2022

Design Document

SECURE SOFTWARE DEVELOPMENT

TEAM BUILDER: DOUG LEECE, HAROUN FUJAH, ADRIAN BOSCU, SAMIYA NOVA.

Table of Contents

Core Application and Security Requirements:

The International Space Station (ISS) governs cooperation between the multiple space agencies through memoranda of understanding, bilateral agreements, and U.N. treaties (Avveduto, 2019). While the orbiting laboratory facilitates scientific data interchange between nations, often made publicly available (Warren, 2020), space agencies may also mark data proprietary (Farand, 2001). Private enterprises utilizing ISS research capabilities for profit (Johnson, 2021) require commercial interest protection data privacy assurance to protect commercial interests. Consequently, the secure repository application must support the ability to retain specific data as confidential and the option to share data with members of one or more groups.

ISS research data also requires integrity and availability protection as experiments very costly to replicate. Despite 600 megabit-per-second data rates (Peters, 2019), station-to-ground communications challenges are unavoidable as satellite-based transmissions (Schlesinger et al., 2017) have latency and jitter. The secure repository must function over satellite links, ensuring priority mission safety and operational data is unaffected by file transfer activity. (Gray, 2017).

ISS networking technology, initially highly controlled and proprietary (Mallary & WhiteLaw, 1987), evolved to internetworked station modules and readily available 802 wireless (Saenz-Otero, 2005). Despite using similar protocols and commercial off-the-shelf equipment (COTS), ISS networks are not hostile like a university campus or the internet; crew members are well-vetted and recognized professionals (Working in Space, 2015; Greenstone 2018). Therefore, within a threat modeling framework like STRIDE (Shostack, 2014; Conklin, 2022), denial of service and elevation of privilege attack classes can be disregarded. Conversely, ISS ground teams and affiliated agency members increase the probability of information disclosure or elevated privilege abuse due to increased threat community size (Open Group, 2021), warranting additional control rigor.

STRIDE CLASS	ISS THREAT	GROUND THREAT	KEY IDENTIFIED CONTROLS
SPOOFING IDENTITY	Very unlikely	Unlikely	Digital Identity enrollment process implemented external to HTTP application (Grassi et al., 2017) Near-real time intrusion monitoring,
TAMPERING WITH DATA	Very unlikely	Unlikely	Flask MVC framework predefined SQL sanitization (Llantos, 2017; OWASP, 2018) MVC application does not require direct O.S. file system access Microservice containers run as non-privileged user (Yasrab, 2021) HTTPS encrypted client to server network transport
REPUDIATION THREATS	Very unlikely	Possible	NIST AAL1 digital identity requirements (Grassi et al., 2017) User & administrative activity logging (Kent & Souppaya, 2006)

			Application threat specific logging, E.G., automatic owner assignment on new file creation event (OWASP, 2018) Database transaction logging (Fowler,2016)
INFORMATION DISCLOSURE	Unlikely	Possible	Database injection attack controls using FLASK framework and secure coding practices (OWASP, 2018) Secure credential storage Data backup encryption Transparent Data Encryption for data at rest (Liu, 2015)
DENIAL OF SERVICE	Very unlikely	Possible	Satellite network transport, private ground networks, limited inter-agency connection, fault-tolerant systems, rate-limiting enabled for HTTP service (Dalili et al., 2022)
ELEVATION OF PRIVILEGES	Very unlikely	Possible	Design decision (process) segregation of duties and interfaces. I.E., data user: HTTP application, administrators: SSH key-based CLI Privileged account activity monitoring (Kent & Souppaya,2006; Fowler, 2016)

Scale	**Indication**
Very Unlikely	This type of activity has never been observed in this environment.
Unlikely	This type of activity has seldom been observed in this environment.
Possible Occurrence	This type of may be observed within the environment at least once every 10 years.
Likely	This type of may be observed within the environment at least once every 3 years.
Extremely Likely	This type of activity is repeatedly observed in this environment.

STRIDE Class	Ground Threat Risk Indication	Threat Control Rationale
Repudiation Threats	Possible	Adversaries employ threat shifting (Blank & Gallagher, 2012) through compromising ground station logging system inserting perplexing data into unprotected logs (Shostack, 2014) relative to bypassing arduous user authentication. Ground station log files are documented locally then forwarded to central logging location enabling comparison of both sets, restricting malicious activity (Cobb, 2011).
Information Disclosure	Possible	Encryption of data backups & system credentials circumvents possible risk of unauthorized alteration of confidential scientific research data (Manulis et al, 2020).
Denial of Service	Possible	Attackers exploit unsecure ground station networks (Manulis et al., 2020) overpowering secure repository application server, denying service to authorized ground station users (Tarandach &Coles, 2020).
Elevation of Privileges	Possible	RegEx input validation on all user inputs safeguards against SQL injection attacks where adversaries manipulate data in repository application backend database (Kranthikumar & Leela Velusamy, 2020).

Solution Design Summary

Selected an HTTP application for data users based on the following assumptions:

- HTTP browser ubiquity, multi-lingual capability, transport encryption
- Familiar authentication process, well-tested development patterns (OWASP, 2018)
- User error feedback patterns, linkable support pages (Figure 1)
- HTTP 1.1 maturity ensures multiple client, server offerings

- Rate limiting and server timeout configuration options addressing latency and jitter
- Viable logging formats for intrusion monitoring (Marty, 2005)

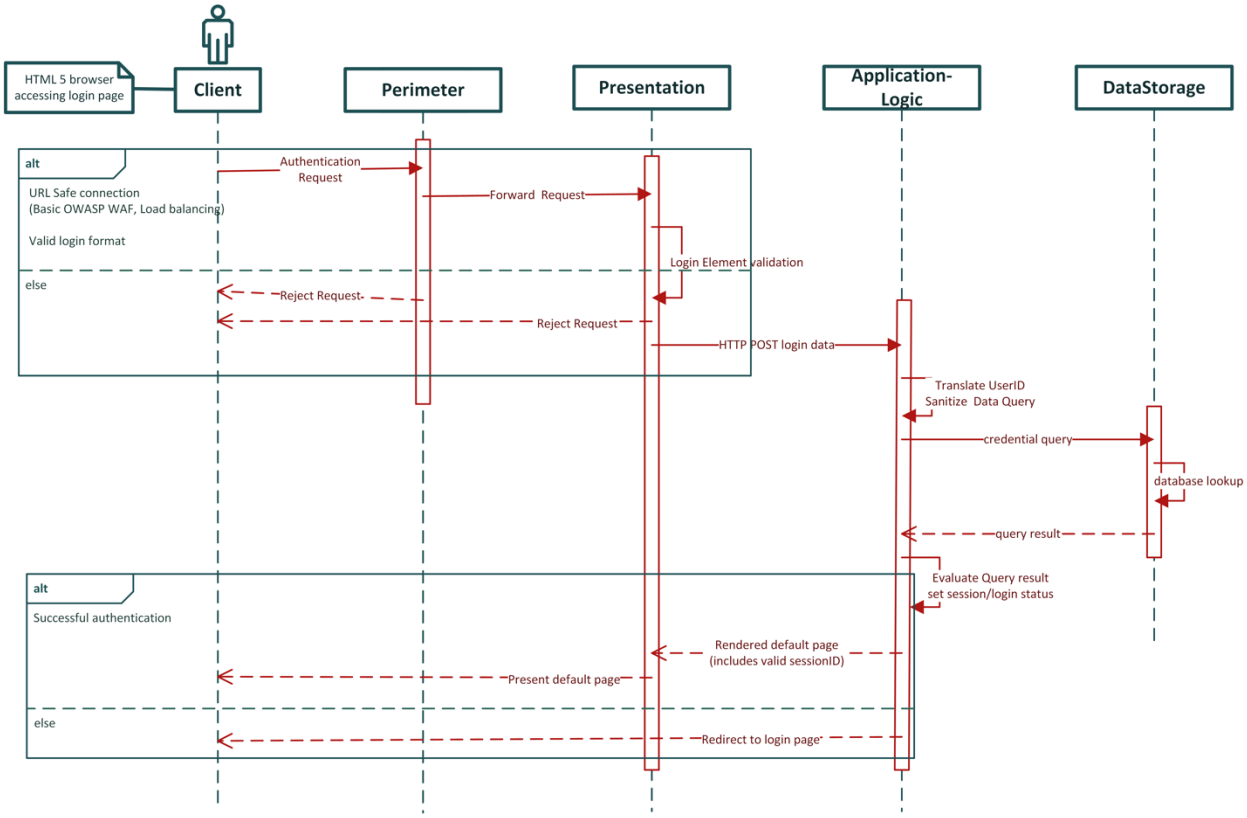

Fig. 1

CRUD requirements implemented through authorization model modifying stored file metadata (Figure 3). Group membership access control rather than individual user due to ISS operating model assumptions:

- Data sharing within research team membership rather than space agency (Figure 2)
- On mission changes new group members access all previously shared data
- Removes individual access to application data upon removal from group or account termination
- Minimal identity change burden on administrators periodically modifying access via SQL commands, eliminates authorization-based privilege elevation threats from role-based pattern implementation errors (OWASP.org 2017, Llantos, 2017) (Figure 5).

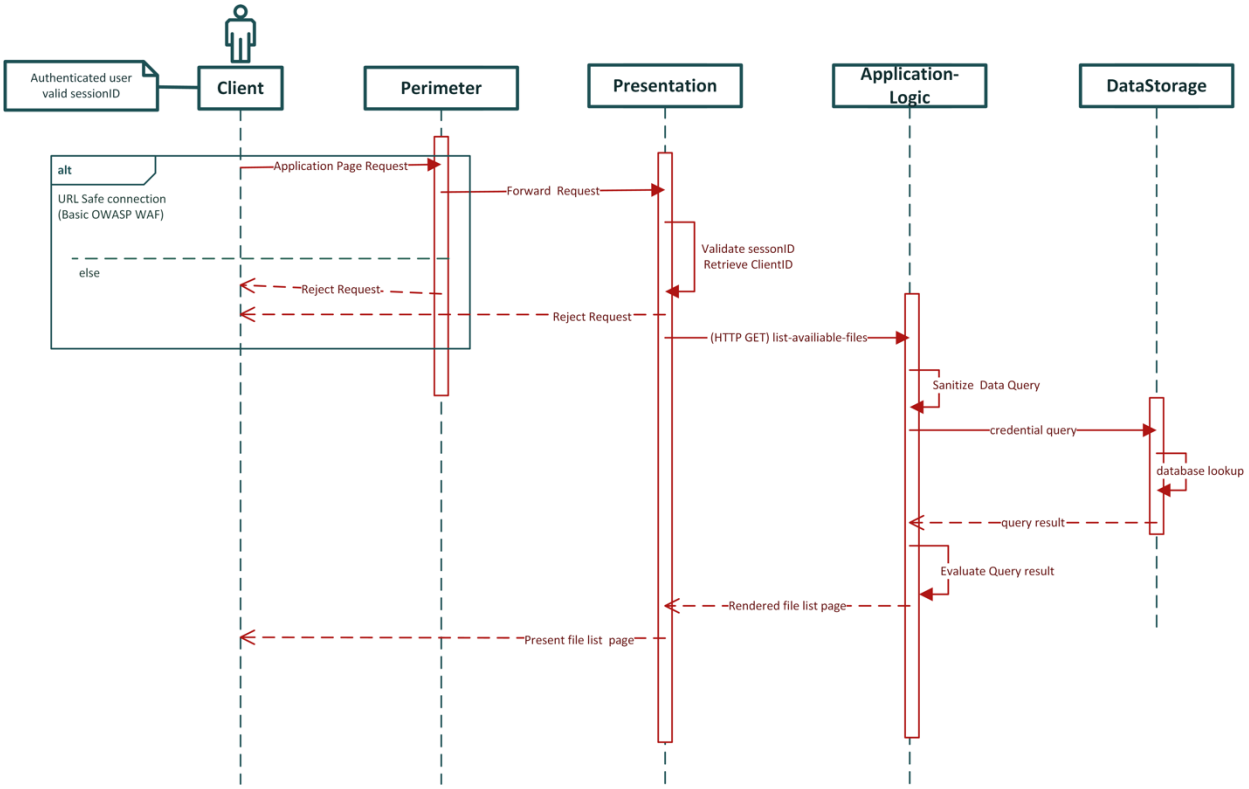
Authenticated user
valid sessionID
Client
Perimeter
Presentation
Application-
Logic
DataStorage
alt
URL Safe connection
(Basic OWASP WAF)
else
Application Page Request
Forward Request
Validate sessonID
Retrieve ClientID
Reject Request
Reject Request
(HTTP GET) list-availiable-files
Sanitize Data Query
credential query
database lookup
query result
Evaluate Query result
Rendered file list page
Present file list page

Fig. 2

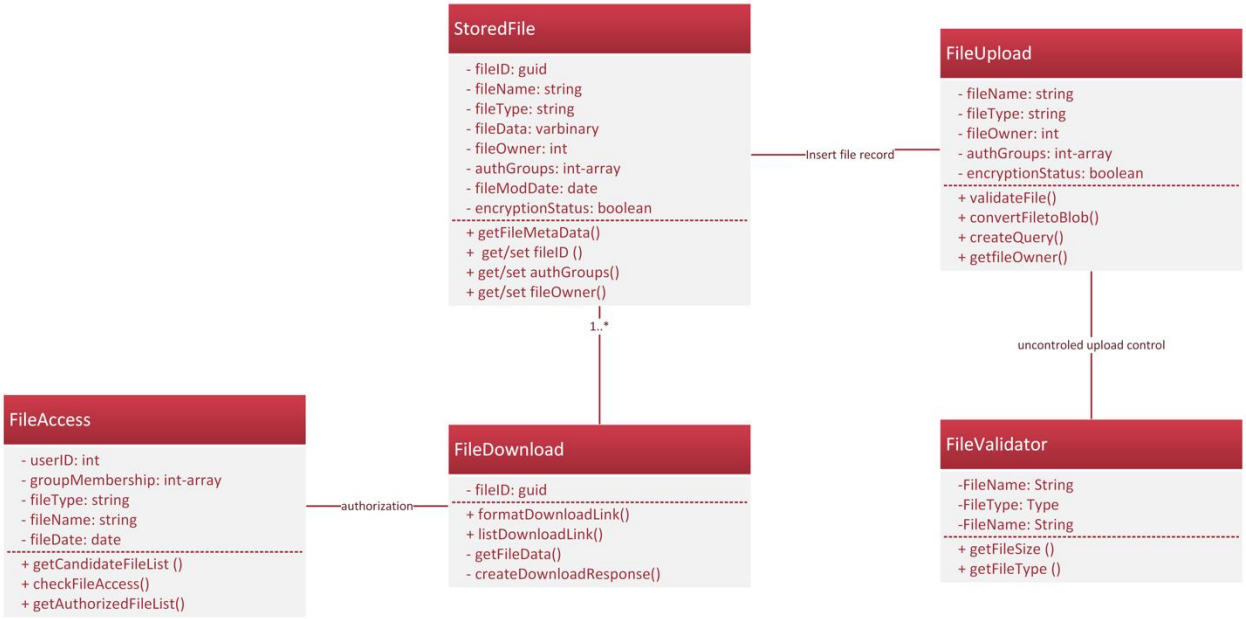
StoredFile
- fileID: guid
- fileName: string
- fileType: string
- fileData: varbinary
- fileOwner: int
- authGroups: int-array
- fileModDate: date
- encryptionStatus: boolean
+ getFileMetaData()
+ get/set fileID ()
+ get/set authGroups()
+ get/set fileOwner()
Insert file record
FileUpload
- fileName: string
- fileType: string
- fileOwner: int
- authGroups: int-array
- encryptionStatus: boolean
+ validateFile()
+ convertFiletoBlob()
+ createQuery()
+ getfileOwner()
1..*
uncontroled upload control
FileAccess
- userID: int
- groupMembership: int-array
- fileType: string
- fileName: string
- fileDate: date
+ getCandidateFileList ()
+ checkFileAccess()
+ getAuthorizedFileList()
authorization
FileDownload
- fileID: guid
+ formatDownloadLink()
+ listDownloadLink()
- getFileData()
- createDownloadResponse()
FileValidator
-FileName: String
-FileType: Type
-FileName: String
+ getFileSize ()
+ getFileType ()

Fig. 3

Application will use database binary object storage versus file system storage access for the following reasons:

- Assuming 1 TB storage footprint acceptable onboard ISS
- No direct application interaction with O.S. filesystem, closing data tampering and command injection attack vectors resulting in O.S. access (Zhong, 2022)
- Resilience, performance, network optimization (Figure 4) and secure storage options via database replication technology and transparent data encryption
- Simplifies network-based data tier required for MVC design pattern based distributed system design utilizing microservices (Figure 5)

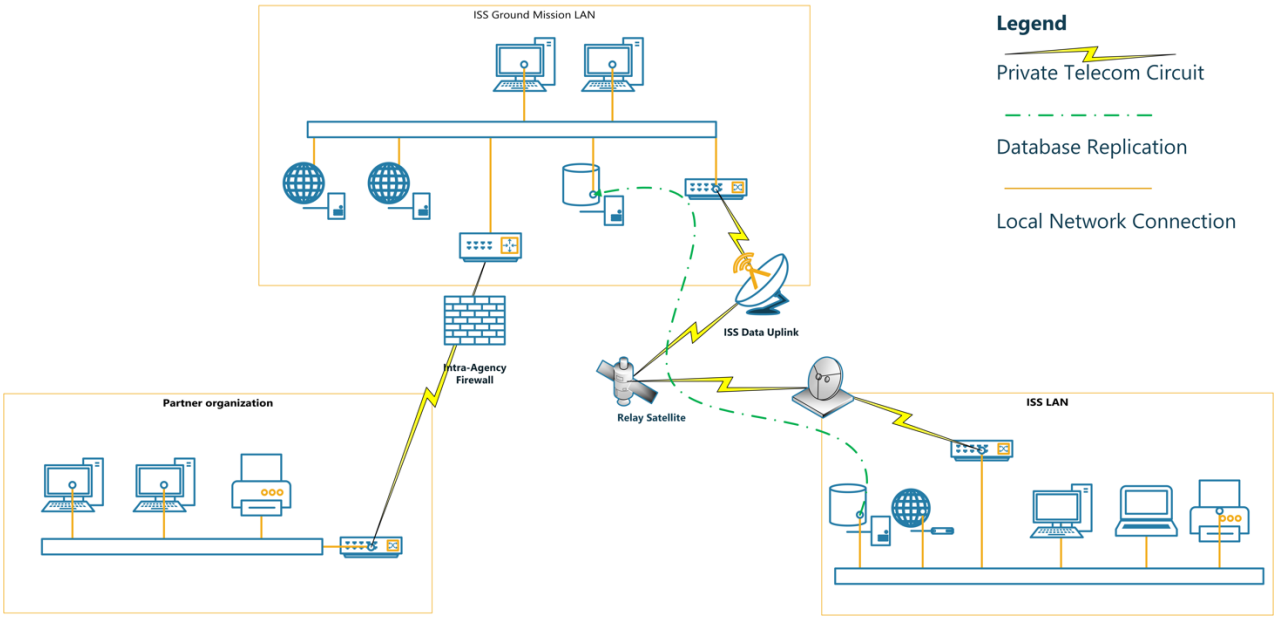

Fig. 4

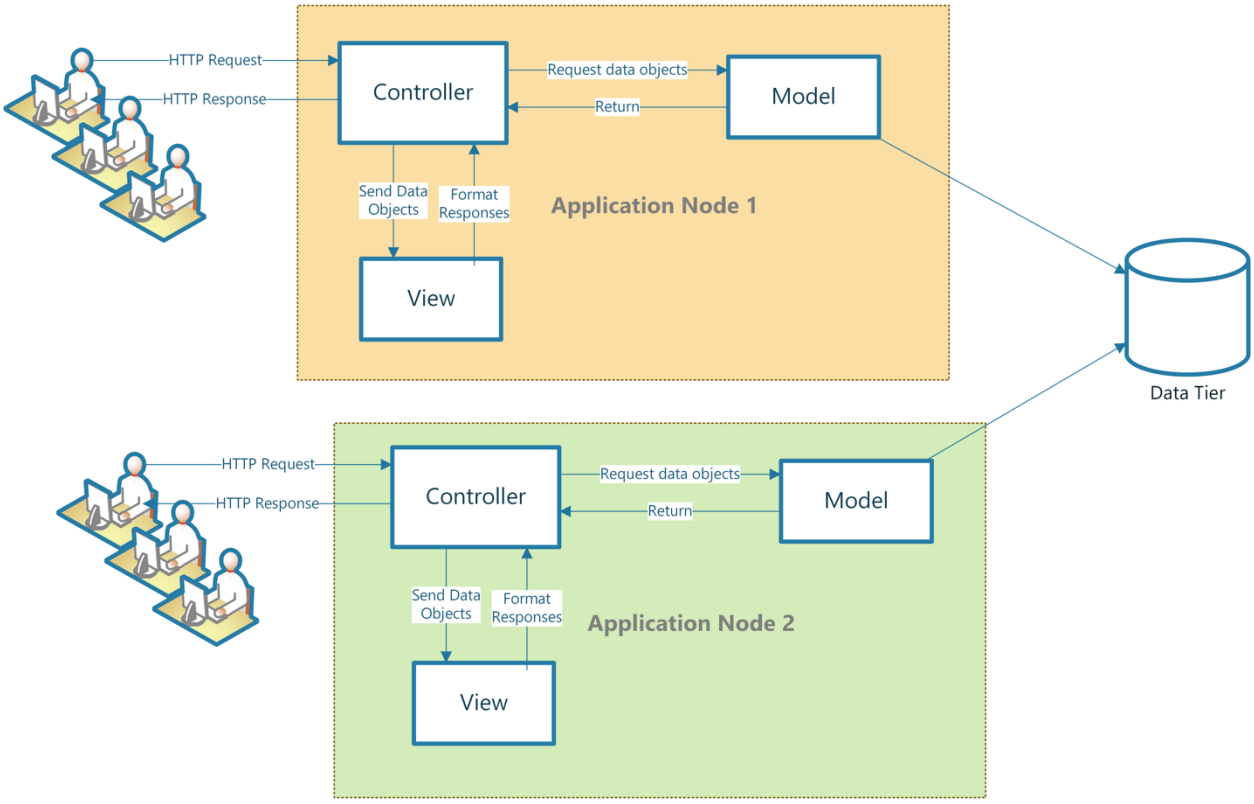

Fig. 5

Solution Component Summary

Software vulnerability management is essential to maintaining application resilience. A comprehensive list of system components allows stakeholders to use the disclosure to exploit dwell time (Figuser 6) to advantage (Carmody et al., 2021).

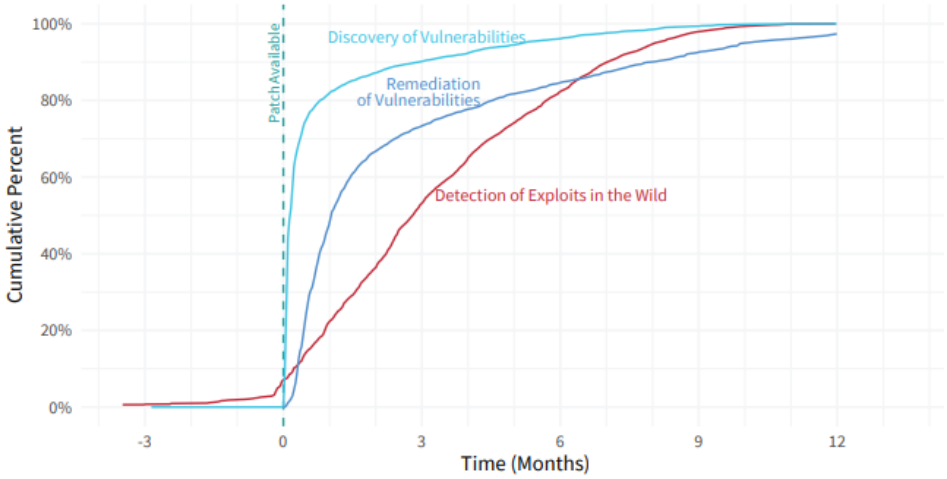

Fig. 6

COMPONENT	VERSION	FUNCTIONALITY	Selection Justification
PYTHON PROGRAMING LANAGUAGE	3.8.10	General-purpose programing language	modern object-oriented open-source programming language (Lin, J.W.B., 2012)
FLASK	2.0.2	Web application micro-framework	A collection of functions and libraries to simplify implementation of secure design patterns (Pallets, 2010)
WERKZUEG	2.0.2	Flask supporting library	Close integration with framework
FLASK-LOGIN	0.5.0	Flask supporting library	Well vetted functions designed for web application login patterns
FLASK-SQLALCHEMY	2.5.1		Abstraction layer between framework and database management system
CRYPTOGRAPHY	36.0.1	Symmetric encryption	Actively supported Python library supporting modern encryption algorithms
NGINX	1.18.0	HTTP proxy, TLS termination	Actively supported HTTP server with multiple network management and security enhancement options
MYSQL	8.0.27	Data storage, replication, and transparent data encryption	Actively supported database management system with encryption, replication, and transaction logging capabilities
UBUNTU	20.04.3 LTS	Computer Operating System	Ubuntu long-term support (LTS) ensures the operating system will have feature improvements for 5 years and security patches for 10 years.
DOCKER	20.10.12	Microservice Container management	Docker facilitates replication of a customized application on multiple systems, Minimizes attack surface when properly implemented (Combe et al. l, 2016)
SYSLOG-NG	3.35.1	Security event monitoring	Central security monitor

Identified System Constraints

Multi-user applications reliant on shared database access raise deadlock possibilities therefore database transaction management (MySQL, N.D.), increases programming effort. Storage constraints not observed with file system storage include:

- 4-gigabyte file size limitation
- Decreased performance processing large files (Spahiu et al, 2009; BinAlshikh, 2018)
- Potentially large database size relative to record count
- Additional application and software configuration required for large files

MVC framework and Docker container Implementation errors can lead to unauthorized information disclosure despite data tier hardening (Llantos, 2017; Combe et all, 2016).

References:

Avveduto, R. (2019) Past, present, and future of intellectual property in space: old answers to new questions. *Washington International Law Journal* 29(1): 203-246. Available from: https://digitalcommons.law.uw.edu/wilj/vol29/iss1/7/ [Accessed 7 February 2022]

BinAlshikh, I. (2018) Storing Files in Server. Filer System or Database?. Available from: https://ibrahim-2017.blogspot.com/2018/07/storing-files-in-server-file-system-or.html [Accessed 11 February 2022]

Blank, R. & Gallagher, P. (2012) *Guide for Conducting Risk Assessments*. Gaithersburg, Maryland: National Institute of Standards and Technology. Available from: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf [Accessed 12 February 2022].

Carmody, S., Coravos, A., Fahs, G., Hatch, A., Medina, J., Woods, B. & Corman, J. (2021) Building resilient medical technology supply chains with a software bill of materials. *npj Digital Medicine* 4(34) DOI: https://doi.org/10.1038/s41746-021-00403-w

Cobb, M. (2011) Best practices for audit, log review for IT security investigations. Available from: https://www.computerweekly.com/tip/Best-practices-for-audit-log-review-for-IT-security-investigations [Accessed 12 February 2022].

Combe, T., Martin, A. & Di Pietro, R. (2016) To docker or not to docker: A security perspective. *IEEE Cloud Computing*, *3*(5): 54-62 Available from: https://doi.org/10.1109/MCC.2016.100

Conklin, L. (2022) Threat Modeling Process. Available from: https://owasp.org/www-community/Threat_Modeling_Process [Accessed 11 February 2022].

Cyentia Institue. (2020) Prioritization to Prediction: Volume 6- The Attacker-Defender Divide. Available from:

https://library.cyentia.com/report/report_006413.html [Accessed 12 February 2022]

Dalili, S., Wetter, D. & Mayo, L. (2022) Unrestricted File Upload. Available from: https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload [Accessed 12 February 2022]

Farand, A. (2001) The Code of conduct for International Space Station crews. Available from: https://www.esa.int/esapub/bulletin/bullet105/bul105_6.pdf [Accessed 5 February 2022]

Fowler, K. (2016) *Data Breach Preparation and Response.* 1st ed. Cambridge: Syngress

Johnson, M. (2021) NASA. Commercial and Marketing Pricing Policy. Available from: https://www.nasa.gov/leo-economy/commercial-use/pricing-policy [Accessed 8 February 2022].

Grassi, P., Fenton, J., Netwon, E., Perlner, R., Regenscheid, A., Burr, W. & Richer, J. (2017) *Digital Identity Guidelines: Authentication and Lifecycle Management*. Gaithersburg, Maryland: National Institute of Standards and Technology. Available from: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-63b.pdf [Accessed 12 February 2022].

Gray, E. (2017) Data Relay Network. Available from: https://svs.gsfc.nasa.gov/12720 [Accessed 11 February 2022].

Greenstone, A. (2018) Ethics and public integrity in space exploration. *Acta Astronautica*, 143: 322-326. DOI: Available from: http://dx.doi.org/10.1016/j.actaastro.2017.10.031

Kent, K. & Souppaya, M. (2006) *Guide to Computer Security Log Management*. Gaithersburg, Maryland: National Institute of Standards and Technology. Available from: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-92.pdf [Accessed 12 February 2022].

Kranthikumar, B. & Leela Velusamy, R. (2020) SQL injection detection using REGEX classifier. *Journal of Xi'an University of Architecture & Technology* 12(6): 800-809. Available from: http://xajzkjdx.cn/gallery/78-june2020.pdf [Accessed 12 February 2022].

Lin, J.W.B. (2012) Why Python is the next wave in earth sciences computing. *Bulletin of the American Meteorological Society*, *93*(12): 1823-1824. DOI: https://doi.org/10.1175/BAMS-D-12-00148.1 [Accessed 12 February 2022]

Llantos, O. (2017) Function: The Foundation of Secure Web Development. *Book of abstracts of the CENTERIS 2017* 2017: 234-237 Available from: https://www.researchgate.net/publication/320933756_Function_The_Foundation_of_Secure_Web_Development

Lui, J. (2015) 'Preventing Leakages of Business Secrets from Encrypt Data Stored in the Cloud', *IEEE 17th International Conference on High Performance Computing and Communications (HPCC). New Y*ork, 24-26 August. Los Alamitos: IEEE. 1488-1494.

Mallary, W. & Whitelaw, V. (1987) Space Station Data Management System Architecture. *Proceedings of the IEEE* 75(3): 320-328. DOI: 10.1109/PROC.1987.13739 [Accessed 9 February 2020]

Manulis, M., Bridges, C., Harrison, R., Sekar, V. & Davis, A. (2020) Cyber security in New Space. *International Journal of Information Security* 20(3): 287-311. Available from: https://link.springer.com/article/10.1007/s10207-020-00503-w [Accessed 12 February 2022].

Marty, R. (2011) 'Cloud application logging for forensics' *ACM Symposium on Applied Computing*. TaiChung Taiwan, 21-25 March. New York: Association for Computing Machinery. 178-184. Available from: https://dl.acm.org/doi/abs/10.1145/1982185.1982226 [Accessed 12 February 2022].

Mysql. (N.D.) 13.3.1 START TRANSACTION, COMMIT, and ROLLBACK Statements. Available from: https://dev.mysql.com/doc/refman/8.0/en/commit.html [Accessed 12 February 2022].

NASA. (2015) Working in Space. Available from: https://www.nasa.gov/audience/foreducators/stem-on-station/ditl_working [Accessed 11 February 2022].

Owasp.org. (2017) OWASP Top Ten 2017 | A2:2017-Broken Authentication. Available from: https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication [Accessed 11 February 2022].

OWASP. (2018) OWASP Proactive Controls. Available from: OWASP Proactive Controls | OWASP Foundation [Accessed 10 February 2022]

Pallets. (2010) Flask web development one drop at time. Available from: https://flask.palletsprojects.com/en/2.0.x/ [Accessed 8 February 2022]

Peters, M. (2019) Data Rate Increase on the International Space Station Supports Future Exploration, NASA. Available from: https://www.nasa.gov/feature/goddard/2019/data-rate-increase-on-the-international-space-station-supports-future-exploration [Accessed 11 February 2022].

Saenz-Otero, A. (2005) Design principles for the development of space technology maturation laboratories aboard the international space station. Ph. D. Thesis, Massachusetts Institute of Technology. Available from: https://www.mit.edu/~alvarso/thesis-phd/ThesisBook.pdf

Schlesinger, A., Davidson, S., Willman, B., Pitts, L. & Pohlchuck, W. (2017) 'Delay/Disruption Tolerant Networking for the International Space Station (ISS)', *2017 IEEE Aerospace Conference.* Big Sky Montana, 4-11 March. Los Alamitos: IEEE. 1-14 DOI: 10.1109/AERO.2017.7943857

Shostack, A. (2014) *Threat Modeling: Designing for Security*. 1st ed. Indianapolis: John Wiley & Sons.

Spahiu, C.S., Stanescu, L., Burdescu, D.D. & Brezovan, M. (2009) 'File Storage for a Multimedia Database Server for Image Retrieval', *Fourth International Multi-Conference on Computing in the Global Information Technology.* Cannes/La Bocca France, 23-29 August. Los Alamitos: IEEE. 35-40

Tarandach, I. and Coles, M., (2020). *Threat Modeling: A Practical Guide for Development Teams*. 1st ed. Sebastopol: O'Reilly.

The Open Group. (2021) *Risk Analysis (O-RA), Version 2.0.1.* Berkshire, United Kingdom: The Open Group Available from: O-RA 2.0.1 (opengroup.org) [Accessed 9 February 2022]

Warren, L. (2020) International Space Station Open-Source Data. *Patterns 1(9)* DOI: https://doi.org/10.1016/j.patter.2020.100172

Yasrab R. (2021), *Mitigating Docker Security Issues.* Oxford: University of Oxford. Available from: https://arxiv.org/pdf/1804.05039.pdf [Accessed 10 February 2022]

Zhong, W. (2022) Command Injection. Available from: https://owasp.org/www-community/attacks/Command_Injection [Accessed 11 February 2022].