March 2022

Design Document

Secure Software Development

Group 2:
Asheoshla Kigbu
Austin Mundy
Gurkan Huray
Michael Geiger
Zihaad Khan

Table of Contents

Core Application and Security Requirements

The International Space Station (ISS) fosters collaboration between multiple space agencies and nations based on the existing U.N treaties and the ISS's inter-governmental agreement, which consists of the United States, Canada, the European Union, Russia, and Japan (Avveduto, 2019). McIntosh et al. (2003) mention that there are contributions from over fifteen nations in designing and implementing this sophisticated programme. The ISS produces a large amount of data, which could be categorised under groups like open source (public), confidential and commercially viable data (Witze, 2014; Warren, 2020; Farand, 2001; von der Lippe, 2000). Due to the data classification options mentioned previously, The ISS's DES (Data Exchange System) requires stricter data security and integrity capabilities, including functionalities like marking data confidential or sharing data across multiple working groups.

Solution Design

A web-based system will be designed for securely exchanging data between the ISS and the ground centre, including the following assumptions:

- Communication must always be maintained. For a backup solution, the data traffic in an emergency must also be possible via the S-band (192 kbps) (Heath, 2016).
- Continuous monitoring of workstations for command and telemetry using anti-virus and spyware protection software (NASA, 2007).
- Securing (end-to-end encryption) communication between components, including protection of privacy (Endeley, 2018).
- Well-known authentication processes and development designs (Kusnardi & Gunawan, 2019).

- Rate limiting and hardening of the underlying infrastructure to mitigate potential latency and jitter (Cheng & Wang, 2011).
- Efficient logging mechanisms to detect intrusions on the infrastructure and network (Roesch, 1999; Salama et al., 2011).
- Achieve the Create, Read, Update and Delete (CRUD) requirements via a designed authorisation model as shown in Figure 1: CRUD Operations.
- Performing moderate-level input validations to address potential validation weaknesses as shown in Figure 2: Authentication Process & Figure 3: Login Validation.
- Exchanging data between research groups (Group Based Access Control) instead of individual access.
- Access must be revoked in the following conditions:
    a) User removed from research group.
    b) User's employment terminated.
    c) Inactivity of user accounts greater than 90 days.
- A 'least privilege principle' is enforced (Ma et al., 2011).
- All critical systems have remote access.
- Direct access to the Operating System (OS) is disabled and thus minimising the risk of tampering and code injection.
- All access to critical systems and specific resources is managed by Privileged Access Security (PAS) mitigating malicious user activities (Bierens & Czaszynski, 2020).
- System will be built to be highly available (database replication), fault tolerant and resilient as shown in Figure 4: Network Diagram

- Adhering to the monolithic system requirements as shown in Figure 5: Monolithic Application Diagram.

The server-side system requirements based on Thomson (2013) and Reimers (2020) are listed in Table 1: System Requirements and can be found in the Appendix.

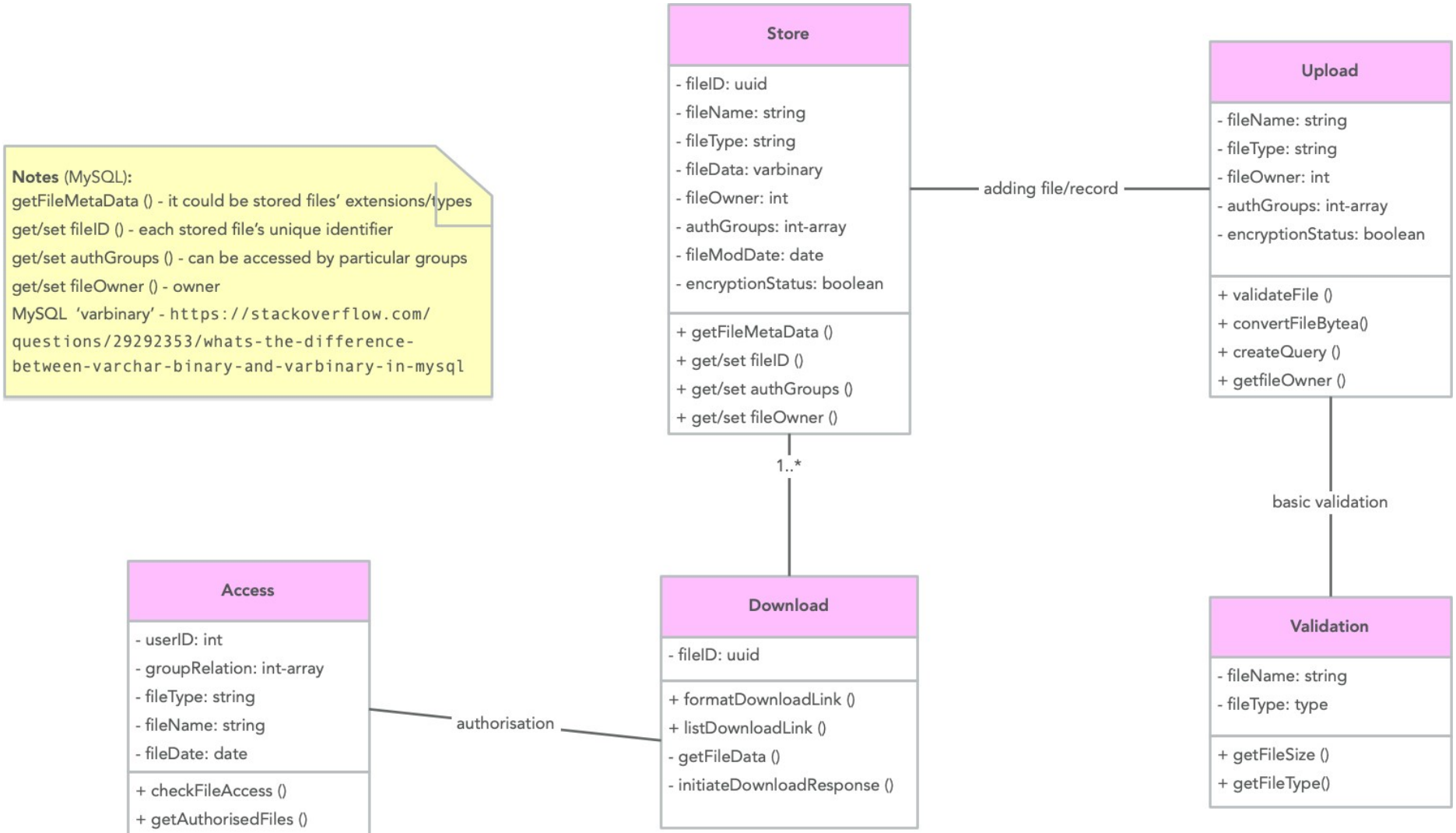

Figure 1: CRUD Operations

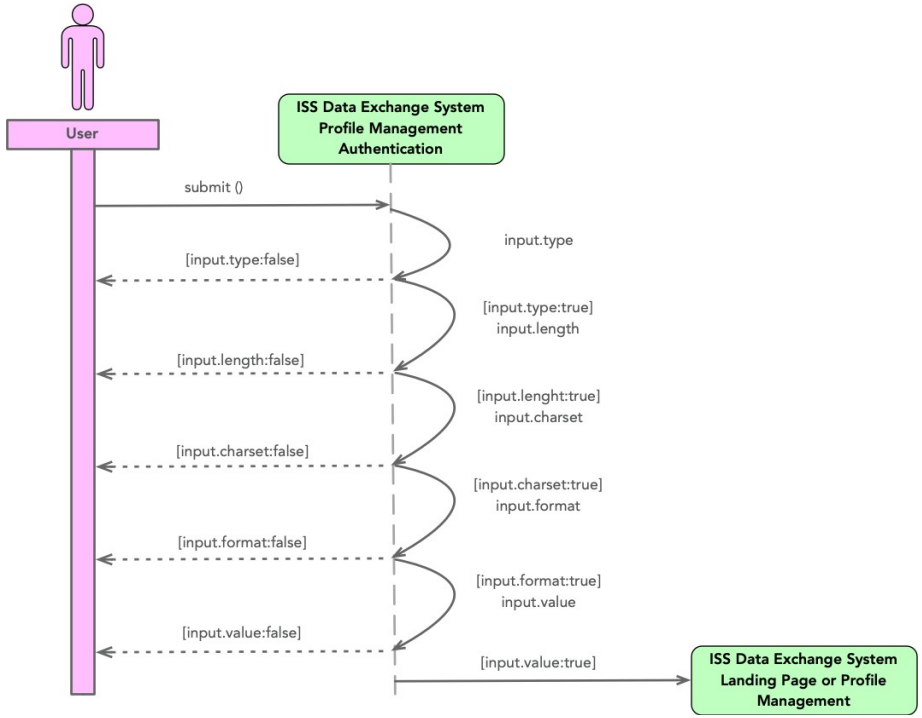

Figure 2: Authentication Process

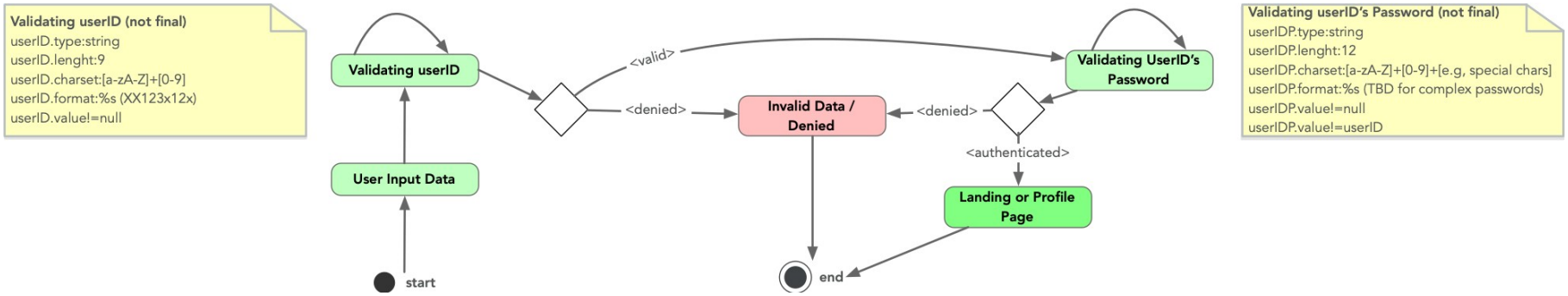

Figure 3: Login Validation

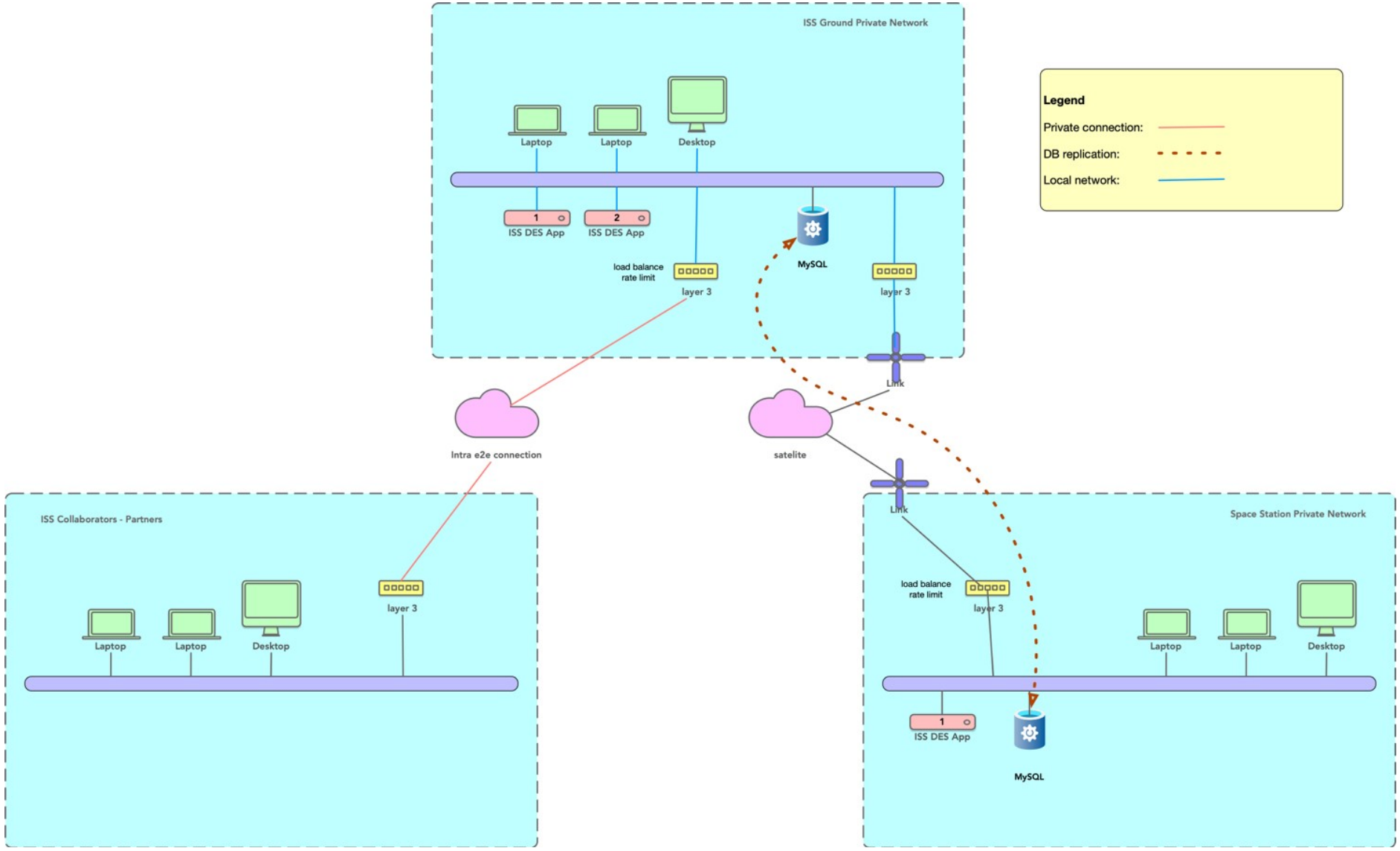

Figure 4: Network Diagram

Scaling DES App:
- The application can be scaled by putting additional copies/instances behind a load balancer

Database Replication:
- Adding resiliency and secure storage capabilities in an encrypted manner
- Storing data within a database
- No direct access to operating system file system; minimising the risk of tampering and code injection

Assumptions:
- Decent storage capacity
- Decent bandwidth (uplink/downlink)

Limitations:
- File size limitations for a single file
- Unprecedented increase in data volume
- CPU, memory and IO limitations - limited scaling capability

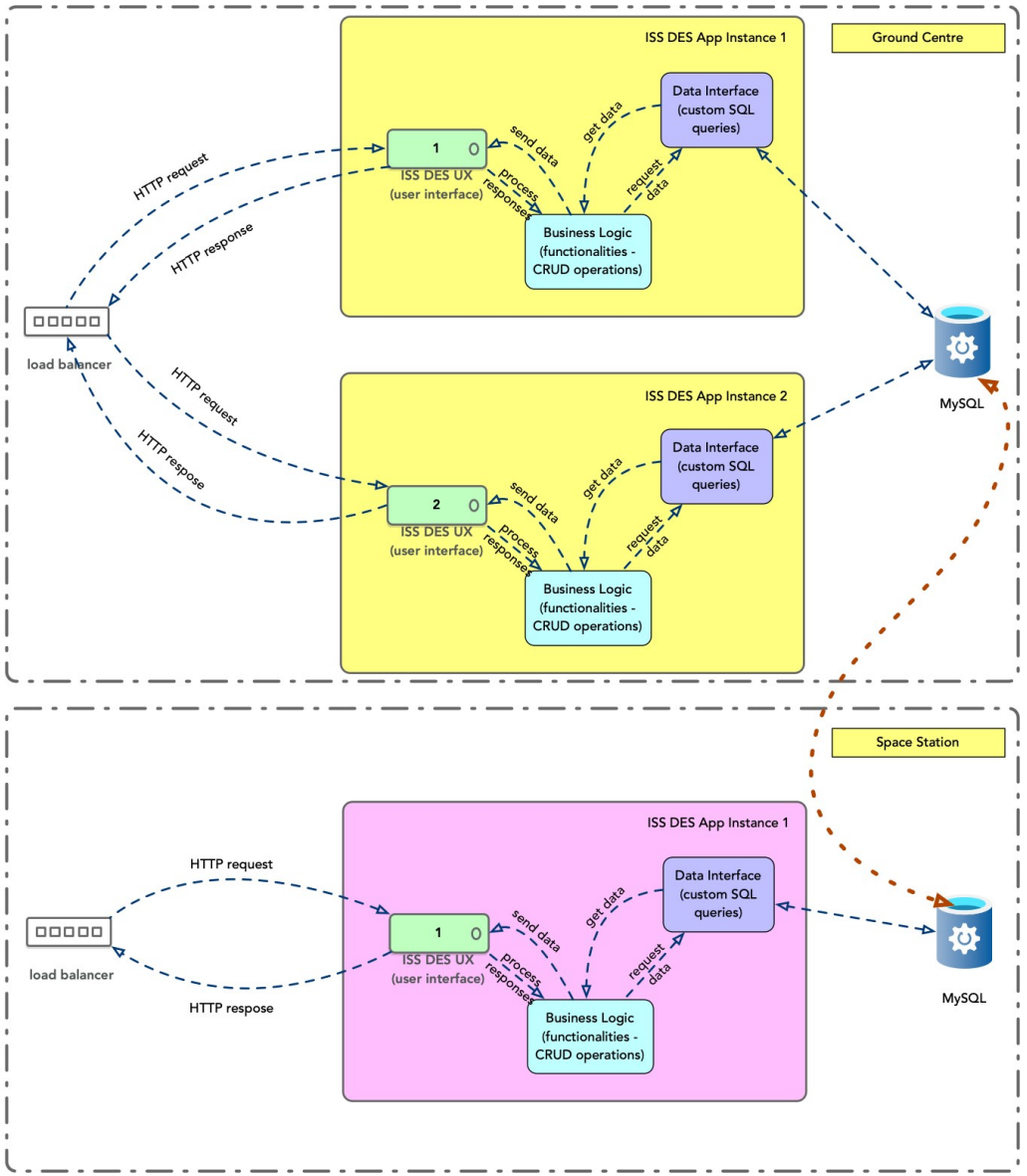

Figure 5: Monolithic Application Diagram

Security Challenges

The security challenges and mitigating factors (OWASP, 2021) are listed in Table 3: Threats and Mitigations and can be found in the Appendix.

System Limitations

Software vulnerabilities are weaknesses within a system that attackers can exploit to attack, damage, or take over a system (Jimenez et al., 2010). As shown in Table 3: Threats and Mitigations, there is a multitude of threats that can be used to compromise an application. Identifying these threats can assist with improving the software design and implementing threat mitigations. The mitigations mentioned in the table below can be implemented to better secure the application. Lastly, further testing could be done with a code-based vulnerability scanner to detect any leftover vulnerabilities.

System Tools and Libraries

Table 2: Test tools and Python libraries listed in the Appendix represents the list of test tools as well as python libraries that will be used to develop and test the ISS DES application.

References

Avveduto, R. (2019) Past, present, and future of intellectual property in space: old answers to new questions. *Wash. Int'l LJ*, *29*: 203. Available from: https://digitalcommons.law.uw.edu/wilj/vol29/iss1/7/ [Accessed 26 March 2022].

Bhat, S. (2018) Introduction to Containerization. In: Practical Docker with Python. Apress, Berkeley, CA. 1-8. Available from: https://doi.org/10.1007/978-1-4842-3784-7_1 [Accessed 22 March 2022].

Bierens, R. & Czaszynski, B. (2020) Malicious behavior detection based on CyberArk PAS logs through string matching and genetic neural networks. Available from: https://www.os3.nl/_media/2019-2020/courses/rp2/p15_report.pdf [Accessed 20 March 2022].

Canonical Ltd. (2022) UBUNTU. Available from: https://ubuntu.com/ [Accessed 22 March 2022].

Cheng, L., Wang, C.L. & Di, S. (2011) Defeating network jitter for virtual machines. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing*. 65-72. Available from: https://www.researchgate.net/publication/220895986_Defeating_Network_Jitter_for_Virtual_Machines [Accessed 26 March 2022].

CyberArk (n.d.) *What is Least Privilege Access? PoLP Explained*. CyberArk. Available from: https://www.cyberark.com/what-is/least-privilege/ [Accessed 23 March 2022].

Dierbach, C. (2014) Python as a first programming language. *Journal of Computing Sciences in Colleges*. 29(3): 73. Available from: https://dl.acm.org/doi/abs/10.5555/2544322.2544337 [Accessed 22 March 2022].

Endeley, R.E. (2018) End-to-end encryption in messaging services and national security—case of WhatsApp messenger. *Journal of Information Security*. *9*(1): 95. Available from: https://www.scirp.org/journal/paperinformation.aspx?paperid=81897 [Accessed 26 March 2022].

F5 Networks Inc. (2022) NGINX. Available from: https://www.nginx.com/products/nginx/load-balancing/ [Accessed 22 March 2022].

Farand, A. (2001) The code of conduct for international space station crews. *ESA bulletin*. *105*: 64-68. Available from: https://www.esa.int/esapub/bulletin/bullet105/bul105_6.pdf [Accessed 26 March 2022].

Grinberg, M. (2018) Flask web development: developing web applications with python. O'Reilly Media, Inc. Available from: https://books.google.de/books?hl=de&lr=&id=cVlPDwAAQBAJ&oi=fnd&pg=PT25&dq=Grinberg,+M.+(2018)+Flask+web+development:+developing+web+applications+with+python.+O%27Reilly+Media,+Inc.+&ots=xOF_dr0jfR&sig=I3KSoL6UB3ER15RdXvteoaunFl4&redir_esc=y#v=onepage&q=Grinberg%2C%20M.%20(2018)%20Flask%20web%20development%3A%20developing%20web%20applications%20with%20python.%20O'Reilly%20Media%2C%20Inc.&f=false [Accessed 26 March 2022].

Heath, N. (2016) From Windows 10, Linux, iPads, iPhones to HoloLens: The tech austronauts use on the ISS. TechRepublic. Available from: https://www.techrepublic.com/article/from-windows-10-linux-ipads-iphones-to-hololens-the-tech-space-station-astronauts-use/ [Accessed 14 March 2022].

Jimenez, W., Mammar, A. & Cavalli, A. (2010) *Software Vulnerabilities, Prevention and Detection Methods: A Review 1.*Available from: https://www.researchgate.net/publication/253704494_Software_Vulnerabilities_Prevention_and_Detection_Methods_A_Review_1 [Accessed 15 March 2022].

Kaur, N. & Kaur, P. (2014) Mitigation of SQL Injection Attacks using Threat Modeling. *ACM SIGSOFT Software Engineering Notes*. 39(6): 1–6. Available from: https://dl.acm.org/doi/abs/10.1145/2674632.2674638 [Accessed 27 March 2022].

Krekel, H. (2015) pytest: helps you write better programs. Available from: https://docs.pytest.org/en/7.1.x/ [Accessed 22 March 2022].

Krogh J.W. (2018) Connecting to MySQL. In: MySQL Connector/Python Revealed. Apress, Berkeley, CA. 47-82. Available from: https://doi.org/10.1007/978-1-4842-3694-9_2 [Accessed 22 March 2022].

Kusnardi, K. & Gunawan, D. (2019) Guillou-quisquater protocol for user authentication based on zero knowledge proof. *Telkomnika*. *17*(2): 826-834. Available from: https://www.academia.edu/43784631/Guillou_quisquater_protocol_for_user_authentication_based_on_zero_knowledge_proof [Accessed 26 March 2022].

Ma, X., Li, R., Lu, Z., Lu, J., & Dong, M. (2011) Specifying and enforcing the principle of least privilege in role-based access control. Concurrency and Computation: Practice and Experience, 23(12): 1313-1331. Available from: https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.1731?saml_referrer [Accessed 27 March 2022].

McIntosh, D.M., Elcott, S., Betts, B.J. & Mah, R.W. (2003) Paper Session II-C-Data Access and Procedure Generation for the International Space Station. Available from:

https://commons.erau.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1067&context=space-congress-proceedings [Accessed 26 March 2022].

NASA (2007) Final Report of the International Space Station Independent Safety Task Force.Available from: https://www.nasa.gov/pdf/170368main_IIST_%20Final%20Report.pdf [Accessed 15 March 2022].

Nokeri, T.C. (2022) Deploying a Web App on the Cloud. In: Web App Development and Real-Time Web Analytics with Python. Apress, Berkeley, CA. Available from: https://doi.org/10.1007/978-1-4842-7783-6_12 [Accessed 22 March 2022].

One Identity LLC. (2022) syslog-ng. Available from: https://www.syslog-ng.com/ [Accessed 22 March 2022].

OWASP (2021) OWASP Top Ten. Available from: https://owasp.org/www-project-top-ten/ [Accessed 15 March 2022].

Pallets (2022) Werkzeug. Available from: https://palletsprojects.com/p/werkzeug/ [Accessed 22 March 2022].

Pradeep, S. & Sharma, Y.K. (2019) 'A Pragmatic Evaluation of Stress and Performance Testing Technologies for Web Based Applications'. 2019 Amity International Conference on Artificial Intelligence (AICAI). Dubai, 4-6 February 2019. USA: IEEE. 399-403. Available from: https://ieeexplore.ieee.org/abstract/document/8701327 [Accessed 26 March 2022].

Python Software Foundation (2022) The Python Standard Library. Available from: https://docs.python.org/3/library/index.html [Accessed 20 March 2022].

Reitz, K. (2021) The Hitchiker's Guide to Python. Available from: https://docs.python-guide.org/scenarios/crypto/ [Accessed 22 March 2022].

Roesch, M. (1999) November. Snort: Lightweight intrusion detection for networks. In *Lisa.* 99(1): 229-238. Available from: https://www.usenix.org/legacy/publications/library/proceedings/lisa99/full_papers/roesch/roesch.pdf [Accessed 26 March 2022].

Salama, S.E., Marie, M.I., El Fangary, L.M. & Helmy, Y.K. (2011) Web Server LogsPreprocessing for Web Intrusion Detection. *Comput. Inf. Sci.*, *4*(4): 123-133. Available from: https://pdfs.semanticscholar.org/6988/e80b6bf5683e53211f60739341edfdc4bf25.pdf [Accessed 26 March 2022].

von der Lippe, J.K. (2000) Entrepreneurial Initiatives to Use the ISS for Profit. In *International Space Station: The Next Space Marketplace: Symposium Proceedings, International Symposium 26-28 May 1999, Strasbourg, France.* 4: 63. Springer Science &

Business Media. Available from: https://books.google.de/books?hl=de&lr=&id=--z46rsmv2gC&oi=fnd&pg=PR13&dq=von+der+Lippe,+J.K.+(2000)+Entrepreneurial+Initiatives+to+Use+the+ISS+for+Profit.+In+International+Space+Station:+The+Next+Space+Marketplace:+Symposium+Proceedings,+International+Symposium+26-28+May+1999,+Strasbourg,+France.+4:+63.+Springer+Science+%26+Busin&ots=1-zA28baJR&sig=hazBKzqLdOE0RhYm1lPZPEWlEFA&redir_esc=y#v=onepage&q=von%20der%20Lippe%2C%20J.K.%20(2000)%20Entrepreneurial%20Initiatives%20to%20Use%20the%20ISS%20for%20Profit.%20In%20International%20Space%20Station%3A%20The%20Next%20Space%20Marketplace%3A%20Symposium%20Proceedings%2C%20International%20Symposium%2026-28%20May%201999%2C%20Strasbourg%2C%20France.%204%3A%2063.%20Springer%20Science%20%26%20Busin&f=false [Accessed 26 March 2022].

Vyas, M. (2020) How to Protect Against Buffer Overflow Attack. Secure Coding. Available from: https://www.securecoding.com/blog/how-to-protect-against-buffer-overflow-attack/ [Accessed 23 March 2022].

Vyshnavi, V.R. & Malik, A. (2019) Efficient Way of Web Development Using Python and Flask. *International Journal of Recent Research Aspects*. 6(2): 16-19. Available from: https://www.ijrra.net/Vol6issue2/IJRRA-06-02-05.pdf [Accessed 22 March 2022].

Warren, L.E. (2020) International Space Station Open-Source Data. *Patterns*, *1*(9): 100-172. Available from: https://www.sciencedirect.com/science/article/pii/S2666389920302361 [Accessed 26 March 2022].

Witze, A. (2014) Space-station science ramps up. *Nature*. *510*(7504): 196-197. Available from: https://www.nature.com/articles/510196a [Accessed 26 March 2022].

Appendix

Table 1: System Requirements

Hardware Component	Requirement
Storage / hard disk	3 TB (annual data transfer amount plus additional storage)
RAM	64 GB
CPU	6 cores
Tracking and Data Relay Satellite System (TDRSS)	Ku-band: 600 Mbps / S-band: 192 kbps

Table 2: Test tools and Python libraries

Component/Library	Version	Description/Capabilities	Rationale
Python Specific			
Python Language	3.10.3	Programming Language	Interactive open-source programming language supporting object-oriented paradigms (Dierbach, 2014)
PyCharm Community Edition	2022.1	Integrated development environment for Python	Useful for developing, testing, and debugging Python web frameworks (Nokeri, 2022)
hashlib	-	Python hashing library	Library used for encrypting passwords (secure hashes and message digests) includes SHA256 encryption Algorithm (Python Software Foundation, 2022)
getpass	-	Portable password input	Library used to prompt the user for a password without echoing (Python Software Foundation, 2022)
mysql.connector	-	MySQL DB connector	Enables Python programs to access MySQL databases, takes 4 parameters i.e. Host, User, Password, Database Name (Krogh, 2018)
cryptography & fernet	36.0.2	Symmetric encryption	library for python that provides cryptographic recipes (Reitz, 2021)
Web Development - Flask			
Flask	2.0.3	Web application micro-framework	A lightweight micro-framework providing tools and libraries to build web-based applications (Vyshnavi & Malik, 2019)
Flask-Login	-	Flask extension	Flask user authentication and session management (Grinberg, 2018)
Flask-Sqlalchemy	-	Flask extension	Used for database management with Flask (Grinberg, 2018)
Werkzeug	-	Flask dependency	Implements WSGI, the standard Python interface between applications and servers (Pallets, 2022)

Testing Tools			
pytest	7.1.1	python testing tool	Widely used test framework, fully featured python testing tool (Krekel, 2015).
Bandit	1.7.4	python testing tool	Tool designed to find common security issues in Python code (Python Software Foundation, 2022)
pylint	2.12.2	python testing tool	Tool designed to look for programming errors and enforcing coding standards (Python Software Foundation, 2022)
Python Locust	2.8.3	Modern load testing framework	Open-source high performance load testing tool, evaluates behaviour of code (Pradeep & Sharma, 2019)
Architecture, DB & OS Distribution			
NGINX	1.20.2	HTTP proxy, TLS termination	Open-source, web server and load balancer designed for high performance & stability (F5 Networks Inc., 2022)
MySQL	8.0.28	MySQL database used for data storage, replication, and data encryption	Powerful open-source relational database commonly used with python (Krogh, 2018)
UBUNTU	21.10	Computer Operating System	Open-source Linux distribution based on Debian (Canonical Ltd., 2022)
Docker	-	Open-source containerization platform	Container platform - provides OS level virtualization and uses resource isolation of the Linux kernel (Bhat, 2018)
SYSLOG-NG	3.31.2	Security event monitoring	Open-source logging application with advance features such as content-based filtering (One Identity LLC., 2022)

Table 3: Threats and Mitigations

Threat	Cause	Mitigation
Broken Access Control	Failure to enforce access control policies.	Enforce rule of least privilege by ensuring RBAC are implemented (Cheng et al., 2011).
Injection Attack	Lack of input validation, filtering, or sanitization.	Threat modelling to identify and classify threats and clearly define entry and exit points within the system architecture (Jimenez et al., 2010).
Buffer Overflow	Insufficient bounds checking.	Configure checks to validate that any user input is legitimate and compatible with configured thresholds (Vyas, 2020).
Insecure Design	Architectural flaws relating to the lack of proper threat modeling and design architectures.	Create use and misuse cases for each whilst implementing adequate testing with an agile approach.
Identification and Authentication Failures	Insufficient implementation of methods relating to securing users' identity, authentication methods, and session management.	Configure authentication and session management controls in alignment with current best practice to ensure all use activity is authorized.