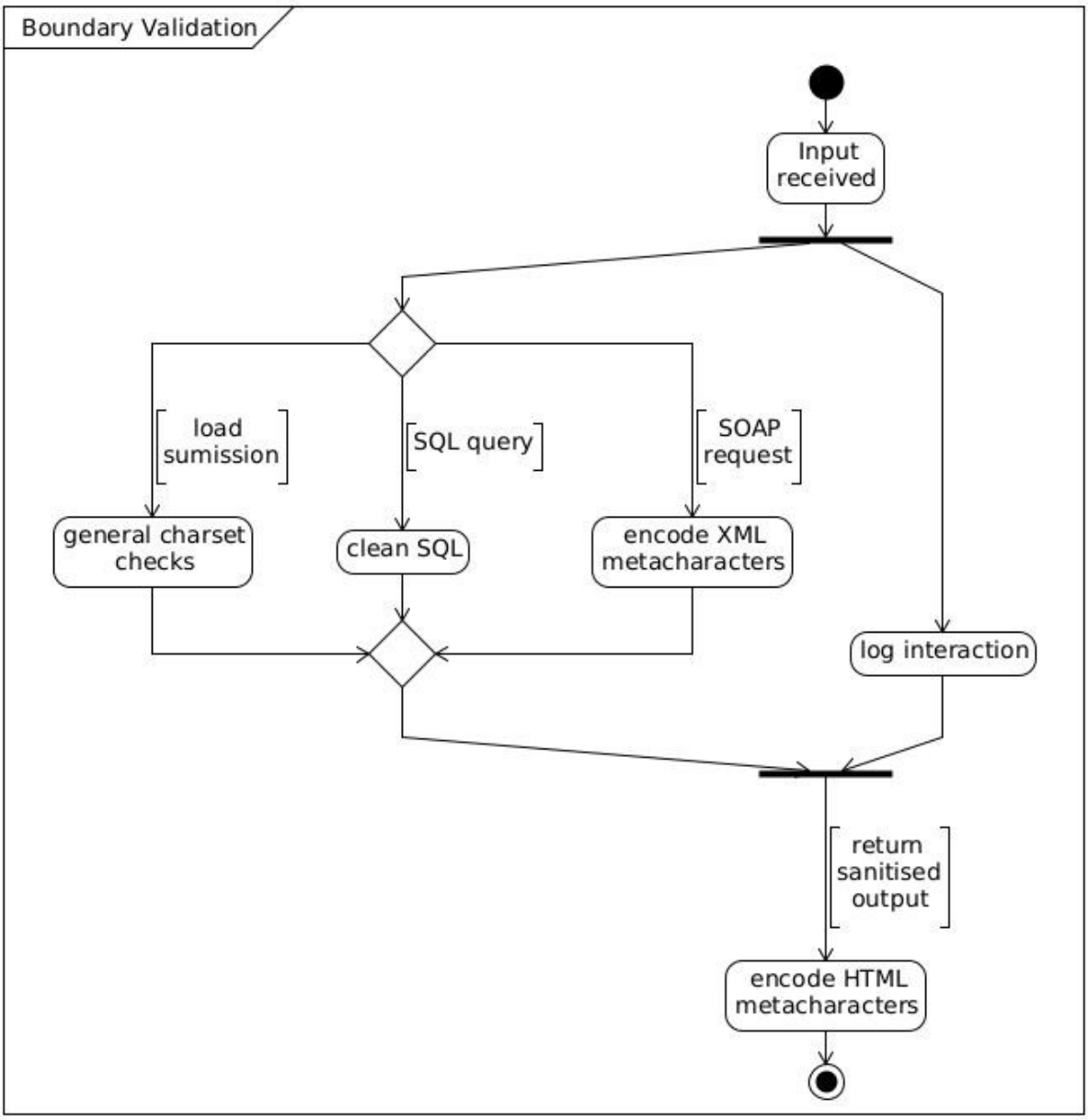

Figure 4: Boundary Validation Activity Diagram

4.3 Testing Requirements

Command line penetration testing shall be undertaken at unit and end-to-end levels with attack-specific malicious input (Table 4; Pinto & Stuttard, 2011; Stuttard, 2011; Tony, 2022).

Table 4: Malicious Code

Vulnerability	Malicious Code
Cookie tampering	Burpsuite decoder
Cross-site scripting	`"><script >alert(document.cookie)</script >`
Denial-of-service	hping
Logic defeat	`' OR 1=1--`
XML injection	`test<foo/>, <!--[...]!-->`

5. Design Schedule

Figure 5 proposes a schedule for project completion.

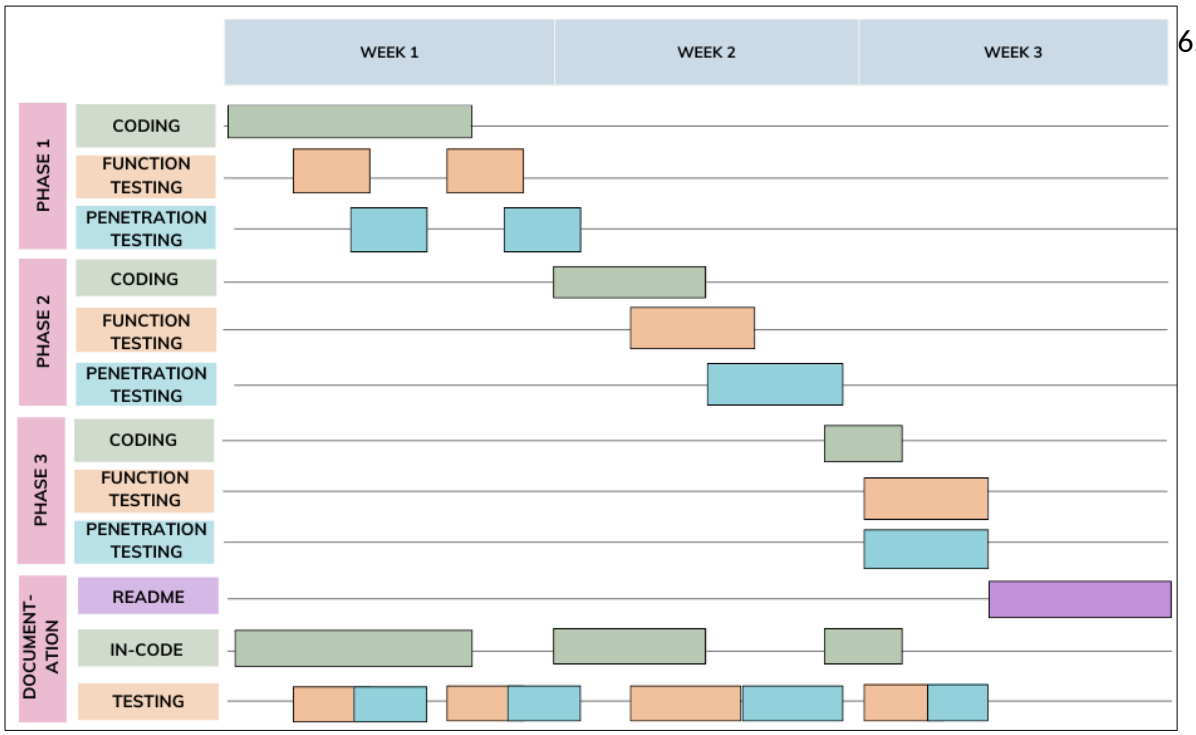

Figure 5: Design Schedule

6. Conclusion

This proposal has sought to present the design of a secure repository for report sharing between the International Space Station and NASA's Ground Control. Technical requirements, security concerns and mitigations, as well as a tentative design schedule, have been presented and discussed. The plan is meant to provide a blueprint for implementation.

7. References

Avi Networks (2019) *DataScript : HTTP Cookie Encryption Gateway*, *Avi Documentation*. Available at: https://avinetworks.com/docs/21.1/http-cookie-encryption-gateway/ (Accessed: March 27, 2023).

Brownlee, J. (2022) *Thread-safe Logging in Python*, *Super Fast Python*. Available at: https://superfastpython.com/thread-safe-logging-in-python/ (Accessed: March 27, 2023).

GDPR (2018a) *Art. 25 GDPR – Data Protection by design and by default*, *General Data Protection Regulation (GDPR)*. Available at: https://gdpr-info.eu/art-25-gdpr/ (Accessed: March 27, 2023).

GDPR (2018b) *Art. 32 GDPR – Security of processing | General Data Protection Regulation (GDPR)*. [online] Available at: https://gdpr-info.eu/art-32-gdpr/.

GDPR (2018c) *Art. 15 GDPR – Right of access by the data subject | General Data Protection Regulation (GDPR)*. [online] Available at: https://gdpr-info.eu/art-15-gdpr/.

GDPR (2018d). *General Data Protection Regulation (GDPR)*. [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/.

Gooch, R. (2005) *Overview of the Linux Virtual File System — The Linux Kernel documentation*. [online] Available at: https://www.kernel.org/doc/html/latest/filesystems/vfs.html.

Hof, H.-J. and Pohl, C. (2015) Secure Scrum: Development of Secure Software with Scrum. In: *The Ninth International Conference on Emerging Security Information, Systems and Technologies*. Venice, Italy: Securware.

IISTF (2007) *Final Report of the International Space Station Independent Safety Task Force*. rep.: 1–111.

Jain, R. (2022) *Unit Testing vs End-to-End Testing - Key Differences in 2023*, *Testsigma Blog*. Available at: https://testsigma.com/blog/unit-test-vs-e2e-test/#What_is_Unit_Testing (Accessed: March 27, 2023).

Krekel, H. (2015) *pytest: Helps You Write Better Programs*, *Pytest*. Available at: https://docs.pytest.org/en/7.2.x/ (Accessed: March 27, 2023).

Mitre (2021) Common attack pattern enumeration and classification, CAPEC. Available at: https://capec.mitre.org/data/definitions/437.html (Accessed: March 27, 2023).

NIST (2018) *Framework for Improving Critical Infrastructure Cybersecurity*. rep.: 1–48.

NumFOCUS (2023) *Pandas*. Available at: https://pandas.pydata.org/ (Accessed: March 27, 2023).

Pinto, M. & Stuttard, D. (2011) The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. 2nd Ed. Indianapolis, USA: Wiley.

PyPi (2023) *Pylint 2.17.1*, *PyPi*. Available at: https://pypi.org/project/pylint/ (Accessed: March 27, 2023).

Python Software Foundation (2023) *Logging - Logging Facility for Python*, *Python Documentation*. Available at: https://docs.python.org/3/library/logging.html (Accessed: March 27, 2023).

Red Hat (2019) *What is SELinux?* [online] Redhat.com. Available at: https://www.redhat.com/en/topics/linux/what-is-selinux.

SmartBear (2020) *SOAP vs REST. What's the Difference?*. [online] SmartBear.com. Available at: https://smartbear.com/blog/soap-vs-rest-whats-the-difference/.

SQLite (2022) *Appropriate Uses for SQLite*, *SQLite*. Available at: https://www.sqlite.org/whentouse.html (Accessed: March 27, 2023).

SQLite (n.d.) *SQLite Encryption Extension*. [online] Available at: https://sqlite.org/com/see.html (Accessed: March 27, 2023).

Stribny, P. (2020) *Scaling Relational SQL Databases*, *Software Development and Beyond*. Available at: https://stribny.name/blog/2020/07/scaling-relational-sql-databases/ (Accessed: March 27, 2023).

Stuttard, D. (2011). *Breaking Encrypted Data Using Burp*. [online] Available at: https://portswigger.net/blog/breaking-encrypted-data-using-burp (Accessed: March 27, 2023).

Synk (2023a) *SQLITE3 5.0.11 vulnerabilities: Snyk*, *Synk Vulnerability DB*. Available at: https://security.snyk.io/package/npm/sqlite3/5.0.11 (Accessed: March 27, 2023).

Synk (2023b) *Popular Python Packages Index*, *Snyk Advisor*. Available at: https://snyk.io/advisor/packages/python/popular (Accessed: March 27, 2023).

Tony (2022) *Linux - How to Simulate and Mitigate DDOS Attacks*, *Medium*. Dev Genius. Available at: https://blog.devgenius.io/linux-how-to-simulate-and-mitigate-ddos-attacks-62a3cb2f5978 (Accessed: March 27, 2023).