Software Security for Maritime

CSP009_SA_M

PRESENTATION BY:
PINELOPI KYRANOUDI, TECHNICAL UNIVERSITY OF CRETE
SPIROS BOROTIS, MAGGIOLI SPA

CyberSecPro

CyberSecPro creates cutting-edge education and training materials and courses to advance competencies and professionalism in EU cybersecurity.

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

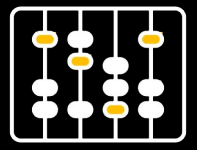
Project Agreement no. 101083594

Goals: Who-What-Why you need to take this training

WHO

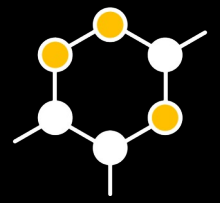
Professionals involved in maritime software development, cybersecurity, and system integration, including engineers, technical staff, and decision-makers

WHAT



Advanced seminar on maritime software security covering secure development, threat modelling, and testing across the software lifecycle

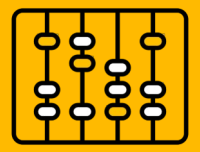
WHY



Providing participants with the knowledge and skills to understand maritime software security principles, practices, and methodologies.

CSP Training Logistic: When-Where-How

WHEN



Time schedule: To be posted in DCM platform

WHERE

ONLINE-ONSITE-Location Information

HOW

Delivery Method

Value Propositions

Benefits to Participants

- Level of Training Module: Advance
- Cybersecurity Professional Training
- Rooted with European Cybersecurity Skills Framework
- Cutting-edge insights from industry-academic experts
- Certificate of the completion
- Helps with skills development and career advancement

WHO

Profile of Training Participants

- Managers and Leaders
- Working-life professionals
- SMEs and Public Sector Employees
- Cybersecurity practitioners and enthusiasts

WHO

Profile of Trainers

- Pinelopi Kyranoudi has obtained her master's degree in Security of Information and Communication Systems from the University of the Aegean (Dept. of Information and Communication Systems Engineering). She served as Network and Information Security Officer at the European Union Agency for Cybersecurity (ENISA) contributing: to five publications in the areas of Cybersecurity in Maritime, e-Health, and National Cybersecurity Strategies; to the creation of web tools, and to the organization of EU Cybersecurity events. During her 10-year career she also worked in various positions and roles, such as Web and Application Developer, IT Security Engineer, Cybersecurity Researcher, and Lecturer in companies and organizations such as Express Publishing SA, Cosmote SA, Maggioli SpA, and Aegean College respectively, among others. She is currently conducting her Ph.D studies in the Cybersecurity field at the University of Piraeus (Dept. of Informatics). She holds a position of Cybersecurity Researcher at the Technical University of Crete (School of Electrical and Computer Engineering), contributing to EU research projects. Her research interests are in the field of Cyber-Physical Security, particularly in Maritime, OT/IoT, and Threat Intelligence.

- Dr. Spiros Borotis holds a Bachelors Degree in Mathematics, an MSc in Information Systems and a PhD in e-Learning. He works more than 20 years in the areas of research and consultancy on learning, innovation and human resources development in national and European level. He has collaborated with various European and national organizations as a senior research and project manager, organizing and running the research activities of the organizations working for, leading and contributing in parallel to the preparation and submission of various proposals and tenders for national and European funded / co-funded projects in the aforementioned areas of interest. In parallel, he has worked as a policy consultant in the areas of education and employment, orienting to the development of policies and modernization of educational strategies in all levels of formal education and the workplace learning. Currently, he works as a Senior Project Manager / Analyst for the Maggioli R&D, coordinating and implement projects funded under the Horizon Europe Programme in the areas of Artificial Intelligence, Extender Reality, CyberSecurity and Advanced Skills Development.

WHAT

Training Topics

- Introduction to Software Security in the Maritime Industry
- Principles of secure software development lifecycle (SDLC) tailored for the maritime industry
- Understanding the process of threat modelling for identifying and mitigating maritime software security risks
- Overview of software security testing methodologies, including static and dynamic analysis
- Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector.

WHY

Learning Outcomes

Knowledge:

- Understanding of software security concepts and their relevance to the maritime industry.
- Familiarity with common software vulnerabilities and threats specific to maritime systems.
- Knowledge of secure software development frameworks, standards, and guidelines applicable to maritime software development.
- Understanding of threat modelling techniques and their application in identifying and mitigating software security risks in maritime environments.
- Awareness of software security testing methodologies, including static and dynamic analysis, and their application to maritime software systems.
- Overview of emerging technologies such as IoT, AI, and blockchain and their impact on maritime software security.

Training Outline

Topic-1: Introduction to Software Security in the Maritime Industry

Overview of software security concepts and their importance in the maritime sector

Introduction to common software vulnerabilities and threats specific to maritime systems and applications

Understanding the role of software security in ensuring the safety and integrity of maritime operations - real-world scenarios and incidents

Topic-2: Principles of secure software development lifecycle (SDLC) tailored for the maritime industry

Introduction to secure software development frameworks, standards, guidelines, and practices relevant to maritime software development

SDLC models, challenges and best practices

Best practices for secure coding, including input validation, secure authentication, and error handling

Training Outline

Topic-3: Understanding the process of threat modelling for identifying and mitigating maritime software security risks

Application of threat modelling techniques to maritime software systems and applications

Case studies and examples of threat modelling in the maritime sector

Secure Deployment and Configuration Management

Topic-4: Overview of software security testing methodologies, including static and dynamic analysis

Types of tests (e.g., penetration testing, DAST, MAST)

~~Application of security testing techniques to maritime software systems~~

~~Hands-on exercises and simulations to reinforce software security concepts and techniques~~

Training Outline

Topic-5: Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector

Group discussions to explore software security challenges and solutions in maritime scenarios

Discussion on the impact of IoT, AI, and blockchain on maritime software security

Consideration of future challenges and opportunities for enhancing software security in maritime operations

Background Knowledge and Prerequisites

Background knowledge:

Basic understanding of software development and networking

Familiarity with common software vulnerabilities and security concepts

General awareness of cybersecurity principles

Prerequisites:

Basic knowledge of software development and cybersecurity concepts

Resources:
Books and Reference Material

1. OWASP Foundation, OWASP Top 10: The Ten Most Critical Web Application Security Risks, 2021
2. OWASP Foundation, OWASP API Security Top 10, 2019
3. OWASP Foundation, OWASP Secure Coding Practices – Quick Reference Guide, v2.0
4. OWASP Foundation, OWASP Application Security Verification Standard (ASVS), v4.0
5. OWASP Foundation, OWASP Web Security Testing Guide, v4.2
6. NIST, Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1, 2018
7. NIST, Risk Management Framework (RMF), SP 800-37 Rev. 2
8. NIST, Guide to Industrial Control Systems (ICS) Security, SP 800-82 Rev. 2
9. NIST, Secure Software Development Framework (SSDF), SP 800-218
10. ENISA, ENISA Threat Landscape, 2024
11. MITRE, ATT&CK Framework
12. Center for Internet Security (CIS), CIS Critical Security Controls, Version 8
13. ISO/IEC 27001, Information Security Management Systems — Requirements, 2022
14. ISO/IEC 27002, Information Security Controls, 2022
15. IEC 62443, Industrial Communication Networks – Network and System Security
16. SANS Institute, Secure Software Development and Cybersecurity Resources
17. McGraw, G., Software Security: Building Security In, Addison-Wesley
18. Viega, J., and McGraw, G., Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley
19. Howard, M., and LeBlanc, D., Writing Secure Code, Microsoft Press
20. Bishop, M., Computer Security: Art and Science, Addison-Wesley
21. Stallings, W., Effective Cybersecurity: A Guide to Using Best Practices and Standards, Addison-Wesley

Course progress

01. Introduction to Software Security in the Maritime Industry

02. Principles of secure software development lifecycle (SDLC) tailored for the maritime industry

03. Understanding the process of threat modelling for identifying and mitigating maritime software security risks

04. Overview of software security testing methodologies, including static and dynamic analysis

05. Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector

Introduction to Software Security in the Maritime Industry

Understanding Software Security

Software security encompasses the strategies, practices, and technologies designed to protect software applications from external threats and internal vulnerabilities. It is an integral part of the cybersecurity framework that ensures the confidentiality, integrity, and availability of software systems.

Why It Matters for the Maritime Industry

The maritime industry, with its increasing reliance on digital technologies for navigation, communication, and operational management, faces unique cybersecurity challenges. Software security in this sector is crucial for safeguarding against attacks that could disrupt maritime operations, lead to environmental hazards, or compromise sensitive data.

Scope of This Presentation

We'll explore the core concepts of software security, delve into common vulnerabilities and threats specific to maritime systems, and discuss the role of software security in ensuring the safety and integrity of maritime operations. Through real-world scenarios and incidents, we'll highlight the importance of proactive and comprehensive software security measures.

The Significance of Software Security in Maritime Operations

Crucial Definition and Objectives

Software security aims to ensure that software systems are resilient against unauthorized interference, maintaining the integrity, confidentiality, and availability of the system's data and functionalities. This goal is paramount in maritime systems, where the stakes for data integrity and system availability are exceptionally high.

Maritime Software Security Landscape

The maritime sector increasingly relies on interconnected software systems for navigation, cargo management, and communication. These systems are prime targets for cyber threats that could disrupt global trade routes and compromise sensitive information.

The Significance of Software Security in Maritime Operations

Security Objectives for Maritime Systems

Ensuring that navigational data remains confidential and accessible only to authorized entities, maintaining the integrity of operational commands, and guaranteeing system availability even under cyber attack conditions are fundamental security objectives for maritime software systems.

Balancing Security with Operational Needs

In the maritime domain, the challenge is to secure software systems while maintaining operational efficiency. This involves a delicate balance between locking down systems for security and ensuring they are available and performant for essential maritime operations.

The Pivotal Role of Software Security

Beyond protecting from external threats, software security in maritime operations also encompasses safeguarding against internal vulnerabilities, such as software bugs or misconfigurations, that could be exploited by attackers.

The Importance of Software Security

Vulnerability Categories

Key vulnerability categories relevant to maritime software include Memory Management Vulnerabilities, API Vulnerabilities, and Race Condition Vulnerabilities. These vulnerabilities pose significant risks to maritime operations, potentially leading to unauthorized access, system failures, or compromised data integrity.

Preventing Vulnerabilities

The maritime industry must prioritize preventive measures, leveraging language design and type systems that inherently reduce the risk of introducing vulnerabilities. For example, memory-safe languages help prevent a vast array of memory management vulnerabilities, which are common entry points for attackers.

Detection and Mitigation

Continuous detection efforts through static and dynamic analysis help identify vulnerabilities within maritime software systems. Meanwhile, mitigation strategies like Runtime Detection of Attacks and Automated Software Diversity ensure that even if vulnerabilities exist, their exploitation is significantly hindered, protecting maritime operations from potential cyber threats.

The Importance of Software Security

Securing the Maritime Industry

Implementing rigorous software security practices is not just about protecting data but ensuring the safety and reliability of maritime navigation and operations systems. From cargo management systems to navigation and communication networks, every aspect of maritime operations relies on secure and robust software systems.

Call to Action

Maritime industry stakeholders, including developers, security professionals, and operational managers, must collaborate to embed security deeply within the software development lifecycle. Adopting secure coding practices, conducting regular vulnerability assessments, and embracing continuous security training are pivotal steps toward fortifying maritime software against evolving cybersecurity threats.

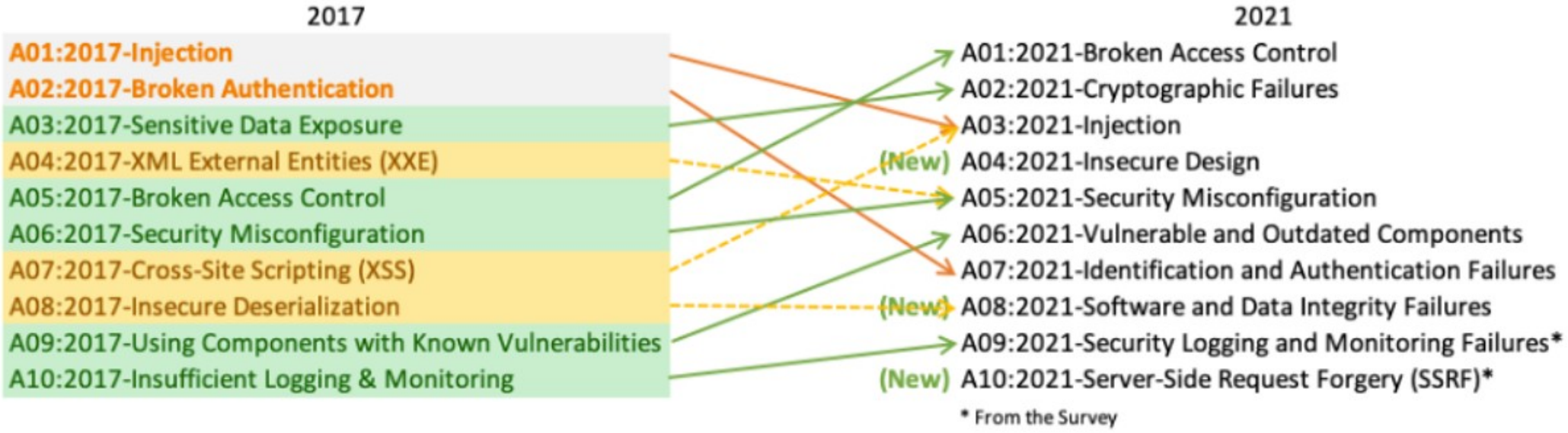
Top 10 Web Application Security Risks

Overview

- **Injection:** Flaws like SQL, NoSQL, OS, and LDAP injection
- **Broken Authentication:** Poorly implemented authentication and session management
- **Sensitive Data Exposure:** Failure to properly protect sensitive data
- **XML External Entities (XXE):** External entities in XML document
- **Broken Access Control:** Improperly enforced restrictions on authenticated users
- **Security Misconfiguration:** Insecure default configurations
- **Cross-Site Scripting (XSS):** Injection of malicious scripts into web pages
- **Insecure Deserialization:** Risks from deserializing untrusted data
- **Using Components with Known Vulnerabilities:** Use of vulnerable libraries or frameworks
- **Insufficient Logging & Monitoring:** Lack of proper logging and monitoring

Top 10 Web Application Security Risks

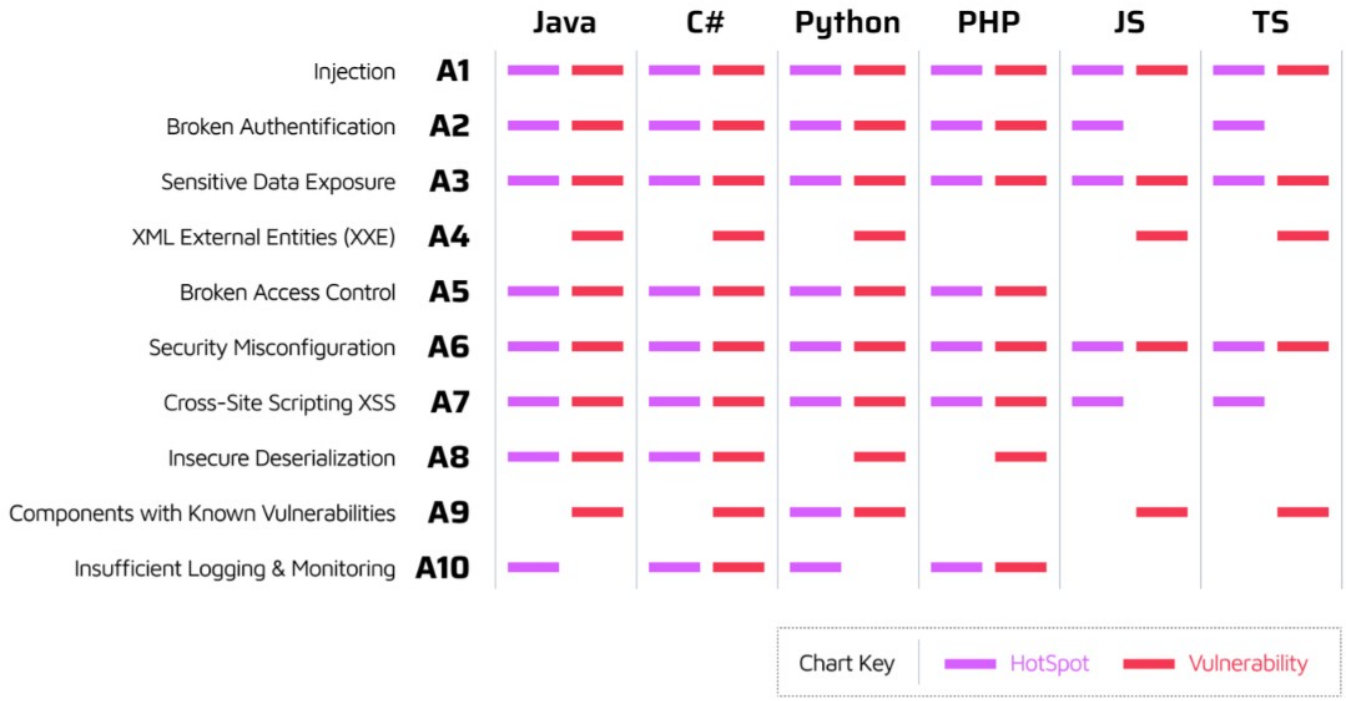
There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



https://owasp.org/www-project-top-ten/

Top 10 Web Application Security Risks

By informing about security vulnerability issues to developers early in the process, it helps companies protect their systems, data and users.



https://www.sonarsource.com/solutions/security/owasp

Maritime Industry Context

Crucial Role of Maritime Trade

The maritime industry is the backbone of global commerce, responsible for transporting over 80% of the world's goods by volume. This vast network of ships, ports, and logistics systems connects markets, fuels economies, and supports millions of jobs worldwide.

Unprecedented Cybersecurity Challenges

The maritime sector's growing digitalization introduces complex cybersecurity challenges. Critical systems that control navigation, communication, and cargo management are increasingly targeted by cyber threats, posing risks to operational safety and global supply chains.

Navigation Systems Under Threat

GPS spoofing and other cyber-attacks can lead to severe navigational errors, endangering vessels and their cargoes. The integrity of these systems is paramount for the safe passage of ships across global trade routes.

Maritime Industry Context

Communication Systems at Risk

Secure and reliable communication channels are essential for the coordination of maritime operations. Cyber threats compromising these systems can disrupt maritime logistics and emergency responses, leading to significant operational setbacks.

Cargo Management Systems Vulnerabilities

The digital systems managing cargo operations are prime targets for cyber-attacks. Breaches in these systems can result in cargo theft, loss, or misrouting, impacting the global supply chain's efficiency and security.

The High Stakes of Cybersecurity in Maritime Operations

A successful cyber-attack in the maritime domain can have far-reaching consequences, affecting not only the targeted company but also the global economy, environment, and public safety. The imperative to secure maritime operations against cyber threats has never been more critical.

Common Software Vulnerabilities in Maritime Systems

Memory Management Vulnerabilities

In maritime software systems, memory management vulnerabilities arise primarily due to improper handling of memory allocation, access, and deallocation by programming languages like C and C++. These vulnerabilities, such as buffer overflows (spatial vulnerabilities) and use-after-free errors (temporal vulnerabilities), can lead to unauthorized access or denial of service, jeopardizing the safety and security of maritime operations.

The inherent risk is exacerbated in maritime contexts, where such vulnerabilities can allow attackers to corrupt navigational data, manipulate cargo management systems, or gain control over communication systems.

API Vulnerabilities

APIs are integral to maritime software systems, facilitating communication between different software components, including navigation, cargo tracking, and communication systems. Vulnerabilities in API design or implementation can expose sensitive operational data or allow attackers to manipulate these systems.

For example, insecure API calls in a maritime tracking system could allow attackers to access or alter sensitive information about cargo shipments, leading to financial losses, safety risks, or environmental hazards.

Common Software Vulnerabilities in Maritime Systems

Preventive Measures

Addressing memory management vulnerabilities requires a combination of secure coding practices, such as using memory-safe languages or adopting libraries that provide safer memory management functions. For maritime systems, this also means rigorous testing and validation of software components that handle critical operations.

Mitigating API vulnerabilities involves designing APIs with security in mind, implementing proper authentication, authorization, and data validation mechanisms. In the maritime industry, ensuring the security of APIs that handle navigational and operational data is paramount for maintaining the integrity of maritime operations.

Conclusion

Recognizing and addressing common software vulnerabilities is crucial in protecting maritime systems against cyber threats. By implementing robust security measures and adopting secure coding practices, the maritime industry can enhance the resilience of its operations against potential cyber-attacks.

Common Software Vulnerabilities in Maritime Systems

CVE-2023-45225 Detail

Description

Zavio CF7500, CF7300, CF7201, CF7501, CB3211, CB3212, CB5220, CB6231, B8520, B8220, and CD321 IP Cameras with firmware version M2.1.6.05 are vulnerable to multiple instances of stack-based overflows. While parsing certain XML elements from incoming network requests, the product does not sufficiently check or validate allocated buffer size. This may lead to remote code execution.

Relating CVE-2023-45225 to the Maritime Industry

Surveillance and Security Applications

Maritime operations, including ports, ships, and offshore platforms, extensively use IP cameras for surveillance and security monitoring. These cameras are pivotal for ensuring the safety and security of maritime facilities, monitoring cargo loading/unloading, overseeing ship movements, and safeguarding restricted areas.

Impact of Vulnerability

Unauthorized Access: Exploiting this vulnerability could allow attackers to gain control over IP cameras, possibly leading to unauthorized surveillance or access to sensitive areas within maritime operations.

Data Integrity and Privacy: Remote code execution could compromise the integrity of video feeds, allowing for tampering with or falsification of surveillance footage, which is critical for security audits and incident investigations.

Threats Specific to Maritime Software Applications

GPS Spoofing Threats to Navigation

GPS spoofing represents a critical threat to maritime navigation systems, wherein attackers manipulate GPS signals to provide false positioning information. This can mislead vessels off their intended course, potentially leading to dangerous encounters, groundings, or violations of territorial waters. The integrity of GPS data is foundational for safe maritime navigation, highlighting the need for enhanced security measures to detect and mitigate spoofing attempts.

Malware in Cargo Tracking Systems

Cargo tracking systems are essential for monitoring the location and status of goods as they move through the global supply chain. However, these systems are vulnerable to malware attacks, which can disrupt operations by altering or deleting cargo data. Such breaches not only jeopardize cargo security but also can cause significant logistical chaos, financial losses, and undermine trust in maritime transport services.

Threats Specific to Maritime Software Applications

Impact of Cyber Threats on Maritime Operations

The threats facing maritime software applications are not limited to data breaches but extend to the potential compromise of vessel safety, cargo security, and environmental protection. Cyberattacks like GPS spoofing and malware in cargo systems can have dire consequences, including collisions, environmental disasters, and loss of life.

Beyond immediate operational impacts, these threats also pose long-term challenges to the maritime industry, such as increased insurance costs, regulatory scrutiny, and the need for substantial investments in cybersecurity measures.

Strengthening Defenses Against Maritime Cyber Threats

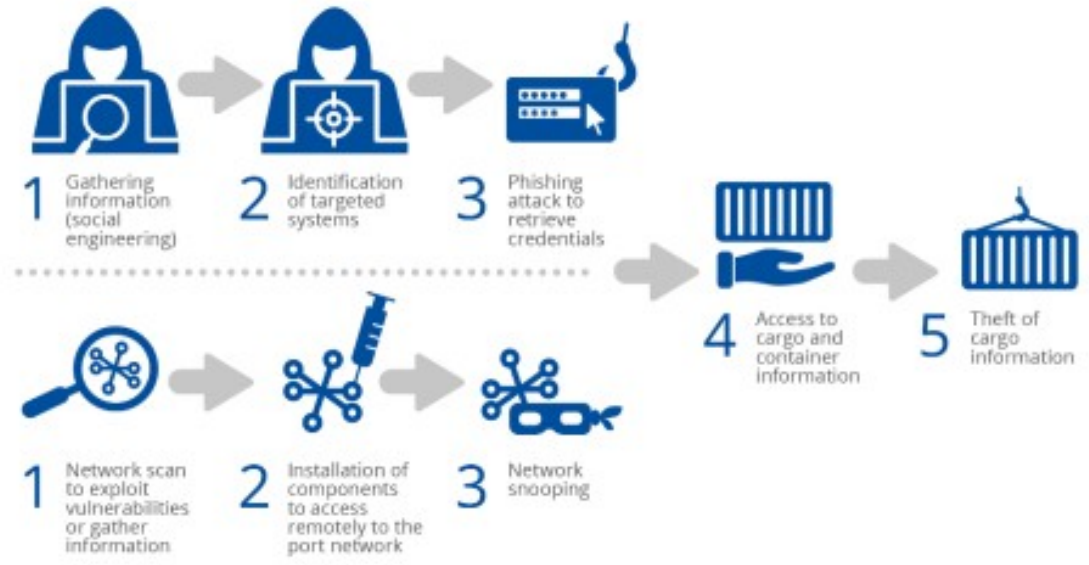
Protecting maritime software applications from these specific threats requires a multi-layered security approach. This includes the deployment of intrusion detection systems, implementation of encryption and secure communication protocols, regular software updates and patches, and comprehensive cybersecurity training for maritime personnel.

Example of Attack Scenario

Scenario A – Compromising of critical data to steal high value cargo or allow illegal trafficking through a targeted attack

This scenario is a sophisticated and targeted attack on port systems (Advanced Persistent Threat): attackers must have in-depth knowledge of port systems and networks (social engineering, network scan), port processes and port infrastructure (physical intrusion) to perform cargo and container theft. An example of such an attack occurred at one of Antwerp's port terminals in Belgium[46].

Figure 7: Compromising of critical data to steal high value cargo or allow illegal trafficking scenario



Impacts
Cargo and goods stealing
Illegal trafficking
Tarnished reputation

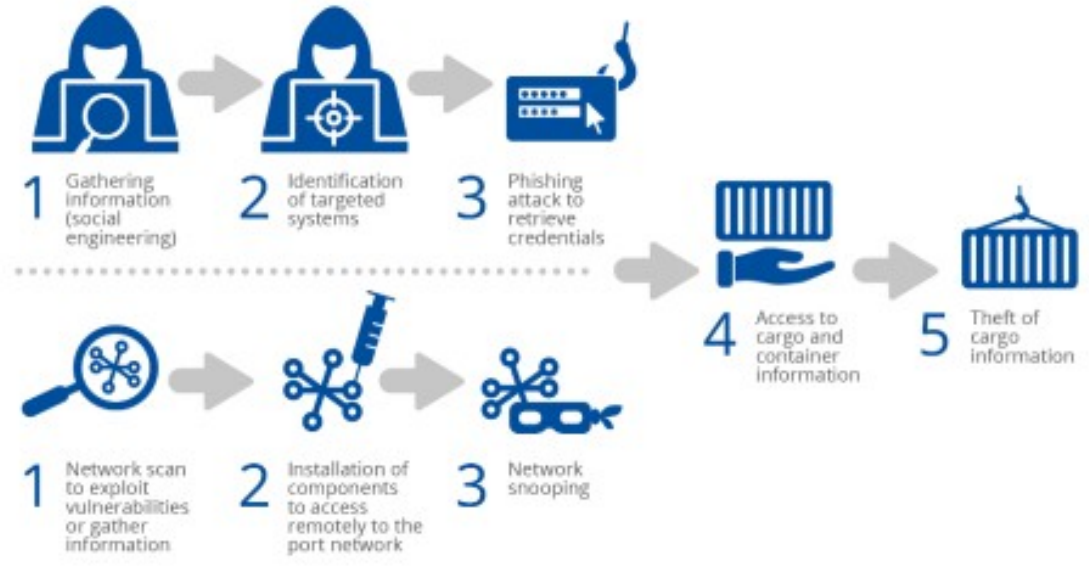
Assets Affected	Stakeholders Involved
Cargo Community Systems (CCS)	Terminal Operators
Networks	Police
Email	Shipping and maritime freight companies
People	Senders and consignees

Example of Attack Scenario

Scenario A – Compromising of critical data to steal high value cargo or allow illegal trafficking through a targeted attack

This scenario is a sophisticated and targeted attack on port systems (Advanced Persistent Threat): attackers must have in-depth knowledge of port systems and networks (social engineering, network scan), port processes and port infrastructure (physical intrusion) to perform cargo and container theft. An example of such an attack occurred at one of Antwerp's port terminals in Belgium[46].

Figure 7: Compromising of critical data to steal high value cargo or allow illegal trafficking scenario



Attack details

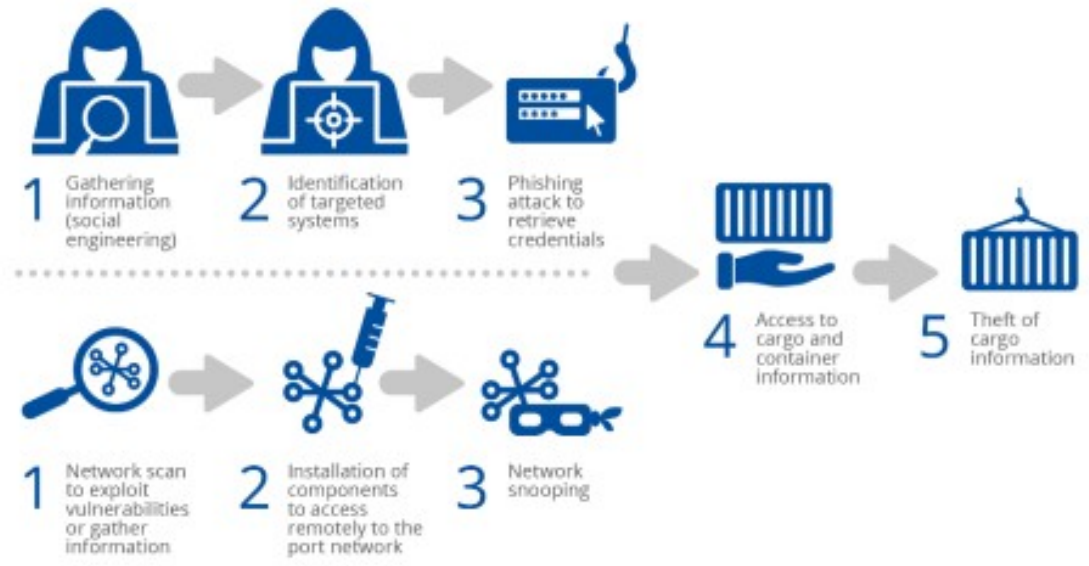
- On the one hand, attackers identify and retrieve authentication data (credentials) to get access to useful systems:
- Attackers gather information on port systems through social engineering o Then they identify the targeted systems used for cargo and container management and identity of the people using them
- Once systems and their operators/users are identified, attackers launch phishing attacks to retrieve credentials to access to those systems
- On the other hand, attackers install components to gain access remotely to the port network and bypass network security:
- Attackers scan the port networks to find vulnerabilities to exploit them and gather information
- They install, if necessary, through physical intrusion, components to access remotely to the port networks (wireless access point)
- To ensure constant access and adapt to each network and infrastructure changes in the long term, they spy on networks
- Attackers have now access to the freight tracking systems and other relevant port systems and they can access critical information on containers they want to steal (localisation, content, pick-up code, etc.) from outside of the port facilities.
- Attackers can then steal the cargo before the official pickup date

Example of Attack Scenario

Scenario A – Compromising of critical data to steal high value cargo or allow illegal trafficking through a targeted attack

This scenario is a sophisticated and targeted attack on port systems (Advanced Persistent Threat): attackers must have in-depth knowledge of port systems and networks (social engineering, network scan), port processes and port infrastructure (physical intrusion) to perform cargo and container theft. An example of such an attack occurred at one of Antwerp's port terminals in Belgium[46].

Figure 7: Compromising of critical data to steal high value cargo or allow illegal trafficking scenario



Main Security Measures
OP-10: Security awareness raising program
OP-16: Creation of a Cybersecurity Operations Centre (SOC)
TP-01: Network segmentation
TP-02: Regular network scans
TP-08: Multi-factor authentication

Preventing Vulnerabilities in Maritime Systems

The Foundation of Secure Maritime Software

At the heart of preventing vulnerabilities in maritime systems lies a commitment to secure coding practices. This involves a proactive approach to software development, where security considerations are integrated from the outset and throughout the development lifecycle.

Regular Code Reviews and Updated Coding Standards

Implementing regular code reviews is a critical practice for identifying and rectifying potential security flaws before they become vulnerabilities. Coupled with adherence to updated coding standards that incorporate the latest security practices, maritime software developers can significantly reduce the risk of vulnerabilities.

For maritime applications, where software reliability can mean the difference between safe passage and disaster, such practices ensure that systems are robust against both accidental errors and malicious attacks.

Preventing Vulnerabilities in Maritime Systems

Secure API Design for Maritime Systems

APIs serve as the connective tissue between different maritime software systems, enabling them to communicate and share data. Designing these APIs with security in mind is paramount. This includes:

- ***Authentication***: Ensuring that only authorized users and systems can access the API.
- ***Encryption:*** Protecting data in transit between systems to prevent interception or tampering.
- ***Access Control Measures:*** Defining and enforcing what data and system functionality can be accessed via the API, based on user or system privileges.

The Impact of Secure Practices

By adopting secure coding practices and secure API design, the maritime industry can protect its software systems from a wide array of vulnerabilities, from data breaches to operational disruptions. These preventative measures are not just about safeguarding data; they are about ensuring the continuity of maritime operations, protecting cargo, and preserving human life at sea.

Navigating Tomorrow: The Future of Maritime Software Security

Emerging Threats on the Horizon

As maritime operations become increasingly digitalized, the sector faces new cybersecurity challenges. Notably, AI-driven attacks present a sophisticated threat, with the potential to adapt and overcome traditional security measures. These attacks can analyze patterns and exploit vulnerabilities faster than ever before.

The push towards greater connectivity, while enhancing operational efficiency, also broadens the attack surface. This connectivity exposes maritime systems to new vulnerabilities, as each connected device or system can become a potential entry point for cyber attackers.

The AI Shield: Utilizing AI in Defence

In the face of AI-driven threats, the maritime industry has the opportunity to harness AI for its defense. AI and machine learning can be deployed to monitor systems in real-time, detect anomalies, and predict potential threats based on data analysis. This proactive approach to cybersecurity can significantly enhance the industry's ability to thwart sophisticated cyber attacks.

Navigating Tomorrow:
The Future of Maritime Software Security

Fostering International Collaboration

Cybersecurity in the maritime domain is a global concern, transcending individual companies or nations. International collaboration emerges as a key opportunity for strengthening maritime cybersecurity. Sharing insights, threat intelligence, and best practices among international stakeholders can lead to the development of robust, unified defence strategies against common threats.

Preparing for the Future

The future of maritime software security is not without its challenges, but it is also ripe with opportunities for innovation and collaboration. By embracing new technologies like AI for defence and committing to international cooperation, the maritime industry can safeguard itself against emerging threats and ensure the safety and security of its operations in the years to come.

Navigating Tomorrow:
The Future of Maritime Software Security

TOP 10 EMERGING SECURITY THREATS
Supply chain compromise of software dependencies
Advanced disinformation campaigns
Rise of digital surveillance authoritarianism/loss of privacy
Human error and exploited legacy systems within cyber-physical ecosystems
Targeted attacks enhanced by smart device data
Lack of analysis and control of space-based infrastructure and objects
Rise of advanced hybrid threats
Skills shortage
Cross-border ICT service providers as a single point of failure
Artificial intelligence abuse

Course progress

01. Introduction to Software Security in the Maritime Industry

02. Principles of secure software development lifecycle (SDLC) tailored for the maritime industry

03. Understanding the process of threat modelling for identifying and mitigating maritime software security risks

04. Overview of software security testing methodologies, including static and dynamic analysis

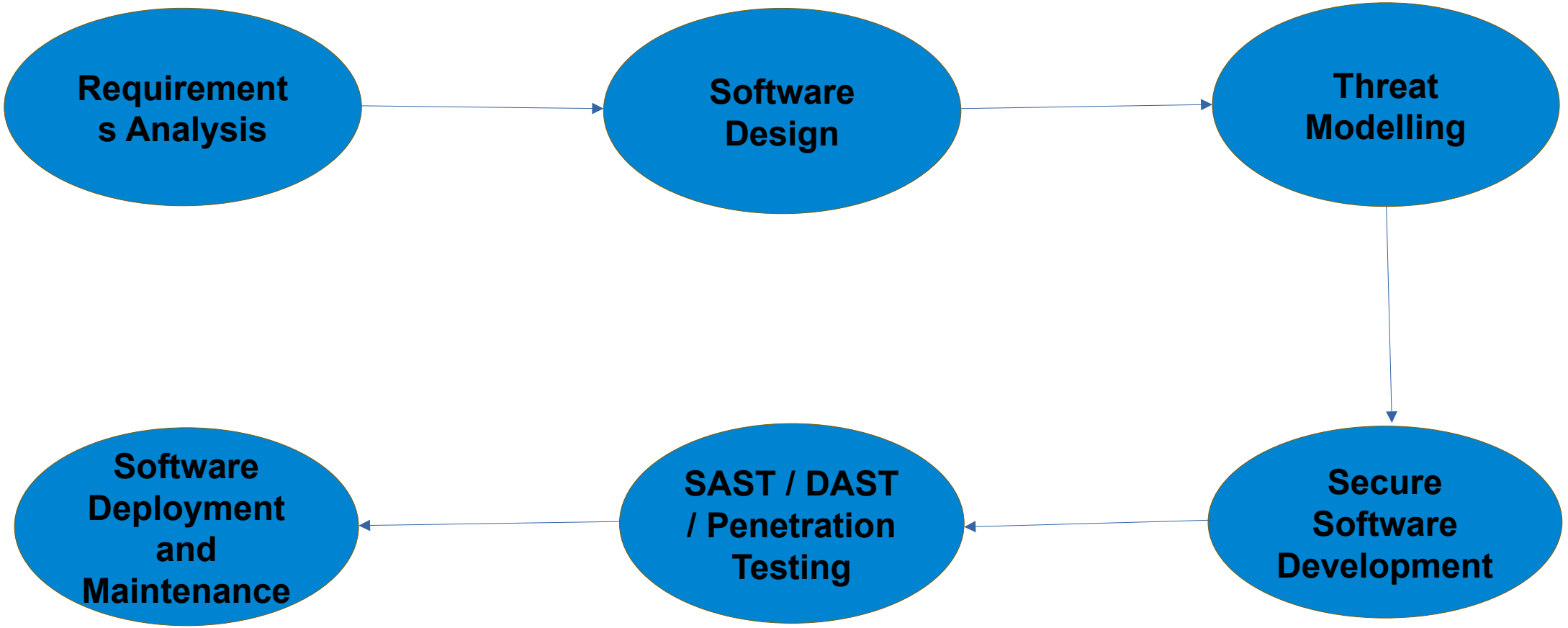
05. Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector

Secure Software Development

Essential for all industries

- High Costs of Security Failures
- Pioneering Secure Development Models
- Adopting Secure Software Lifecycle Processes
- Customization for Specific Industry Needs
- Evaluating Secure Development Practices

Software Development Life Cycle (SDLC)



Secure Software Development for the Maritime Industry

Essential for Maritime Security

In the maritime industry, securing the digital infrastructure through secure software development is paramount. The Secure Software Lifecycle domain underscores the critical need for security to be woven through every phase of software development, from conception to deployment. This holistic approach not only strengthens defences against cyber threats but is also integral to the digital progression of maritime operations.

High Costs of Security Failures

The substantial financial and reputational damages tied to security breaches are a stark reminder of the stakes involved. Emphasizing secure software development practices emerges as a prudent, cost-effective strategy for maritime organizations. This focus not only mitigates risks but also minimizes the potential for catastrophic losses, underscoring the value of security investments.

Pioneering Secure Development Models

The Microsoft Security Development Lifecycle (SDL) is heralded for its groundbreaking role in illustrating the efficacy of secure development practices. This model has significantly influenced the industry by demonstrating how such practices can drastically reduce software vulnerabilities, serving as a guiding light for secure development methodologies.

Secure Software Development for the Maritime Industry

Adopting Secure Software Lifecycle Processes

Emphasizing the importance of secure software lifecycle processes, these methodologies are vital for embedding security within product development. Models like Microsoft SDL, Touchpoints, and SAFECode are highlighted for their industry-wide applicability, tailored to suit the diverse technological needs within the maritime sector, from agile/DevOps environments to mobile, cloud computing, and IoT.

Customization for Maritime Needs

The adaptability of secure software lifecycle processes to the maritime industry's varied technological domains is a key focus. This flexibility ensures that such methodologies are not only relevant but highly effective across different platforms and technologies, catering specifically to the industry's unique needs.

Evaluating Secure Development Practices

To fortify cybersecurity measures within the maritime sector, the adoption of comprehensive assessment frameworks like the Software Assurance Maturity Model (SAMM) and the Building Security In Maturity Model (BSIMM) is critical. These frameworks serve as the cornerstone for organizations aiming to critically assess and enhance their secure software development methodologies.

Software Development Lifecycle (SDLC) – Maritime Challenges and Best Practices

SDLC Models Overview

In maritime software development, the choice of SDLC models—Waterfall, Agile, and DevOps—is critical. Each model offers distinct advantages: Waterfall for its structured approach, Agile for flexibility and rapid iteration, and DevOps for enhancing collaboration between development and operations. These models serve as the backbone for incorporating security throughout the development lifecycle, ensuring that software meets the high-security demands of maritime operations.

Maritime-Specific Challenges

The maritime industry faces unique challenges that impact software development, including strict adherence to international regulations like IMO and SOLAS, ensuring seamless interoperability among a variety of systems, and protecting against sophisticated cyber threats in a globally interconnected environment. Addressing these challenges within the SDLC is crucial for developing secure, reliable maritime software solutions.

Software Development Lifecycle (SDLC) – Maritime Challenges and Best Practices

Best Practices for Secure Coding

Input Validation: Critical for defending against attacks such as SQL injection and cross-site scripting, input validation involves scrutinizing all incoming data against predetermined criteria, effectively blocking malicious inputs that could compromise maritime systems.

Error Handling: Proper error handling practices are vital for preventing the disclosure of sensitive information and ensuring the stability of maritime operations. Secure error handling maintains operational continuity, even in the event of unexpected system errors.

Secure Authentication: The implementation of robust authentication mechanisms, including multi-factor authentication and enforcing the principle of least privilege, is fundamental. These measures are essential for controlling access to maritime systems, thereby safeguarding sensitive data and operations.

Incorporating Security Across the SDLC

Integrating security from the initial stages of development and throughout is non-negotiable. From conducting thorough threat modelling to identify and mitigate potential vulnerabilities early in the design phase to adhering to secure design principles and conducting regular security testing, these practices are pivotal for early detection and remediation of security issues.

Adapting to Evolving Threats

The constantly changing landscape of cyber threats targeting maritime operations necessitates a dynamic approach to security. Continuous security assessments, timely software updates, and ongoing staff training ensure that maritime software development practices remain effective and responsive to new and emerging threats.

Secure Software Development Frameworks

Trustworthy Software Framework (TSFr)

An integration of good practices, existing guidance, and relevant standards covering safety, reliability, availability, resilience, and security, TSFr offers a structured approach ensuring software trustworthiness across various facets, pivotal for maritime systems that demand high reliability and safety.

NIST's Cybersecurity Framework

Focuses on managing and reducing cybersecurity risk in a prioritized, flexible, and cost-effective manner. This framework's adaptability makes it suitable for securing maritime operations against evolving threats, enhancing the resilience of maritime infrastructure.

Software Engineering Institute (SEI) Guidance

SEI's contributions to software assurance offer comprehensive resources for building security into software systems from the ground up, crucial for creating resilient maritime software capable of withstanding cyber threats.

Secure Software Development Frameworks

UK National Cyber Security Centre (NCSC)

Provides extensive resources for secure software development, including application development and secure deployment guidance. This is particularly relevant for the maritime industry, where securing complex and interconnected systems is paramount.

OWASP's Secure Software Development Lifecycle (S-SDLC)

Defines a standard SDLC with a security focus, offering invaluable resources for maritime software developers to incorporate best practices at each development phase, ensuring robust security from design to deployment.

Tailoring SDLC Models for Maritime Software Security

Introduction to SDLC Models

The maritime software landscape requires a strategic approach to software development, where the choice of SDLC model—Waterfall, Agile, or DevOps—plays a pivotal role. Selecting the appropriate model is crucial, depending on the project's requirements, the dynamics of the development team, and the specific security needs of maritime systems.

Waterfall Model

The Waterfall model is characterized by its linear and sequential approach, dividing the project into distinct phases: Requirements, Design, Implementation, Verification, and Maintenance. This model's strength lies in its clear structure and milestones, making it ideal for projects with well-defined requirements and scope.

- ***Pros:*** Offers a straightforward approach with clear milestones, beneficial for projects with fixed requirements.
- ***Cons:*** Its rigidity limits adaptability to changes, posing challenges in the dynamic maritime security environment where requirements can evolve.

Tailoring SDLC Models for Maritime Software Security

DevOps Approach

DevOps merges software development (Dev) with IT operations (Ops), emphasizing automation, continuous delivery, and rapid feedback cycles. This approach is designed to reduce the time from development to operation, critical for responding swiftly to security threats in maritime systems.

Pros: Facilitates rapid deployment and shorter development cycles, crucial for maritime systems requiring immediate updates and security patches.

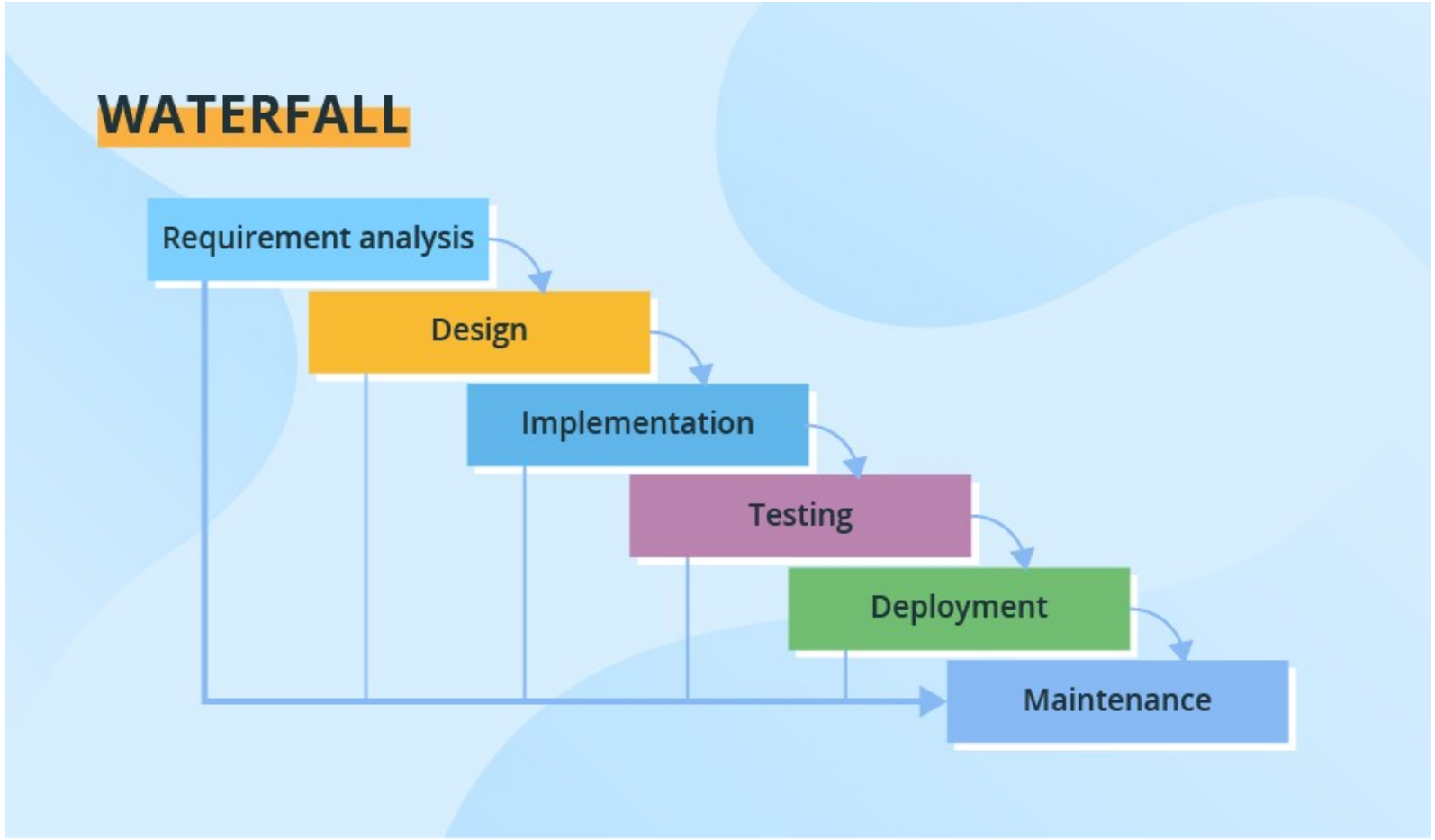
Cons: Demands a significant cultural shift and tight coordination between development and operations, which can be challenging to establish.

Relevance to Maritime Software Development

Tailoring these SDLC models to the maritime industry involves addressing unique challenges such as regulatory compliance, the security of critical navigation and communication systems, and the integration with existing onboard legacy systems. Embedding security practices at each stage of the selected SDLC model is imperative for developing secure, resilient maritime software solutions.

Software Development Lifecycle (SDLC)

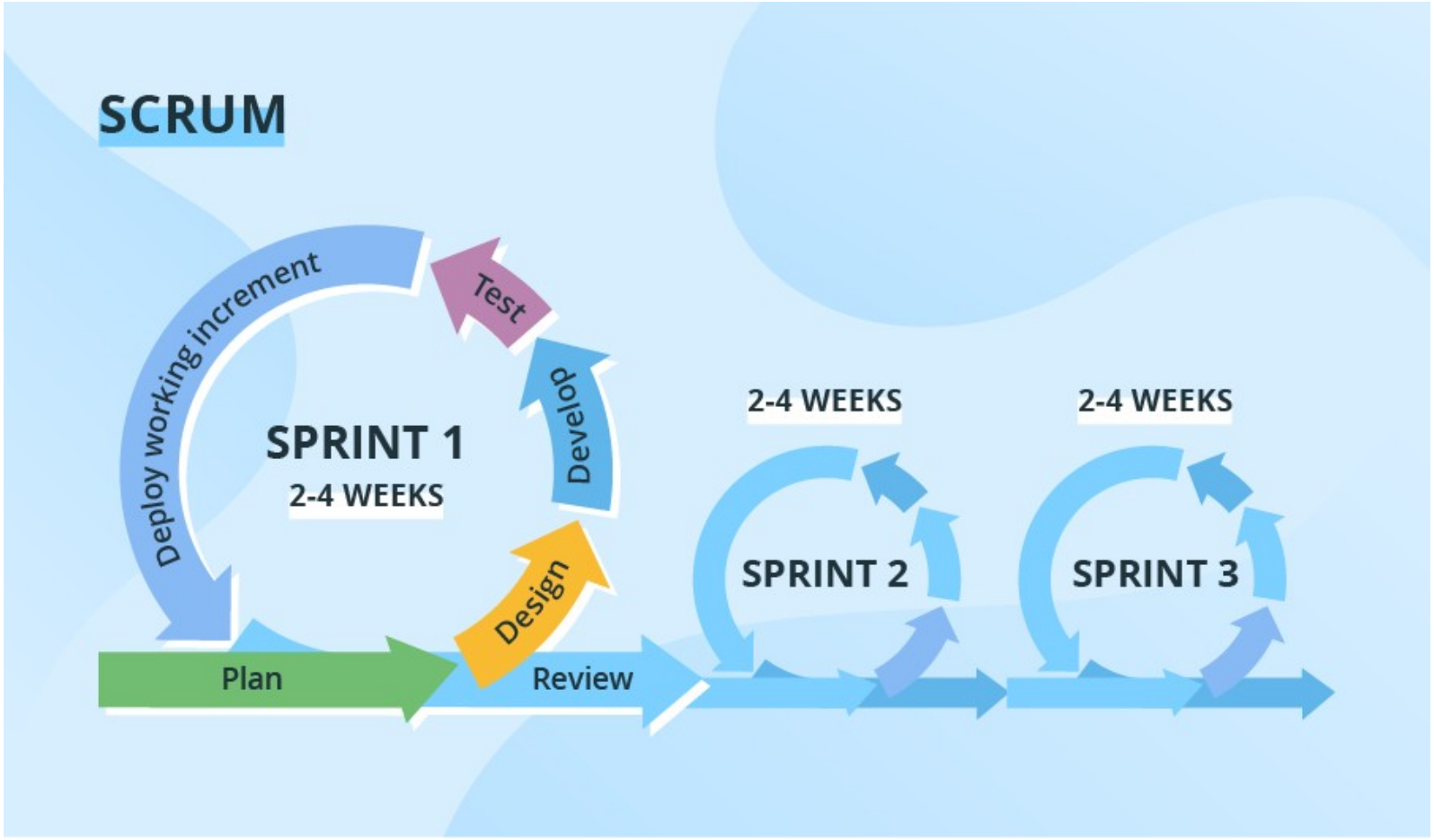
Waterfall



https://www.scnsoft.com/software-development/software-development-models

Software Development Lifecycle (SDLC)

Agile Group



https://www.scnsoft.com/software-development/software-development-models

Best Practices for Integrating Security in SDLC

Training and Awareness

All team members, including developers, project managers, and product managers, require continuous cybersecurity training. This ensures everyone is equipped with up-to-date knowledge on security threats and best practices, emphasizing the evolving nature of cyber threats and the importance of secure coding practices.

Defining Security Requirements

Security requirements are critical from the onset, influenced by the system's functional needs, compliance obligations, and known threats. Utilizing frameworks like Security Quality Requirements Engineering (SQUARE) aids in integrating security into the initial stages, ensuring a solid foundation for the development lifecycle.

Threat Modelling

A structured approach to identifying and addressing potential security issues within the design. This includes analyzing the system's architecture to identify and prioritize threats, using methodologies such as STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Escalation of privilege) to systematically evaluate potential vulnerabilities.

Best Practices for Integrating Security in SDLC

Static Analysis Security Testing (SAST)

Automated tools review code for common security issues, enabling early detection of vulnerabilities. SAST can be integrated into the development pipeline to ensure ongoing adherence to secure coding standards.

Dynamic Analysis Security Testing (DAST)

Assessing the application in its running state to identify vulnerabilities that manifest during execution. Techniques like fuzzing automate the discovery of security flaws, complementing SAST by uncovering runtime issues.

Penetration Testing

Manual testing simulates cyber-attacks to identify vulnerabilities, including both design flaws and coding errors. This critical assessment provides a real-world perspective on the application's security posture.

Best Practices for Integrating Security in SDLC

Standard Incident Response Plan (IRP)

Preparing for inevitable security incidents with a well-defined response plan. This includes protocols for rapid vulnerability mitigation, effective communication strategies, and swift deployment of fixes. Incorporating lessons learned back into the SDLC is essential for continuous improvement.

Integration and Continuous Improvement

Security practices must be woven into each phase of the SDLC, from planning to deployment and beyond. Continuous assessment and adaptation of security measures ensure resilience against evolving threats.

Fortifying Maritime Operations with Secure Coding Practices

Critical Importance

In the maritime industry, where software increasingly underpins navigation, communication, and operational management, secure coding practices are vital. They prevent code-level vulnerabilities typically introduced through programmer mistakes, forming a strong cybersecurity foundation.

Proactive Prevention and Detection

The adoption of coding standards and the selection of secure programming languages, frameworks, and libraries are crucial. Leveraging their security features helps pre-emptively address vulnerabilities, a practice critical for maritime systems where security lapses can have significant operational impacts.

Automated Tools and Manual Reviews

Utilizing automated analysis tools in conjunction with manual code reviews significantly enhances the ability to identify and rectify security vulnerabilities. This dual approach is essential in making maritime software systems resilient to cyber threats, ensuring the safety and efficiency of maritime operations.

Fortifying Maritime Operations with Secure Coding Practices

Standards and Guidelines

Embracing established secure coding standards and guidelines, such as the OWASP Secure Coding Practices Quick Reference Guide and the SEI CERT Coding Standards, provides developers with a comprehensive checklist. This ensures the development of secure maritime software that adheres to the highest security benchmarks.

Error Handling and Third-Party Risks

Properly managing unanticipated errors through controlled handling mechanisms and addressing security risks posed by third-party components are paramount. These practices prevent vulnerabilities within the maritime software ecosystem, maintaining operational integrity and safety.

Continuous Improvement

The journey toward secure maritime software development is continuous, necessitating the incorporation of lessons learned from security incidents and the adoption of new security findings. Regularly updating practices to counter evolving threats is crucial for sustaining the security and integrity of maritime operations.

Sembcorp Cybersecurity Incident: A Third-Party Software Challenge

Date of Incident: 31 August.

Challenge Identified: Unauthorized access through third-party software.

Information Accessed: Employee data & non-critical operational info.

Expert Intervention: Security experts engaged immediately for risk mitigation.

Root Cause Analysis: Detailed analytics performed to flush out breaches.

Enhanced Security Measures: Strengthened core IT infrastructure and systems.

Continuous Monitoring: Ongoing assessment and improvement of security defences.

Input Validation Techniques

Essential for Security

Input validation stands as a cornerstone of security within maritime software applications, acting as the primary defense against various input-based attacks such as SQL Injection, Cross-Site Scripting (XSS), and buffer overflows. These vulnerabilities pose significant risks to maritime operations, threatening the integrity and security of critical data.

Best Practices

Whitelist Validation: Prioritize allowing only data that fits a predefined list of acceptable values, significantly reducing the risk of malicious input exploitation.

Type, Length, and Syntax Checks: Implement checks to ensure input data aligns with expected types, lengths, and formats, preventing the processing of incorrect or potentially harmful data.

Use of Secure Frameworks and Libraries: Leverage frameworks and libraries equipped with built-in input validation capabilities to minimize the chance of developer errors and streamline the application of secure coding practices.

Input Validation Techniques

Best Practices

Escaping Special Characters: When user inputs are incorporated into SQL queries or HTML content, properly escape special characters to safeguard against injection and scripting attacks.

Regular Expression for Pattern Matching: Employ regular expressions to rigorously define and enforce acceptable input patterns, effectively filtering out unsanctioned data formats.

Challenges in Maritime Context

The intricate and interconnected nature of maritime software systems, spanning navigation, cargo management, and beyond, amplifies the necessity for stringent input validation. This is crucial to averting system-wide failures and security breaches resulting from compromised input data.

Continuous Validation and Testing

To keep pace with the dynamic threat landscape, it's imperative to frequently update validation criteria and rigorously test input validation mechanisms. Employing contemporary security testing methodologies ensures sustained protection against emerging threats and attack vectors.

Strengthening Maritime Software: Secure Error Handling and Logging Practices

Critical Role of Error Handling in Maritime Software

Effective error handling is vital for maritime software systems, where reliability and safety are paramount. Ensuring that software applications behave predictably, especially under adverse conditions, is crucial for systems involved in navigation, communication, and operational management at sea.

Best Practices for Error Handling in Maritime Context

Develop error handlers that maintain applications in a secure state, crucial for operations where errors can have significant safety implications. Design error handling to not disclose sensitive or operational information, preventing potential exploitation by attackers.

Graceful error handling is essential, ensuring that users are informed without compromising security or exposing system vulnerabilities, a balance critical for the complex environments of maritime operations.

Strengthening Maritime Software: Secure Error Handling and Logging Practices

Importance of Logging for Maritime Security

Logging is indispensable for the oversight and diagnosis of maritime software behavior. It provides insights into both regular and suspicious activities, facilitating rapid response to security incidents which is critical for maintaining the safety and security of maritime operations.

Implement secure logging practices to capture and manage logs without recording sensitive data, preventing unauthorized access or manipulation, and supporting compliance with maritime security regulations.

Adapting Secure Logging to the Maritime Industry

Ensure that logging mechanisms and policies are robust, with a focus on securing against tampering and unauthorized access, reflecting the high stakes of maritime security. Use logs not just for problem resolution but also for understanding and improving the security posture of maritime software systems.

Leverage industry resources like the OWASP Logging Cheat Sheet, applying its principles to maritime contexts. This involves adjusting guidelines to address the specific threats and operational requirements of maritime software applications, ensuring enhanced security and compliance.

Course progress

01. Introduction to Software Security in the Maritime Industry

02. Principles of secure software development lifecycle (SDLC) tailored for the maritime industry

03. Understanding the process of threat modelling for identifying and mitigating maritime software security risks

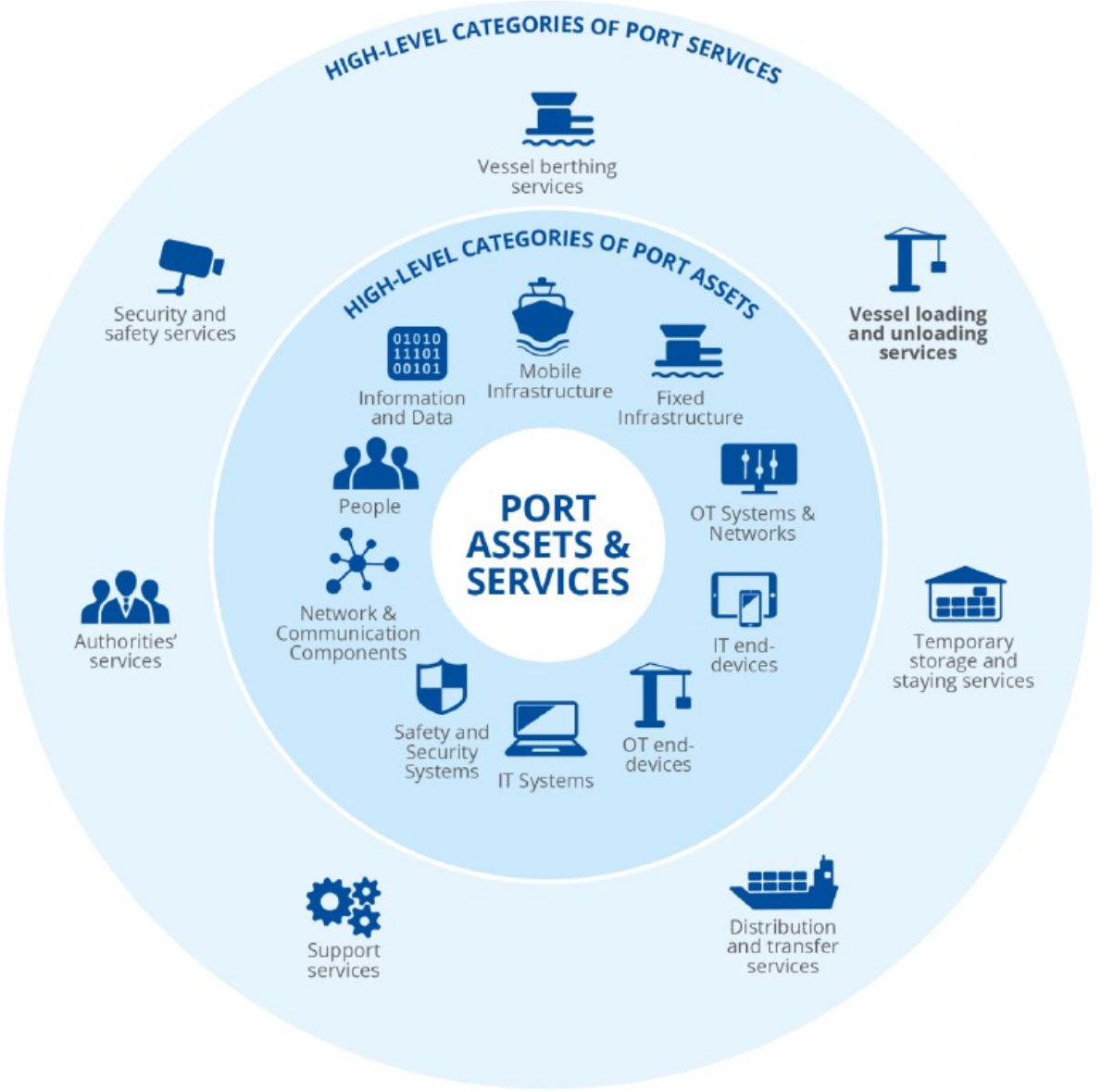
04. Overview of software security testing methodologies, including static and dynamic analysis

05. Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector

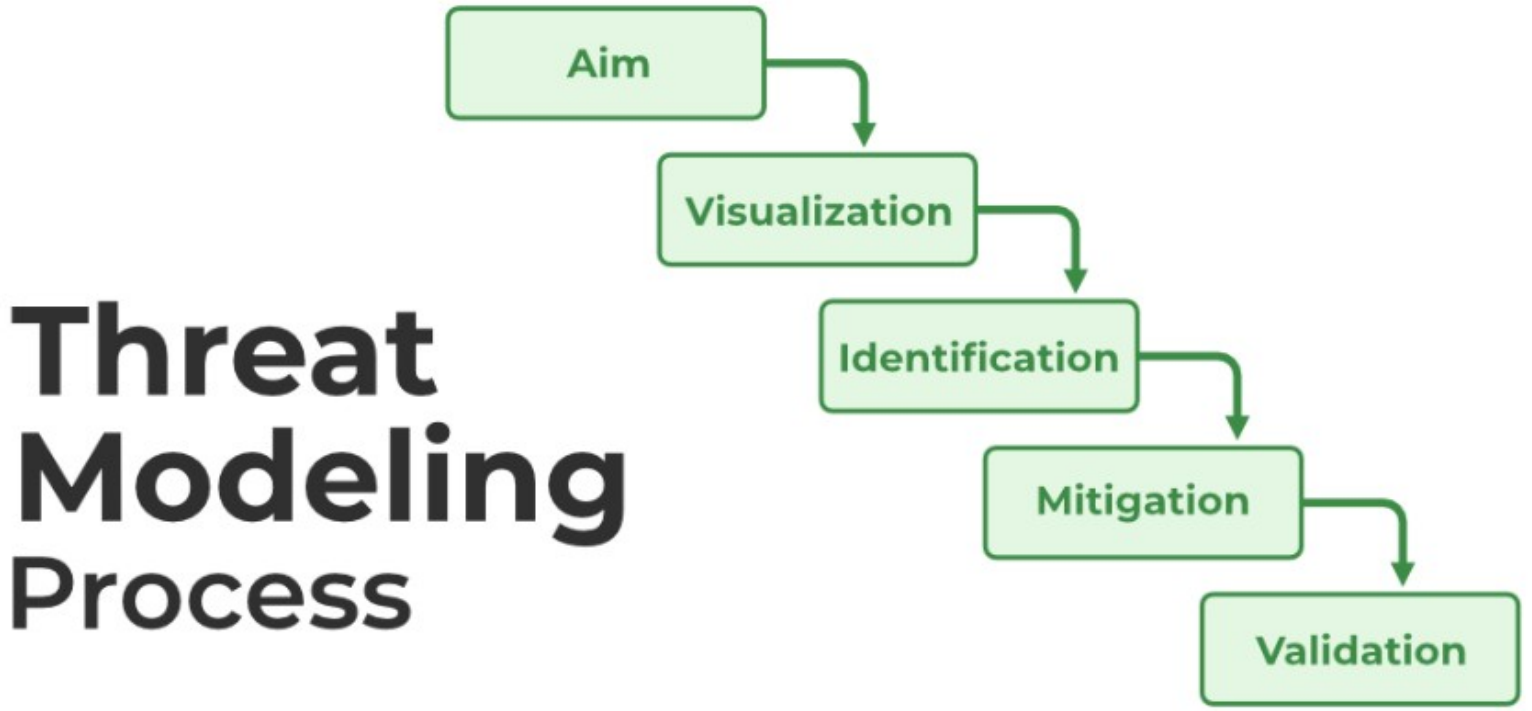
Understanding the value of threat modelling

Why it is important in the Maritime Industry

- Maritime:
  - critical sector - key to international supply chains
  - plethora of interconnected assets, processes, sensitive data
  - all of them are accompanied by software or firmware
- Threat modelling:
  - provides a systematic way to assess the security of a system at a given point
  - requires (thus, guides towards the creation of) inventories of areas of interest
  - identifies ways to effectively mitigate risks



Understanding the process of threat modelling



Understanding the process of threat modelling

STRIDE in more detail (1/2)

- Initially presented in 1999 by Microsoft staff members Loren Kohnfelder and Praerit Garg.
- A powerful brainstorming tool for threat modelling, that puts someone in the attacker's shoes, by simply asking 6 diverse questions:
    - **Spoofing:** How can I impersonate the users?
    - **Tampering:** How can I modify users' data without permission?
    - **Repudiation**: How can I hide my tracks?
    - **Information disclosure:** How can I leak users' secrets?
    - **Denial of service:** How can I affect the availability of their system?
    - **Escalation of privilege:** How can I gain additional access?
- Pros: Simple, beginner-friendly, helpful brain teaser
- Cons: Attacker-centric approach, also not a methodology

Understanding the process of threat modelling

STRIDE in more detail (2/2)

Threat Category	Violates	Examples
Spoofing	Authenticity	An attacker steals the authentication token of a legitimate user and uses it to impersonate the user.
Tampering	Integrity	An attacker abuses the application to perform unintended updates to a database.
Repudiation	Non-repudiability	An attacker manipulates logs to cover their actions.
Information Disclosure	Confidentiality	An attacker extract data from a database containing user account info.
Denial of Service	Availability	An attacker locks a legitimate user out of their account by performing many failed authentication attempts.
Elevation of Privileges	Authorization	An attacker tampers with a JWT to change their role.

What is violated

is also

the security control.

https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

Understanding the process of threat modelling

Other threat modelling approaches, complementary to STRIDE

- **Software-centric:** giving priority to software components, communication, data flows, trust boundaries
- **Asset-centric:** identifying and analyzing assets and their likely exposure
- **Threat-centric:** focusing on processes and technologies, identifying attack vectors and their impact
- All come with benefits and limitations
- A combined approach would be comprehensive

Understanding the process of threat modelling

Effective threat modelling overview

1) **Definition of analysis' scope and depth:** boundaries setting, expected outcomes identification.

2) **Data gathering**: architectural descriptions, software design documents, API specifications, end user documentation, source code, and other related artifacts which point high-risk areas.

3) **System modelling:** high-level diagram creation of all the in-scope (or out-of-scope, if needed), adjacent, or connecting components, their connections and their protocols.

4) **Threat analysis:** possible realistic system threats enumeration to the system – focus on data, dataflow, platform, and operational security.

5) **Risk analysis:** traceability matrix creation and calculation of risk for the attack paths that either cannot be supported by a control or the control can be bypassed by an attacker – also security weaknesses monitoring and/or penetration testing conduction if needed.

6) **Reporting:** documentation of models, system descriptions, potential risks, actual risks and their associated likelihood and impact.

Data Flow Diagrams in threat modelling

Data Flow Diagrams (DFD) introduction

- **Origin:** Developed in the 1970s by system engineers to visualize data movement, storage, and manipulation within systems.
- **Evolution:** In the early 2000s, security professionals introduced the concept of trust boundaries to make DFDs applicable to threat modelling.

Data Flow Diagrams in threat modelling

DFD purpose

- Used to **identify** broad categories of threats
- Often employs the **STRIDE threat classification scheme** (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege)

Data Flow Diagrams in threat modelling

DFD-Based threat modelling steps

- View the System **as an Adversary**
- **Characterize the System:** Identify data flows, processes, and trust boundaries
- **Determine** the threats

Data Flow Diagrams in threat modelling

DFD Challenges and Limitations

- DFDs do not accurately depict the application's **design and flow**
- **The list of threats** generated is often incomplete and can be a weak starting point for modelling
- Focus on **data movement** rather than user interaction
- **No standard approach,** leading to inconsistent threat modelling outputs for the same scenario or problem
- **Overall** useful for visualizing data flow, but not ideal as a sole approach for threat modelling.

Data Flow Diagrams in threat modelling

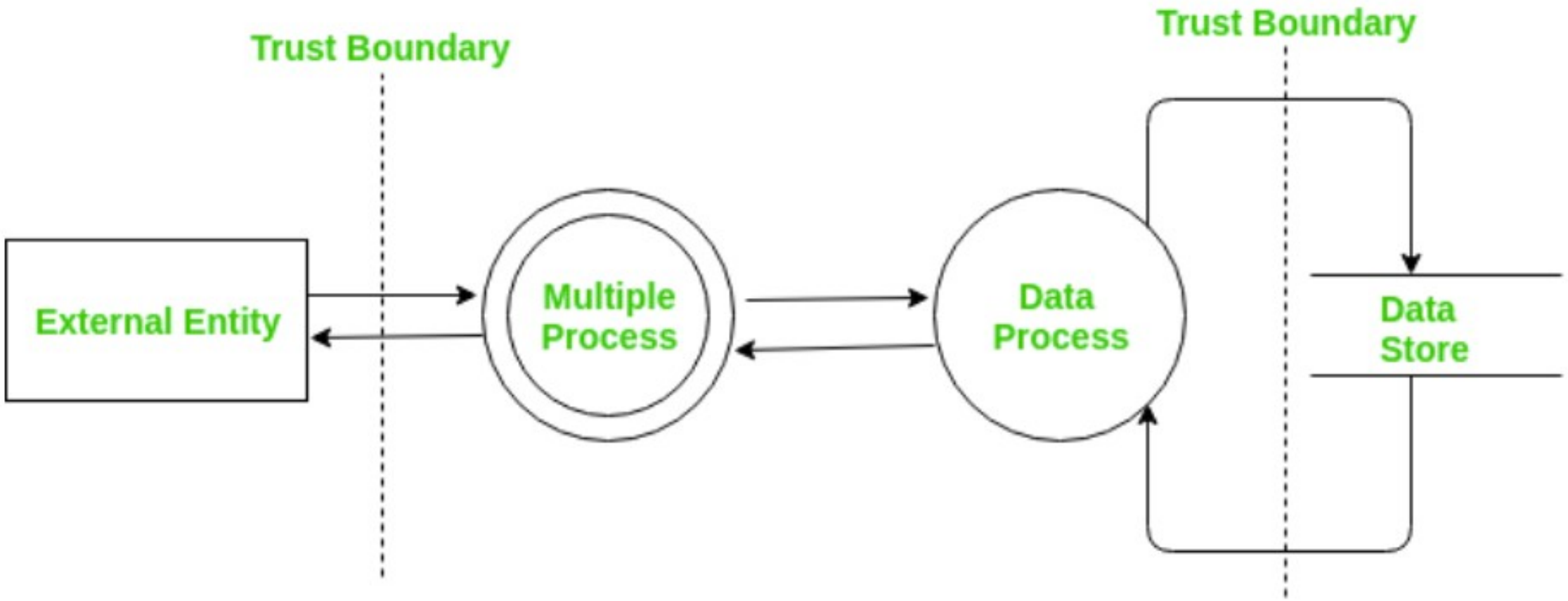
DFD Symbols

Symbol	Name	Description
	External Entity	The external entity shape is used to represent any entity outside the application that interacts with the application via an entry point.
	Process	The process shape represents a task that handles data within the application. The task may process the data or perform an action based on the data.
	Multiple Process	The multiple process shape is used to present a collection of subprocesses. The multiple process can be broken down into its subprocesses in another DFD.
	Data Store	The data store shape is used to represent locations where data is stored. Data stores do not modify the data, they only store data.
	Data Flow	The data flow shape represents data movement within the application. The direction of the data movement is represented by the arrow.
	Privilege Boundary	The privilege boundary (or trust boundary) shape is used to represent the change of trust levels as the data flows through the application. Boundaries show any location where the level of trust changes.

https://owasp.org/www-community/Threat_Modeling_Process

Data Flow Diagrams in threat modelling

DFD Example

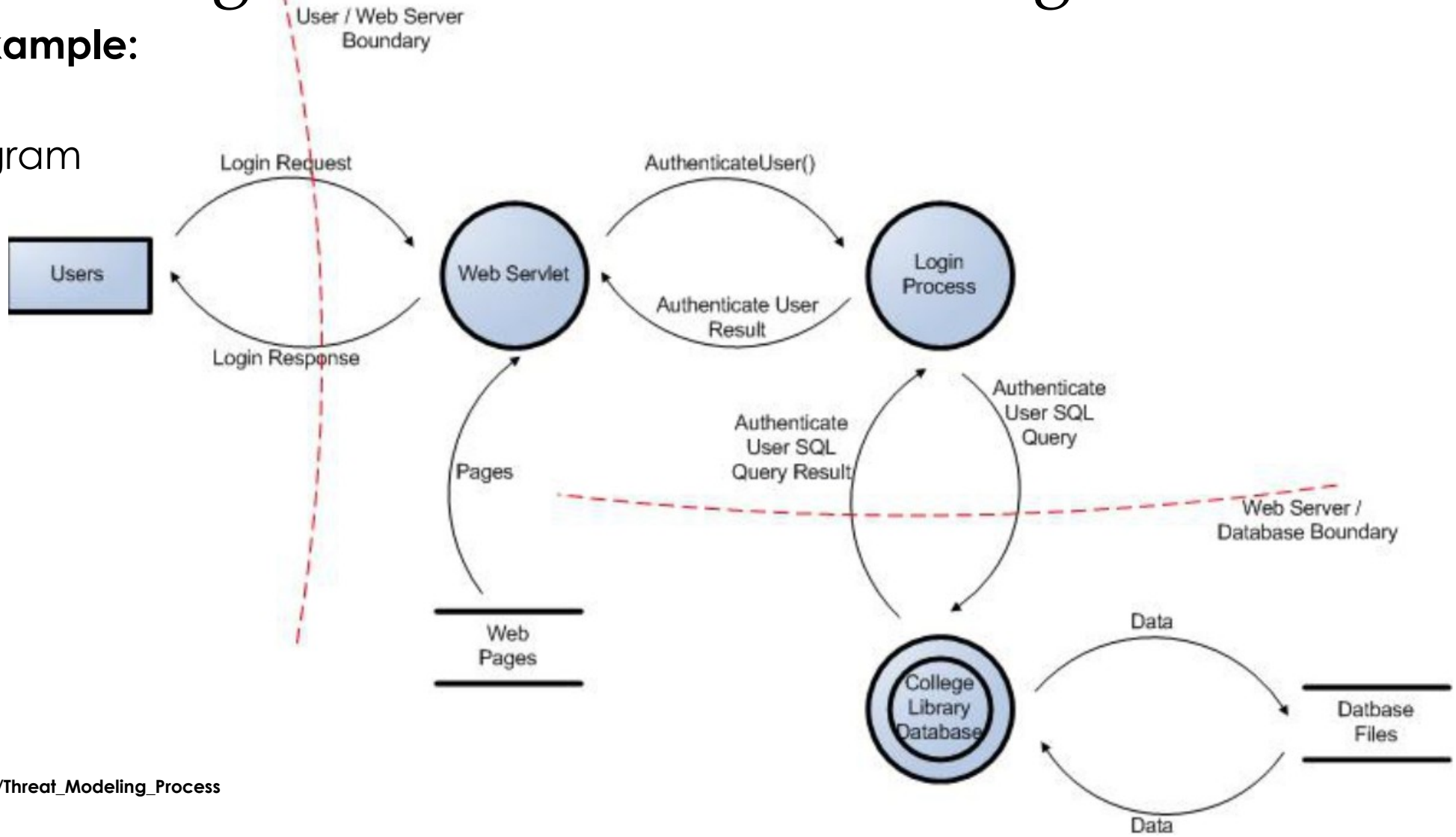


https://www.geeksforgeeks.org/threat-modelling/

Data Flow Diagrams in threat modelling

DFD Specific Example:

User Login Data Flow Diagram for the College Library Website



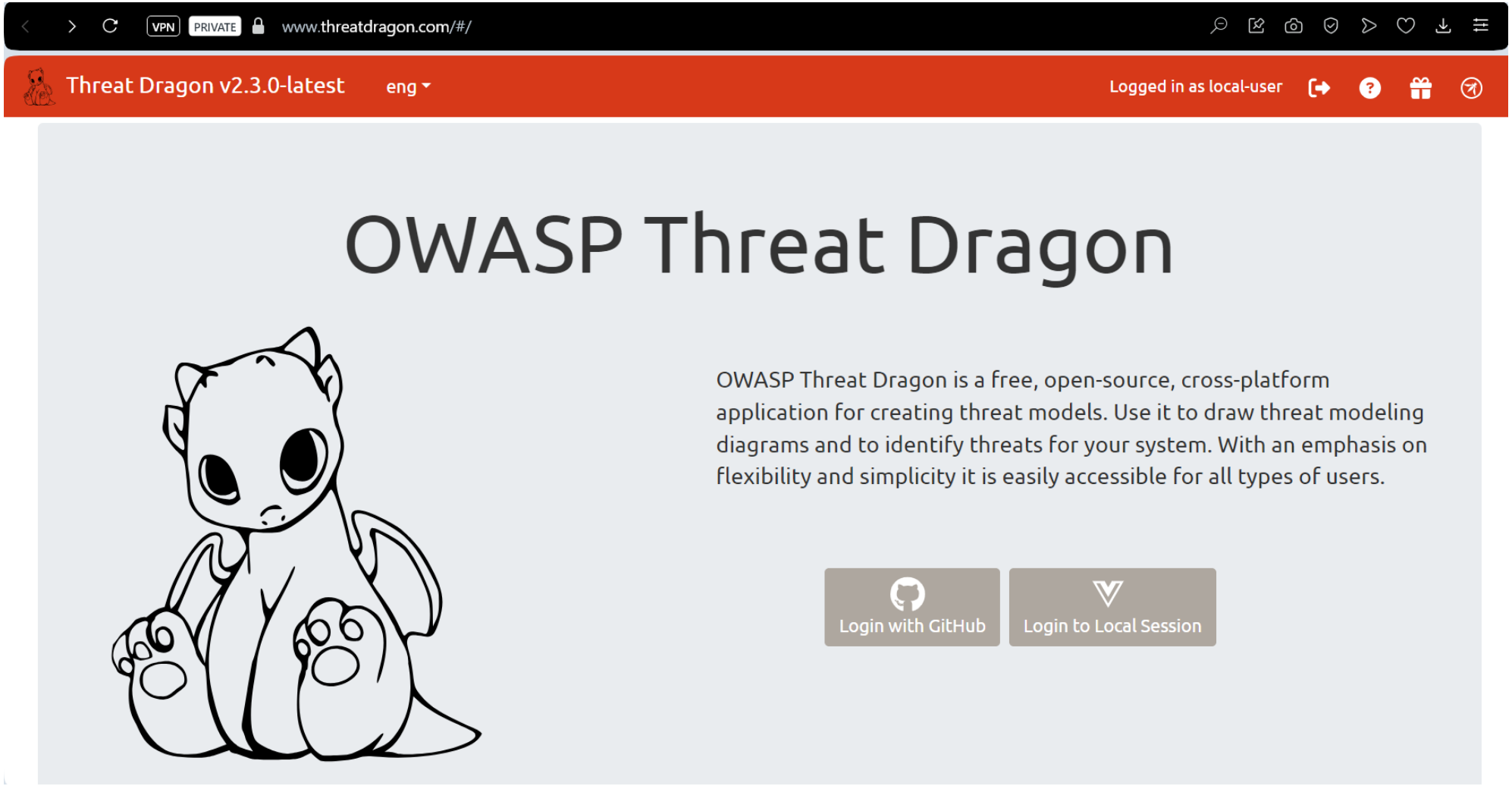
https://owasp.org/www-community/Threat_Modeling_Process

Threat Modelling tool introduction

OWASP Threat Dragon

Threat Modelling tool introduction

OWASP Threat Dragon



Threat Modelling Practical Exercise

Threat Dragon practice

- Divide into groups
- Create a data flow diagram for an online ticket booking system for maritime transport
- Write down potential threats and mitigation measures

A high-level example of Maritime cyber risk management methodology by ENISA

...based on threat modelling principles

Online tool: https://www.enisa.europa.eu/cyber-risk-management-for-ports#/

Secure deployment and configuration management

Enhancing Software Security for Maritime Operations

What are they?

Secure deployment involves implementing processes to ensure that software and systems are installed and configured securely.

Configuration management involves managing the settings and parameters of software and systems to maintain security.

Why are they important?

Secure deployment and configuration management are essential to mitigate security risks and ensure the resilience of maritime software systems against cyber threats.

Common software deployment strategies

- **Blue-Green deployment:** running two identical production environments, but only one of them being live each time, which serves all production traffic.
    - Risk: sensitive data could be exposed, lost or corrupted during the switchover – security measures must be duplicated.
- **Canary deployment:** gradually rolling out a small subset of users before the entire infrastructure deployment.
    - Risk: vulnerability exploitation could affect a subset of users or even leak to the main production environment.
- **Feature toggles / flags:** enabling or disabling features in a live environment, hidden from users until ready for release.
    - Risk: potential unfinished/untested features to end-users – vulnerabilities exposure.
- **Rolling deployment:** serially deploying across all servers/nodes – always a version available to users.
    - Risk: versioning desynchronization – potential vulnerabilities.
- **A/B testing deployment:** concurrent run of two software versions, separating the users who are directed to them into two pools to test the performance before the full release.
    - Risk: versioning desynchronization – exposure to different levels of risk.

General software deployment potential risks

Code tampering: unauthorized access and code alteration for malicious purposes.

- Mitigation measures: code access control, version control / track changes.

- **Third-party vulnerabilities:** third-party libraries and components vulnerabilities.
  - Mitigation measures: third-party libraries and components updates, audits, and/or patches.
- **Configuration errors:** software system configuration errors that can lead to malfunctions.
  - Mitigation measures: thorough functional and configuration testing.
- **Inadequate environment isolation:** deficient isolation of the test, staging and production environments - data leakage or unintentional deployment of untested code to the production environment.
  - Mitigation measures: separate servers for each environment, access control implementation.
- **Lack of monitoring:** leads to difficulty of issues identification and rectification.
  - Mitigation measures: error tracking and performance monitoring.

Best secure software deployment practices

- Code / Peer reviews
- Automated security scans
- Environment hardening
- Principle of least privilege (PoLP)
- Secure configuration management
- Immutable deployments

Best secure software deployment practices

- Code / Peer reviews
- Automated security scans
- Environment hardening
- Principle of least privilege (PoLP)
- Secure configuration management
- Immutable deployments

Course progress

01. Introduction to Software Security in the Maritime Industry

02. Principles of secure software development lifecycle (SDLC) tailored for the maritime industry

03. Understanding the process of threat modelling for identifying and mitigating maritime software security risks

04. Overview of software security testing methodologies, including static and dynamic analysis

05. Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector

Overview of Software Security Testing

- **Software security testing ensures that vulnerabilities are identified before and after deployment**, especially in safety-critical maritime systems.
    - **Focuses on identifying weaknesses** in code, runtime behavior, and system integration
    - **Applies across the entire lifecycle** (development, testing, deployment, operation)
    - **Critical for systems** such as navigation, communication, and port platforms
    - **Must account for limited downtime and operational constraints**

Categories of Security Testing

- **Security testing methods differ based on how they analyze the system.**
  - **Static Application Security Testing (SAST)** analyzes source code without execution
  - **Dynamic Application Security Testing (DAST)** evaluates system behavior at runtime
  - **Penetration Testing (Pentest)** simulates real attacker actions
  - **Software Composition Analysis (SCA)** focuses on third-party libraries and dependencies
  - **Mobile Application Security Testing (MAST)** evaluates the security of mobile applications during execution
- **Tool examples per category:**
  - **SAST:** SonarQube, Fortify, Checkmarx
  - **DAST:** OWASP ZAP, Burp Suite, Acunetix
  - **PenTest:** Metasploit, Nmap, Kali Linux
  - **SCA:** Snyk, OWASP Dependency-Check, Black Duck
  - **MAST:** MobSF, Drozer, QARK

Static Application Security Testing (SAST)

- **Analyzes source code to identify vulnerabilities early in development, before the system is executed.**
  - Identifies insecure coding patterns, logic flaws, and unsafe function usage
  - Applied during development, often as part of automated pipelines
  - Enables early detection and correction of issues before they propagate
- **Maritime use case:** Detecting improper input validation in navigation or control software before deployment
- **Limitations:**
  - Does not detect runtime issues such as misconfigurations or environment-specific vulnerabilities
  - May produce false positives that require manual validation
  - Limited visibility into system behavior and interactions between components

Dynamic Application Security Testing (DAST)

- **Tests the system while it is running to identify vulnerabilities that appear during execution.**
  - Detects authentication issues, API flaws, and misconfigurations
  - Evaluates how the system behaves under real usage conditions
  - Simulates external interaction, similar to an attacker probing the system
- **Maritime use case:** Testing a port management web system to identify exposed endpoints or weak authentication mechanisms
- **Limitations:**
  - Does not provide visibility into source code or internal logic
  - May miss vulnerabilities that are not exposed through the running interface
  - Requires a deployed and functional system to be effective

Penetration Testing (Pentest)

- **Simulates real-world attacks to evaluate how vulnerabilities can be exploited in practice.**
    - Identifies weaknesses that automated tools may not detect
    - Focuses on attacker perspective and realistic attack paths
    - Typically performed before deployment or during scheduled security assessments
- **Maritime use case:** Testing remote vessel access interfaces to identify unauthorized access or control risks
- **Limitations:**
    - Time-consuming and dependent on tester expertise
    - Not suitable for continuous or frequent execution
    - May be constrained in live environments due to operational risks

Software Composition Analysis (SCA)

- **Analyzes third-party libraries and dependencies to identify known vulnerabilities.**
  - Detects publicly disclosed vulnerabilities in open-source components
  - Provides visibility into software supply chain risks
  - Supports secure dependency management and updates
- **Maritime use case:** Detecting vulnerable third-party libraries in cargo or port management systems
- **Limitations:**
  - Limited to known vulnerabilities and existing databases
  - Does not identify issues in custom-developed code
  - Requires continuous updates to remain effective

Mobile Application Security Testing (MAST)

- **Evaluates the security of mobile applications during execution on actual devices.**
  - Focuses on mobile-specific risks such as insecure storage and communication
  - Analyzes application behavior in real usage conditions
  - Combines static and dynamic techniques
- **Maritime use case:** Testing mobile applications used by crew or port personnel for secure communication and operations
- **Limitations:**
  - Limited to mobile platforms
  - Requires access to real devices or emulators

Combining Testing Methods

- **No single testing method provides complete coverage of system security.**
    - Static testing identifies issues early in development
    - Dynamic testing validates behavior during execution
    - Penetration testing evaluates real-world exploitability
    - Composition analysis addresses third-party risks
    - Mobile application testing addresses platform-specific risks
- **Maritime use case:** A vessel communication platform is analyzed with static testing during development, validated with dynamic testing before release, assessed through penetration testing prior to deployment, and supported by mobile testing for crew-facing applications.
- **Effective security testing requires combining complementary methods across the lifecycle.**

Applying Testing in Maritime Systems

- **Security testing in maritime environments must adapt to operational and safety constraints.**
  - Critical systems are thoroughly tested before deployment
  - Live systems onboard vessels have limited testing windows
  - Continuous monitoring supports validation during operation
- **Maritime examples:** Navigation systems are fully validated prior to installation, while port management platforms are tested and monitored continuously during operation
- Balancing thorough testing with system availability and operational continuity

Exercise: Selecting a Testing Method

- **A maritime communication system connects onboard systems with satellite and port services.**
- **Task:**
  - Identify the most critical components (e.g. communication interfaces, APIs, authentication mechanisms)
  - Determine possible vulnerabilities (e.g. weak authentication, exposed endpoints, insecure data handling)
  - Select appropriate testing methods (SAST, DAST, PenTest, SCA)
  - Explain why each method fits the selected components
- **Approach:**
  - Start from system exposure (what is externally accessible?)
  - Focus on high-impact components first (communication and access control)
  - Match testing method to system layer (code, runtime, interfaces, dependencies)

Exercise's Example Solution

- **Critical components:**
  - Communication interfaces (ship ↔ satellite ↔ port)
  - APIs for data exchange
  - Authentication mechanisms
  - Data handling components
- **Potential vulnerabilities:**
  - Weak or missing authentication
  - Exposed or insecure APIs
  - Unprotected communication channels
  - Improper input validation
  - Vulnerable third-party libraries
- **Selected testing methods:**
  - SAST for identifying code-level issues
  - DAST for testing exposed interfaces and APIs
  - Pentest for simulating real-world attacks
  - SCA for detecting vulnerabilities in dependencies
- **Reasoning:**
  - External exposure requires dynamic testing and attack simulation
  - Code-level risks require static analysis
  - Third-party components require dependency analysis
  - Critical communication paths require combined testing methods

Course progress

01. Introduction to Software Security in the Maritime Industry

02. Principles of secure software development lifecycle (SDLC) tailored for the maritime industry

03. Understanding the process of threat modelling for identifying and mitigating maritime software security risks

04. Overview of software security testing methodologies, including static and dynamic analysis

05. Exploration of future trends and emerging technologies shaping the future of software security in the maritime sector

Digital Transformation & Software Complexity

- **Maritime operations are increasingly driven by complex and interconnected software systems:**
  - Growing use of distributed and cloud-based software architectures
  - Integration of onboard applications with port and external platforms
  - Increased reliance on real-time data processing and remote software access
- **Implications for software security:**
  - Expanded attack surface across software components and interfaces
  - Increased complexity in securing software integrations and dependencies
  - Strong reliance on software integrity and availability for safe operations

Impact of Emerging Technologies on Software Security

- **Emerging technologies introduce new capabilities but also increase software security risks:**
  - **IoT** introduces distributed software components on devices with limited security controls, increasing exposure through device interfaces and communication channels
  - **AI** introduces software components that rely on data-driven logic, creating risks in model integrity, input manipulation, and lack of transparency in decision-making
  - **Blockchain** shifts trust from centralized systems to distributed validation logic, introducing new risks in smart contracts, key management, and transaction validation
  - **Cloud systems** rely on external APIs for identity, data storage, and service integration, increasing exposure to API-level vulnerabilities
- **As maritime software systems become more distributed and interconnected, vulnerabilities in one component can affect the security of the entire system.**

Emerging Software Security Risks in Maritime Systems

Modern maritime software introduces new categories of security risks.

- **Risk Drivers:**
  - Autonomous and software-driven vessel operations
  - Integration of multiple software systems and external services
  - Use of third-party software components and libraries
  - Complex software supply chains and update mechanisms
- **Potential Risks:**
  - Exploitation of vulnerabilities in maritime software applications
  - Insecure APIs enabling unauthorized access to system functionality
  - Compromise of software update mechanisms or dependencies
  - Manipulation of software controlling critical vessel functions
- **Potential Impacts:**
  - Service disruption affecting maritime operations
  - Loss of integrity or availability of critical software systems
  - Safety incidents due to compromised software behavior
  - Cascading effects across interconnected software systems

Future Software Security Challenges

- **Maritime organizations will need to address evolving software security challenges:**
  - Securing complex and distributed software architectures
  - Managing vulnerabilities in third-party software components
  - Ensuring secure software updates in long-lived systems
  - Detecting and responding to advanced attacks targeting software
- **Challenges:**
  - Maintaining visibility and control over distributed software components
  - Ensuring consistent security across integrated systems and environments
  - Balancing software innovation with safety and regulatory requirements

Building Resilient Maritime Software Systems

- **Improving software security requires changes in how systems are designed and maintained:**
  - Security-by-design in software architecture and development
  - Continuous testing and validation of software components
  - Secure management of dependencies and software updates
  - Collaboration across stakeholders for secure software integration
- **Opportunities:**
  - More resilient and secure software-driven maritime operations
  - Improved trust in interconnected software systems
  - Stronger software supply chain security practices

Discussion: Future of Maritime Software Security

- **Scenario:** A software-driven vessel relies on integrated onboard applications, cloud services, and external platforms.

- **Discussion points:**

  - How does increasing software complexity affect security risks?
  - What vulnerabilities emerge from software integration and dependencies?
  - What software security practices should be prioritized in future systems?
  - How can organizations ensure secure software updates and maintenance?

Thank you

Please send all questions to:
pkyranoudi@tuc.gr
spiros.borotis@maggioli.gr