CRITICAL SOFTWARE

TN01: Literature Review and State-of-the-art Study on Cybersecurity for Space or Similar Domains

CYBERSECURITY FOR SPACE

CONTRACT REFERENCE: NOT APPLICABLE.

DATE: 2024-09-06
PROJECT CODE: CSEC4SPACE
DOC. REF.: CSW-2024-TNR-01442
STATUS: APPROVED
PAGES: 53
INFORMATION CLASSIFICATION: PUBLIC
VERSION: 1.1

DISCLAIMER -

The work described in this report was performed under the Master's degree research titled "Cybersecurity for space domain". Responsibility for the contents resides in the author or organization that prepared it.

PARTNERS:

Polytechnic University of Coimbra

APPROVAL

VERSION	NAME	FUNCTION	SIGNATURE	DATE
1.1	Nuno Silva	Industry Supervisor		2024-09-06
1.1	João Carlos Cunha	Academic Supervisor		2024-09-06

AUTHORS AND CONTRIBUTORS

NAME	DESCRIPTION	DATE
Pedro Miguel Sousa	Author	2024-03-19
Nuno Silva	Reviewer	2024-05-30

COPYRIGHT

REVISION HISTORY

VERSION	DATE	DESCRIPTION	AUTHOR
0.1	2024-03-19	First revision of the technical note.	Pedro Sousa
0.2	2024-04-17	Updated Technical Note according to internal review.	Pedro Sousa
1.0	2024-07-31	Updated document according to internal review comments. Document Approved for delivery.	Pedro Sousa
1.1	2024-09-06	Updated according to internal review. Approved for delivery.	Pedro Sousa

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

1. INTRODUCTION

As humanity pushes further into space exploration, ensuring the security of space systems becomes a paramount concern. Emerging since the "Cold War" era with the launch of the first satellite, Sputnik, in 1957, space exploration was dominated by a select group of states developing satellites with long lifecycles. At that time, "security through obscurity" was a common security practice, tightly restricting information to hinder the rival nation´s space program.

Advancements in technology - particularly in computing with smaller sizes, reduced energy demands, and more processing power - led to a significant increase in the number of satellites orbiting the Earth, and therefore elevating the likelihood of a cyberattack. While critical infrastructure on Earth receives increasing cybersecurity attention, space systems have been neglected. This neglect stems from a confluence of factors. Firstly, the perception that space systems were too complex or remote for attackers to target. Secondly, the rapid expansion of private companies in the space industry has prioritized rapid development and affordability. This focus on speed and cost may have unintentionally led to less emphasis on robust cybersecurity measures. Finally, the challenge of legacy systems adds another layer of complexity. Upgrading these systems, built and used for a long time, can be expensive and technically challenging.

This technical note presents a review of the state-of-the-art in cybersecurity in general, including standards, frameworks and good practices and the current use of secure development lifecycle. It then analyses the current state-of-the-art in cybersecurity applications for space systems and similar domains, by examining existing research papers, news articles, and industry blogs.

1.1. OBJECTIVE

This technical note aims to review the current state of cybersecurity for space systems and similar domains (railway, aviation and automotive) by examining existing literature.

1.2. SCOPE

This technical note examines the current state of cybersecurity for space systems, focusing on software used in ground, space, and link segments. The analysis draws upon information from academic papers, news articles, and blogs.

1.3. AUDIENCE

The audience of this technical note includes: Critical Software S.A., Coimbra Institute of Engineering, and Engineers/Researchers interested in the field of cybersecurity.

1.4. DEFINITIONS AND ACRONYMS

Table 1 presents the list of definitions used throughout this document.

NAME	DESCRIPTION
Availability	Ensuring timely and reliable access to and use of information.
Authentication	Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.

NAME	DESCRIPTION
Attack Vector	Attack vector is a path or means by which an attacker or hacker can gain access to a computer or network server to deliver a payload or malicious outcome.
Cybersecurity	Prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, and confidentiality.
Confidentiality	Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
Critical Infrastructure	Critical infrastructure refers to the systems, facilities and assets that are vital for the functioning of society and the economy.
Integrity	Guarding against improper information modification or destruction and includes ensuring information non-repudiation and authenticity.
Threat	Any circumstance or event with the potential to adversely impact organizational operations, organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability.
Threat Actor	Threat actors, also known as cyberthreat actors or malicious actors, are individuals or groups that intentionally cause harm to digital devices or systems by exploiting vulnerabilities in computer systems, networks and software.
Vulnerability	Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.
Reference Document	A document is considered a reference if it is referred but not applicable to this document. Reference documents are mainly used to provide further reading.

Table 1: Definitions

Table 2 presents the list of acronyms used throughout this document.

ACRONYM	DESCRIPTION
ABAP	Advanced Business Application Programming
AJAX	Asynchronous JavaScript and XML
AIP	Application Intelligence Platform
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network

ACRONYM	DESCRIPTION
ASH	Automated Security Helper
ASP	Active Server Pages
ASPM	Application Security Posture Management
ASST	Automated Software Security Toolkit
ATV	Automated Transfer Vehicle
AUTOSAR	Automotive Open System Architecture
AVM	Application Vulnerability Mitigation
AWS	Amazon Web Services
B2B	Business to Business
BBC	British Broadcasting Corporation
BSIMM	Building Security Maturity Model
BSP	Board Support Package
CCPA	California Consumer Privacy Act
CCRA	Common Criteria Recognition Arrangement
CCSDS	Consultative Committee for Space Data Systems
CFML	ColdFusion Markup Language
CI/CD	Continuous Integration and Continuous Delivery/Deployment
CIA	Confidentiality, Integrity, Availability
CISO	Chief Information Security Officer
CLI	Command Line Interface
CMITSE	Common Methodology for Information Technology Security Evaluation
CMS	Content Management System
COBOL	Common Business Oriented Language
COTS	Commercial-of-the-shelf
CSW	Critical Software, S.A.
CUDA	Compute Unified Device Architecture
DAST	Dynamic Application Security Testing

ACRONYM	DESCRIPTION
DES	Data Encryption Data
DOC	Document
DSL	Domain-specific Language
EASA	European Union Aviation Safety Agency
ECSS	European Cooperation for Space Standardization
EDR	Endpoint detection and response
ENVISAT	Environmental Satellite
EOS	Earth Observing System
ESA	European Space Agency
FAA	Federal Aviation Administration
FCA	Financial Conduct Authority
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GCA	Global Cybersecurity Agenda
GDPR	General Data Protection Regulation
GEO	Geosynchronous Equatorial Orbit
GPL	General Public License
GPS	Global Positioning System
HKTM	Housekeeping Telemetry
HTML	Hyper Text Markup Language
IACS	Industrial Automation and Control Systems
IAST	Interactive Application Security Testing
IBM	International Business Machine Corporation
IDE	Integrated Development Environment
IDP	Individual Development Plan
IEC	International Electronic Commission
IEEE	The Institute of Electrical and Electronics Engineers

ACRONYM	DESCRIPTION
IMT	International Mobile Telecommunications
ISA	International Standards on Auditing
ISBN	International Standard Book Number
ISE	Identity Services Engine
ISMS	Information Security Management System
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JSON	JavaScript Object Notation
JSP	Java Servlet Pages
LAPSE	Lightweight Analysis for Program Security in Eclipse
LDAP	Lightweight Directory Access Protocol
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
MISRA	Motor Industry Software Reliability Association
MITRE	Massachusetts Institute of Technology Research Establishment
MVC	Model, View, Controller
MXML	Magic eXtensible Markup Language
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
NMAP	Network Map
NTSP	NASA Technical Standards Program
OSS	Open-Source Vulnerability Scanning
OWASP	Open Worldwide Application Security Project
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
REST	Representational State Transfer
SAE	Society of Automotive Engineers

ACRONYM	DESCRIPTION
SAMM	Software Assurance Maturity Model
SARIF	Static Analysis Results Interchange Format
SASDS	Security Architecture for Space Data Systems
SAST	Static Application Security Testing
SAT	Satellite
SBOM	Software Bill of Materials
SCA	Static Code Analysis
SDL	Security Development Lifecycle
SEC	Securities and Exchange Commission
SIEM	Security information and event management
SOAR	Security orchestration, automation, and response
SPICE	Software Process Improvement and Capability Determination
SQL	Structured Query Language
SSDL	Secure Software Development Lifecycle
STD	Standard
SwCA	Software Composition Analysis
TARA	Threat Analysis and Risk Analysis
VMDR	Vulnerability Management, Detection and Response
WAP	Web Application Protection
XDR	Extend detection and response
XML	Extensible Markup Language
XSS	Cross-site Scripting
XXE	XML external entity injection
YAML	Yet Another Markup Language
ZAP	Zed Attack Proxy

Table 2: Acronyms

1.5. DOCUMENT STRUCTURE

Section 1 (Introduction) presents this document.

Section 2 presents the state of the art of cybersecurity standards, frameworks and regulations used by space industry and similar domains.

Section 3 provides the state of the art of cybersecurity for the space domain.

Annex A provides two tables of vulnerability scanning and source code analysis tools.

1.6. REFERENCE DOCUMENTS

Table 3 presents the list of reference documents.

REFERENCE DOCUMENT	DOCUMENT NUMBER
[RD-1] International Telecommunication Union	https://www.itu.int/en/action/cybersecurity/Pages/default.aspx, visited on 2023-12-07.
[RD-2] The current structure of the European space manufacturing sector	https://eurospace.org/wp-content/uploads/2021/01/structure-of-the-space-manufacturing-sector-v2021.pdf, visited on 2023-12-14.
[RD-3] Falco, Gregory (2018). Cybersecurity Principles for Space Systems. Journal of Aerospace Information Systems	(PDF) Cybersecurity Principles for Space Systems (researchgate.net), visited on 2023-12-14.
[RD-4] The CubeSat revolution changing the way we see the world	https://www.bbc.com/news/business-48533945, visited on 2023-12-20.
[RD-5] Nanosats Database	https://www.nanosats.eu/,visited on 2023-12-20.
[RD-6] Protecting Space Systems from Cyber Attack	https://medium.com/the-aerospace-corporation/protecting-space-systems-from-cyber-attack-3db773aff368, visited on 2023-12-20.
[RD-7] HACK-A-SAT 4	https://hackasat.com/,visited on 2024-03-01.
[RD-8] 10 Defining Moments in Cybersecurity and Satellite in 2013	https://interactive.satellitetoday.com/via/january-february-2024/10-defining-moments-in-cybersecurity-and-satellite-in-2023/, visited on 2024-03-01.
[RD-9] Spacenews – Space Critical Infrastructure Breaking the Binary Debate and Call for Space Council Action	https://spacenews.com/space-critical-infrastructure-breaking-the-binary-debate-and-a-call-for-space-council-action/, visited on 2024-05-23.
[RD-10] SatelliteToday – Debate Over Space as a Critical Infrastructure in Europe Takes Center Stage at Cysat	https://www.satellitetoday.com/cybersecurity/2024/04/24/debate-over-space-as-a-critical-infrastructure-in-europe-takes-center-stage-at-cysat/, visited on 2024-05-23.
[RD-11] Falco, Gregory & Boschetti, Nicolò. (2021). A Security Risk Taxonomy	(PDF) A Security Risk Taxonomy for Commercial Space Missions (researchgate.net), visited on 2024-03-01.

REFERENCE DOCUMENT	DOCUMENT NUMBER
for Commercial Space Missions	
[RD-12] ECSS-E-ST-40C - Software	https://ecss.nl/standard/ecss-e-st-40c-software-general-requirements/, visited on 2024-04-09.
[RD-13] ECSS-Q-ST-80C – Software Product Assurance	https://ecss.nl/standard/ecss-q-st-80c-rev-1-software-product-assurance-15-february-2017/, visited on 2024-04-09.
[RD-14] The Consultative Committee for Space Data Systems	https://public.ccsds.org/default.aspx, visited on 2024-03-20.
[RD-15] IEEE SA - Standard for Space System Cybersecurity	https://standards.ieee.org/ieee/3349/11182/, visited on 2024-03-21.
[RD-16] ISA – ISA/IEC 62443	https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards, visited on 2024-03-21.
[RD-17] Aerospace – The Private Sector´s Assessment of U.S. Space Policy and Law	https://aerospace.csis.org/the-private-sectors-assessment-of-u-s-space-policy-and-law/, visited on 2024-03-21.
[RD-18] NIST - Lightweight Crypto, Heavyweight Protection	https://www.nist.gov/blogs/taking-measure/lightweight-crypto-heavyweight-protection, visited on 2024-03-22.
[RD-19] ESA – The Protection of Space Missions: Threats and Cyber Threats	https://cybersecurity.uniroma1.it/sites/default/files/ZATTI%20-18%20March%202019.pdf, visited on 2024-03-22.
[RD-20] ESA – Frequently asked questions on Galileo	https://www.esa.int/Applications/Navigation/Frequently_asked_questions_on_Galileo, visited on 2024-03-22.
[RD-21] Microsoft SDL practices	https://www.microsoft.com/en-us/securityengineering/sdl/practices, visited on 2024-03-23.
[RD-22] Book review: Software Security: Building Security in – Part II: Seven Touchpoints for Software Security	https://adriancitu.com/tag/security-touchpoints/, visited on 2024-03-23.
[RD-23] SAFECode – Our History	https://safecode.org/our-history/, visited on 2024-03-23.
[RD-24] NIST SP 800-218 – Secure Software Development Framework	https://csrc.nist.gov/pubs/sp/800/218/final, visited on 2024-03-23.
[RD-25] ISO – ISO/SAE 21434	https://www.iso.org/standard/70918.html, visited on 2024-03-23.
[RD-26] Cyber Security Process and Methods	Cyber Security Processes and Methods: A Guide (careerkarma.com), visited on 2024-03-23.
[RD-27] Understanding and Investigating Adversary Threats and Countermeasures in the	(PDF) Understanding and Investigating Adversary Threats and Countermeasures in the Context of Space Cybersecurity (researchgate.net), visited on 2024-03-23.

REFERENCE DOCUMENT	DOCUMENT NUMBER
Context of Space Cybersecurity	
[RD-28] Frontegg – What is access management? Risk, Technology and Best Practices	https://frontegg.com/guides/access-management, visited on 2024-03-24.
[RD-29] IBM – What is incident Response	https://www.ibm.com/topics/incident-response, visited on 2024-03-24.
[RD-30] Antunes, M. Rodrigues, B. Introdução à Cibersegurança [Introduction to Cybersecurity]. (2022)	ISBN 9789727229246, visited on 2024-03-24.
[RD-31] Sprinto – Top 16 cybersecurity tools you must know in 2024	Top 16 Cyber Security tools You Must Know in 2024 - Sprinto, visited on 2024-03-24.
[RD-32] Jürgen, Dobaj & Salamun Alen (2021) Cybersecurity Verification and Validation Testing in Automotive	(PDF) Cybersecurity Verification and Validation Testing in Automotive (researchgate.net), visited on 2024-03-24.
[RD-33] OWASP - SAMM	https://owasp.org/www-project-samm/, visited on 2024-03-24.
[RD-34] Synopsys - Building Security Maturity Model	https://www.synopsys.com/software-integrity/software-security-services/bsimm-maturity-model.html, visited on 2024-03-24.
[RD-35] Common Criteria	https://www.commoncriteriaportal.org/index.cfm, visited on 2024-03-24.
[RD-36] Common Criteria – Common Methodology for Information Technology Security Evaluation	https://www.commoncriteriaportal.org/cc/index.cfm, visited on 2024-03-24.
[RD-37] Common Criteria – Common Criteria Recognition Arrangement	https://www.commoncriteriaportal.org/ccra/index.cfm, visited on 2024-03-24.
[RD-38] OWASP – Vulnerability Scanning Tools	https://owasp.org/www-community/Vulnerability_Scanning_Tools, visited on 2024-03-24.
[RD-39] OWASP – Source Code Analysis Tools	https://owasp.org/www-community/Source_Code_Analysis_Tools, visited on 2024-03-24.
[RD-40] Via Satellite – Satellite Operators Respond to Cyber Threats in a Rapidly Changing Environment	https://interactive.satellitetoday.com/via/october-2022/satellite-operators-respond-to-cyber-threats-in-a-rapidly-changing-environment/, visited on 2024-03-24.
[RD-41] The Maritime Executive – Mass GPS Spoofing attack in Black Sea?	https://maritime-executive.com/editorials/mass-gps-spoofing-attack-in-black-sea, visited on 2024-03-24.
[RD-42] TN02: Analysis of Practices and International	CSW-2024-TNR-01442, visited on 2024-03-24.

REFERENCE DOCUMENT	DOCUMENT NUMBER
Standards Related to Cybersecurity	

Table 3: Reference documents

2. STATE OF THE ART

This section introduces the standards, frameworks and regulations used in various domains and in the space industry, as well as three examples of software secure lifecycles.

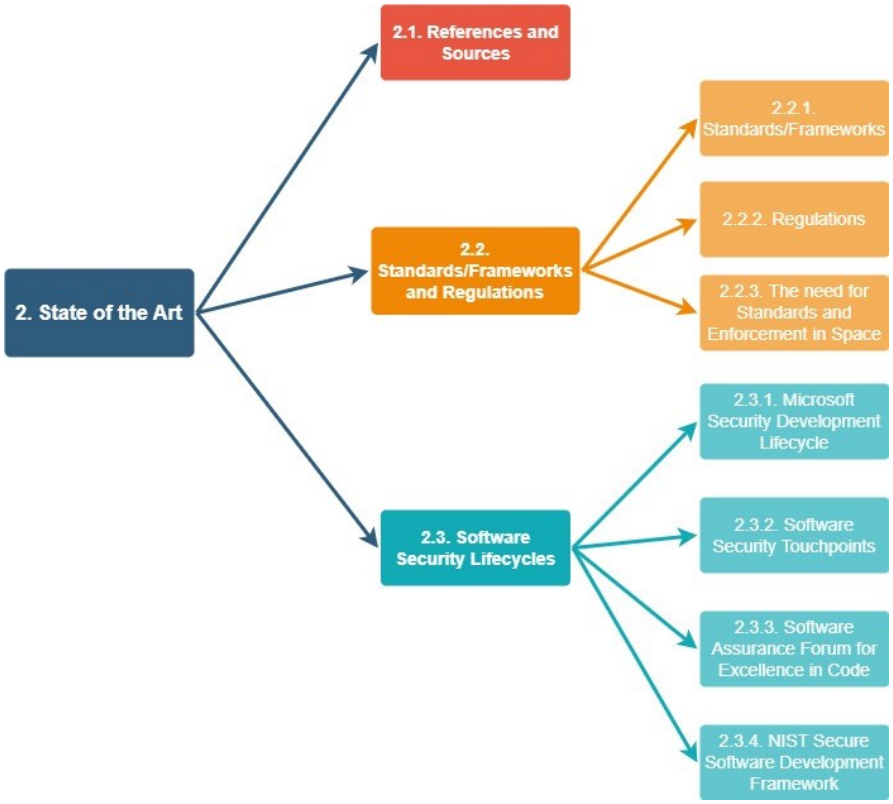

Figure 1: Overview of Section 2

2.1. REFERENCES AND SOURCES

The references presented in Table 3 primarily consist of industrial and academic papers, blogs, and online news articles. Papers authored or co-authored by Space cybersecurity expert Gregory Falco were particularly influential and closely followed for inspiration.

2.2. STANDARDS, FRAMEWORKS AND REGULATIONS

The lack of industry-specific cybersecurity standards and regulations is a major challenge to securing space systems. However, positive developments are underway.

2.2.1. STANDARDS/FRAMEWORKS

The European Space Agency (ESA) is revising its ECSS standards, particularly E40 [RD-12] and Q80 [RD-13], to incorporate a stronger cybersecurity focus. Similarly, the United States has a long-standing focus on cybersecurity, particularly through NASA Technical Standards Program (NTSP).

On the world-wide front, the Consultative Committee for Space Data Systems (CCSDS) plays a critical role. Comprised of leading space communication experts from 28 nations, CCSDS serves as a flagship for developing standards in space communication and data handling [RD-14]. Recognizing the growing need for robust cybersecurity, the Institute of Electrical and Electronics Engineers (IEEE) established a dedicated working group in 2023. This group is tasked with developing the first-ever IEEE International Standard for Space System

Cybersecurity. This standard aims to provide secure-by-design technical specifications, addressing the shortcomings of outdated policies, regulations, and doctrines related to space security [RD-15].

While dedicated space cybersecurity standards are lacking, some generic standards can be leveraged for the sector. These include the ISO 27000 family from the International Organization for Standardization (ISO) and standards from the National Institute of Standards and Technology (NIST) in the United States. For example, NIST's FIPS 140 standard focuses on cryptographic module security requirements. Table 4 presents a summarized version of generic cybersecurity standards that can be applied to the Space sector, and will be explored in more detail in the technical note [RD-42] about international standards related to cybersecurity.

Domain	Standard	Description
Cryptography	FIPS 140	The standard "Security Requirements for Cryptographic Modules" purpose is to outline the security criteria that must be met by a cryptographic module employed in a security system.
Security categorization	FIPS 200	FIPS 200 outlines the minimum-security requirements for information and information systems.
Information security	NIST SP 800-53	The purpose of the SP 800-53 is to enhance cybersecurity practices and protect sensitive information by providing a security and privacy controls for information systems and organizations.
Information security	NIST SP 800-171	This standard concentrates on safeguarding information confidentiality shared between the federal government and its external service providers (e.g., financial, web, electronic mail, healthcare) as well as educational institutions such as colleges and universities, by the recommendation of security requirements.
Information security / Evaluation of IT security	ISO/IEC 15408	ISO/IEC 15408, commonly known as the Common Criteria (CC), is an international standard consisting of five parts, developed to provide a framework for evaluating and certifying the security features of information technology products.
Information security / Evaluation of IT security	ISO/IEC 18045	This standard is responsible to support the evaluation of the ISO/IEC 15408 series, by providing the minimum actions to be performed by an evaluator.
Automotive	ISO/IEC 21434	ISO/IEC 21434 provides a comprehensive set of requirements for cybersecurity risk management throughout the lifecycle of electrical and electronic (E/E) systems in road vehicles, including their components and interfaces, from the initial concept stage to decommissioning.
Information security	ISO/SAE 27000 family	The ISO/IEC 27000 family of standards is a set of international standards for information security management systems (ISMS).
Industrial automation and control systems	ISA/IEC 62443	Cybersecurity standard for industrial automation and control systems (IACS), provides a comprehensive set of requirements and guidance that can help

Domain	Standard	Description
		organizations to protect their systems from cyberattacks.
Power plant I&C programmable digital systems	IEC 62645	This standard is used by the nuclear sector with the premise of establishing cybersecurity requirements needed to prevent and/or mitigate the impact of a cyberattack against the digital Instrumentation and Control systems.

Table 4: Applicable Non-Space Cybersecurity Standards/Framework

Table 5 presents Space industry-specific cybersecurity standards developed by NASA, ECSS and ISO, that will be explored in more detail in the technical note about international standards related to cybersecurity [RD-42]. These standards include both currently implemented standards and those undergoing revision with a focus on cybersecurity.

Domain	Standard	Description
Space system protection	NASA-STD-1006A	Establishes protection requirements across the agency, ensuring that all missions, programs and projects are resilient to cyber threats.
Space software assurance and safety	NASA-STD-8739.8B	Defines requirements for implementing a systematic approach to independent assurance, safety, verification, and validation for software created, acquired, provided, used, or maintained by or for NASA.
Space software engineering	ECSS-E-ST-40C	Assists stakeholders in formulating their software requirements by leveraging the ECSS-E-ST-40C framework, which combines methods and requirements from other branches of ECSS.
Space system software product assurance	ECSS-Q-ST-80C	Through the Q-80 standard, users leverage a set of software product assurance requirements for developing and maintaining the software component of firmware in space systems, including space, launch, and ground segments.
Security architecture for space data systems	ISO 20214	Security Architecture for Space Data Systems (SASDS), provides a high-level system engineering reference specifically tailored to securing space systems.

Table 5: Space Industry Specific Standards

2.2.2. REGULATIONS

Regulating the space sector, particularly for multinational agencies like the ESA, presents a significant challenge. The fragmented regulatory landscape can be difficult to navigate, potentially causing delays in regulating companies. This complexity stands in stark contrast to the rapid pace of technological innovation in the space industry, creating difficulties and needs such as [RD-17]:

- **Outdated Regulatory Frameworks** – existing policies stopped in time and aren't adapting to the changing commercial space landscape.
- **Need for Agile Licensing Process** – currently, the licensing processes are too complex and need to be more agile.

Overall, the space industry needs regulations that are adaptable, efficient, and encourage innovation while ensuring security. There's a need to balance fostering a thriving commercial space economy with robust regulatory frameworks.

Table 6 lists some regulatory entities that regulate specific domains.

Domain	Name	Description
Telecommunications	Federal Communications Commission (FCC)	Regulates interstate and international telecommunications by providing licenses, establishing rules for broadcasters, and fostering competition.
	International Telecommunication Union (ITU)	A specialized agency of the United Nations that coordinates global telecommunication regulations and policies.
Aviation	Federal Aviation Administration (FAA)	Oversees all aspects of civil aviation in the United States, including airworthiness certification, air traffic control, and safety regulations.
	European Union Aviation Safety Agency (EASA)	Ensures a high level of uniform safety within civil aviation across the European Union.
Finance	Securities and Exchange Commission (SEC)	Protects investors, maintains fair markets, and facilitates capital formation in the United States.
	Financial Conduct Authority (FCA)	Regulates the financial services industry in the United Kingdom to protect consumers and promote financial stability.
Data Privacy	General Data Protection Regulation (GDPR)	Regulates how personal data is collected, used, and stored within the European Union.
	California Consumer Privacy Act (CCPA)	Grants California residents certain rights regarding the access, deletion, and sale of their personal information.

Table 6: Regulatory Entities from Other Domains

2.2.3. THE NEED FOR STANDARDS AND THEIR ENFORCEMENT IN SPACE

Two key gaps hinder effective cybersecurity in space systems: first, there's no way to enforce the use of existing standards, and second, there's a lack of dedicated standards tailored to the unique risks faced by space systems. This reality can have impacting consequences, namely large financial costs (economic assets) and loss of lives.

For the aviation industry, for example, as listed above, the FAA in the United States and EASA in Europe regulate all aspects of civil aviation. This includes certifying repair shops, aircrews, and mechanics. Importantly, they also

ensure that organizations comply with various standards, particularly those related to cybersecurity and best practices. This comprehensive approach aims to protect aircraft, traffic control systems, and other critical infrastructure.

Without some entity enforcing the compliance of standards, organizations might adopt varying levels of cybersecurity depending on budget, expertise, and risk perception. This creates vulnerabilities across the entire ecosystem. Additionally, the lack of a common baseline for secure design and operation makes identifying and addressing vulnerabilities a much more complex process. Finally, the absence of dedicated space security standards can lead to continued use of outdated security protocols and technologies, leaving them more susceptible to known attacks, potentially disrupting critical services and causing widespread economic damage and societal chaos.

In response to this growing concern, initiatives like the Global Cybersecurity Agenda (GCA) are emerging. Launched in 2007 by the ITU, the GCA aims to establish enforcement mechanisms for existing standards. It serves as an international cooperation framework that bolsters confidence and security in the information society by emphasizing cooperation, efficiency, and collaboration among relevant partners. This framework leverages existing initiatives to avoid redundancy.

The impact of the GCA, highlighted through initiatives like the Global Cybersecurity Index measuring nation-state cybersecurity capabilities, Standardizing Security practices to instill confidence in the use of Information and Communication Technologies (ICTs), and efforts securing radio communications for International Mobile Telecommunications (IMT) networks (3G, 4G, 5G and soon 6G), underscores its pivotal role in global cybersecurity. This framework also issues recommendations on security aspects in network management architecture for digital satellite systems and enhances transmission control protocol performance [RD-1].

In conclusion, the lack of dedicated cybersecurity standards for the space sector hinders the development of robust and effective defences for space systems. Unlike the aviation industry with its established cybersecurity standards (DO-326/ED-202, DO-356A/ED-203A), the space sector remains largely unguarded. However, positive changes are emerging. The European Space Agency are revising existing standards like ECSS-E-40 and ECSS-Q-80, with a specific focus on incorporating cybersecurity requirements.

2.3. SOFTWARE SECURITY LIFECYCLES

The space sector, like many other sectors, relies heavily on software to operate. In fact, software permeates not only industry but also our daily lives. From powering spacecraft to writing technical notes, secure software is essential for ensuring systems' availability for authorized action only, integrity (free from unauthorized system alterations) and confidentiality (free from unauthorized disclosure of information). The Software Security Development Lifecycle (SSDL) is a framework that integrates security considerations throughout the entire software development process. This approach, often referred to as "security by design," ensures that security is a core concern for developers from the very beginning of a project.

The following subsections provide a short description of three of the most common SSDL.

2.3.1. MICROSOFT SECURITY DEVELOPMENT LIFECYCLES

The Microsoft SSDL (also known as Microsoft SDL) began to be developed in 2002 as a result of the Trustworthy Computing initiative and was full integrated into their (Microsoft) standard software development practices in 2004. This initiative aimed to develop a security model that empowers developers to implement secure coding practices [RD-21].

Recent updates to the Microsoft SDL prioritize simplicity, automation, and enhanced developer guidance. Additionally, the SDL has been adapted to address the evolving threat landscape, advancements in cloud computing and AI/ML, and ever-changing regulatory demands.

The Microsoft SDL consists of twelve practices that support the integration of security requirements and compliance throughout the development process. It also assists developers in building robust software, reducing vulnerabilities, and lowering development costs, as shown in Table 7.

Microsoft SDL Practices	
1. **Cybersecurity related training** – educate personnel in cybersecurity best practices.	2. **Define Security Requirements** – ensure the security requirements are updated when changes of functionality and threat landscape occur.
3. **Define Metrics, and Compliance Reporting –** define the minimum acceptable level of security quality.	4. **Apply threat Modeling** – identify security vulnerabilities, determine risks and create mitigations.
5. **Establish Design Requirements** – selection of set of security features that all engineers should use.	6. **Define and use Cryptography Standards** – choosing the right cryptography solution for the protection of the data.
7. **Manage Security Risk when Using 3ª Party Components** – develop a plan to assess third-party components for vulnerabilities.	8. **Use Approved Tools** – selection of approved tools and associated security checks.
9. **Perform Static Analysis Security Testing** – check the compliance of good coding practices.	10. **Perform Dynamic Analysis Security Testing** – testing the security of the code in running software application.
11. **Perform Penetration Testing** – identify potential vulnerabilities from coding errors, poor system configuration or other operational weakness.	12. **Establish a Standard Incident Response Process** – prepare an incident response plan to address new threats that can emerge over time.

Table 7: Twelve Microsoft SDL Practices [RD-21]

2.3.2. SOFTWARE SECURITY TOUCHPOINTS

Developed by Gary MacGraw, the main objective of this security lifecycle was to implement security practices across all software development cycle. The term “touchpoints” is a mix of two activities, constructive and destructive, taking inspiration in the Yin and Yang symbol in one of his books [RD-22]. The constructive activity refers to building defences and security architectures and in the other hand, the destructive activity aims to assault those defences by exploring vulnerabilities. Table 8 presents the seven touchpoints forming this security lifecycle.

Touchpoint	Development Phase	Activity
1. Abuse cases	Requirements and Use Cases	Destructive
2. Security Requirements	Requirements and Use Cases	Constructive
3. Risk Analysis	Requirements and Use Cases / Architecture and Design	Constructive
4. Risk-Based Security Tests	Test Plans	Constructive and Destructive
5. Code Review (Tools)	Code	Constructive
6. Penetration Testing	Tests and Test Results / Feedback from the Field	Destructive
7. Security Operations	Feedback from the Field	Constructive

Table 8: Seven Touchpoints of Gary MacGraw [RD-22]

2.3.3. SOFTWARE ASSURANCE FORUM FOR EXCELLENCE IN CODE (SAFECODE)

Founded in 2007, SAFECode is a global industry forum comprised of business leaders and experts who collaborate to develop security programs with some well-known names in the cybersecurity community. The forum has been led by prominent cybersecurity personalities, including Paul Kurtz, the late Howard Schmidt, and today, Steven Lipner, who is widely recognized in the technology community as the “Father of the SDL”. Their focus on

cooperation is a step in the right direction, especially within cybersecurity, where collaboration is key to success. They recommend security practices in eight steps, shown as follows [RD-23]:

1. Application Security Control Definition (security requirements).
2. Design.
3. Secure Coding Practices (code standards, safe language).
4. Manage Security Risk Inherent in the Use of 3rd party components.
5. Testing and Validation.
6. Manage Security Findings (from previous steps).
7. Vulnerability Response and Disclosure.
8. Planning the Implementation and Deployment of Secure Development (plan at organization level).

2.3.4. NIST SECURE SOFTWARE DEVELOPMENT FRAMEWORK

As a more recent SSDL, NIST introduced a well-crafted Secure Software Development Lifecycle framework [RD-24]. This framework, published in February 2022, aims to address the shortcomings of the previous lifecycles by placing a strong emphasis on detailed software security practices.

The NIST SSDL recommends a core set of high-level security practices that can help software producers achieve several key goals. These goals include reducing the number of vulnerabilities in their software, mitigating the potential impact of zero-day vulnerabilities by addressing their root causes, and fostering improved communication by establishing a common vocabulary for secure software development.

The practices are gathered in four groups, as shown in Table 9.

Group	Practice
Prepare the Organization	• Define Security Requirements for Software development. • Implement roles and responsibilities. • Implement Supporting Toolchains (automation). • Define and Use Criteria for Software Security Checks. • Implement and Maintain Secure Environments for Secure Development.
Protect the Software	•Protect all Forms of Code from Unauthorized Access and Tampering. •Provide a Mechanism for Verifying Software Release Integrity. •Achieve and Protect Each Software Release.
Produce Well-Secure Software	•Design Software to meet Security Requirements and Mitigate Security Risks. •Review the Software Design to Verify Compliance with Security and Risk Information. •Verify Third-party Software Complies with Security Requirements. •Reuse Existing, Well-Secured Software wen feasible. •Create Source Code by Adhering to Secure Coding Practices. •Configure the Compilation, Interpreter, and Build Process to Improve Executable Security. •Review and/or Analyse Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements. •Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements. •Configure Software to Have Secure Settings by Default.
Respond to Vulnerabilities	•Identify and Confirm Vulnerabilities on an Ongoing Basis. •Assess, Prioritize, and Remediate Vulnerabilities. •Analyse Vulnerabilities to Identify Their Root Causes.

Table 9: NIST SP 800-218 SSDL Framework [RD-24]

3. CURRENT LANDSCAPE OF SPACE SYSTEM

This section provides a view of the current landscape of space systems in what concerns cybersecurity, as depicted in Figure 2.

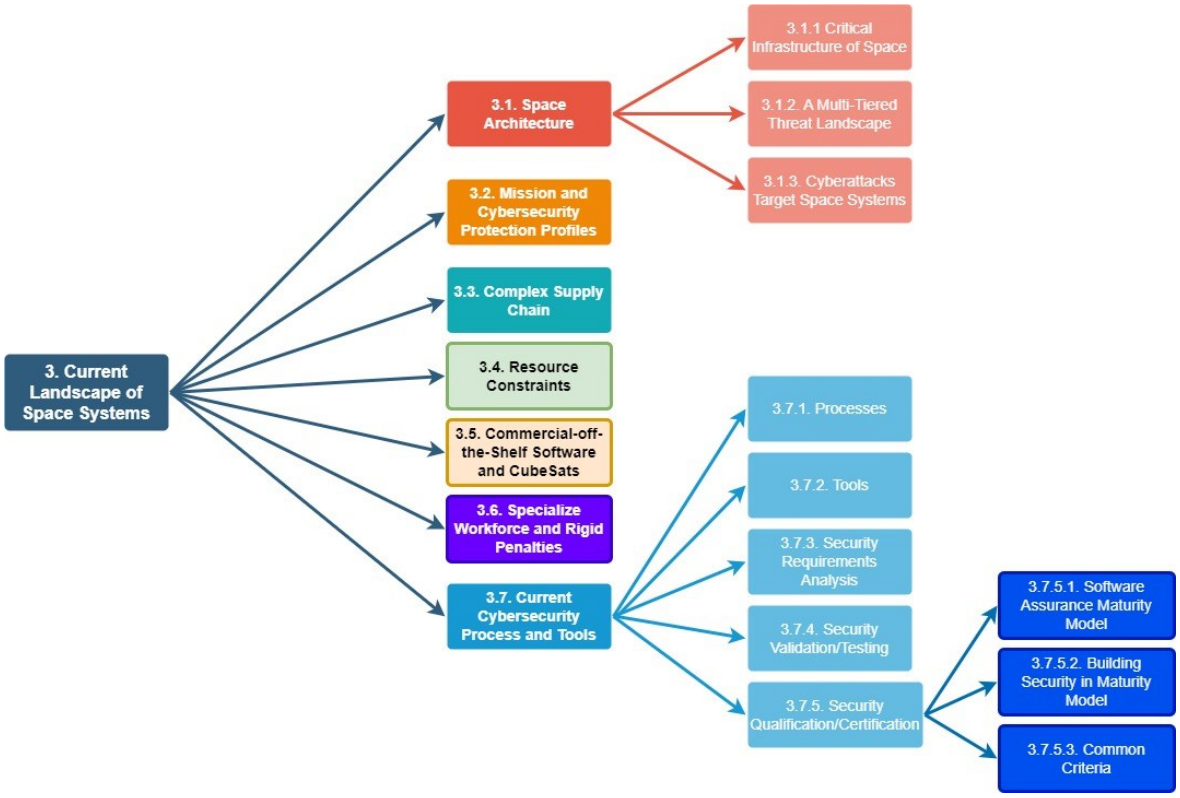

Figure 2: Overview of the Section 3

Early space systems relied heavily on analogue technology, even onboard spacecrafts. These systems were primarily mechanical and lacked the extensive digital infrastructure that now permeates modern space assets. As a result, cybersecurity concerns were not a major focus. At that time, the term “cybersecurity” didn´t exist, as it wasn´t until the 1970s when the first computer program called Creeper was created and could move across ARPANET´s network, a connectivity network developed prior to the internet. Security for the technology launched into space relied on tightly controlled access to its documentation and the highly complexity of the systems themselves, the so called “security by obscurity”.

As satellites shifted towards digital and software-driven systems, relying on "security by obscurity" became unfeasible. This transition offered numerous benefits like heightened productivity, quicker execution of processes, and enhanced product quality. Yet, it also introduced notable vulnerabilities, given that these interconnected systems were now accessible via the internet, leaving them exposed to potential threats from the public.

With digitization came the emergence of various protocols facilitating communication between network points. Each producer and operator of industrial control systems had their own protocols, ensuring what they believed to be sufficiently secure but susceptible to compromise. The lack of cybersecurity standards and regulations further exacerbated the security of space systems.

Today, challenges in space system cybersecurity revolve around how space systems are a single point of failure for various industry sectors, lack of imposing cybersecurity standards and regulations, complex supply chains, extended asset lifecycles, Commercial off-the-shelf (COTS) technology, resource constraints and specialized workforce and rigid penalties.

3.1. SPACE ARCHITECTURE

The operations of every industry sector are underpinned, to varying degrees, by space systems. Whether it's telecommunications, navigation, weather forecasting, military endeavors, or other industries, their infrastructures are considered crucial to both society and the nation. Recognizing their significance, substantial investments in their cybersecurity are essential to prevent any potential disruptions.

Figure 3 depicts a simplified architecture of a space system. Each component within it is considered a potential attack vector, and they are generally designed, manufactured, operated, and supported by different entities (supply chain vulnerabilities). The fact that sometimes each component has its own cybersecurity strategy means that not all threat intelligence is shared among the different interconnected components and therefore there is an increased risk that one of these components can impact the whole space system.

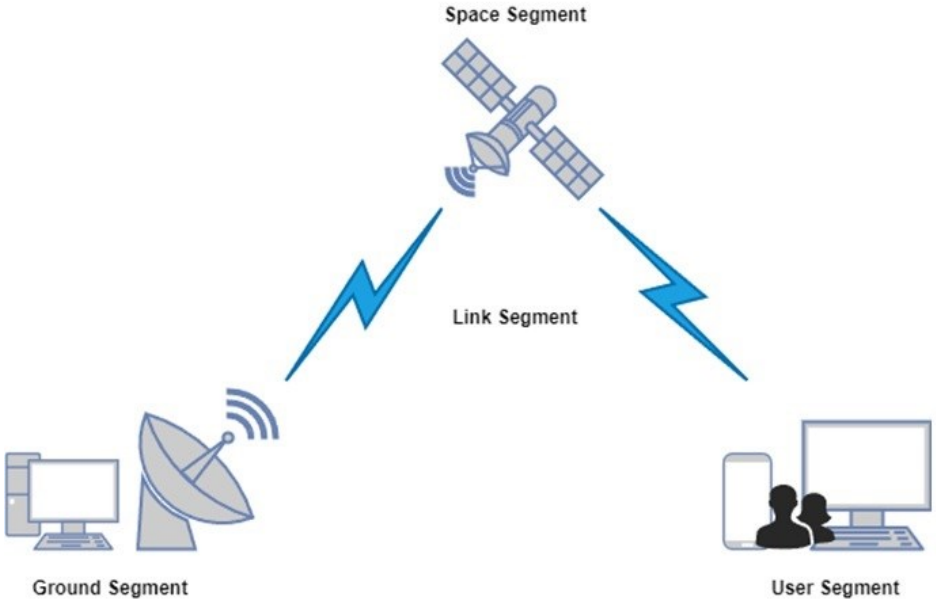

Figure 3: Common space system architecture

The Space Segment is the physical spacecraft itself, orbiting Earth or venturing further out. It houses the dedicated equipment for the mission's purpose, called the payload. This payload can range from scientific instruments like telescopes to communication antennas for relay satellites. Onboard computers process information, control the spacecraft's operations, and enable communication with the ground segment.

The Ground Segment encompasses all the infrastructure needed to communicate with and control the spacecraft. Ground stations act as communication hubs, exchanging data with the spacecraft through commands (telecommands) and receiving data transmissions (telemetries). Mission control centers monitor the spacecraft's health and location, analyze the received data, and send commands to control its operations. Launch facilities, where the spacecraft is prepared and launched into orbit, can also be considered part of this segment.

The Link Segment acts as a bridge between the ground and the space segment, and between the user and the space segments. It encompasses the methods used to transmit information between them, primarily relying on communication channels that utilize radio waves of specific frequencies. Factors like distance and data transmission rates influence the chosen frequencies. Tracking systems, often employing radar or optical telescopes, pinpoint the spacecraft's location and trajectory in orbit, allowing for accurate communication and mission planning.

The User Segment comprises the entities or individuals who ultimately benefit from the data or services provided by the space system. These users can include universities or researchers studying climate change who utilize data from Earth observation satellites. Similarly, users can rely on GPS technology to receive precise location information and use satellites to communicate with others.

3.1.1. CRITICAL INFRASTRUCTURE OF SPACE

The immense destructive potential of a cyberattack on Space Systems, such as the ability to wipe out an entire network of satellites, has become an appealing tactic for adversaries, especially during times of war. This was evident in the conflict between Russia and Ukraine, where there was a growing focus on developing and exploiting those capabilities.

Until recently, cybersecurity efforts for the ground segment primarily focused on its infrastructure. However, it's crucial to consider the entire segment as critical infrastructure. An attack disrupting or destroying its control system communications, which are essential for flight dynamics, orbit maintenance, and fuel and power management, could trigger severe negative economic impacts worldwide [RD-8].

An attack on one critical infrastructure has impacts on other critical infrastructures. Currently the USA and Europe are considering designating space-based assets as a critical infrastructure. This is crucial because Space sector, not only interacts with individuals but mainly with business, military and other critical infrastructures [RD-9] [RD-10].

3.1.2. A MULTI-TIERED THREAT LANDSCAPE

Adversaries employ a range of methods to achieve their nefarious goals, which vary depending on the system target (space, link, ground segments). Brandon Baily, a cybersecurity senior project leader at The Aerospace Corporation, classifies these methods into tiers based on the attackers' skills and the level of threat they pose [RD-6].

In this tiered system, depicted in Table 10, there are seven tiers, starting from tier 1, characterized by "script kiddie" attacks carried out by simple individuals, to tier 7, representing the most capable state actors conducting nation-state level attacks (e.g. supply chain intrusions). These tiers are depicted using a colour scheme, with yellow denoting a low threat level and shifting to dark red indicating a higher threat level.

Tier	Category	Skill
1	Script Kiddies	VERY LOW
2	Hacking for hire	LOW
3	Small hacker groups; Non-state actors	MODERATE
4	Insider threats	LOW HIGH
5	Large, well organized teams; non-state or state actors	HIGH
6	Highly capable state actors	VERY HIGH
7	Most capable state actors	VERY HIGH

Table 10: Threat tiers and Level of Sophistication

The communication link serves as the sole entry point for any satellite, used by the ground segment and the satellite itself for communication. For instance, if the communication link, such as the Command and Control (C2) link, has weak protection or lacks encryption, the satellite becomes vulnerable to two primary risk sources: seizure of control, and data corruption and interception [RD-11].

To highlight the importance of addressing cybersecurity in space systems, we have an example of a public hacking exercise called Hack-A-Sat [RD-7]. Currently (2023) in its fourth iteration, this event focuses on leveraging the talents of the cybersecurity community to develop strategies for hacking a satellite, ultimately identifying vulnerabilities and proposing solutions to enhance the security of space systems. Notably, the fourth iteration involved hacking a satellite in orbit, with the winning team completing the challenge in matter of hours [RD-8].

3.1.3. CYBERATTACKS TARGET SPACE SYSTEMS

The examples presented hereafter demonstrate the high level of vulnerability of each segment within a space system to cyberattacks, potentially putting businesses and, in the worst case, lives at risk. The specific segment affected helps determine the attribution of the attack.

Space segment:

In 2008, the Terra EOS AM-1 Earth observation satellite was hijacked twice (June 20th and October 22nd) for durations of 2 and 9 minutes, respectively. While believed to be the work of Chinese hackers, no commands were reportedly issued to the satellite [RD-3].

Ground segment:

At the beginning of the invasion of Ukraine by Russia, an attack targeted Viasat's KA-SAT ground station. This attack compromised tens of thousands of modems across Ukraine and parts of Europe, rendering them inoperable [RD-40]. The attack resulted in millions of dollars in damages and reputational harm to the company.

Link segment:

In 2017, the U.S. Maritime Administration reported a GPS spoofing attack targeting ships located in the Black Sea. During the attack, the ships' navigation tools were unable to display their correct location, simply showing "lost GPS fixing position." At one point, the spoofed location even indicated a position 25 nautical miles away, near Gelendzhik Airport [RD-41].

3.2. MISSION AND CYBERSECURITY PROTECTION PROFILES

Cybersecurity implementations cannot be a one-size-fits-all approach for space missions. Each mission carries unique threats that necessitate specific security requirements. The identification of these requirements is influenced by the mission itself. For example, in a manned mission, safety-of-life applications will take absolute priority.

According to the European Space Agency (ESA), space missions are categorized into five groups based on their associated risks.

Category	Security Requirements
Scientific	Ranging from orbiting Earth missions to interplanetary deep space missions, cybersecurity is a concern when the spacecraft is still in LEO.
Earth Observation	Meteorology, study of the oceans and wildlife missions can be put into this category. Typical requirements use in this category are the secure telecommand uplink, and in some cases include payload data and telemetry encryption.
Navigation	GPS and Galileu constellations fly in MEO and are consider as a critical infrastructure, therefore all aspects of security requirements are applied [RD-20].
Communication	Generally, all communication satellite providers are located at the most far away orbit GEO. In those cases, the high-speed connections and high bandwidth are the focus, not cybersecurity.
Manned Spaceflight and Exploration	In these missions all aspects of cybersecurity must be accounted for: confidentiality, integrity, and availability (CIA principles). They need flawless communications by voice or video.

Table 11: Mission Categories and Security Requirements

Depending on the mission type, a cybersecurity protection profile can be identified. ESA developed five profiles for TC and TM protection presented in the Table 12 [RD-19].

Profile	Description	Security Features	Examples
0. No specific Security	Lowest security level	•No TC authentication or encryption •No telemetry or science data encryption •Standard terrestrial link security (firewalls, IDP, SIEM)	•ERS/ENVISAT •Earth Explorers
1. Static Telecommand Protection	Basic TC protection	•TC authentication and anti-replay •Authentication key pre-loaded onboard •TC authentication can be enabled/disabled automatically or by ground	•MetOp •ATV
2. Dynamic Telecommand Protection	Enhanced TC protection	•TC authentication and anti-replay •Authentication keys loaded by ground using pre-installed Master Keys for TC encryption •TC authentication can be enabled/disabled automatically or by ground	•Sentinels
3. Dynamic Telecommand + Payload Data Protection	Protects TC and payload data	•Payload data encryption •Uses 4 key types: Master key, TC authentication key, payload data encryption key, TC encryption key •Payload data encryption can be enabled/disabled automatically or by ground	
4. Dynamic Telecommand + Payload + HKTM Data Protection	Highest security level	•Encrypts TC, payload data, and housekeeping telemetry (HKTM) data •Uses 5 key types: Master key, TC authentication key, data encryption key, HKTM data encryption key, TC encryption key - HKTM data encryption can be enabled/disabled automatically or by ground	

Table 12: Cybersecurity Profiles for TC and TM Protection

3.3. COMPLEX SUPPLY CHAIN

The space system comprises multiple components, and the space program is structured with a multi-layer contract. Before reaching the hands of the customer, the lower tier companies (subcontractors) specialized in technology, manufacture their products, which are then sent to the prime contractor who is responsible to manage, ensuring the procurement and integrate all systems components. This complexity in the supply chain not only pertains to physical aspects but also encompasses cybersecurity considerations. For example, the European space industrial supply chain is made up of the big four (Airbus, Thales, Leonardo and Safran) and other smaller groups such as Kongsberg, Dassault, RUAG. In total, the supply chain is comprised of more than 50000 people [RD-2]. This situation carries a potential risk due to the involvement of multiple companies and individuals. Information about the components might be vulnerable to interception by hackers through methods such as phishing attacks or other cyber threats. By analysing these components and pinpointing their vulnerabilities, a hacker could potentially compromise all system upon integration of the component.

The diagram below depicts the intricacies of the risk and responsibilities for a sample satellite project.

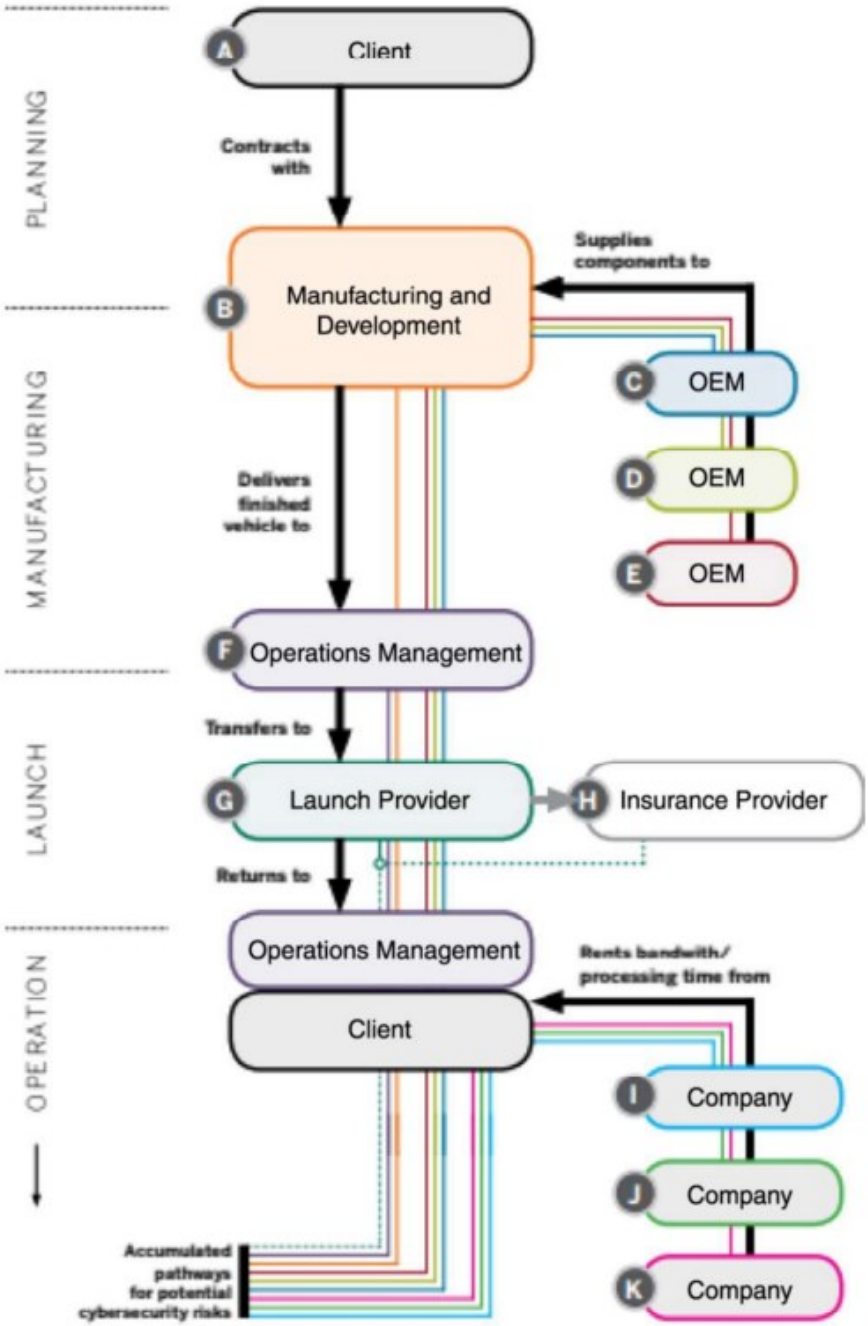

Figure 4: Cybersecurity responsibility landscape [RD-3]

The reading of the Figure 4 is as follows: **company A** give green light to develop a new satellite to the **company B (prime contractor),** responsible for the cybersecurity of the satellite. To acquire the necessary components **company B** subcontracts **companies C, D** and **E** who have their own responsibilities in cybersecurity. **Company B** assemble all the components and give the final product to the client. Then the client (**company A**) contracts another company (**company F**) to manage the operations of the satellite, who then assumes all the responsibilities about cybersecurity.

To put the satellite in space, **company F** contracts **company G** responsible for that purpose and as well the responsibility of cybersecurity during launch, but normally that responsibility is transferred to an insurance provider (**company H**).

When the satellite is in orbit and is ready to fulfil the mission purpose, the responsibility of cybersecurity for the operation shifts back to the **company F**.

Given a company's core focus on profitability, the client (**company A**) aims to optimize the satellite's utility. This involves maximizing bandwidth and processing capabilities for other companies like **I, J**, and **K**, who share responsibility for cybersecurity alongside their utilization of the satellite's services [RD-3].

3.4. RESOURCE CONSTRAINTS

Securing satellites presents a unique challenge due to limited power, environment, memory, and other constraints. Demanding processing power for robust cybersecurity measures can significantly impact a satellite's ability to perform its mission and maintain its orbit. This includes unplanned maneuvers to avoid debris or other satellites, which further strain the power budget.

Satellite energy management prioritizes survivability, often directing power to critical components during emergencies. Consequently, many satellite communication protocols are designed to be lightweight, sacrificing some security features like stronger encryption algorithms, for faster transmission speeds and lower power consumption. However, the optimal balance between performance and security depends on the specific mission objectives.

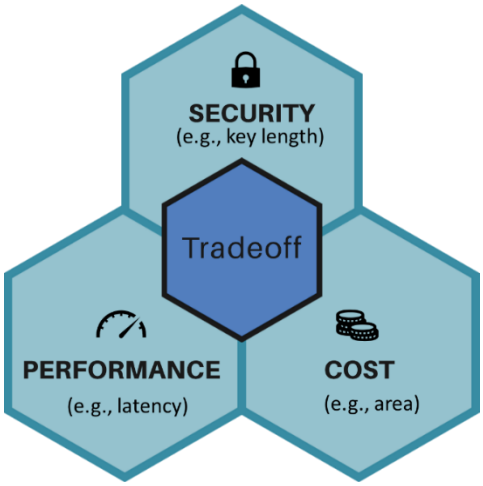

Figure 5: Trade-off between Security, Performance and Cost Dilemma [RD-18]

Limited budgets within companies can significantly impact their cybersecurity posture. This often leads to the prioritization of certain measures over others, potentially leaving vulnerabilities unaddressed. Furthermore, resource constraints can complicate compliance with industry-specific regulations, such as budget limitations for training and technology investments, making it challenging for companies to navigate and adhere to these standards.

Addressing one of this challenges, NIST (National Institute of Standards and Technology) launched a lightweight cryptography competition in 2018 [RD-18]. Aiming to address the needs of resource-constrained environments, the competition solicited submissions for new, more efficient cryptographic algorithms. With participation from 25 countries and over 50 submissions, the competition spurred significant innovation in this field.

The influx of innovative submissions has a wider impact than simply selecting a new standard. These advancements benefit not only resource-constrained environments like satellites and internet-of-things (IoT) devices, but also hold the potential to improve the overall efficiency and security of cryptographic algorithms across the board.

3.5. COMMERCIAL-OFF-THE-SHELF SOFTWARE AND CUBESATS

Throughout much of its history, only nations boasting robust financial resources and advanced technology could develop satellites equipped with intricate systems capable of implementing stringent security protocols. However, in the past decade, the space sector has witnessed a staggering total investment of 177.7 billion U.S. dollars distributed among 1,343 distinct companies. Predominantly, the United States and China stand as the primary contributors, accounting for 75% of this cumulative investment.

This substantial investment has facilitated the advancement of new technologies and the emergence of specialized technology companies, fostering the development of Commercial-off-the-Shelf (COTS) solutions within the space industry.

The commercial-off-the-shelf products brought with them some advantages mainly to the smaller companies in the sector, which are:

- Acquisition of COTS Software is cheaper.
- Easy implementation.
- Reviews of the product are online for everyone to see.

But also introduce some disadvantages in the form of security vulnerabilities:

- Long term costs (licenses and updating costs).
- Several individuals contribute to the code behind open-source technology, raising the potential for the insertion of backdoors or vulnerabilities into the software.
- Scalability and security aspects are the responsibility of the COTS manufacturers, but they are not the only ones suffering from the effects of unscalable and unsecure systems.

This software sees the most use among CubeSats due to their construction cost and launch services typically cost around 100,000 U.S dollars [RD-4]. However, employing Commercial Off-The-Shelf (COTS) components in these small satellites could expand vulnerability points, posing risks to both military and commercial assets. As per the nanosats Database [RD-5], 952 nanosats remain operational, with an anticipated launch of 2080 nanosats by 2027.

3.6. SPECIALIZED WORKFORCE AND RIGID PENALTIES

Critical positions within a company, especially those related with information security, require qualified individuals. To achieve this, a dedicated entity similar to the Securities and Exchange Commission (see section 2.2.2) could be established.

This entity would focus on:

- Providing training on cybersecurity to ensure that personnel can effectively manage and safeguard sensitive information.
- Investigating whether companies comply with established cybersecurity standards and best practices.

The SolarWinds incident in 2020, where a Chief Information Security Officer (CISO) oversaw inadequate security measures and misled the public, caused the company to be the target of a cyberattack leading to a breach in their servers [RD-8]. This highlights the importance of both thorough personnel training and strong accountability.

To ensure effective information security, there should be a combined approach: **robust training programs** alongside **substantial consequences** for those who fail to fulfill their cybersecurity responsibilities.

3.7. CURRENT CYBERSECURITY RELATED PROCESSES AND TOOLS

Cybersecurity is critical for guaranteeing the protection of systems, networks, and data. Organizations employ various strategies to identify, prevent, and respond to cyber threats. The following sections explore some of the essential tools that form the backbone of a robust cybersecurity posture, from firewalls and encryption to cutting-edge threat detection systems.

3.7.1. PROCESSES

Cybersecurity related processes are essential steps and practices that cybersecurity analysts implement to protect systems, networks, and data from cyber threats. Processes help prevent and defend against unauthorized access, attacks, and breaches. They follow a set of stages that are intertwine with each other [RD-26]:

- **Identify the Assets** – identify the assets that need to be protected against cyber threats.
- **Protect the Assets** – implement measures to safeguard the previously identified assets. On the technical side, cybersecurity analysts will recommend appropriate VPNs, encryption algorithms, and anti-malware solutions, among others. On the organizational side, employee awareness training should be employed.
- **Monitor System** – with the protection means implemented on the assets, a proactive attitude must be taken to monitor and test the system to see if there are vulnerabilities that can be favour to hackers.
- **Respond to Incidents** – despite the best efforts to protect systems, a cybersecurity incident response plan is essential in case of a breach. It contains a series of actions to be implemented to mitigate the effects of the cyberattack.
- **Recovery** – in the worst-case scenario of sensitive information leaks or operational disruptions, having backups and a recovery plan in place is crucial.

Table 13 presents some examples the space industry uses in protecting their projects.

Processes	Description	Examples
Risk Management	Manage the risk of a given threat by determining the probability (vulnerability **vs** threat intent **vs** threat capability) of a cyberattack target the three principles of cybersecurity. The next step is to calculate the technical, economic, and social impact of each attack and design a strategy to reduce the risk to an acceptable level, based on a cost-benefit analysis or feasibility assessment [RD-27].	• NIST Cybersecurity Framework • Risk Management Framework • ISO/IEC 27001
Access Management	Focuses on controlling a monitoring access to data, resources, and systems within an organization through a series of key steps [RD-28]: • **Authentication** – verifying the identity of a user, through a username and password or more modern methods like biometrics or multi-factor authentication. • **Authorization** – a given resource or information a user is authorized to access, based by role, permissions, and access levels. • **Access Control** – management of granting or denying access to a given resource or information taking into account the user´s authorization. There are three mechanism and include control lists, role-based and attribute-based access controls. • **User Management** – creating or deleting user accounts and modifying by managing attributes such roles, permissions, and access level. Policies for password management also take place in the user management. • **Preventing Unauthorized Access** – preventing access to sensitive resources or information by implementing multi-factor authentication, physical security measures, monitoring user access, and conduct analysis.	• IBM Identity and Access Management • Microsoft Identity and Access Management • CyberArk Certified Delivery Engineer

Processes	Description	Examples
Incident Response	Organizations develop a plan that utilizes tools and procedures to detect and respond effectively to cyberattacks, minimizing or preventing their impact. These plans usually include the following [RD-29]: • Roles and responsibilities of each member of the security team. • Software, hardware, and other security tools install across the organization. • Plan for restoring critical affected systems and data in the eventuality of a disruption attack. • Roadmap that guides organizations through effectively identifying, containing, eradicating, and recovering from a cyberattack. • Transparency communication to all contributors and customers of the company in case of an attack. • Document the attack and how it came to be, to a later date do an analysis and correct the vulnerabilities.	• Security information and event management (SIEM) • Security orchestration, automation, and response (SOAR) • Endpoint detection and response (EDR) • Extend detection and response (XDR)
Encryption	By using mathematical techniques, we can transform data to make it unreadable to the naked eye. Only those who possess the key can decrypt and understand it. There are two main types of cryptographic keys [RD-30]: **symmetric** and **asymmetric**. Symmetric keys use the same key for both encryption and decryption, offering better performance but posing a higher risk if compromised. Asymmetric keys utilize two separate keys: a public key for encryption, which can be widely distributed, and a private key for decryption, which is kept secret by authorized individuals.	• Data Encryption Standard (DES) • Advanced Encryption Standard • TripleDES • International Data Encryption Algorithm
Vulnerability Management	Ongoing process that helps organizations identify, assess, prioritize, and remediate vulnerabilities in their systems, networks, and applications, reducing organization´s overall risk exposure. It follows a set of phases: • **Discovery** – identify vulnerabilities in your asset inventory. • **Prioritization of assets** – prioritization of the assets by its criticality in the organization view. • **Assessment** – identify and prioritize the risk that need to be address first. • **Reporting** – Develop a security plan and report knows vulnerabilities. • **Remediation** – solve the vulnerabilities that are more concern to the organization. • **Verification and monitoring** – regular audits to ensure the threats are eliminated.	• Microsoft Defender Vulnerability Management • Tenable Nessus • Rapid7 InsightVM • Qualys VMDR

Table 13: Cybersecurity Processes

3.7.2. TOOLS

Organizations heavily rely on information technology, necessitating proactive measures to safeguard their confidential data. Cybersecurity tools play a vital role by continuously monitoring computer systems and networks, helping companies and individuals maintain data privacy and security.

Depending on the threats, cybersecurity tools can be tailored to counter them. Figure 4 depicts examples of cybersecurity tools.

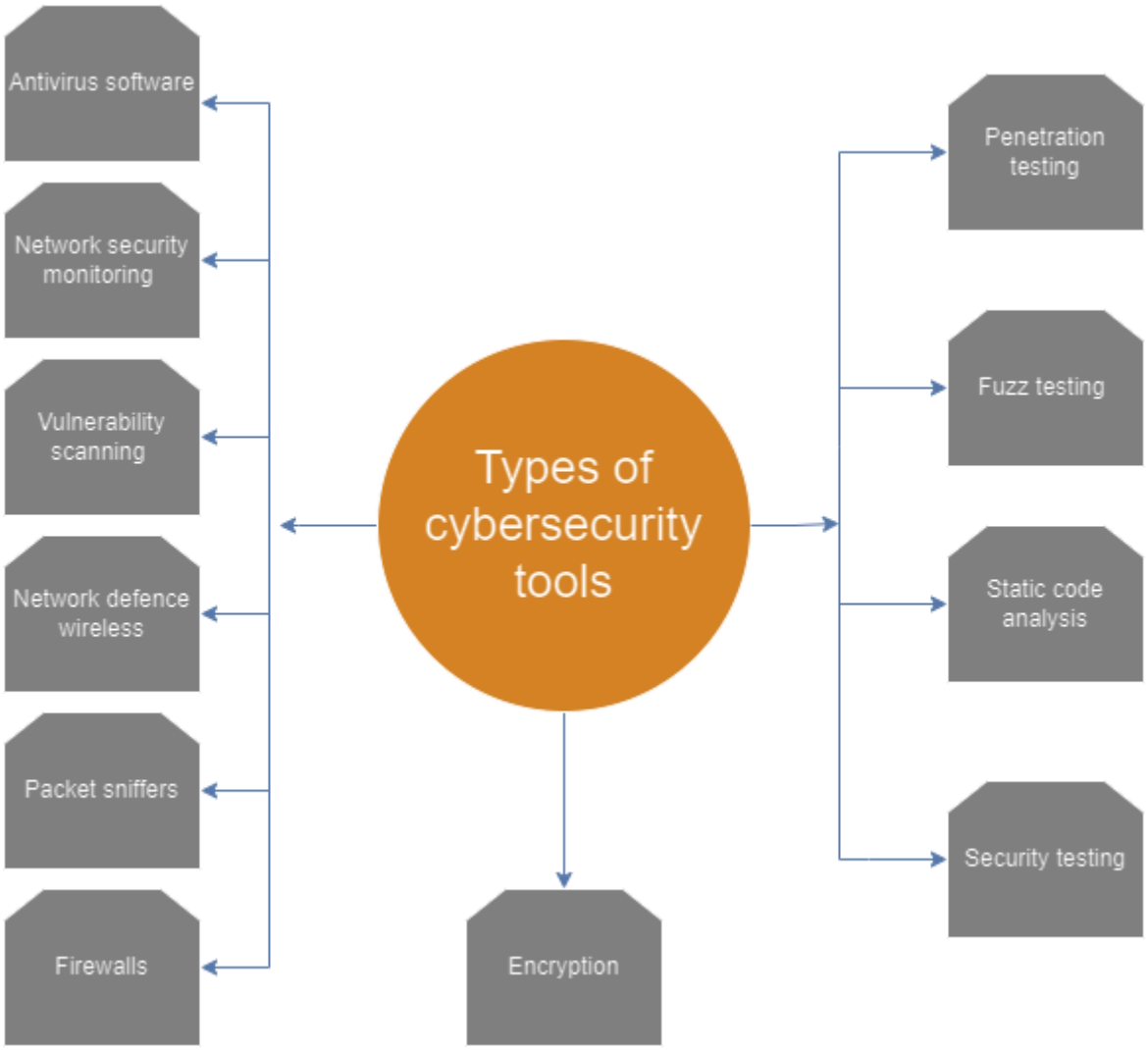

Figure 6: Types of cybersecurity tools adapted from [RD-31]

Table 14 provides some details of the different types of tools [RD-31].

Tool	Description	Examples
Network security monitoring	Identify external threats by detecting and preventing attacks. Use in organizational level + ground segment.	• Cisco Stealthwatch • Darktrace • Suricata
Vulnerability scanning	Identify weaknesses in software and hardware components. Use in organizational level + ground segment.	• Splunk • OpenVAS • Acunetix
Network defence wireless-tools	Protect data while maintaining the network´s usability and integrity. Use in organizational level + ground segment.	• KisMAC • OpenWIPS • Cisco ISE
Encryption	Scramble data to ensure confidentiality during transmission or storage. Use in organizational level + ground and space segment.	• Advanced Encryption Standard • FIPS 140 • Rivest-Shamir-Adleman

Tool	Description	Examples
Firewalls	Control incoming and outgoing traffic, acting as a barrier between trusted and untrusted networks. Use in organizational level + ground segment.	• Fortinet Fortigate • CiscoASA • Palo Alto Networks Firewall
Packet sniffers	Applications that analyse the network and gather data to be analyse by technicians. Use in organizational level + ground segment.	• Wireshark • tcpdump • Fiddler
Antivirus software	Monitor, block and remove any malicious software that compromises the computer functionality. Use in organizational level + ground segment.	• Windows Defender • Bitdefender • Norton
Penetration testing	Detect vulnerabilities by staging an attack to their systems. Use in organizational level + ground and space segment.	• Kali Linux • Metasploit
Fuzz testing	Software testing to uncover implementation bugs by injecting “trash” input and analyse the reaction. Use in organizational level + ground and space segment.	• PortSwigger Burp Suite Professional • OWASP WSFuzzer • Synopsys Defensics
Static code analysis	Analysis the source code to identify potential issues, bad practices and vulnerabilities. Use in organizational level + ground and space segment.	• Astrée • OWASP WSFuzzer • MISRA
Security testing	Comprise with performance tools, simulated attacks, buffer overflow, DoS testing and equivalence class partitioning test. Use in organizational level + ground and space segment.	• Kali Linux • Metasploit

Table 14: Cybersecurity Tools

A more extensive list of tools can be found in Annex A.

3.7.3. SECURITY REQUIREMENTS ANALYSIS

Security requirements analysis plays a crucial role in safeguarding information and systems from cyber threats. Therefore, developing well-defined security requirements in the project design phase is imperative. This systematic process identifies, documents, and refines the security needs of a system or organization. The first step involves identifying and classifying systems to assess what kind of threats and vulnerabilities they may have. With this list of threats, a risk assessment is then conducted to evaluate the likelihood and potential impact of each threat, identifying the most concerning ones.

Based on the identified threats and risks, specific security requirements are established. Commonly used standards that focus on cybersecurity can be leveraged, such as IEC 62443 [RD-16].

Consider IEC 62443-3-3, which is described in more detail in the technical note Analysis of Practices and International Standards Related to Cybersecurity [RD-42]. This standard outline seven foundational requirements, each containing several specific security requirements, depicted in Table 15.

Foundational Requirement	Security Requirements
Identification and Authentication Control	•Human user identification and authentication •Software process and device identification and authentication •Account management

Foundational Requirement	Security Requirements
Use Control	•Authorization enforcement •Wireless use control •Use control for portable and mobile devices
System Integrity	•Communication integrity •Malicious code protection •Security functionality verification
Data Confidentiality	•Information confidentiality •Information persistence •Use of cryptographic
Restricted Data Flow	•Network segmentation •Zone boundary protection •General purpose person-to-person communication restrictions
Timely Response to Events	•Audit log accessibility
Resource Availability	•Denial of service protection •Resource management •Control system backup

Table 15: IEC 62443-3-3 Foundational and Security Requirements

3.7.4. SECURITY VALIDATION/TESTING

Security validation/testing is a systematic process of evaluating the effectiveness of security controls in a system or organization. It serves as a form of proactive defence, testing the security of a system to identify weaknesses before attackers exploit them. This allows for the implementation of appropriate security controls to address these vulnerabilities.

The IEC 62443-4-1 standard includes a section that focuses on documenting the outcomes of all security testing. This documentation helps verify if the security requirements are being met, ensuring the product is effectively maintained throughout its real-world use.

Security Verification and Validation Testing	Examples
Security Requirements Testing	•Functional testing of security requirements •Performance and scalability testing •Boundary/edge condition, stress and malformed or unexpected input test
Threat Mitigation Testing	•Create and execute mitigation plans •Create and execute plans for attempting to thwart each mitigation
Vulnerability Testing	•Attack surface analysis •Black box known vulnerability scanning •Dynamic runtime resource management testing
Penetration Testing	•Password cracking •Privilege escalation •Exploiting software vulnerabilities
Independence of Testers	•Security consulting firms •Freelance security researchers •Internal audit teams with dedicated security expertise

Table 16: IEC 62443-4-1 Security verification and validation testing

In the automotive industry, cybersecurity homologation assessment using the Automotive SPICE tool became mandatory in July 2022. To verify and validate cybersecurity requirements, the industry began conducting threat modeling to identify critical interfaces, data, and asset groups.

Following threat modeling, a Threat Analysis and Risk Assessment (TARA), as detailed in ISO 21434 [RD-25], is performed to measure threat and impact levels. This allows for the calculation of the security level (threat + impact) and the definition of cybersecurity goals.

These security goals are then broken down into system requirements, critical functions, and software data requirements. To validate the identified requirements, penetration testing teams utilize a structured test strategy based on attack patterns derived from the MITRE framework [RD-32].

3.7.5. SECURITY QUALIFICATION/CERTIFICATION

Security qualification and certification are two critical processes that help organizations build trust and confidence in the security posture of their systems and software.

Security Qualification involves rigorous testing and evaluation of a system or software product against a set of predetermined security requirements in areas like access control, encryption, data integrity, and vulnerability management.

Security Certification builds upon security qualification. A software product successfully undergoes a qualification process and is awarded a formal certification by a recognized certification body. This certification signifies that the product meets a specific set of security criteria and demonstrates a level of assurance regarding its security posture.

There are different assessment approaches to evaluate the maturity of secure development, for example:

- Software Assurance Maturity Model (SAMM)
- Building Security In Maturity Model (BSIMM)
- Common Criteria (CC)

3.7.5.1. SOFTWARE ASSURANCE MATURITY MODEL

One way to qualify and certify software is by using the Software Assurance Maturity Model (SAMM), developed by OWASP. SAMM's main purpose is to provide an effective and measurable analysis of an organization's secure development lifecycle (SDL) and to recommend improvements.

SAMM is an open framework, risk-driven, and constantly evolving. It helps organizations formulate and implement strategies for improving their software security posture [RD-33].

Figure 7 illustrates the areas covered by the framework.

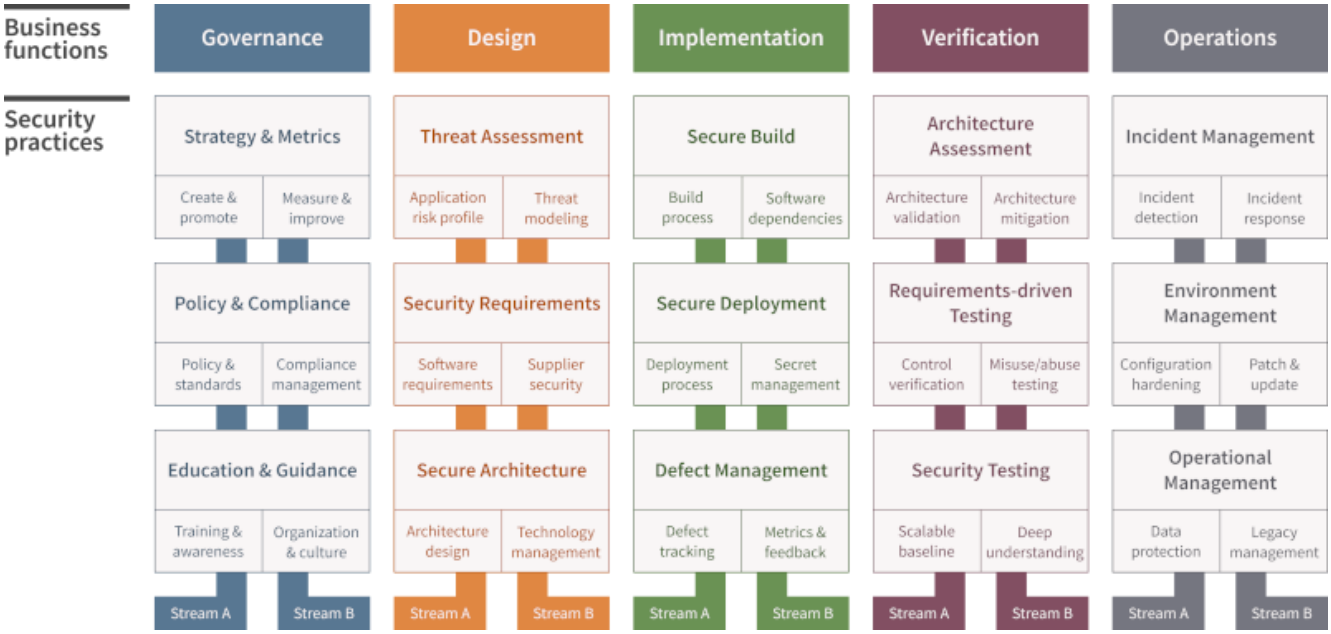

Figure 7: OWASP SAMM v2 model [RD-33]

3.7.5.2. BUILDING SECURITY IN MATURITY MODEL.

Gaining an external perspective on a company's security posture is crucial. The Building Security In Maturity Model (BSIMM) from Synopsys can be a valuable tool in this regard.

BSIMM analyzes and benchmarks software security programs across various industries. It does this by comparing a company's security needs against established best practices in software security programs. This analysis helps identifying strengths and weaknesses, enabling the company to prioritize improvements and evolve its security posture.

Ultimately, BSIMM facilitates building trust with stakeholders by allowing you to share a clear picture of the company's security posture.

Additionally, BSIMM helps organizations identify the most effective security practices to achieve a strong security posture, considering available resources, time, and budget [RD-34].

3.7.5.3. COMMON CRITERIA

The third and final example of security qualification/certification is the Common Criteria for Information Technology Security Evaluation (CC). The CC is an internationally recognized framework for evaluating the security of Information Technology (IT) products [RD-35].

It has two companion documents:

- Common Methodology for Information Technology Security Evaluation (CMITSE) [RD-36]: This document is divided into five parts, covering the introduction and general models, security functional requirements, security assurance requirements, a framework for specifying evaluation methods and activities, and predefined packages of security requirements.
- Common Criteria Recognition Arrangement (CCRA) [RD-37]: This arrangement outlines four objectives that are shared by its signatories.
    - **Maintain high evaluations standards**: Ensure IT product and protection profile evaluations are rigorous, consistent, and instill confidence in their security.
    - **Increase availability of secure products**: Promote the development of and availability of thoroughly evaluated IT products with robust security features.

- **Eliminate Duplicate Evaluations**: Prevent redundant evaluations by establishing a globally recognized framework.
- **Enhance efficiency and cost effectiveness**: Continuously improve the evaluation and certification process to be more efficient and cost effective for all stakeholders.

ANNEXES

ANNEX A. TOOLS

This section explores various tools for vulnerability scanning and source code analysis.

A.1 VULNERABILITY SCANNING TOOLS

Table 17 presents a variety of vulnerability scanning tools recommended by OWASP. These tools primarily focus on web applications and aim to identify security vulnerabilities.

Name	Owner	License	Platforms	Note
Acunetix	Acunetix	Commercial	Windows, Linux, MacOS	Free (Limited Capability)
Akto	Akto	Commercial	SaaS or On-Premises	150+ tests, free trial available
APIsec	APIsec	Commercial	SaaS	Free limited API Pen Test
App Scanner	Trustwave	Commercial	Windows	
AppCheck Ltd.	AppCheck Ltd.	Commercial	SaaS	Free trial scan available
AppScan	HCL Software	Commercial	Windows	
AppScan on Cloud	HCL Software	Commercial	SaaS	
AppSpider	Rapid7	Commercial	Windows	
AppTrana	Indusface	Free	SaaS	
Aptori	Aptori	Commercial	SaaS or On-Premises	
Arachni	Arachni	Free	Most platforms supported	Free for most use cases
Astra Security Suite	Astra Security	Free	SaaS	Paid Option Available
Beagle Security	Beagle Security	Commercial	SaaS	Free (Limited Capability)
beSECURE	Beyond Security	Commercial	SaaS	Free (Limited Capability)
Blacklock	Blacklock Security	Commercial	Any	14-day trial
BlueClosure BC Detect	BlueClosure	Commercial	Most platforms supported	2-week trial
BREACHLOCK Dynamic Application Security Testing	BREACHLOCK	Commercial	SaaS	
Burp Suite	PortSwigger	Commercial	Most platforms supported	Free (Limited Capability)
CI Fuzz CLI	Code Intelligence	Open Source	Most platforms supported	
CloudDefense	CloudDefense	Commercial	SaaS or On-Premises	CloudDefense DAST integrates with any CI/CD with just 1 line of code. It supports multiple authentication types. Perform deep DAST scans with ease.
Code Intelligence App	Code Intelligence	Commercial	SaaS or On-Premises	

Name	Owner	License	Platforms	Note
Contrast	Contrast Security	Commercial	SaaS or On-Premises	Free (Full featured for 1 App)
Crashtest Security	Crashtest Security	Commercial	SaaS or On-Premises	
Cyber Chief	Audacix	Commercial	SaaS or On-Premises	
Deepfence ThreatMapper	Deepfence	Open Source	Linux	Apache v2
Deepfence ThreatStryker	Deepfence	Commercial	Linux, Windows	
Detectify	Detectify	Commercial	SaaS	
Digifort- Inspect	Digifort	Commercial	SaaS	
Edgescan	Edgescan	Commercial	SaaS	
Escape	Escape	Commercial	SaaS	Run thousands of GraphQL security scans
GamaScan	GamaSec	Commercial	Windows	
GoLismero	GoLismero Team	Open Source	Windows, Linux and Macintosh	GPLv2.0
Grabber	Romain Gaucher	Open Source	Python 2.4, BeautifulSoup and PyXML	
GraphQL Security	Escape	Free	SaaS	Free. No account required.
Grendel-Scan	David Byrne	Open Source	Windows, Linux and Macintosh	
Harbor	Harbor	Open Source	Cloud-native (Kubernetes and Docker)	Is a powerful tool for managing and securing container images, ensuring compliance, and maintaining performance across cloud-native platforms like Kubernetes and Docker.
Heyhack	Heyhack	Commercial	SaaS or On-Premises	Free trial available
Holm Security	Holm Security	Commercial	Saas or On-Premises	
HostedScan.com	HostedScan.com	Commercial	SaaS	Free Forever
IKare	ITrust	Commercial	N/A	
ImmuniWeb	High-Tech Bridge	Commercial	SaaS	Free (Limited Capability)
Indusface Web Application Scanning	Indusface	Commercial	SaaS	Free trial available
InsightVM	Rapid7	Commercial	SaaS	Free trial available
Intruder	Intruder Ltd.	Commercial		
Invicti, formerly Netsparker	Invicti	Commercial	Windows	

Name	Owner	License	Platforms	Note
IOTHREAT	IOTHREAT	Commercial	SaaS	Free (View Partial Results). Full report (PRO) - 50% discount for the OWASP community with 'OWASP50'.
K2 Security Platform	K2 Cyber Security	Commercial	SaaS/On-Premises	Free trial available
Kayran	Kayran	Commercial	All Web Applications Supported	Automatic Penetration Testing for Web Applications & API Schema Penetration Testing
Mayhem for API	ForAllSecure	Commercial	SaaS	30-day Free Trial
N-Stealth	N-Stalker	Commercial	Windows	
Nessus	Tenable	Commercial	Windows	
Nexploit	NeuraLegion	Commercial	SaaS	
Nexpose	Rapid7	Commercial	Windows/Linux	Free (Limited Capability)
Nikto	CIRT	Open Source	Unix/Linux	
Nmmapper Tool Collections	Nmmapper	Commercial	SasS	Great Collection of Kali Tool hosted online
Nuclei	ProjectDiscovery	Open Source	Windows, Unix/Linux, and Macintosh	Fast and customisable vulnerability scanner based on simple YAML based DSL.
OnSecurity Protect	OnSecurity LLP	Commercial	SaaS	Free tier and free trial available.
OpenApi Security	Escape	Free	SaaS	Free. No account required.
OpenVAS by Greenbone	greenbone	Open Source	Linux	Open source full-featured vulnerability scanner, developed and maintained by Greenbone Networks GmbH.
OSTE Meta Scanner	OSTEsayed	Open Source	Linux	OSTE meta scanner is a comprehensive web vulnerability scanner that combines multiple DAST scanners, including Nikto Scanner, ZAP, Nuclei, SkipFish, and Wapiti.
Pentest-Tools.com Website Scanner	Pentest-Tools.com	Commercial or Free	SaaS	Finds vulnerabilities such as XSS (testing using real browsers), Server-Side Template Injection, Code Injection (with out of band detection) and other OWASP Top 10, and more high-risk vulnerabilities. Even newer vulnerabilities such as Client-Side Prototype Pollution are included.
Probely	Probely	Commercial	SaaS	Free (Limited Capability)
Proxy.app	Websecurify	Commercial	Macintosh	

Name	Owner	License	Platforms	Note
purpleteam	OWASP	Open Source	CLI and SaaS	GNU-AGPL v3
QualysGuard	Qualys	Commercial	N/A	
ReconwithMe	Nassec	Commercial	SaaS	Paid Option Available
Retina	BeyondTrust	Commercial	Windows	
Ride (REST JSON Payload fuzzer)	Adobe, Inc.	Open Source	Linux / Mac / Windows	Apache 2
ScanRepeat	Ventures CDX	Commercial	SaaS	
ScanTitan Vulnerability Scanner	ScanTitan	Commercial	SaaS	Free (Limited Capability)
Sec-helpers	VWT Digital	Open Source or Free	N/A	
SecOps Solution	SecOps Solution	Commercial or Free	Windows/Linux/SaaS/On-Premises	An agent-less full-stack vulnerability and patch management platform for identifying and remediating vulnerabilities in servers, network devices, endpoints, web application and mobile application
SecPoint Penetrator	SecPoint	Commercial	N/A	
SecretScanner	Deepfence	Open Source	Linux	Find secrets (tokens, keys, passwords, etc) in containers and filesystems, supporting approx. 140 different secret types
Security For Everyone	Security For Everyone	Commercial	SaaS	Free (Limited Capability)
Securus	Orvant, Inc	Commercial	N/A	
Secyour Scanner	Secyour	Commercial	Windows, Linux, MacOS	Free (Limited Capability)
Sentinel	WhiteHat Security	Commercial	N/A	
SmartScanner	SmartScanner	Commercial	Windows	Free (Limited Capability)
SOATest	Parasoft	Commercial	Windows / Linux / Solaris	
SOOS DAST	SOOS	Commercial	SaaS	Free 30-day Trial. Includes DAST & SCA (OSS Vuln detection, Licenses, Policies, SBOM). Affordable flat rate price.
StackHawk	StackHawk	Commercial	SaaS	
ThreatMapper	Deepfence	Open Source	Linux	Open-source vulnerability discovery and prioritization for Kubernetes, Docker, Serverless and host-based workloads

Name	Owner	License	Platforms	Note
Threatspy	Secure Blink	Commercial or Free	SaaS	Threatspy is the heuristic Application Security Management Platform. Discovering Both Known and Unknown Vulnerabilities, Strategically Prioritized via the Reachability Framework, with Built-in Automated Remediation.
Tinfoil Security	Synopsys	Commercial	SaaS or On-Premises	Free (Limited Capability)
Trustkeeper Scanner	Trustwave SpiderLabs	Commercial	SaaS	
Vega	Subgraph	Open Source	Windows, Linux and Macintosh	
Vex	UBsecure	Commercial	Windows	
Vulners	Vulners Inc	Commercial or Free	Linux / Windows / Plugins / SaaS	SAST and DAST software vulnerability scanner based on the Vulners database. Including different integrations for administration and security tools, such as NMAP, Burp, Ansible and more.
VulnSign	VulnSign	Commercial	SaaS or On-Premises	
w3af	w3af.org	Open Source	Linux and Mac	GPLv2.0
Wapiti	Informática Gesfor	Open Source	Windows, Unix/Linux and Macintosh	
Web Security Scanner	DefenseCode	Commercial	On-Premises	
WebApp360	TripWire	Commercial	Windows	
WebCookies	WebCookies	Free	SaaS	
WebInspect	Micro Focus	Commercial	Windows	
WebReaver	Websecurify	Commercial	Macintosh	
WebScanService	German Web Security	Commercial	N/A	
Websecurify Suite	Websecurify	Commercial	Windows, Linux, Macintosh	Free (Limited Capability)
Website Security Check	CyberAnt	Commercial	SaaS	20% off with OWASP20
WPScan	WPScan Team	Commercial	Linux and Mac	Free options
Zed Attack Proxy	The ZAP Development Team	Open Source	Windows, Unix/Linux, and Macintosh	Apache-2.0

Table 17: Vulnerability Scanning Tools [RD-38]

A.2 SOURCE CODE ANALYSIS TOOLS

Table 18 presents a variety of source code analysis tools, also known as Static Application Security Testing, recommended by OWASP. These tools analyse source code or compiled code to help identify security vulnerabilities.

Name	Owner	License	Platforms	Note
.NET Security Guard		Open Source or Free		.NET, C\#, VB.net
42Crunch		Commercial		REST API security platform that includes Security Audit (SAST), dynamic conformance scan, runtime protection, and monitoring.
ABOM Scanner	Vulert	Free	SaaS	ABOM is an online SwCA (Software Composition Analysis) tool that scans your application for open-source vulnerabilities using only a manifest file. Covering PHP, JavaScript, Rust, Python, and other top languages.
Agnitio		Open Source or Free	Windows	ASP, ASP.NET, C\#, Java, Javascript, Perl, PHP, Python, Ruby, VB.NET, XML
Aikido Security	Aikido Security	Commercial or Free	SaaS	Aikido Security is a developer-friendly software security platform. It scans your code, containers & cloud in 9 different ways, showing you which security issues and vulnerabilities are important to solve. We speed up triaging massively by cutting out the false-positives and making things human-readable.
APIsecurity.io Security Audit		Open Source or Free		online tool for OpenAPI / Swagger file static security analysis
AppSweep	Guardsquare	Open Source or Free	SaaS	Mobile application security testing tool for compiled Android apps with support of CI/CD integration
Arnica	Arnica.io	Commercial or Free	SaaS	Arnica is an end-to-end security solution that includes SAST, SCA, IaC, Licensing, Reputation, Hardcoded Secrets Mitigation, Permissions and more. Arnica's pipelineless approach empowers developers through automated mitigations and real time feedback.
Automated Security Helper	AWS	Open Source or Free		ASH is a one stop shop for security scanners and does not require any installation. It will identify the different frameworks, and download the relevant, up to date tools. ASH is running on isolated Docker containers, keeping the user environment clean, with a single aggregated report. The following frameworks are supported: Git, Python, Javascript, Cloudformation, Terraform and Jupyter.
Bandit		Open Source or Free		Bandit is a comprehensive source vulnerability scanner for Python
Bearer CLI	Bearer	Open Source or Free	CLI on Windows, MacOS, Linux, Docker, CI/CD integration	Static Application Security Testing (SAST) to discover, filter and prioritize security and privacy risks using sensitive data flow analysis. Currently supports Java, Ruby, JavaScript and TypeScript.

Name	Owner	License	Platforms	Note
Betterscan CE (Community Edition)	Marcin Kozlowski	Open Source		Code Scanning/SAST/Static Analysis/Linting using many tools/Scanners with One Report. Currently supports: PHP, Java, Scala, Python, Ruby, Javascript, GO, Secret Scanning, Dependency Confusion, Trojan Source, Open Source and Proprietary Checks (total ca. 1000 checks). Supports also Differential analysis. Goal is to have one report using many tools/scanners
Beyond Security beSOURCE	Beyond Security	Commercial		Static application security testing (SAST) used to be divorced from Code quality reviews, resulting in limited impact and value. beSOURCE addresses the code security quality of applications and thus integrates SecOps into DevOps.
BlueClosure BC Detect	BlueClosure	Commercial		Analyses client-side JavaScript.
Brakeman		Open Source or Free		Brakeman is an open-source vulnerability scanner specifically designed for Ruby on Rails applications
bugScout	Nalbatech, Formerly Buguroo	Commercial		
CAST AIP		Commercial		Performs static and architectural analysis to identify numerous types of security issues. Supports over 30 languages. [AIP's security specific coverage is here](https://www.castsoftware.com/solutions/application-security/cwe#SupportedSecurityStandards).
clj-holmes	clj-holmes	Open Source	Linux and MacOs	A CLI SAST (Static application security testing) tool which was built with the intent of finding vulnerable Clojure code via rules that use a simple pattern language.
CloudDefense	CloudDefense	Commercial	SaaS or On-Premises	CloudDefense provides holistic threat intelligence across all attack surfaces - Containers, Kubernetes, Code, Open-Source Libraries, APIs and more...
Codacy		Commercial		Offers security patterns for languages such as Python, Ruby, Scala, Java, JavaScript and more. Integrates with tools such as Brakeman, Bandit, FindBugs, and others. (free for open-source projects)
CodeScan Cloud		Commercial		A Salesforce focused, SaaS code quality tool leveraging SonarQube's OWASP security hotspots to give security visibility on Apex, Visualforce, and Lightning proprietary languages.
CodeSonar	GrammaTech	Commercial		tool that supports C, C++, Java and C# and maps against the OWASP top 10 vulnerabilities.
CodeThreat	CodeThreat	Commercial or Free	SaaS or On-Premises	Developer-friendly SAST and SCA solutions, Integration with CI/CD pipelines complement a robust DevSecOps strategy, and AI-powered features provide actionable insights with code fix suggestions and potential attack scenarios, helping developers remediate identified issues promptly.
Codiga	Codiga	Commercial	SaaS or On-Premises	Codiga scans your code and find security, safety, design, performance and maintainability issues in your code at each push or pull request. It integrates with GitHub, GitLab and Bitbucket.
CoGuard	Heinle Solutions Inc.	Commercial	SaaS or On-Premises	A SAST tool for infrastructure configuration analysis. Support for common web servers, databases, streaming

Name	Owner	License	Platforms	Note
				services, authentication services, container orchestration and Infrastructure-as-Code tools.
Contrast Assess		Commercial		Contrast performs code security without doing static analysis. Contrast does Interactive Application Security Testing (IAST), correlating runtime code & data analysis. It provides code level results without relying on static analysis.
Coverity Static Analysis	Synopsys	Commercial		Apex, C/C++, C#, CUDA, Java#, JavaScript, PHP, Python, .NET Core, ASP.NET, Objective-C, Go, JSP, Ruby, Swift, Fortran, Scala, VB.NET, iOS, Android, TypeScript, Kotlin
CxSAST	Checkmarx	Commercial	Saas, or on-premises. Windows and Linux with CI/CD and IDE plugin integration	Run full or incremental source code security scans. Supported languages include Javascript, Java, Apex, PHP, Python, Swift, Scala, Perl, Groovy, Ruby, C++, C#.NET, PL/SQL, VB.NET, ASP.NET, HTML 5, Windows Mobile, Go, and Kotlin.
Cycode | Complete ASPM		Commercial	SaaS, On-Premises, IDE Plugin	Cycode is a complete ASPM that also has its own native scanners tools from code to cloud, including native SAST and native SCA scanners.
Dawnscanner		Open Source or Free		Dawnscanner is an open-source security source code analyzer for Ruby, supporting major MVC frameworks like Ruby on Rails, Padrino, and Sinatra. It also works on non-web applications written in Ruby.
DeepSource	DeepSource Corp.	Commercial	SaaS or On-Premises	DeepSource helps companies ship clean and secure code with powerful static analysis, OWASP Top 10 compliance, and Autofix. Supports all major programming languages.
DerScanner	DerScanner Ltd.	Commercial		Capable of identifying vulnerabilities and backdoors (undocumented features) in over 30 programming languages by analyzing source code or executables, without requiring debug info.
Enlightn	Enlightn Software	Open Source		Enlightn is a vulnerability scanner specifically designed for Laravel PHP applications that combines SAST, DAST, IAST and configuration analysis techniques to detect vulnerabilities.
Find Security Bugs		Open Source or Free		Java, Scala, Groovy
FindBugs		Open Source or Free		Find bugs (including a few security flaws) in Java programs [Legacy - Not Maintained - Use SpotBugs (see other entry) instead]
FindSecBugs		Open Source or Free		A security specific plugin for SpotBugs that significantly improves SpotBugs's ability to find security vulnerabilities in Java programs. Works with the old FindBugs too.
Flawfinder		Open Source or Free		Scans C and C++.
Fluid Attack's Scanner	Fluid Attacks	Open Source		SAST, DAST and SCA vulnerability detection tool with perfect OWASP Benchmark score.

Name	Owner	License	Platforms	Note
Fortify	Micro Focus	Commercial	Windows, Linux, and MacOSX	Free trial scan available. Supported languages include: ABAP/BSP, ActionScript/MXML (Flex), APEX, ASP.NET, VB.NET, C\# (.NET), C/C++, Classic ASP (w/VBScript), COBOL, ColdFusion CFML, Go, HTML, Java (including Android), JavaScript/AJAX, JSP, Kotlin, Objective-C, PHP, PL/SQL, Python, Typescript, T-SQL, Ruby, Scala, Swift, Visual Basic (VB.NET), Visual Basic 6, VBScript, XML
GitGuardian — Automated Secrets Detection		Commercial	SaaS or On-Premises	Secure your software development with automated secrets detection & remediation for private or public source code.
GitHub Advanced Security	GitHub	Open Source or Free	SaaS or On-Premises	GitHub Advanced Security uses CodeQL for Static Code Analysis, and GitHub Secret Scanning for identifying tokens. GitHub code scanning can import SARIF from any other SAST tool
GitLab	GitLab	Commercial	SaaS, Linux, Windows	
GolangCI-Lint		Open Source or Free		A Go Linters aggregator - One of the Linters is [gosec (Go Security)](https://github.com/securego/gosec), which is off by default but can easily be enabled.
Google CodeSearchDiggity		Open Source or Free		Uses Google Code Search to identify vulnerabilities in open-source code projects hosted by Google Code, MS CodePlex, SourceForge, Github, and more. The tool comes with over 130 default searches that identify SQL injection, cross-site scripting (XSS), insecure remote and local file includes, hard-coded passwords, and much more. *Essentially, Google CodeSearchDiggity provides a source code security analysis of nearly every single open-source code project in existence – simultaneously.*
Graudit		Open Source or Free	Linux	Scans multiple languages for various security flaws. Basically security enhanced code Grep.
HCL AppScan CodeSweep - GitHub Action	HCL Software	Open Source or Free		Scan the new code on a push/pull request using a GitHub action. Findings are highlighted in the `Files Changed` view and details about the issue and mitigation steps can be found in the `Actions` page. Unrestricted usage allowed with a free trial account. The tool currently supports Python, Ruby, JS (Vue, React, Node, Angular, JQuery, etc), PHP, Perl, COBOL, APEX & a few more.
HCL AppScan CodeSweep - IDE	HCL Software	Open Source or Free		This is the first Community edition version of AppScan. It is delivered as a VS Code [https://hclsw.co/codesweep] and JetBrains [https://hclsw.co/codesweep-jetbrains] (IntelliJ I, CLion, GoLand, PhpStorm, PyCharm , Rider, RubyMine, WebStorm) plugin and scans files upon saving them. The results show the location of a finding, type and remediation advice. The tool currently supports Java, .Net, Go, Python, Ruby, JS (Node, Angular, JQuery, etc) , PHP, Perl, COBOL, APEX & a few more. Auto-fix for some of the issues is available with a free trial.
HCL AppScan on Cloud	HCL Software	Open Source or Free		Apex, ASP, C, C++, COBOL, ColdFusion, Go, Java, JavaScript(Client-side JavaScript, Kotlin, NodeJS, and AngularJS), .NET (C#, ASP.NET, VB.NET), .NET Core, Perl, PHP, PL/SQL, Python, Ruby, T-SQL, Swift, Visual Basic 6
HCL AppScan Source	HCL Software	Commercial		Android, Apex, ASP, C, C++, COBOL, ColdFusion, Go, Java, JavaScript(Client-side JavaScript, NodeJS, and

Name	Owner	License	Platforms	Note
				AngularJS), .NET (C#, ASP.NET, VB.NET), .NET Core, Perl, PHP, PL/SQL, Python, Ruby, T-SQL, Visual Basic 6
Hdiv Detection	Hdiv Security	Commercial		Hdiv performs code security without doing static analysis. Hdiv does Interactive Application Security Testing (IAST), correlating runtime code & data analysis. It provides code-level results without relying on static analysis.
Horusec		Open Source or Free		C#, Java, Kotlin, Python, Ruby, Golang, Terraform, Javascript, Typescript, Kubernetes, PHP, C, HTML, JSON, Dart, Elixir, Shell, Nginx, Swift
HuskyCI		Open Source or Free		HuskyCI is an open-source tool that orchestrates security tests inside CI pipelines of multiple projects and centralizes all results into a database for further analysis and metrics. HuskyCI can perform static security analysis in Python (Bandit and Safety), Ruby (Brakeman), JavaScript (Npm Audit and Yarn Audit), Golang (Gosec), and Java(SpotBugs plus Find Sec Bugs)
Insider CLI	InsiderSec	Open Source or Free		An open-source Static Application Security Testing tool (SAST) written in GoLang for Java Maven and Android), Kotlin (Android), Swift (iOS), .NET Full Framework, C#, and Javascript (Node.js).
Kiuwan	a division of Idera, Inc.	Commercial		provides an application security testing and analytics platform – including SAST and SCA solutions – that reduces risk and improves change management and DevOps processes
Klocwork	Perforce	Commercial		Static Code Analysis for C, C++, C#, Java, JavaScript, Python, Kotlin
Kroogal		Commercial		C, C++
L3X	VulnPlanet	Open Source or Free	GitHub	L3X detects vulnerabilities in Rust and Solidity code based on patterns and AI code analysis. Various LLMs act as validators for vulnerabilities detected by patterns and validate each other's results in AI code analysis. Vulnerabilities are confirmed when they receive confirmation from a majority of validators.
Lucent Sky AVM	Lucent Sky	Commercial	SaaS or On-Premises	Automatically finds and fixes application vulnerabilities in source code. Supports .NET, ASP, Android, C and C++, ECMAScript, Go, iOS, Java, PHP, Python, Ruby, and Visual Basic applications.
LunaTrace by LunaSec	LunaSec	Open Source or Free	SaaS or On-Premises	Software Composition Analysis (SwCA) tool to generate SBOMs, identify vulnerabilities in dependencies, and generate patches. Leverages Static Analysis to reduce false positives by filtering non-exploitable CVEs.
Mend SAST	Mend	Commercial		Static security analysis for 27+ languages.
Microsoft FxCop		Open Source or Free		.NET
Microsoft PREFast		Open Source or Free		C, C++
MobSF		Open Source or Free		Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment

Name	Owner	License	Platforms	Note
				framework capable of performing static and dynamic analysis.
MobSF		Open Source or Free	Windows, Unix	Android Java, Objective C, Swift
NaiveSystems Analyze	Naive Systems Ltd.	Open Source or Free	Windows, MacOS and Linux	NaiveSystems Analyze ensures compliance with functional safety and coding standards including MISRA, AUTOSAR, and Google C++ Style Guide. Supports C, C++, Java, and more languages.
NextGen Static Analysis	ShiftLeft	Commercial	SaaS	Free version available. Currently supports Java, JavaScript, C\#, TypeScript, Python, and Terraform. Create your free account at https://shiftleft.io/register.
nodejsscan		Open Source or Free	Unix	Node.js
Nucleaus Core	Nucleaus	Commercial	SaaS	Scans Git repos daily and provides a web-based dashboard to track code and dependency vulnerabilities. Handles team-based access patterns, vulnerability exception lifecycle, and is built on API first principles.
Offensive360		Commercial		SAST technology that attacks the source code from all corners it has all in one. Malware, SCA, License, and deep source code analysis.
Oversecured	Oversecured Inc	Commercial	iOS, Android	Enterprise vulnerability scanner for Android and iOS apps. It offers app owners and developers the ability to secure each new version of a mobile app by integrating Oversecured into the development process.
OWASP ASST (Automated Software Security Toolkit)	Tarik Seyceri & OWASP	Open Source or Free	Ubuntu, MacOSX and Windows	An Open Source, Source Code Scanning Tool, developed with JavaScript (Node.js framework), Scans for PHP & MySQL Security Vulnerabilities According to OWASP Top 10 and some other OWASP's famous vulnerabilities, and it teaches developers of how to secure their codes after scan.
OWASP Code Crawler	OWASP	Open Source		.NET, Java
OWASP LAPSE Project	OWASP	Open Source		Java
OWASP Orizon Project	OWASP	Open Source		Java
OWASP WAP (Web Application Protection)	OWASP	Open Source		PHP
ParaSoft		Open Source or Free		C, C++, Java, .NET
Parasoft Test	Parasoft	Commercial		Test tools for C/C++, .NET, Java
phpcs-security-audit		Open Source or Free		A set of PHP_CodeSniffer rules to finds flaws or weaknesses related to security in PHP and its popular CMS or frameworks. It currently has core PHP rules as well as Drupal 7 specific rules.

Name	Owner	License	Platforms	Note
PITSS.CON	PITSS	Commercial		Scans Oracle Forms and Reports Applications
Pixeebot	Pixee	Commercial or Free	GitHub	Pixeebot finds security and code quality issues in your code and creates merge-ready pull requests with recommended fixes.
PMD		Open Source or Free		PMD scans Java source code and looks for potential code problems (this is a code quality tool that does not focus on security issues).
Polyspace Static Analysis Tools		Commercial		C, C++, Ada
PreFast	Microsoft	Open Source or Free		PREfast is a static analysis tool that identifies defects in C/C++ programs. Last update 2006.
Progpilot		Open Source or Free		Progpilot is a static analyzer tool for PHP that detects security vulnerabilities such as XSS and SQL Injection.
Psalm	Vimeo, Inc.	Open Source		Static code analysis for PHP projects, written in PHP.
PT Application Inspector	Positive Technologies	Commercial		Combines SAST, DAST, IAST, SCA, configuration analysis and other technologies for high accuracy. Can generate special test queries (exploits) to verify detected vulnerabilities during SAST analysis. Supports Java, C\#, PHP, JavaScript, Objective C, VB.Net, PL/SQL, T-SQL, and others.
Puma Scan	Puma Security	Commercial		A .NET C\# static source code analyzer that runs as a Visual Studio IDE extension, Azure DevOps extension, and Command Line (CLI) executable.
Puma Scan Professional		Open Source or Free		.NET, C\#
PVS-Studio		Open Source or Free		C, C++, C\#
PVS-Studio Analyzer	PVS-Studio	Commercial		Static code security analysis for C, C++, C#, and Java. A commercial B2B solution but provides several free [licensing options](https://www.viva64.com/en/b/0614/).
Pyre		Open Source or Free		A performant type-checker for Python 3, that also has [limited security/data flow analysis](https://pyre-check.org/docs/pysa-basics.html) capabilities.
reshift		Commercial		A CI/CD static code security analysis tool for Java that uses machine learning to give a prediction on false positives.
SecureAssist	Synopsys	Commercial		Scans code for insecure coding and configurations automatically as an IDE plugin for Eclipse, IntelliJ, and Visual Studio, etc. Supports Java, .NET, PHP, and JavaScript.
Security Code Scan		Open Source or Free		Static code analyzer for .NET. It will find SQL injections, LDAP injections, XXE, cryptography weakness, XSS and more.

Name	Owner	License	Platforms	Note
Seeker	Synopsys	Commercial		Seeker performs code security without doing static analysis. Seeker does Interactive Application Security Testing (IAST), correlating runtime code & data analysis with simulated attacks. It provides code level results without relying on static analysis.
Semgrep		Open Source or Free		Semgrep is a fast, open-source, static analysis engine for finding bugs, detecting vulnerabilities in third-party dependencies, and enforcing code standards. Semgrep analyzes code locally on your computer or in your build environment: code is never uploaded.
Semgrep Supply Chain		Commercial		Semgrep Supply Chain's reachability analysis lets you quickly find and remediate the 2% of dependency vulnerabilities that are reachable.
Sentinel Source	Whitehat	Commercial		Static security analysis for 10+ languages.
ShiftLeft Scan		Open Source or Free		A free open-source DevSecOps platform for detecting security issues in source code and dependencies. It supports a broad range of languages and CI/CD pipelines by bundling various open-source scanners into the pipeline.
Snyk Cloud	Snyk Limited	Commercial or Free	SaaS, IDE Plugin	Detects cloud security issues as soon as developers start designing configurations, providing expert guidance to cloud, platform, and security teams in the tools and workflows they use every day.
Snyk Code	Snyk Limited	Commercial or Free	SaaS, IDE Plugin	AI-powered code checker that analyzes your code for security issues, providing actionable advice directly from your IDE to help you fix vulnerabilities quickly

Table 18: Source Code Analysis Tools [RD-39]

Critical software

We are CMMI Maturity Level 5 rated.
For a list of our certifications & standards visit our website.

www.criticalsoftware.com