

Chapter 2

LITERATURE REVIEW

This chapter provides an elucidation of the significance of DevSecOps. Next, we thoroughly examine and analyze the data, methodology, and the ways in which this research contribute to the development of methodologies for enhancing the security of DevSecOps in both the Security Testing Assessment and the Data Security Assessment.

2.1 PT XYZ

PT XYZ is progressing its business strategy by shifting to a digital-centric model, focusing on creating solutions that facilitate government digital initiatives upon request, while also crafting software that improves operational efficiency for a variety of organizations. As the company embarks on this digital evolution, the importance of both development speed and system security grows increasingly significant. Delivering digital products quickly is crucial to meet ever-changing market requirements, but this must be coupled with a rigorous focus on safeguarding data.

The transformation journey requires PT XYZ to deliver digital products rapidly while ensuring they are designed with strong security measures in place. In today's landscape of escalating cyber risks and frequent data breaches, the security of these solutions must be a top priority. It is critical that these platforms protect sensitive information, especially when managing personal details or confidential customer data. Any vulnerability or breach could jeopardize trust and have severe implications for both the company and its clients.

To mitigate these risks, PT XYZ needs to establish comprehensive security protocols throughout the development cycle. DevSecOps plays a vital role in automating security controls within the CI/CD pipeline, enabling continuous monitoring and reducing the manual effort required for security testing. Automated tools like Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST), along with other vulnerability scanning tools, are integrated into the development and deployment workflows to ensure that security is continuously

validated. This approach helps organizations address security issues early, enabling faster remediation and reducing the risk of vulnerabilities being introduced into production.

By shifting security left—embedding it early in the development process—and automating critical security checks, DevSecOps helps organizations achieve a higher level of security maturity. It aligns the goals of development, security, and operations teams, promoting a culture of shared responsibility for security and ensuring that applications are secure by design.

The primary focus of this thesis be the implementation of DevSecOps framework at XYZ, as previously discussed. This research encompass defining clear objectives, assessing the security maturity of applications developed using DevSecOps framework practices, and conducting evaluations to ensure that the applications created by PT XYZ meet the predetermined security maturity benchmarks.

2.2 Policy, Law and Regulation

Organizations have established security policies in place to make sure that the application when go to production already secured in the first place. However, product owners often overlook or disregard these policies, leading to vulnerabilities in products when they are released into production. Internal policies and standardized documents, must be diligently followed and enforced to ensure compliance with the organization's rules and security standards.

Security and privacy regulations are of critical importance, as they are designed to safeguard sensitive financial and personal data from unauthorized access, breaches, and misuse. The recent enactment of the Personal Data Protection Law (UU No. 27 Tahun 2022) marks a significant milestone in enhancing data privacy in Indonesia. This legislation provides a comprehensive framework for the protection of personal data, reflecting the global focus on securing consumer information in the digital age. These Indonesian regulations align with international standards while adapting security and privacy practices to the country's specific legal and cultural context. By doing so, they ensure that the organizational ecosystem remains resilient, secure, and responsive to local requirements.

2.2.1 Internal Security Policy

The organization has established its own security policies, which must be strictly enforced across all departments and applications. These security policies are outlined in the "Peraturan Direksi" (Director's Regulation) and serve as a mandatory framework that every unit, especially those involved in application development, must adhere to. Each application, without exception, is required to comply with these security standards to ensure a consistent and unified approach to safeguarding the organization's assets and data. It is the responsibility of every development team to integrate these policies into their processes and ensure full compliance throughout the development lifecycle.

2.2.2 UU PDP no 27 Tahun 2022

As highlighted by the Personal Data Protection Act (UU PDP) No. 27 of 2022 is a crucial legal milestone in Indonesia, aimed at safeguarding the confidentiality and integrity of individuals' personal data. This act is intended to ensure that personal information is treated with care and protected against unauthorized access or misuse (Indonesia 2022).

The legislation sets forth a comprehensive structure for the collection, processing, and distribution of personal data. By establishing stringent guidelines, it ensures that organizations are held accountable and comply with strict data protection measures, enhancing overall data security across various sectors.

2.3 Security Maturity

The term "security maturity" describes an organization's security posture in relation to its risk landscape and tolerance levels. Risk scenarios can differ significantly based on the organization's specific environment, as each entity has its own unique security risk culture. Therefore, an organization's maturity is measured by how effectively it implements security controls, processes, and reporting. (Presland 2024)

2.3.1 Security Maturity Framework

Assessing an organization's security maturity is crucial. With the global surge in security threats—both in volume and complexity—it's no longer a question of

if a business attacked, but when and how frequently. To ensure strong prevention, detection, and response capabilities, organizations must adopt a security-first mindset. A mature evaluation model defines a five-level capability maturity classification based on assessments, utilizing process maturity metrics to evaluate both the processes and outputs of the software development lifecycle (Ge et al. 2011). The Systems Security Engineering Capability Maturity Model (SSE-CMM) (ISO/IEC 21827), the Federal Information Security Management Framework (NIST), and the Control Objectives for Information and Related Technologies (COBIT) management guidelines are just a few of the maturity evaluation models that are out there.

2.3.1.1 Systems Security Engineering Capability Maturity Model (SSE-CMM)

The Systems Security Engineering Capability Maturity Model (SSE-CMM) or ISO 21827 framework, along with its appraisal method, is specifically designed to provide engineering organizations with a structured tool for evaluating their security engineering practices. Table 2.1 indicates the levels on the Systems Security Engineering Capability Maturity Model (SSE-CMM). This model allows organizations to systematically assess their current security processes, identify gaps or inefficiencies, and define targeted areas for improvement. By using the Systems Security Engineering Capability Maturity Model (SSE-CMM), organizations can gain valuable insights into the maturity of their security practices, enabling them to implement necessary enhancements, strengthen their overall security posture, and align more effectively with industry standards and best practices. The model also serves as a guide for continuous process improvement, helping organizations evolve their security capabilities over time (International Organization for Standardization 2008).

Table 2.1: Levels in Systems Security Engineering Capability Maturity Model (SSE-CMM)

Maturity Level	Description
Level 1	Base practices are performed informally
Level 2	Base practices are planned and tracked
Level 3	Base practices are well defined
Level 4	Base practices are quantitatively controlled
Level 5	Base practices are continuously improving

According to (Almuhammadi & Alsaleh 2017) The NIST Cybersecurity Framework (NIST CSF) is a flexible and scalable guide designed to help organizations manage cybersecurity risks. It focuses on high-level information security requirements, ensuring that essential security functions like data protection, threat detection, and incident response are addressed comprehensively. This framework can be shown on the Table 2.2 is especially valuable in developing robust information security programs and policies, offering a structured approach through its five core functions: Identify, Protect, Detect, Respond, and Recover (National Institute of Standards and Technology (NIST) 2014). Organizations can tailor these guidelines to their specific needs, creating effective security policies that align with their operational and risk environments, while strengthening their overall cybersecurity posture.

Table 2.2: Levels in NIST Cybersecurity Framework (NIST CSF)

Maturity Level	Description
Level 1: Policies	Formal documented and updated policies are communicated to all employees. Policies establish a continuous risk assessment and implementation and cover major facilities and operations. Clearly assign roles, responsibilities, and measures.
Level 2: Procedures	Formal documented and updated policies cover all security controls. Procedures clearly identify where, how, when, when and on what the procedure to be performed. Procedures document controls implementation.
Level 3: Implementation	Procedures are communicated to respective individuals and controls are implemented in consistent manner. Initial testing is performed.
Level 4: Test	Adequate tests are routinely performed to ensure that all policies, procedures, and controls are acting as intended. Effective corrective actions, self-assessments, and independent audits are performed.
Level 5: Integration	Policies, procedures, implementations, and tests are continually reviewed and improved. IT security program is embedded into culture and an integrated practice. Continuous cost-benefit analysis is performed.

2.3.1.3 Control Objectives for Information and Related Technologies (COBIT)

COBIT is a comprehensive IT governance and management framework that addresses high-level strategic decisions in IT management without delving into technical specifics. It provides a collection of best practices for managing IT resources, infrastructure, processes, responsibilities, and controls. COBIT includes 34 IT processes, each with overarching control objectives (COs) and a detailed set of control objectives (DCOs), totaling 318 DCOs across these processes (R et al. 2016). It

is an effective solution for managers seeking an integrated framework that operates cohesively on its own, without the need to be combined with other IT governance frameworks Table 2.3 shown the Levels in Cobit framework.

Table 2.3: Levels in Control Objectives for Information and Related Technologies (COBIT)

Maturity Level	Description
Level 0 – Non-existent	The organization does not recognize the need for IT security. There is a complete lack of a recognizable system security administration process.
Level 1 – Initial / Ad hoc	The organisation recognizes the need for IT security. But the organization considers IT risks in an ad hoc manner, without following defined processes or policies.
Level 2 – Repeatable but intuitive	Responsibilities and accountabilities for IT security are assigned to an IT security coordinator. There is an emerging understanding that IT risks are important and need to be considered. Some approach to risk assessment exists, but the process is still immature and developing.
Level 3 – Defined process	Security awareness exists and is promoted by management. An organization-wide risk management policy defines when and how to conduct risk assessments. Risk assessment follows a defined process that is documented and available to all staff through training.
Level 4 – Managed and measurable	Responsibilities for IT security are clearly assigned, managed and enforced. The assessment of risk is a standard procedure and exceptions to following the procedure would be noticed by IT management.
Level 5 – Optimized	IT security is a joint responsibility of business and IT management and is integrated with corporate security business objectives. Risk assessment has developed to the stage where a structured, organisation-wide process is enforced, followed regularly and managed well.

2.4 Information Security Principles

Information security principles serve as essential guidelines and best practices for organizations to safeguard their information from unauthorized access, alteration, or destruction, as highlighted by Devopedia (2020). As shown on Figure 2.1 CIA triad is an information security model guiding organizations in identifying vulnerabilities and maintaining robust security policies. These principles encompass the tools and processes designed to uphold the confidentiality, integrity, and availability of information—collectively known as the CIA triad.

2.4.1 Confidentiality

The term "confidentiality" originates from the Latin verb *confidere*, meaning "to have full trust or reliance." It is a fundamental principle of information security, rooted

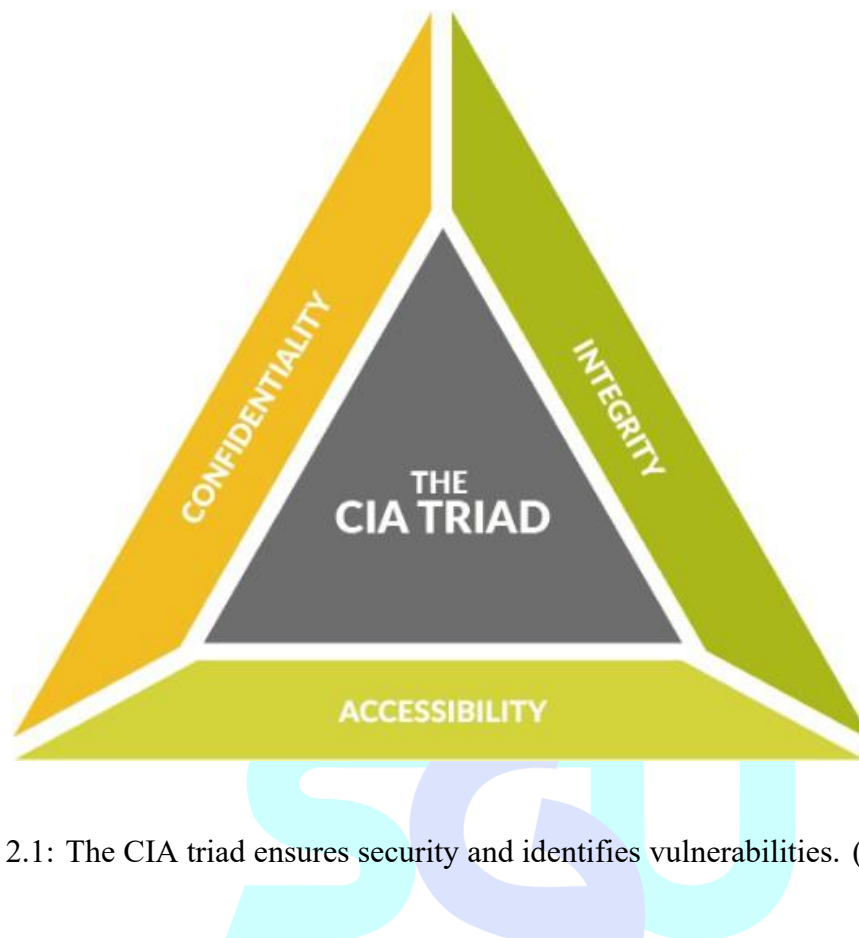


Figure 2.1: The CIA triad ensures security and identifies vulnerabilities. (Threatscape 2024)

in the military's approach of maintaining strict authority and control. This principle ensures that access to information is granted only on a need-to-know basis, preserving its secrecy and trustworthiness (Samonas & Coss 2014).

As mentioned on the Chai & Zolkipli (2021) confidentiality refers to the safeguarding of information to ensure it remains inaccessible to unauthorized individuals, preventing data breaches and unauthorized disclosure. This principle is vital for maintaining trust, protecting sensitive information, and upholding the security of systems and processes.

2.4.2 Integrity

Integrity, on the other hand, is to ensure robust protection against unauthorized modifications to information, as maintaining data integrity is just as critical as preventing unauthorized access. Even if an adversary is unable to view or access the

data directly, they can still pose a significant threat by corrupting it or making subtle, selective changes (Devopedia 2020). Such alterations can compromise the reliability and accuracy of the information, leading to misinformation, operational disruptions, or long-term damage. Addressing this risk is essential to preserve the trustworthiness of data and to prevent malicious actors from exploiting altered information for harmful purposes later on.

2.4.3 Integrity

Ensure protection against denial of access to critical information, as this is a key aspect of maintaining system availability. Even if an adversary cannot access or alter your data, they may attempt to block your ability to access or utilize it effectively (Devopedia 2020). Such attacks can take various forms, including disrupting communication channels, overloading network resources, or shutting down data servers entirely. These actions can lead to significant operational disruptions, impede decision-making, and even cripple essential services. Implementing strong safeguards, such as redundancy, robust incident response protocols, and traffic management systems, is essential to counter these threats and ensure continuous access to vital information.

2.5 DevOps

DevOps focuses on promoting cross-functional collaboration between development and operations teams. A key aspect of this approach is the use of semi-automation and full automation in tasks such as building, deployment, and testing, which significantly enhances software delivery performance. However, organizations adopting DevOps practices often encounter difficulties in managing the tension between development teams' goal of shortening development cycles and delivering features more rapidly, and operations teams' emphasis on maintaining stability.

A more pressing concern is the tendency of both teams to overlook security vulnerabilities. As a result, potential weaknesses in the system may go unnoticed, providing opportunities for threat actors to exploit. Balancing the demands of speed, stability, and security remains a key challenge in the successful implementation of DevOps practices (Naidoo & Möller 2022).

2.6 DevSecOps

One of the main drivers behind this shift was the rapid growth in the use of open-source software components, leading to a rise in data breaches and vulnerability exploits. Information security professionals could no longer remain detached from the development process, nor could security validation be postponed until just before production. Security had to be embedded in every stage of development and deployment, transforming from a secondary concern to a shared responsibility across the entire team.

DevSecOps is essential for the smooth integration of security considerations into all areas of DevOps practices. DevOps encompasses the principles and practices that integrate software development (Dev) and information technology operations (Ops) across the whole product lifecycle (Bitra & Achanta 2021). DevSecOps integration entails the automated creation of security and compliance artifacts at every stage of the development process (Watrobski et al. 2021). The importance of this technique is emphasized by numerous crucial factors:

1. Streamlined Vulnerability Mitigation: DevSecOps significantly reduces vulnerabilities, malicious code, and other security issues in software that has been released. Crucially, it accomplishes this without hindering the speed at which code is produced and released. DevSecOps entails the early integration of security measures into the software development lifecycle to mitigate vulnerabilities in the product and decrease the expenses associated with correcting them (Anjaria & Kulkarni 2021).
2. Thorough Mitigation: DevSecOps mitigates the potential effect of vulnerability exploitation at multiple phases by addressing vulnerabilities across the application lifecycle, including code development and program execution on dynamic hosting platforms. DevSecOps aims to seamlessly integrate security throughout the entire software lifecycle. While developers may view security reviews and audits as hindrances to swift development and feature releases, neglecting these issues poses a risk. Failing to address security flaws promptly can lead to exploitation, resulting in more substantial losses in terms of time and economic costs (Floris 2021).

3. Identification and resolution of the underlying cause: DevSecOps aims to address the underlying reasons for vulnerabilities by implementing strategies such as strengthening testing tools and techniques throughout the toolchain. DevSecOps is a holistic and cooperative methodology for software development, guaranteeing that security is an indispensable component of the development process (Ness et al. 2023). Furthermore, it improves coding methodologies and operational protocols to mitigate the reoccurrence of security vulnerabilities.
4. Minimization of Friction: An essential benefit of DevSecOps is its capacity to minimize conflicts across development, operations, and security teams. By adopting this collaborative strategy, the business may effectively maintain the necessary speed and agility to accomplish its purpose, all the while taking use of contemporary and inventive technologies. Within the manufacturing, the technological components employ a continuous integration/continuous delivery (CI/CD) pipeline. The culture and process of DevSecOps are primarily embedded inside the development, management, and stakeholder mindset (Morales et al. 2020). Ensuring a harmonious collaboration between teams is crucial for attaining maximum efficiency and security in the development and deployment procedures. The key elements of the DevSecOps itself is include collaboration, automation, measurement, and monitoring (Moyón et al. 2020).

2.6.1 DevSecOps Culture

According to GitHub (2024) the DevSecOps culture involves the 4 aspect that's People, Process, Technology and Governance. People Refers to the human resources within the company, including employees, teams, and stakeholders, whose skills, expertise, and collaboration drive the success of the organization. In the context of DevSecOps, the focus is on fostering cross-functional teams that include developers, security experts, and operations personnel working together to achieve shared goals.

Processes: These are the structured steps, workflows, and actions that an organization follows to meet its objectives. Processes guide how tasks are executed, from planning and development to testing and deployment. Effective processes ensure that the company's goals are met efficiently, with minimal delays or errors, while integrating security considerations early in the development lifecycle.

Technology: This encompasses all the tools, platforms, and techniques that individuals and teams must utilize to implement the processes effectively. Technology in a DevSecOps framework includes automation tools, code repositories, testing frameworks, and security software. These tools enable continuous integration, delivery, and security checks, streamlining workflows and ensuring that quality and security are maintained throughout the software development lifecycle (Pylayeva 2024).

Governance: In a DevSecOps environment, governance focuses on continuous improvement and requires a culture of ongoing measurement and evaluation. This involves setting up frameworks to track performance, security, and compliance metrics. Governance ensures that all processes are aligned with business objectives and that teams are consistently improving by analyzing data, refining strategies, and making informed decisions that enhance both productivity and security.

2.6.2 “Shift Left” of DevSecOps

One of the foundational principles of the DevSecOps methodology is the shift-left approach. This concept was first applied in the testing phase, where waiting until the end of the development cycle to run tests often created inefficiencies.

By pushing testing earlier in the process, the shift-left principle helps to prevent bottlenecks that can slow down development and lead to increased costs and delays in product release. This proactive approach enhances overall efficiency and reduces risks throughout the software development lifecycle.

2.6.3 Automated Security Testing Assessment

Leveraging automation to identify vulnerabilities at an early stage is crucial in the software development lifecycle. This involves utilizing a variety of tools and methodologies designed for different aspects of security testing. Static Application Security Testing (SAST) analyzes source code and binaries without executing the program, allowing developers to catch vulnerabilities during the coding phase. Dynamic Application Security Testing (DAST), on the other hand, tests the application while it is running, simulating attacks to identify potential weaknesses in real-time (Jammeh 2020).

In addition to SAST and DAST, Software Composition Analysis (SCA) helps

identify vulnerabilities in third-party components and libraries, ensuring that any external dependencies do not introduce security risks. Interactive Application Security Testing (IAST) combines elements of both static and dynamic testing, providing insights while the application is in use, which helps identify vulnerabilities that may not be apparent through other methods. Additionally, Mobile Application Security Testing (MAST) focuses specifically on securing mobile applications, while Runtime Application Self-Protection (RASP) provides real-time security by monitoring the application during execution.

Vulnerability Assessment (VA) rounds out this suite of tools by offering a comprehensive analysis of the system's overall security posture. By employing this range of automated solutions early in the development process, organizations can proactively identify and remediate vulnerabilities, ultimately leading to more secure software and reducing the risk of potential exploits. Conduct manual penetration testing to identify vulnerabilities that may have been overlooked during automated scanning. According to Heilmann (2020) in penetration testing, an application is deliberately attacked with malicious intent to uncover potential security flaws and configuration issues in a fully deployed, production environment. While many organizations perform penetration tests periodically, it is recommended that this practice be carried out continuously rather than in (half-)yearly cycles. Continuous testing ensures that new vulnerabilities are detected and addressed promptly, maintaining a higher level of security over time.

2.6.4 Compliance Automation

Security compliance automation plays a critical role in ensuring that DevSecOps environments consistently adhere to regulatory requirements and industry standards. By automating compliance processes, organizations can integrate security checks throughout the development lifecycle, ensuring that all necessary protocols are followed without slowing down the workflow. This helps maintain compliance with key regulations such as GDPR, HIPAA, PCI-DSS, UU PDP and other industry-specific mandates (Tatinen 2023).

In addition to enforcing these regulations, automation also generates auditable records and reports, making it significantly easier to demonstrate compliance during

audits. These records provide a transparent, traceable history of security checks, configurations, and updates, allowing organizations to quickly respond to auditors' requests for proof of adherence to standards. This reduces the manual effort typically associated with compliance documentation and minimizes the risk of non-compliance, ultimately contributing to a more efficient and secure DevSecOps environment.

2.6.5 Security Monitoring

To effectively address malicious activity, it is essential to build an adaptive defense mechanism centered on comprehensive monitoring. Monitoring is the cornerstone of security incident management, providing continuous visibility into system activities and potential threats. Integrating tools that offer advanced monitoring capabilities is critical for ensuring a proactive security approach (Kumar & Goyal 2020).

Granular logging is a key component of this strategy. It involves capturing detailed logs of system events, which can be invaluable in identifying anomalous behavior or early signs of an attack. By tracking user activities, file changes, network traffic, and system processes in detail, security teams can detect and respond to potential threats more swiftly and accurately. Overall, a robust monitoring system, supported by granular logging, auto-service discovery, and advanced analytics, forms the backbone of an adaptive defense strategy, allowing organizations to detect, analyze, and mitigate security threats in real time.

2.7 Vulnerability Assessment & Penetration Testing

Vulnerability Assessment and Penetration Testing is a systematic process conducted in stages. Vulnerability assessment involves scanning systems, software, or networks to identify weaknesses and potential loopholes. These vulnerabilities could serve as backdoors for attackers to compromise the target. Common vulnerabilities include access control flaws, boundary condition issues, input validation errors, authentication weaknesses, configuration misconfigurations, and exception handling problems (Varma Vegesna 2022).

Penetration testing follows vulnerability assessment as the next critical step. It involves authorized attempts to exploit the identified vulnerabilities to uncover potential risks in the system. Penetration testers operate with permission, intentionally testing

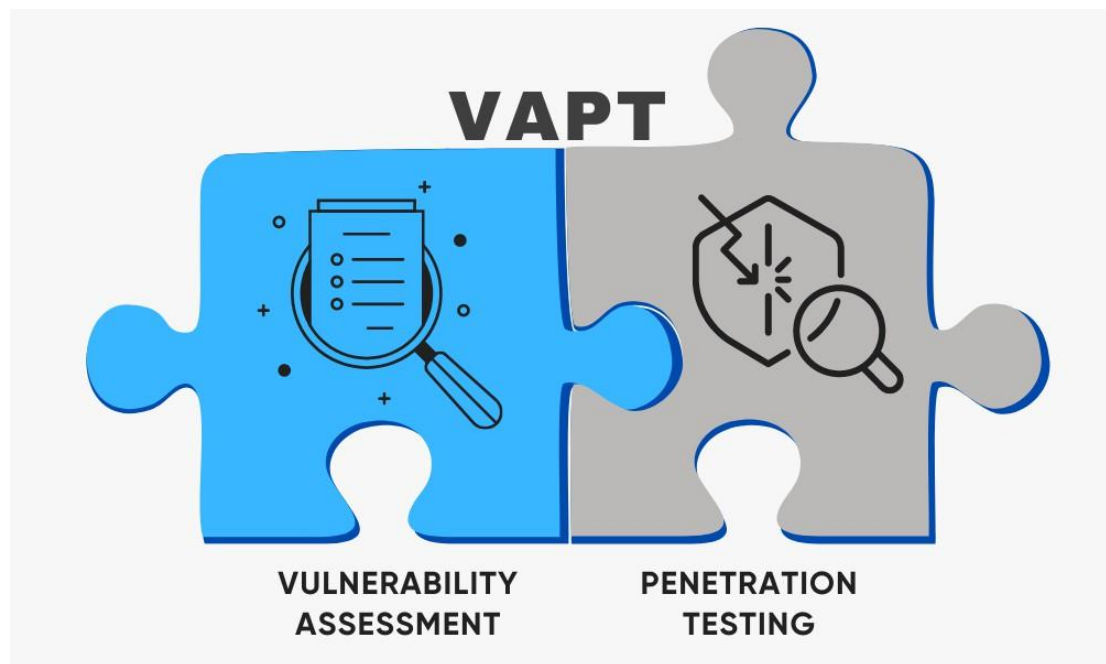


Figure 2.2: VAPT is a crucial practice for protecting against cyber threats. (Kalsariya 2023)

the system's defenses to identify exploitable weaknesses. This process helps assess the system's resilience against potential attacks.

2.7.1 Penetration Testing Strategies

VAPT testers employ three penetration testing strategies, tailored to the scope and type of auditing needed (Shah & Mehre 2024):

- **Black Box Testing:** In this approach, the testers do not have prior knowledge of the target system. They are entirely relying on their experience and expertise to identify vulnerabilities. The primary focus is on auditing the external security boundary of the target, simulating the actions and methods of a real attacker. This attacker would operate from an external location, outside the system's boundary, with no prior information about the target.
- **Gray Box Testing:** Gray Box Testing shares similarities with Black Box Testing. In this approach, the tester is assumed to have partial knowledge of the application's session management. This knowledge aids in assessing whether key features, such as logout and timeout functionalities, are properly secured (Meucci & Muller 2014).

- **White Box Testing:** Takes an opposite approach to Black Box Testing. In this method, testers are given full access to all necessary information such as credentials related to the target system and also the source code. The focus is on evaluating the internal security mechanisms, simulating the behavior of an insider threat, such as a malicious employee operating within the system's boundaries. This approach provides a thorough assessment of internal vulnerabilities and potential risks.

As shown on the Figure 2.2 the VAPT process typically follows a two-step approach: first conducting a Vulnerability Assessment (VA), followed by Penetration Testing (PT). The purpose of starting with VA is to identify potential vulnerabilities in the system through automated scanning and analysis. Once these vulnerabilities are identified, the PT phase is carried out to validate and verify the findings from the VA process. This ensures that the vulnerabilities identified are not false positives and are exploitable in a real-world scenario. By combining VA and PT, the VAPT process delivers more accurate and reliable results, providing a comprehensive understanding of the system's security posture.

Based on the Kalsariya (2023) VAPT should be conducted regularly, especially after significant system changes, to maintain a strong security posture. It is also essential to perform VAPT before deploying new systems or applications to ensure they are secure from the outset. Additionally, after a security incident, VAPT can help identify exploited vulnerabilities and prevent similar attacks in the future.

2.8 Question Development Framework

Researchers can leverage question development frameworks to construct well-structured research questions (Augustus C. Long Health Sciences Library 2024). These frameworks serve as step-by-step guides, helping researchers break down and refine their questions into clear, focused components. By running an initial idea through a question development framework, researchers can sharpen its focus and highlight the key concepts to address. Various frameworks exist, each with unique strengths and limitations, making them suitable for different types of research inquiries.

2.8.1 PICO

PICO represents Patient/Population, Intervention, Comparison, and Outcome. Using a PICO question within a guideline development process inherently emphasizes an experimental or observational epidemiological approach, aligning the inquiry with evidence-based methodologies that prioritize measurable outcomes and systematic analysis (Booth et al. 2019).

2.8.2 PEO

The PEO framework is well-suited for developing qualitative research questions and stands for Population, Exposure, and Outcome(s) (University of Plymouth Library 2024). "Exposure" refers to a risk factor, condition, illness, screening, or service. PEO is particularly useful when investigating potential outcomes associated with a specific condition or exposure, as well as exploring correlations.

2.8.3 SPICE

SPICE is a qualitative research framework used to evaluate the outcomes of interventions or projects. It stands for Setting (study location), Perspective (focus group or stakeholder), Intervention/Exposure/Interest, Comparison, and Evaluation (success measures or specific outcomes) (Augustus C. Long Health Sciences Library 2024).

2.9 Gap Analysis

A gap analysis using COBIT 2019 focuses on identifying the current state of organizational IT governance and management processes and comparing them against the framework's objectives to highlight areas for improvement. COBIT 2019 is structured around five key domains:

- EDM (Evaluate, Direct, Monitor): Ensures that IT processes align with strategic goals, enabling effective decision-making and performance oversight. The analysis may reveal gaps such as misaligned IT strategies or unclear performance metrics.
- APO (Align, Plan, Organize): Focuses on aligning IT capabilities with organizational objectives through planning, risk management, and resource

optimization. Common gaps in this domain include undefined security policies, insufficient risk assessments, or a lack of resource planning.

- BAI (Build, Acquire, Implement): Covers the development and deployment of IT solutions to meet organizational needs. This domain often identifies gaps like inadequate testing, poorly implemented processes, or inconsistent delivery of solutions.
- DSS (Deliver, Service, Support): Addresses operational processes to ensure the delivery of reliable services and efficient support. Gaps here might include weak incident management, inadequate service delivery, or lack of user support mechanisms.
- MEA (Monitor, Evaluate, Assess): Focuses on evaluating performance, compliance, and assurance. Identified gaps may involve insufficient monitoring tools, lack of compliance tracking, or poorly defined metrics.

COBIT 2019, developed by ISACA, is a comprehensive framework designed for the governance and management of enterprise IT. It offers a collection of guidelines, best practices, and tools to help organizations effectively manage and govern their information and technology (IT) resources (ISACA 2012). The purpose of this framework is to define a clear set of governance and management objectives that guide the development, implementation, and monitoring of IT processes. By aligning these processes with business goals and strategies, COBIT 2019 facilitates continuous performance monitoring and improvement. Structured around 40 governance and management objectives, the framework covers end-to-end business and IT functions, providing a detailed roadmap for establishing control and enhancing the effectiveness of IT operations.

By addressing the identified gaps and aligning IT governance and management practices with COBIT 2019's framework, the organization ensures that business objectives and IT security-related goals are harmonized. This alignment fosters a cohesive strategy where IT initiatives support overarching business priorities, enhancing operational efficiency and overall organizational success.

2.10 Framework Maturity Comparison

The security maturity framework that used on this thesis is one from the Systems Security Engineering Capability Maturity Model (SSE-CMM), because the model is more related to the software engineering lifecycle and also the model is reviewed every 5 years to makes sure that the framework is still relevant for the related organizations.

Table 2.4 Provides an overview of the maturity models covered in this section.

Table 2.4: Thesis Framework and comparison with other approaches

Maturity Models	Levels	Focuss
SSE-CMM Model	1. Conducted informal design 2. Planned and tracked 3. Well defined 4. Quantitatively controlled 5. Continuous improvement	Focuses on assessing and improving an organization's security engineering practices, emphasizing the integration of security into the systems engineering lifecycle.
COBIT Model	0. Non-existent 1. Initial/ad hoc 2. Repeatable but intuitive 3. Defined process 4. Managed and measurable 5. Optimized	IT governance and management, offering a set of best practices to ensure IT processes align with business goals, while addressing security, risk management, and compliance.
NIST Model	1. Policy 2. Procedure 3. Implementation 4. Testing 5. Integration	Provides detailed security and risk management guidelines, which help organizations manage cybersecurity risks through a set of standards, guidelines, and practices.

In summary, SSE-CMM is focused on improving security engineering processes, COBIT emphasizes IT governance and alignment with business goals, and NIST provides comprehensive cybersecurity risk management practices. Each model offers a unique approach to enhancing security and maturity within an organization. One of the critical steps toward achieving a high level of security maturity is making security a top priority for every application developed, while also integrating automation for security controls across the entire development process. This is where DevSecOps becomes essential, as it focuses on embedding security practices from the very beginning of the software development lifecycle. By incorporating security from the planning phase, DevSecOps ensures that security is not an afterthought but a fundamental part of the process. So, for the measurement of the security maturity in this thesis be using the

Systems Security Engineering Capability Maturity Model (SSE-CMM).

2.11 Related Works

In the "Self-service cybersecurity monitoring as enabler for DevSecops", (Díaz et al. 2019) emphasizes the importance of the monitoring on the IoT systems with self-service features. The author created comprehensive report from the case studies on the implementation of DevSecOps culture on IoT systems, the pros are the paper provides the real study case and evidence and a complete report from the deploying step until the monitoring step. Leveraging virtualization and containerization technologies enables organizations to implement a self-service monitoring approach effectively. These technologies provide a scalable and flexible environment where the authors can deploy, manage, and monitor authors applications and services without relying heavily on centralized IT resources. Virtualization abstracts the underlying hardware, allowing for the creation of isolated virtual environments, while containerization offers lightweight and portable runtime environments that ensure consistency across development and production. Together, authors empower independently monitor resource utilization, application performance, and potential issues, fostering a proactive and efficient operational model. This self-service capability accelerates response times, enhances autonomy, and aligns with modern DevSecOps practices.

The author only uses one case study deployed at the Universidad Politécnica de Madrid, the report limited to one case study and can't be compared by another case study that implements the same system. Also, on the security monitoring the authors do not explain do the attack is legitimate or just false-positive. The authors only mentioned that there are a number of unique attacks received.

In the field of transforming the DevOps to DevSecOps, (Zhou et al. 2023) the author emphasizes that DevSecOps represents an evolved approach to integrating security as a shared responsibility within the entire DevOps process. It seeks to embed security seamlessly into every phase of the software development lifecycle, ensuring that security considerations are not treated as an afterthought but are prioritized from the very beginning. By blending development, operations, and security teams, DevSecOps fosters a collaborative culture where all stakeholders share accountability for building secure and reliable software. Although some companies have adopted this approach,

both industry and academia are still working towards a common understanding of the DevSecOps movement. This thesis also utilize virtualization and containerization for security scanning tools to enhance their effectiveness. This approach eliminates the need to install each security scanning tool individually, streamlining the process and improving efficiency.

While Wambua (2023) focus is on implementing DevSecOps specifically for mobile applications by embedding security practices from the outset. This involves leveraging tools and methodologies such as Dynamic Application Security Testing (DAST), Static Application Security Testing (SAST), and penetration testing. The process emphasizes continuous code refinement and the ongoing enhancement of security measures to ensure robust protection throughout the application lifecycle. The paper of Sandu (2021) primarily focuses on enhancing resilience through resilience testing, rather than emphasizing comprehensive security assessments. While certain steps include vulnerability testing, these are relatively shallow and do not involve in-depth methodologies like penetration testing. This limits the scope of the research in addressing deeper security vulnerabilities, leaving critical aspects of a thorough security evaluation unexplored. These two papers explores various scenarios for implementing DevSecOps, highlighting different areas of focus. While one scenario emphasizes conducting thorough security assessments and reducing vulnerabilities, another prioritizes resilience testing to ensure the application remains stable and operational in production environments. This approach aims to prevent downtime and maintain reliability, even when faced with unexpected challenges or system failures. This thesis primarily focuses on the security perspective rather than delving deeply into resilience. However, tests include scenarios such as Distributed Denial of Service (DDoS) attacks, along with load testing, to assess system performance under stress.

The focus of de Vicente Mohino et al. (2019) on other hand is to evolve the traditional Software Development Life Cycle (SDLC) framework into a Secure SDLC (S-SDLC), ensuring that security is integrated from the application's inception to its end-of-life. A key strength of this approach is that it covers security at both the technical and managerial levels. The approach encompasses essential managerial activities, including kickoff meetings, periodic review sessions, and collaborative planning, to ensure all stakeholders remain aligned on clearly defined security objectives. These

meetings provide a platform to establish priorities, address potential concerns, and maintain continuous communication between development, operations, and security teams. Additionally, it emphasizes the integration of key security practices, such as Dynamic Application Security Testing (DAST), Static Application Security Testing (SAST), and penetration testing, to identify and address vulnerabilities throughout the development process. By combining structured managerial oversight with robust technical practices, this holistic strategy ensures that security considerations are embedded at every stage of the development lifecycle. This proactive and comprehensive approach not only safeguards the application against potential threats but also fosters a culture of security-first thinking within the team, contributing to the delivery of resilient, secure, and high-quality software. Focusing on integrating security into application development is crucial to ensure that security measures are adopted from the early stages of the development lifecycle. This proactive approach helps build more robust and secure applications.

This research adopt a security assessment approach that automates various stages, including Image scanning, secret scanning, dependency scanning, and both Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST), while implementing a "shift left" DevSecOps strategy. The DAST results be validated through Vulnerability Assessment and Penetration Testing (VAPT) to confirm that any identified vulnerabilities are not false positives, and approach to extends monitoring capabilities by adopting a multifaceted perspective. Instead of solely relying on SIEM (Security Information and Event Management) systems, it also incorporates monitoring through threat intelligence. This combination provides a broader and more proactive security posture, enabling the identification and mitigation of potential threats based on real-time intelligence and comprehensive system insights. Additionally, the security assessment be paired with a maturity evaluation using the Systems Security Engineering Capability Maturity Model (SSE-CMM) to ensure that each product developed by PT XYZ aligns with organizational policies and UU PDP requirements, aiming for a minimum maturity score of 2, which indicates a well-defined level. This involve gathering necessary evidence from the product owner for each maturity assessment question. Furthermore, after the product is released, continuous monitoring be implemented to ensure ongoing security and compliance.

This comprehensive approach ensures that all products meet required standards and minimizes vulnerabilities that could be exploited in production.

Table 2.5 summarizes the comparison of this research framework and other related works.

Table 2.5: Thesis Framework and comparison with other approaches

Key Items	Main Focus	Strength	Limitations
Díaz et al. (2019)	Cybersecurity monitoring for IoT systems with self-service features.	Using virtualization and containerization technology to adopt the self-service monitoring	Focussing on the monitoring only and not measure the maturity level.
Zhou et al. (2023)	Transforming DevOps Into DevSecOps on Industry	The authors classify the interpretations of DevSecOps into three core aspects of DevSecOps capabilities, cultural enablers, and technological enablers.	Only do the research from the perspective of the researcher and not implementor of DevSecOps, also not measure the security maturity of the products.
Wambua (2023)	Creating a mobile application development framework that embeds security from the start.	Integrating security from the start while continuously refining code and improving security measures.	Only cover the mobile application, and not including the backend of the mobile application on the development lifecycle and not covering the mobile application security maturity on the process.
Sandu (2021)	Focuses on integrating the security for resilience enhancement.	Primarily focus on the resilience enhancement.	Deep on the resilience testing only and not security testing or the security maturity.
de Vicente Mohino et al. (2019)	Evolving SDLC to S-SDLC to secure the development of an application from its origin to the end of its life cycle	Integrates the security at both application and managerial levels	The framework applies Agile principles to manage scaling challenges through coordinated multi-team efforts and also not covering the security maturity on the application on the life cycle.
This Thesis	Focuses on integrating the security assessment with security maturity level	Identify the security maturity level to the application before going to production	Focusing only on the security assessment and the security maturity to make sure that the application is comply with the UU PDP.