

Chapter 4

RESULTS AND DISCUSSIONS

4.1 GAP Analysis

For the gap analysis, we utilized Focus Group Discussions (FGDs) to engage stakeholders in exploring the organization's goals, including overarching business objectives and specific security targets. These discussions also examined the organization's current state, particularly in relation to its CI/CD practices, and aligned these findings with the COBIT 2019 framework. The purpose of this analysis was to identify discrepancies between existing CI/CD processes and the desired state of the company goals. By doing so, we aimed to uncover gaps and opportunities for improvement, ensuring that the organization's CI/CD practices align with both security goals and industry standards and adopt the DevSecOps.

4.1.1 FGD Subject Discussion

The Focus Group Discussion (FGD) center on several key topics aimed at aligning organizational goals with security objectives and identifying actionable improvements. The main discussion points are as follows:

- **Understanding Organizational and Security Goals:** The FGD explore the overarching objectives of the organization alongside its specific security goals. This includes a thorough discussion of the organization's mission, vision, and priorities, as well as its current security strategies and how they integrate with broader business objectives. Additionally, participants assess the organization's current state to establish a clear baseline for improvement.
- **Identifying Strengths and Weaknesses of the Current CI/CD Implementation:** Participants evaluate the current CI/CD practices to identify areas where the organization excels and areas that need improvement. This involves examining workflows, tools, processes, and the overall alignment of CI/CD with industry

standards and security principles. Understanding these strengths and weaknesses provides a foundation for targeted discussions about enhancement opportunities.

- **Proposing Improvements Based on Identified Weaknesses:** Based on the shortcomings highlighted during the discussions, the group brainstorm and propose specific improvements to address these gaps. This step focuses on actionable solutions, such as adopting new tools, refining workflows, enhancing security measures, or incorporating best practices to strengthen the CI/CD pipeline. The goal is to develop a set of practical recommendations that align with the organization’s security and operational objectives.

The framework employed to shape the FGD questions is the PEO methodology as shown on the Table 4.1, which is particularly effective in qualitative research. This approach centers on three core concepts—Population, Exposure, and Outcome—enabling a structured exploration of the subject matter.

Table 4.1: PEO - Question Formation Frameworks [®]

PEO	Definition	Example
Population	Who is my question focused on?	Security Engineer and DevOps Engineer that worked on PT XYZ.
Exposure	What is the phenomenon or experience being studied?	Security maturity that involved on the DevSecOps.
Outcome	What are the expected outcomes or areas of interest?	Improved security maturity and the challenges in implementation.

Through these structured discussions, the FGD aims to provide a comprehensive analysis of the current state, foster collaborative problem-solving, and lay the groundwork for implementing improvements that enhance both security and efficiency within the CI/CD processes.

4.1.2 FGD Participants

For this Focus Group Discussion (FGD), I consulted four participants to thoroughly explore the organization’s objectives, assess the strengths and weaknesses of the current CI/CD processes, and discuss potential improvements. This collaborative approach

helps ensure that insights gathered are both practical and aligned with the organization's strategic goals. The following individuals were selected to participate in the FGD:

1. Participant A

- Role: Senior DevOps Engineer at PT XYZ.
- Relevant Criteria: As the current Senior DevOps Engineer at PT XYZ, User A is responsible for maintaining and overseeing the organization's CI/CD pipeline, ensuring its stability and efficiency

2. Participant B

- Role: Project Management Officer, PT XYZ.
- Relevant Criteria: As the project manager at PT XYZ who specializes in UU PDP initiatives and is part of the security platform team, User B frequently coordinates and communicates security matters to product owners and also the participant is on Satgas Delta Force per SK. 19 /PS 330/COP-J0000000/2024 of PT. XYZ as shown on the Appendix B.

3. Participant C

- Role: Blue Team, PT XYZ.
- Relevant Criteria: Serving as the blue team member at PT XYZ, User C is tasked with implementing defensive measures—such as WAF, XDR, and SIEM—and also participates in the UU PDP task force, contributing to the creation and review of policies, standards, and guidelines and also the participant is member of Satgas Delta Force per SK. 19 /PS 330/COP-J0000000/2024 of PT. XYZ as shown on the Appendix B.

4. Participant D

- Role: Security Analyst.
- Relevant Criteria: As a member of the blue team at PT XYZ's security platform, User C oversees and reviews the Security Information and Event Management (SIEM) solution and also manages the implementation of both SIEM and WAF solutions.

4.1.3 FGD Results

Table 4.2 presents a comprehensive gap analysis, derived from the FGD findings and the Table 4.3 indicates the action plan that need to be done.



Table 4.2: GAP Analysis Results

COBIT Objective	Current State	Desired State	Gap Identified
EDM01 - Ensured Governance Framework	- No compliance verification against UU PDP is carried out when the application moves into production	- Ensure the application meets UU PDP requirements before going to production (i.e., safeguarding all user data).	- Inadequate or non-existent compliance checks.
DSS05 - Managed Security Services	- The application does not undergo a thorough security assessment. - Data on the application or server is insufficiently protected. - The product owner is unaware that the data they handle contains PII, which requires additional safeguards.	-The application undergoes a comprehensive security assessment. - Data within the application is encrypted and safeguarded. - The product owner can clearly identify PII and implement the necessary measures to secure it.	- Automate and accelerate security assessments to streamline the process. - Implement proactive monitoring to identify potential threats early. - Provide clear documentation on PII data and the necessary protection measures.
MEA01 - Managed Performance and Conformance Monitoring	- Endpoint monitoring licenses are limited. - Some employees and product owner are unaware of the need for monitoring. - Issues arise when installing the monitoring agent on endpoints. - No monitoring is in place to detect credential leaks.	-All endpoints can be monitored continuously, 24/7. - Product owners can install agents on their assets for real-time monitoring. - Dark web monitoring provides insights into potential risks to the assets.	- Provides guidance on implementing agent monitoring for both applications and endpoints. - Includes active monitoring enhanced with threat intelligence integration.

Table 4.3: GAP Analysis Results - Action Plan

COBIT Objective	Action Plan
EDM01 - Ensured Governance Framework	- Create the assessment checklist when application is go to production
DSS05 - Managed Security Services	- Align security assessments and maturity assessment with the deployment process via DevSecOps. - Continuously monitor all assets for potential risks. - Ensure that any application handling PII data has implemented all necessary security protection.
MEA01 - Managed Performance and Conformance Monitoring	- Conduct research on monitoring using open-source tools compatible with all platforms. - Facilitate discussions on installing agents on endpoints.

In keeping with this thesis's objective of enhancing data protection through a DevSecOps approach, the primary emphasis lies in addressing the gaps identified in DSS05—Managed Security Service—while also ensuring EDM01 (Ensured Governance Framework) and continuous monitoring are upheld. Resolving these DSS05 gaps is critical for strengthening data protection within a DevSecOps context. During the discussion, participants confirmed that the identified gaps and proposed actions are aligned with PT XYZ's strategic objectives. Their feedback emphasized the importance of involving management after creating the maturity and security assessments in DevSecOps. By securing management support, the product team can effectively implement the proposed measures, ensuring the application remains secure and compliant with UU PDP. The EDM03 and MEA 03 is not included during this mapping, although relevant to the FGD result because the EDM03 and MEA03 focus more on governance, audits, and high-level compliance tracking, which may not be directly actionable within the DevSecOps framework implementation.

4.2 Framework Creation

This section explain the result of the framework creation and assessing the existing product.

4.2.1 Create Maturity Assessment Checklist

For developing the Maturity Assessment Checklist, we leverage internal policies alongside the relevant Personal Data Protection (PDP) law, ensuring that each component aligns with the core tenets of the CIA triad—Confidentiality, Integrity, and Availability. The checklist is also mapped to ISO 27001 as a benchmark to ensure alignment with PT XYZ’s internal policy. By grounding our checklist in both regulatory requirements and organizational standards, we create a robust, compliance-focused framework that safeguards data and supports continuous improvement in security practices as shown on the figure 4.1.

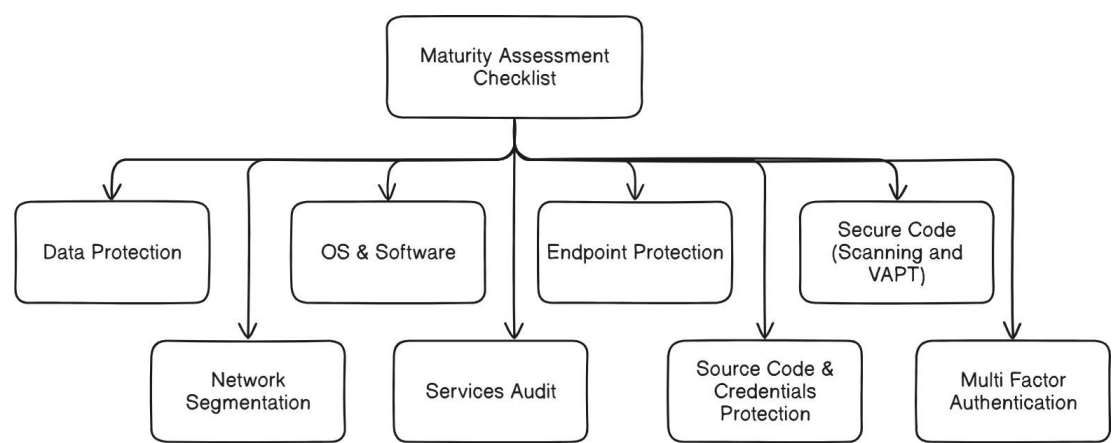


Figure 4.1: Maturity Assessment Checklist

4.2.1.1 Data Protection

In the data protection domain as shown on the Table 4.4, our primary focus centers on both encryption and data backup. We ensure that data is safeguarded through robust, properly selected encryption algorithms, while also designing comprehensive backup processes that detail the appropriate methods, retention periods, schedules, and storage locations. By emphasizing both encryption and backup best practices, we help maintain data security and resilience.

- Data Encryption - Encryption Algorithm: In alignment with recognized industry standards, the recommended encryption approach utilizes the Secure Hash Algorithm (SHA) at a 256-bit strength—commonly referred to as SHA-256 or its Cipher Block Chaining variant (SHA-256-CBC). These specifications are

outlined in the PT. XYZ Encryption Standard version 1.0, which provides detailed guidance on parameters, usage, and best practices. By adhering to these robust cryptographic methods, organizations can ensure that they maintain high levels of confidentiality, integrity, and availability for all critical data assets.

- **Data Encryption - Type of Data Encrypted:** In accordance with the PT. XYZ Data Classification Standard version 1.0, it is crucial to encrypt both personally identifiable information (PII)—such as names, phone numbers, and ID numbers—as well as non-PII data. This comprehensive approach ensures that all information, regardless of its sensitivity level, remains securely protected from unauthorized access or disclosure throughout its lifecycle.
- **Data Encryption - Encryption Implementation:** In addition to determining which data requires encryption, it is equally important to establish the specific layer at which the encryption is applied—whether at the disk level or the table level. This choice can depend on factors such as the data’s sensitivity, regulatory requirements, system architecture, and operational constraints. Disk-level encryption may be simpler to implement for broader coverage, whereas table-level encryption allows more granular control. Ensuring the right encryption layer is crucial for maintaining robust security and optimizing system performance.
- **Backup Data - Backup Testing:** It’s essential to verify the integrity of backup data by performing a restoration test. This process involves attempting to restore the saved data and reviewing the outcome—be it a successful restoration, an error. By testing backups regularly, organizations can confirm that their backup strategy is robust and capable of recovering data accurately when needed.
- **Backup Data - Backup Retention:** It is crucial to determine the retention period for backup data—whether 6 months, 1 year, or even 5 years—before it is disposed of or destroyed. These timelines may differ based on the data category, such as personally identifiable information (PII), transactional data, or non-transactional data. Clearly defining and adhering to retention periods not only ensures compliance with legal and regulatory requirements but also upholds organizational best practices for data governance and security.

- Backup Data - Backup Period: Data backup is typically scheduled at set intervals—hourly, daily, weekly, or monthly—based on the criticality and volatility of the data, ensuring that essential information can be recovered in a timely manner.
- Backup Data - Backup Method: The backup process can be conducted using either a full backup, which copies all data in its entirety, or an incremental backup, which only captures data changes since the most recent backup. This approach can optimize storage requirements and backup efficiency.
- Backup Data - Data and Repo Redundancy: Implementing redundancy for assets that store both source code and data is a critical safeguard against single points of failure. One effective approach is to utilize multiple repository providers—such as hosting code on both GitHub and GitLab—or to maintain replicas across two different cloud service providers like AWS and ABC. By distributing these assets, organizations enhance resiliency, ensuring that if one platform experiences an outage or breach, the other environment remains accessible. This strategy not only mitigates downtime and data loss but also supports overall business continuity and strengthens the organization's security posture.
- Backup Data - Backup Location: Storing backup data in the same location as operational data can undermine the entire backup strategy, since any incident—such as data corruption or a natural disaster—could affect both the primary and backup data. To minimize this risk, keep your backup data in a separate, secure environment that's shielded from threats like floods, fires, and earthquakes. This approach ensures that, in the event of a localized disaster, the backup remains intact and available for recovery.

Table 4.4: Mapping Maturity Assessment Checklist on Data Protection

Domain	Item	Policy	CIA Triad
Data Protection	Data Encryption: Encryption Algorithm	PD: Pasal 15 ISO 27001: A.10.1.1	Confidentiality
	Data Encryption: Type of Data Encrypted	PD: Pasal 15 ISO 27001: A.8.2.3	Confidentiality
	Data Encryption: Encryption Implementation	PD: Pasal 15 ISO 27001: A.8.2.3	Confidentiality
	Backup Data: Backup Testing	PD: Pasal 24 ISO 27001: A.12.3.1	Availability
	Backup Data: Backup Retention	PD: Pasal 24 ISO 27001: A.12.3.1	Availability
	Backup Data: Backup Period	PD: Pasal 24 ISO 27001: A.12.3.1	Availability
	Backup Data: Backup Method	PD: Pasal 24 ISO 27001: A.12.3.1	Availability
	Backup Data: Encrypted Data Backup	PD: Pasal 24 ISO 27001: A.12.3.1	Confidentiality
	Backup Data: Data and Repo Redundancy	PD: Pasal 24 ISO 27001: A.12.3.1	Availability
	Backup Data: Backup Location	PD: Pasal 24 ISO 27001: A.12.3.1	Availability

4.2.1.2 Network Segmentation

The Table 4.5 indicates the maturity assessment checklist on the network segmentation.

- Network segmentation: is implemented for tribes or chapters to manage access to various information sources—such as repositories, financial data, PII data, or other sensitive information—in accordance with PT. XYZ Data Classification Standard version 1.0 or PD 406 of 2018 on information security governance. This strategy helps ensure that each group only accesses the data it needs, thereby enhancing overall security.
- Network Protection System: Various network security tools and mechanisms—such as firewalls, IP whitelisting, Intrusion Detection/Prevention Systems (IDS/IPS), Network Access Control (NAC) solutions like FortiNAC, and VPC peering—are utilized to safeguard the network. These measures help protect against unauthorized access, detect potential threats, and ensure a secure environment for critical data and services.
- Network Segmentation Implementation: Network segmentation is implemented for employees according to their employment level—such as organic, outsourcing, or internship. This may involve assigning distinct IP ranges, device identifiers, or MAC addresses, ensuring that each group has restricted access to only the systems and data appropriate for their role.
- Network Segmentation on Sensitive Information Access: Network segmentation is implemented to restrict access to sensitive data sources—such as financial, customer, and strategic business information—by isolating these critical assets from other network segments. This approach helps reduce the risk of unauthorized access and ensures tighter security measures for high-value data.

Table 4.5: Mapping Maturity Assessment Checklist on Network Segmentation

Domain	Item	Policy	CIA Triad
Network Segmentation	Network Segmentation	PD: Pasal 28 & Pasal 29 ISO 27001: A.9.4.1	Confidentiality
	Network Protection System	PD: Pasal 28 & Pasal 29 ISO 27001: A.9.4.1	Confidentiality
	Network Segmentation Implementation	PD: Pasal 28 & Pasal 29 ISO 27001: A.9.4.1	Confidentiality
	Network Segmentation on Sensitive Information Access	PD: Pasal 28 & Pasal 29 ISO 27001: A.9.4.1	Confidentiality

4.2.1.3 OS and Software

Table 4.6 shown that item need to be done on the OS and Software domain on the maturity assessment.

- Periodic Updates to Tools that Support Operational Activities: Regular updates are applied to both operational tools—such as operating systems (OS), SonarQube, MySQL, Jenkins, JetBrains products, Android Studio, and Node.js—and productivity tools like Microsoft Office, Confluence, and Trello. Keeping software current helps ensure optimal performance, closes security vulnerabilities, and supports the reliability of the overall environment.
- Rollback Mechanism: There is a rollback mechanism in place to revert to a stable version if an update to any tool fails. This approach helps mitigate disruptions,

prevents extended downtime, and ensures continuity of operations by quickly restoring normal functionality.

- **Access Restriction with Local Policy Implementation:** There is a policy that restricts access to the command prompt or terminal, registry, and root folders, and it also enforces a separation of administrative and user privileges. This policy is applied to endpoints both locally—covering hardware USB usage or software constraints—and centrally through Active Directory.

Table 4.6: Mapping Maturity Assessment Checklist on OS & Software

Domain	Item	Policy	CIA Triad
OS & Software	Periodic Updates to Tools that Support Operational Activities	PD: Pasal 22 ISO 27001: A.14.2.1	Confidentiality
	Rollback Mechanism	PD: Pasal 22 ISO 27001: A.14.2.4	Availability
	Archiving Developed Applications.	PD: Pasal 22	Availability

4.2.1.4 Services Audit

On the services audit is more focussing on the logging on the application, database or the system, the details of the services can be shown on the Table 4.7.

- **System, Application, Database Logging:** There is a mechanism in place to log activities across systems, applications, and databases, ensuring that each action is recorded for monitoring, auditing, and accountability purposes.
- **Minimum Items to be Logged (Logging):** The minimum information captured in activity logs includes the user ID or user account, the specific activities performed in the system, the date, time, and details of the event, and the device identity (e.g., laptop name, MAC address, location, operating system, IP address, and protocol used).

- **Event Logging:** The recorded events encompass both successful and unsuccessful attempts to access the system or database, configuration changes to the system, activities involving privilege accounts or privileged access, files accessed and the types of files accessed (including any critical file deletion actions), operations that activate or deactivate security systems (such as antivirus, IDS, IPS, or Windows Defender), and any creation, modification, or deletion of specific account identities.

Table 4.7: Mapping Maturity Assessment Checklist on Services Audit

Domain	Item	Policy	CIA Triad
Services Audit	System, Application, Database Logging	PD: Pasal 25 ISO 27001: A.12.4.1	Integrity
	Minimum Items to be Logged (Logging)	PD: Pasal 25 ISO 27001: A.12.4.2	Integrity
	Event Logging	PD: Pasal 25 ISO 27001: A.12.4.1	Integrity

4.2.1.5 Endpoint Protection

Endpoint protection is to make sure that the endpoint from the users is well-protected including the installation of Antivirus, EDR and the storage encryption as shown on the Table 4.8.

- **Reporting of lost laptops:** There is a designated reporting channel for lost laptop assets, ensuring that such incidents can be promptly reported and properly documented.
- **Active auto-update:** Automatic updates are applied to active user endpoints—covering operating systems, software, and any operational tools—to help mitigate risks and maintain a secure, up-to-date environment.

- Installation/activation of Antivirus, EDR on Each Endpoint: Antivirus and malware detection solutions are utilized to protect against malicious scripts or files, helping identify, quarantine, or remove harmful content before it can compromise systems or data.
- Password Protection: Ensure the use of strong, complex passwords across all accounts, while also considering dual authentication methods (e.g., tokens or fingerprint recognition) to further enhance security and protect against unauthorized access.
- Use of Storage Encryption: Storage encryption is enforced at the storage level through tools such as BitLocker, which encrypt the entire drive and help prevent unauthorized data access.
- Access Restriction with Local Policy Implementation: There is a policy that restricts access to the CMD/Terminal, Registry, and Root Folders, and enforces a segregation of duties between Admin and User roles. This policy is implemented on endpoints both at the local level (covering hardware USB and software) and centrally through Active Directory.
- Limiting Browsing Access: There is a restriction in place that prevents browsing to websites deemed unrelated to productivity, ensuring that browsing activities remain focused on work-related purposes.

Table 4.8: Mapping Maturity Assessment Checklist on Endpoint Protection

Domain	Item	Policy	CIA Triad
Endpoint Protection	Reporting of lost laptops	PD: Pasal 23 ISO 27001: A.8.2.3	Confidentiality
	Active auto-update	PD: Pasal 23	Integrity
	Installation/activation of Antivirus, EDR on Each Endpoint	PD: Pasal 23	Confidentiality
	Password Protection	PD: Pasal 23	Confidentiality
	Use of Storage Encryption	PD: Pasal 23 ISO 27001: A.10.1.1	Confidentiality
	Access Restriction with Local Policy Implementation	PD: Pasal 23	Confidentiality
	Limiting Browsing Access	PD: Pasal 23	Confidentiality

4.2.1.6 Source Code and Credentials Protection

On this domain, the main focus is to make sure that the source code and the credential is protected and restricted and can be accessed by the user that has been assigned from the admin, along with documentation related to UAM and UAR. The details of the each checklist can be shown on the Table 4.9.

- Segregation of Duties: Access is restricted or granted based on the assigned authority or role—covering repositories, resource projects, databases, or object storage—to ensure that only authorized individuals have the appropriate level of permissions.
- Data Encryption at Rest: Data is encrypted in storage at both the table level and the disk level. For disk-level encryption, this applies not only to servers but also to workstations or endpoints. This approach ensures that data remains protected, regardless of where it resides.

- **Data Encryption in Transit:** Encryption—such as TLS v2 or SSL v3—is applied whenever data is transmitted, in alignment with the PT. XYZ Encryption Standard version 1.0. This approach helps maintain confidentiality and integrity during data exchange.
- **User Access Matrix (UAM) Documentation:** User access roles to systems, applications, or databases are documented according to the relevant roles and authorities. By keeping a clear, up-to-date record of permissions, organizations can ensure that users only have access rights appropriate to their responsibilities.
- **User Access Review (UAR) Documentation:** User access roles are periodically reviewed for systems, applications, and databases to ensure that they remain current and reflect each individual’s responsibilities. This process helps maintain appropriate permissions, reduces security risks, and ensures that user privileges align with evolving operational requirements.

Table 4.9: Mapping Maturity Assessment Checklist on Source Code & Credentials Protection

Domain	Item	Policy	CIA Triad
Source Code & Credentials Protection	Segregation of Duties	PD: Pasal 15 ISO 27001: A.6.1.1	Confidentiality
	Data Encryption at Rest	PD: Pasal 15	Confidentiality
	Data Encryption in Transit	PD: Pasal 15	Confidentiality
	User Access Matrix (UAM) Documentation	PD: Pasal 15 ISO 27001: A.9.2	Integrity
	User Access Review (UAR) Documentation	PD: Pasal 15 ISO 27001: A.9.2	Integrity

4.2.1.7 Secure Code (Scanning and VAPT)

Secure Code domain is to make sure that the application is already assessed by the security assessment including VAPT as shown on the Table 4.10

Table 4.10: Mapping Maturity Assessment Checklist on Secure Code (Scanning and VAPT)

Domain	Item	Policy	CIA Triad
Secure Code (Scanning and VAPT)	Follow The Security Assessment on The DevSecOps Pipeline	PD: Pasal 27 ISO 27001: A.14.2.6	Confidentiality

- Follow The Security Assessment on The DevSecOps Pipeline: Available security assessments—such as Docker Scanning, Dependency Check Scanning, SAST, DAST, and VAPT—are conducted to identify and address potential vulnerabilities, ensuring a robust and secure environment throughout the development and deployment processes.

4.2.1.8 Multi Factor Authentication

Table 4.11 checklist is to make sure that the application is enabling the MFA on the critical assets to protect against unauthorized access or the credentials stealer.

- Multi Factor Authentication: Multi-Factor Authentication (MFA) is implemented wherever possible for email services, Virtual Private Networks (VPNs), and other access points, with particular emphasis on securing critical assets. This additional layer of security helps protect against unauthorized access by requiring multiple forms of verification.

Table 4.11: Mapping Maturity Assessment Checklist on Multi Factor Authentication

Domain	Item	Internal Policy	CIA Triad
Multi Factor Authentication	Multi Factor Authentication	PD: Pasal 28 & Pasal 29 ISO 27001: A.9.4.1	Confidentiality

4.3 Assess Existing Product

This step involves conducting a thorough evaluation of the existing product using the maturity assessment checklist we have developed, then reviewing the findings to

gain clear insights into the product’s performance and compliance levels. In addition, it serves as a critical opportunity to gather detailed information about the product and identify any challenges it may be facing. By consolidating these insights, we can pinpoint areas for improvement and devise a focused strategy to enhance the product’s overall maturity and resilience.

4.3.1 Product A

Product A serves as the digital transformation of a key government service, executed by PT XYZ. It incorporates personally identifiable information (PII) and is utilized extensively by the public, underscoring the need for stringent security measures to protect sensitive data and ensure compliance with relevant regulations.

4.3.1.1 First Maturity Assessment

The first assessment we measure with the maturity assessment the product A, as shown on the Table 4.12.

Table 4.12: Product A First Maturity Assessment

No	Domain	Maturity Score	Target	Max Score
1	DATA PROTECTION	1.75	2	5
2	NETWORK SEGMENTATION	2	2	5
3	OS and SOFTWARE	2	2	5
4	SERVICES AUDIT	0	2	5
5	ENDPOINT PROTECTION	0.8	2	5
6	SOURCE CODE and CREDENTIALS	1.20	2	5
7	SECURE CODE	1	2	5
8	MFA	0	2	5

Product owner mentioned that in the past three months, Product A has received a credentials leak report from BSSN, which was verified and confirmed to affect its assets. Recognizing the gravity of this incident, Product A’s team has prioritized remediation

efforts and plans to implement stronger preventive measures to avoid similar breaches in the future.

In light of Product A’s current challenges, the recent credential leaks, the security team advises focusing on key areas to strengthen overall protection. These areas include enhancing network segmentation, conducting a thorough audit of services, implementing robust endpoint protection, adhering to secure coding practices, and enabling multi-factor authentication (MFA). By prioritizing these measures, Product A can more effectively mitigate risks and reinforce its security posture.

4.3.2 Product B

Meanwhile, Product B, developed under PT XYZ, serves as a marketplace application specifically designed for APIs, connecting sellers and buyers who want to trade and utilize various API services. The platform offers three subscription models—Free, Prepaid, and Postpaid—giving users flexible options to match their budget and usage requirements.

4.3.2.1 First Maturity Assessment

Meanwhile, Product B has not yet implemented a maturity assessment because it is still focused on the development phase. An overview of Product B’s maturity assessment can be found in the Table4.13

Table 4.13: Product B First Maturity Assessment

No	Domain	Maturity Score	Target	Max Score
1	DATA PROTECTION	0.9	2	5
2	NETWORK SEGMENTATION	0.5	2	5
3	OS and SOFTWARE	1	2	5
4	SERVICES AUDIT	1	2	5
5	ENDPOINT PROTECTION	0	2	5
6	SOURCE CODE and CREDENTIALS	1.2	2	5
7	SECURE CODE	0	2	5
8	MFA	0	2	5

Nevertheless, the product owner remains fully committed to enhancing both the security and overall maturity of Product B. In pursuit of this goal, they plan to implement a robust security assessment process, prioritizing key areas such as secure code, and access controls. By proactively addressing these concerns, the product owner aims to mitigate risks, maintain regulatory adherence, and ultimately deliver a more secure and resilient application for end users.

4.4 Proposed DevSecOps Framework

The proposed DevSecOps framework, as shown in Figure 4.2 is incorporates several key components designed to enhance security, compliance, and operational efficiency. It includes a comprehensive maturity assessment of the product to evaluate its development and security readiness, ensuring alignment with internal policies and the Personal Data Protection (PDP) law. The proposed DevSecOps framework ensures a balance between rapid development, security assessments, and maturity evaluations. The key measure of success is that applications are deployed on schedule while undergoing comprehensive security assessments, effectively minimizing the risk of data breaches.

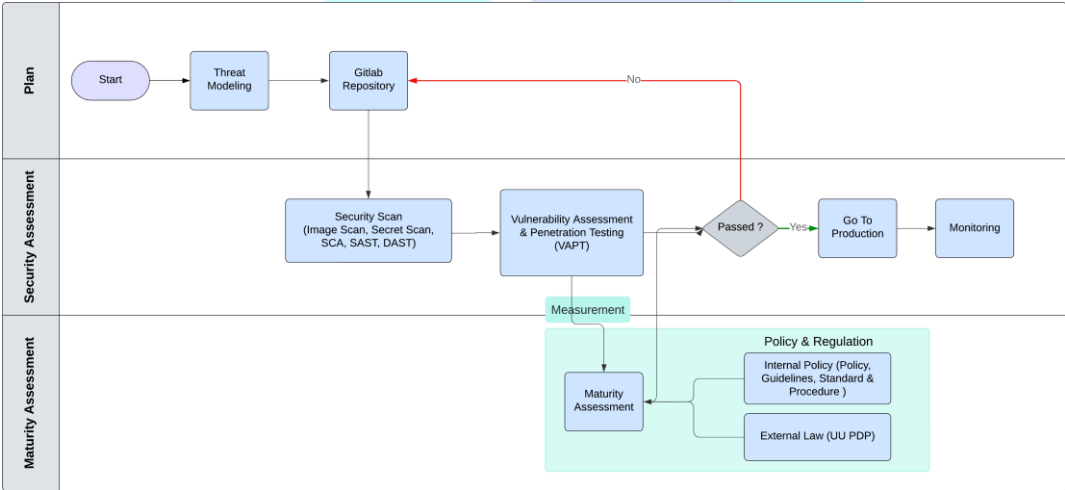


Figure 4.2: Proposed DevSecOps Framework

Additionally, robust security assessments are conducted on applications to identify and address vulnerabilities proactively. Furthermore, the framework integrates active monitoring of red team activities, leveraging threat intelligence applications to detect and analyze potential threats in real time. This is complemented by continuous

monitoring using Endpoint Detection and Response (EDR) solutions, which provide advanced threat detection, investigation, and response capabilities to safeguard the system against evolving cyber risks. These measures collectively ensure that the product meets regulatory standards and internal security requirements while maintaining resilience against cyber threats.

4.4.1 Plan

In the planning stage, the product team conducts threat modeling to pinpoint potential risks and devise countermeasures tailored to the product's specific vulnerabilities. As part of Design for Security (DFW), these measures are identified and integrated to ensure security is embedded into the development lifecycle. Once these safeguards are implemented, the updated code is pushed to the organization's repository, ensuring that security enhancements are properly documented and made available for subsequent development, security review, and deployment.

4.4.2 Security Assessment

In the security assessment phase, the product undergoes multiple scanning processes, including Software Composition Analysis (SCA), image scanning, secret scanning, Dynamic Application Security Testing (DAST), and Static Application Security Testing (SAST). After these scans, the product proceeds to a Vulnerability Assessment and Penetration Testing (VAPT). If the VAPT is successfully passed, and the product is already remediated the security finding that found on the VAPT process, the product moves on to the maturity assessment. On the other hands, there's proactive monitoring to do to make sure that the application is secure on the public functions.

The VAPT process be conducted in six key steps as shown on the Figure 4.3. First, a request for VAPT is submitted, initiating the assessment. Next, the development team conducts a sharing session to provide important technical details and clarify scope. Then, the VAPT itself is performed to identify potential vulnerabilities. Following that, the team compiles a comprehensive report detailing the findings and present the VAPT finding to developer, in this steps the developer also can discuss the proper fix with the pentester to effectively mitigate the issue. Retesting is carried out to confirm that remediations are effective, and finally, a final report is issued to document the complete

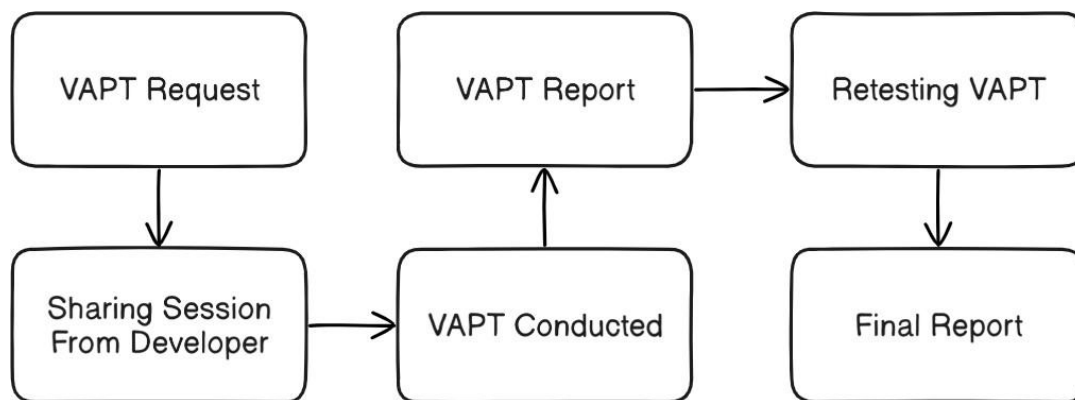


Figure 4.3: VAPT Steps

process and results.

This comprehensive approach ensures that any potential vulnerabilities are identified and addressed early, providing a robust security foundation before the product progresses further in its lifecycle. The recommendation tools to support the security assessment can be shown on Table 4.14.

Table 4.14: Tools that Support Security Assessment

No	Stage	Recommendation Tools
1	Software Composition Analysis (SCA)	Dependency Check, Snyk
2	Image Scanning	Trivy, Snyk, Qualys
3	Secret Scanning	Trufflehog, GitLeaks
4	Static Application Security Testing (SAST)	SonarQube, SemGrep, Bandit
5	Dynamic Application Security Testing (DAST)	Burp Suite, OWASP Zap
6	Vulnerability Assessment and Penetration Testing (VAPT)	Burp Suite, Nessus

4.4.3 Maturity Assessment

In the maturity assessment phase, the product's readiness is evaluated using checklist aligned with the UU PDP regulation and organizational policies. The level of the score can be shown on the Table 4.15.

Table 4.15: Maturity Assessment Measure

Level	Classification	Description
0	Not Implemented	Product not implement the necessary control.
1	Conducted informal design	Product implement the base practice but lack of evidence, or the evidence is not completed.
2	Planned and tracked	Basic requirements is already implemented and have evidence, and can be repeatable.
3	Well defined	The difference from level 2 is that it is planned and can be tracked, besides being repeatable the process is also more mature (there is documentation)
4	Quantitatively controlled	The difference from level 3 is that the processes carried out can be audited
5	Continuous improvement	The key difference at level 4 is that the established processes are subject to regular reviews and continual improvement.

Maturity assessment process verifies the product's compliance with required security standards. If the assessment yields a minimum score of 2, the product can proceed to production and then undergo continuous monitoring. If the product does not meet the threshold, it must implement the recommended remediation measures to achieve compliance with the maturity assessment criteria. The maturity assessment flow can be shown on the Figure 4.4. The maturity assessment help the security team to make sure that product is follow the internal policy, along with complying UU PDP.

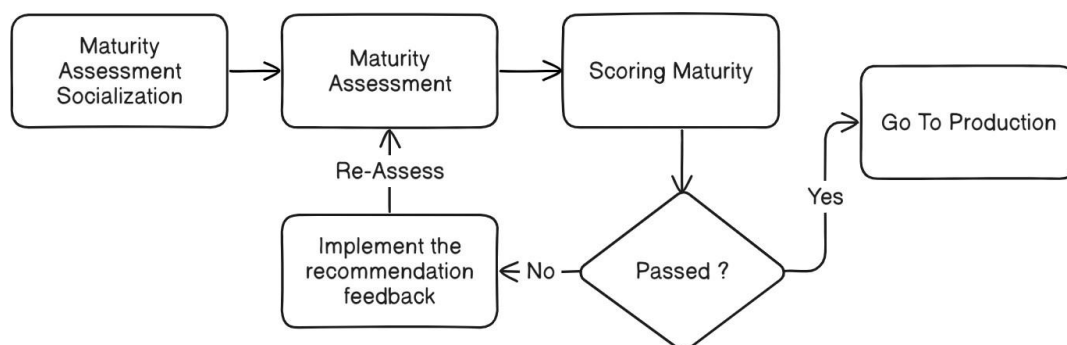


Figure 4.4: Maturity Assessment Flow

4.5 Evaluation

In the Evaluation section, the proposed DevSecOps framework is applied on a limited scale to strengthen the product's security posture. This approach allows for focused enhancements in critical areas, after which the product is re-assessed to confirm improvements and identify any remaining gaps.

4.5.1 Narrow Implementation

The narrow implementation is based on the feedback from the previous maturity assessment, which provides targeted recommendations to help the product reach at least the minimum required score. However, due to constraints in personnel and the product's immediate priorities, the implementation process focus on the most pressing issues facing the product, ensuring that critical security gaps are addressed first while still aiming to improve overall maturity.

4.5.1.1 Product A

The main objective for Product A is to address the recent credentials leak that impacted its assets. Based on the security team's recommendations, the primary focus areas include establishing robust application logging to track and analyze critical events, enforcing clear segregation of duties to reduce unauthorized access or actions. By concentrating on these measures, Product A can strengthen its overall security posture and mitigate the risk of similar incidents in the future.

Application Logging The first priority for the application is implementing a logging feature. Due to recent credential leaks affecting critical assets, it's essential to track user logins by recording details such as who logged in, when the login occurred, the originating IP address, and any actions performed during the session. This functionality not only helps detect suspicious activity—like an unexpected login from an unfamiliar location—but also enables the product owner to investigate and respond to potential security incidents more effectively.

Multi Factor Authentication Multi-Factor Authentication (MFA) is another critical feature to implement on the application, especially in light of recent credential leaks

caused by malware. Acting as a second layer of defense, MFA ensures that even if an attacker obtains a user's credentials, they must also pass an additional verification step—such as a one-time code sent via email or an authentication app. This extra barrier significantly reduces the risk of unauthorized access, as the attacker would be unable to proceed without successfully completing the MFA process.

4.5.1.2 Product B

A top priority for Product B is conducting a comprehensive security assessment, which serves as a critical step in addressing potential vulnerabilities. By proactively identifying and mitigating security risks before the product goes live, Product B can better protect itself against exploitation by malicious actors. This approach not only safeguards user data and maintains trust but also positions the platform for sustained reliability and compliance once it becomes publicly available.

Secure Code As previously noted, Product B places a strong emphasis on security assessments. Consequently, the team plans to conduct a variety of scanning and testing procedures, including image scanning, secret scanning, dependency checks, Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and a full Vulnerability Assessment and Penetration Test (VAPT). By undertaking these comprehensive measures, Product B aims to proactively identify and address potential weaknesses, ensuring a secure product environment before its public release.

Multi Factor Authentication With the rise of sophisticated malware stealers, implementing Multi-Factor Authentication (MFA) has become indispensable in modern security strategies. By requiring additional verification beyond just a password—such as a one-time code, biometric factor, or hardware token—organizations can significantly reduce the risk of unauthorized access. Even if an attacker manages to obtain leaked credentials, the extra MFA layer prevents them from logging into critical systems and accessing sensitive data. This robust defense helps maintain the integrity of the application and safeguards important assets.

4.5.1.3 Proactive Monitor

The Proactive Monitor is a red-team driven activity aimed at uncovering vulnerabilities exposed to the public internet. By conducting realistic attack simulations, the red team goes beyond simply identifying weaknesses in PT XYZ’s products, escalating its findings to demonstrate broader risks and potential business impact. This heightened awareness underscores for senior management the critical importance of prioritizing security—especially in light of UU PDP requirements—and ensures that vulnerabilities are addressed proactively.

In addition to uncovering vulnerabilities, this engagement also leverages threat intelligence tools to gather information on potential credential leaks. By consolidating these findings—including both discovered vulnerabilities and any exposed credentials—the team can provide a comprehensive security overview to inform timely remediation efforts.

This form of monitoring works in tandem with security tools like SIEM or WAF reports. Although SIEM and WAF solutions are invaluable in detecting and blocking many threats, they may not catch every type of attack. By simulating real-world adversarial scenarios, proactive monitoring enhances overall security posture and ensures that all products remain protected. Leveraging insights from both the SIEM/WAF data and proactive monitor findings provides a more comprehensive view of potential risks.

4.5.2 Re-Assess Existing Product

This section provides an overview of the re-assessment conducted on the existing product following the narrow implementation completed previously.

4.5.2.1 Product A

Product A’s current priority is to address the credential leaks identified by BSSN. Efforts are primarily focused on implementing Multi-Factor Authentication (MFA) and robust application logging, enabling the product to better protect its assets from any future credential compromises. The Table 4.16 below identified the second scoring of the Product A.

Table 4.16: Product A Second Maturity Assessment

No	Domain	Maturity Score	Target	Max Score
1	DATA PROTECTION	1.75	2	5
2	NETWORK SEGMENTATION	2	2	5
3	OS and SOFTWARE	2	2	5
4	SERVICES AUDIT	2	2	5
5	ENDPOINT PROTECTION	0.8	2	5
6	SOURCE CODE and CREDENTIALS	1.20	2	5
7	SECURE CODE	1	2	5
8	MFA	2	2	5

4.5.2.2 Product B

Meanwhile, Product B places its emphasis on secure coding practices and Multi-Factor Authentication (MFA) as shown on the Table 4.17. These efforts include conducting thorough Vulnerability Assessments and Penetration Testing (VAPT), as well as other scanning processes, to ensure that any vulnerabilities are identified and resolved before the product is deployed to production.

Table 4.17: Product B Second Maturity Assessment

No	Domain	Maturity Score	Target	Max Score
1	DATA PROTECTION	0.9	2	5
2	NETWORK SEGMENTATION	0.5	2	5
3	OS and SOFTWARE	1	2	5
4	SERVICES AUDIT	1	2	5
5	ENDPOINT PROTECTION	0	2	5
6	SOURCE CODE and CREDENTIALS	1.20	2	5
7	SECURE CODE	2	2	5
8	MFA	2	2	5

4.5.3 Report

In this section, we present a comparative overview of the security maturity levels for both Product A and Product B, Table 4.18 highlighting the state of their security measures before and after implementing recommended improvements. We also outline the findings from proactive monitoring activities, which helped uncover specific security issues that may not have been detected through conventional methods. By examining these results, readers can gain a clear understanding of how both products have evolved in terms of maturity and the role proactive monitoring played in identifying critical vulnerabilities.

Table 4.18: Before and After Narrow Implementation

Products	Before	After
Product A	- No safeguards are in place to prevent attack from credential leaks. - Unable to identify a compromised user. - It's not possible to monitor user actions within the application.	- There is an additional layer of protection for assets in the event of credential leaks. - The system can identify a compromised user. - It is possible to monitor user actions within the application. - The solution can provide an indicator of compromise.
Product B	- There are no protective measures in place to prevent attacks stemming from compromised credentials. - The application's vulnerability remains undiscovered. - The underlying security flaw has yet to be identified.	- In the event of a credential leak, an extra layer of security safeguards critical assets. - The vulnerability has been discovered. - The identified flaw has been resolved.

4.5.3.1 Product A

Before the recommended security measures were implemented, Product A experienced a critical credential leak, as highlighted by a report from the BSSN. This report contained not only usernames and passwords but also a direct link to the Product

A admin panel. Upon further investigation, the Product A team confirmed that these credentials were indeed valid, posing a serious risk of unauthorized access. In response, they took immediate action by resetting all user passwords associated with the admin panel to prevent potential breaches and protect sensitive data.

Following the discovery of the credential leak, the product owner sought guidance from the security team, who recommended implementing detailed application logging and multi-factor authentication (MFA). By adding these features, the organization aims to reduce the likelihood of future credential compromises and gain better visibility into user behavior. Specifically, application logging makes it possible to detect logins from unfamiliar IP addresses and track any unauthorized activities, while MFA adds an extra layer of protection by requiring additional verification steps beyond a simple username and password.

Shortly after implementing MFA and application logging, rumors surfaced on a hacker forum suggesting that unauthorized users had gained access to the application. In response, the product owner immediately consulted with the security team to investigate. Thanks to the new logging capabilities, the team could verify every login attempt, noting the specific user, time, and originating IP address. Additionally, the requirement for multi-factor authentication meant that any unauthorized login attempt would have been flagged. Because no signs of suspicious activity were discovered, the team concluded that the rumors were unfounded, labeling the report as a false-negative. Simultaneously, the security team contacted the individual claiming to have access to the product's assets, posing as an interested buyer. Through this interaction, the team confirmed that the hacker did not actually possess the leaked credentials or other sensitive information, contradicting the claims made in the forum posts.

In conclusion, this targeted implementation—emphasizing credential protection against malware by introducing MFA and comprehensive application logging—significantly enhances overall security. By requiring an additional verification factor and maintaining detailed records of user logins, the system effectively thwarts unauthorized access attempts. At the same time, the logging features offer clear insights into whether user credentials may have been compromised, allowing for swift detection and response. This approach ensures both a more robust defense against attackers and a reliable means of monitoring user activities.

4.5.3.2 Product B

As it prepares for production, the primary focus is on solidifying its security posture. According to the security team's recommendations, Product B should implement secure coding practices—encompassing code scanning and Vulnerability Assessment and Penetration Testing (VAPT)—and promptly address any issues that arise from these assessments. This ensures that the product is free of critical vulnerabilities before it goes live.

During the Static Application Security Testing (SAST) using SonarQube. Product B's source code, 27 code smells were identified. Despite these findings, the code still maintains an A grade in terms of overall maintainability. During the comprehensive Vulnerability Assessment and Penetration Testing (VAPT), the penetration tester identified 17 confirmed security vulnerabilities in Product B, all of which require remediation to ensure a secure launch the details vulnerability can be found on the Table 4.19.

Table 4.19: Vulnerability Found By The Severity

Vulnerability	Critical	High	Medium	Low	Informational
Vulnerability Found	1	6	6	3	1
Vulnerability Closed	1	6	6	3	1
Vulnerability Open	0	0	0	0	0

The details the vulnerability that has been found on the Product B can be found on the Table 4.20.

Table 4.20: Vulnerability Details

No	Findings Name	Severity Level	Status
1	Account Takeover via Cookie Manipulation	Critical – 9.3	Closed
2	Sensitive Information Disclosure in qr-secret	High – 8.2	Closed
3	Sensitive Information Disclosure in qr-image	High – 8.2	Closed
4	2FA Bypass via Response Manipulation	High - 8.2	Closed
5	IDOR Add Users in Access Plans	High – 7.2	Closed
6	IDOR Set Active/Inactive on Plans	High – 7.2	Closed
7	Server-Side Request Forgery (SSRF) in Backend	High – 7.1	Closed
8	No Session Expiration After Log-Out	Medium – 6.3	Closed
9	IDOR in Detail Backend	Medium – 6.3	Closed
10	IDOR in Detail Upstream	Medium – 6.3	Closed
11	HTML Injection on Name at Register Function	Medium – 5.3	Closed
12	IDOR Reset 2FA without Validation	Medium – 5.1	Closed
13	Mass Assignment in Update Profile	Medium – 4.8	Closed
14	Text Injection on Error Response	Low – 2.1	Closed
15	Reuse Reset Password Token	Low – 2.1	Closed
16	Error Backend Disclosure	Low – 2.1	Closed
17	Security Header Misconfiguration	Informational – 0	Closed

Based on the results from the scanning and VAPT engagement, Product B has improved its overall security posture. During testing, the team discovered 17 vulnerabilities in total, including 1 critical issue and 6 high-severity findings, which must be addressed to further strengthen the product before it goes live.

4.5.3.3 Proactive Monitor

In the context of proactive monitoring, the red team conducts penetration tests across all PT XYZ applications, extending well beyond Products A and B. By simulating real-world attack scenarios, they seek to discover vulnerabilities in any product or service within PT XYZ. Once identified, these findings are promptly reported to the

respective product owners. This proactive monitoring serves as a red team engagement, aiming to continuously protect PT XYZ’s assets by evaluating their resilience under various adversarial conditions.

During the engagement, the team performs extensive reconnaissance and harnesses threat intelligence to identify any potential credential leaks. They also employ a range of security scanners—including nuclei, httpx, naabu, nmap, subfinder, Nikto, and Burp Suite—to thoroughly analyze the application’s security posture and pinpoint vulnerabilities. This multi-pronged approach provides a comprehensive overview of potential risks and supports targeted remediation efforts. Table 4.21 indicates the result of the proactive monitor and the Table 4.22 indicates the severity that vulnerability has been found. There is a critical infrastructure-related vulnerability that remains unresolved because attempting to fix it causes service disruptions. Each time the developer deploys a patch, the service breaks and must be rolled back. As a result, the developer has enlisted the help of the DevOps team to collaborate on a stable solution that addresses the vulnerability without compromising service availability. However, right now the vulnerability is still on progress to fix.

Table 4.21: Proactive Monitor Results - Vulnerability & Credential Leaks

No	Findings Category	Total
1	Cross-site Scripting (XSS)	3
2	Improper Authentication	2
3	Insecure Direct Object Reference (IDOR)	1
4	Misconfigured Access-Control-Allow-Origin Header	1
5	OS Command Injection	1
6	SQL Injection	4
7	Credentials Leaks	1

Table 4.22: Vulnerability Found By The Severity Of Proactive Monitor

Vulnerability	Critical	High	Medium	Low	Informational
Vulnerability Found	4	3	6	0	0
Vulnerability Closed	3	3	6	0	0
Vulnerability Open	1	0	0	0	0

Leveraging insights gathered from proactive monitoring, the red team holds a dedicated sharing session to address vulnerabilities requiring immediate attention. This session brings together developers and product owners, ensuring that all stakeholders understand the identified risks and collaborate on strategies to mitigate and prevent similar issues in the future. Because the red team discovered vulnerabilities in the production environment, a clear timeline for remediation has been established as mentioned on the Table 4.23. This approach ensures that any security gaps are addressed promptly, minimizing potential risks and maintaining the integrity of the system.

Table 4.23: Timeline To Remediate The Vulnerability

No	Vulnerability Severity	Time To Remediate
1	Critical	1 Day
2	High	3 Day
3	Medium	7 Day
4	Low	10 Day

In conclusions, through ongoing proactive monitoring and collaborative sharing sessions, PT XYZ can maintain a high level of security across its products and continually strengthen their defenses.

4.6 Validation

The primary goal of the validation process is to confirm that the DevSecOps framework—integrating security maturity and assessment—provides comprehensive protection from multiple angles, with particular emphasis on meeting UU PDP requirements.

4.6.1 Participants Selected for the Validation

For this validation, two experts were invited for interviews to evaluate the proposed DevSecOps framework that include the security maturity model. Their insights covered both the strengths and weaknesses of the approach, highlighted areas needing improvement, and offered feedback on how best to refine and optimize the framework's implementation.

1. Mochamad Akbar Anggamaulana

- Role: Senior Product Security Engineer at Funding Societies | Modalku Group.
- Relevant Criteria: As a Senior Product Security Engineer, Akbar brings extensive hands-on experience with DevSecOps implementations and possesses a broad knowledge base in various security domains. So, that his insight and knowledge be valuable on validate this thesis.

2. Muhammad Yuga Nugraha

- Role: Research And Development Engineer at Practical DevSecOps.
- Relevant Criteria: Yuga serves as a Research and Development Engineer at Practical DevSecOps, where he develops curricula on various DevSecOps domains and provides consulting services to clients implementing DevSecOps within their organizations. His expertise and practical experience make him an ideal candidate to validate this thesis.

4.6.2 SPICE Framework

Because this thesis employ a qualitative approach to evaluate the project's outcomes, the SPICE framework utilized to formulate the research questions as shown on the

Table 4.24. This method provides a structured way to capture the necessary qualitative insights and ensure comprehensive coverage of the study’s objectives.

Table 4.24: SPICE - Question Formation Frameworks

SPICE	Description
Setting	The context is PT XYZ adopting the DevSecOps framework and focusing on security maturity, specifically in compliance with UU PDP.
Perspective	The perspective is drawn from industry professionals, practitioners that experienced in DevSecOps and data security.
Intervention	The focus is on the DevSecOps framework and its integration with security maturity practices.
Comparison	Implicit comparison to the effectiveness of current frameworks and potential adjustments needed for compliance and minimization of data breaches.
Evaluation	Measures success in terms of compliance with UU PDP, minimizing data breaches, and the adaptability of the framework to modern industry needs.

The following questions shown in Table 4.25 were presented during the interview session:

Table 4.25: Mapped Questions

Topic	Question
Evaluation & Compliance (Setting, Perspective, Intervention, Evaluation)	What do you think about the DevSecOps framework in relation to security maturity? Can it comply with UU PDP and effectively minimize data breaches? Why or why not?
Challenges & Mitigation (Setting, Perspective, Intervention, Evaluation)	What are the downsides of the DevSecOps framework when integrated with security maturity? How can we mitigate these challenges?
Experience-Based Insights (Setting, Perspective, Evaluation)	Based on your experience, what aspects should we focus on to ensure minimal data breaches?
Optimization (Setting, Perspective, Intervention, Evaluation)	What steps can we take to optimize the proposed DevSecOps framework that aligns with security maturity?
Industry Relevance (Setting, Perspective, Intervention, Comparison, Evaluation)	What adjustments are necessary to align DevSecOps and security maturity practices with the current needs of the industry?

This question list represents only the main inquiries. During the actual interview, additional follow-up questions be posed as needed to clarify responses and gain deeper insights.

4.6.3 Validation Result

The validation process involved detailed discussions with the participants, including the presentation regarding the framework along with the outcome and then question and answer. The feedback is gathered based on the question that asked during the interview. Table 4.26 and Table 4.27 is summary of the key finding based on the participants responses.

Table 4.26: Cross Case Synthesis - Validation

Question	Akbar	Yuga
Evaluation & Compliance	Incorporating a maturity assessment into the DevSecOps framework significantly reduces the risk of data breaches. This assessment aligns with both UU PDP requirements and internal policies, ensuring robust security measures are in place throughout the development lifecycle.	By conducting a maturity assessment, organizations introduce an added layer of security for public-facing applications, ensuring they adhere to consistent standards and remain better protected against potential threats.
Challenges & Mitigation	One of the primary challenges lies in the time-intensive nature of achieving security maturity. Employing an interview-based approach can help expedite the assessment process, and automating parts of the maturity evaluation would further streamline and optimize this effort.	One potential downside of the maturity assessment is that it can delay deployment, as applications must meet stringent standards before going live. However, providing clear, well-structured guidelines can help streamline this process, making it simpler for teams to comply with the requirements while maintaining momentum.
Experience-Based Insights	Implement continuous monitoring of digital assets through solutions like Endpoint Detection and Response (EDR) and Extended Detection and Response (XDR). Avoid storing sensitive information on public-facing servers, and regularly utilize Google Dorking techniques to proactively identify any unintended data exposures.	Maintain a high level of confidence in your infrastructure by proactively addressing supply chain attacks. For instance, manage your own Docker images and incorporate software attestation to verify the integrity of your software. These measures help secure the development pipeline and prevent the introduction of malicious code.

Table 4.27: Cross Case Synthesis - Validation

Question	Akbar	Yuga
Optimization	Strive to automate as many components of the maturity assessment as possible to minimize the time and effort required. By streamlining repetitive tasks through tooling and automated processes, teams can more efficiently evaluate and improve their security posture.	Develop guidelines and baselines to support the maturity assessment, ensuring that repeatable processes are clearly identified and optimized—whether through automation or streamlined workflows—to continually enhance security and efficiency.
Industry Relevance	It is advisable to schedule regular VAPT engagements to maintain a proactive security posture. Additionally, incorporating a design review alongside threat modeling is highly recommended, as both processes are closely related. By evaluating the architecture during threat modeling, teams can uncover and address security gaps early in the development cycle, ultimately strengthening the overall security of the application. Additionally, align the remediation timelines with an established framework, ensuring that vulnerabilities are addressed within set thresholds.	While image scanning within DevSecOps is not strictly mandatory—organizations may opt to build and harden their own images to use as a secure baseline—certain steps are crucial. In particular, secret scanning, Software Composition Analysis (SCA), Static Application Security Testing (SAST), and Dynamic Application Security Testing (DAST) should be performed and validated through a comprehensive Vulnerability Assessment and Penetration Testing (VAPT). This ensures a robust security posture for applications throughout their development and deployment lifecycle.

Based on the validation interviews, we conclude that integrating a maturity assessment into the DevSecOps process can effectively reduce the risk of data breaches and ensure compliance with UU PDP, while also providing an additional layer of security and standardized practices. However, this approach may prolong the time it takes for applications to reach production. Therefore, stakeholders recommend optimizing the maturity assessment—particularly through increased automation—and providing clear guidelines so that product owners understand the steps required to

remediate any issues discovered during the assessment.

Moreover, offering fundamental resources like a hardened Docker image, along with EDR and XDR monitoring, is crucial for continuous asset protection. Finally, incorporating software attestation further boosts confidence in the supply chain, helping to maintain a robust security posture overall.

