Green Pace

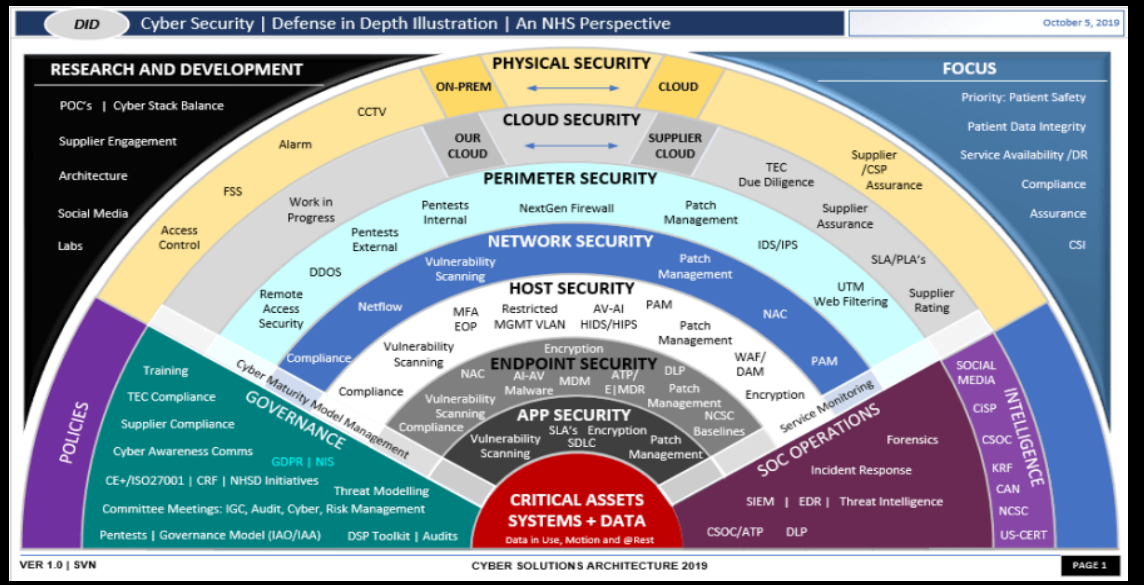
Security Policy Presentation

Developer: *Hong Luu*

This presentation introduces the core security principles, C++ coding standards, and authorization, authentication, and auditing standards, as well as data encryption standards, ensuring consistent and uniform implementation across all software development activities at Green Pace.

OVERVIEW: DEFENSE IN DEPTH

Our new security policy at Green Pace addresses the need for strong measures in the face of evolving cybersecurity threats. It establishes core principles, coding standards, and authentication practices, supporting a defense-in-depth approach. This policy ensures secure development practices from design to maintenance.



Green Pace

THREATS MATRIX

Likely This section highlights high-severity risks that are probable and require immediate attention due to their significant potential impact on security.	**Priority** While less likely, these risks still require monitoring to ensure compliance and readiness.
Low priority Despite being unlikely, these risks necessitate attention to maintain security standards, though with lower urgency.	**Unlikely** These low-severity risks are improbable but should still be addressed over time to enhance security.

Green Pace

10 PRINCIPLES

Principles and Coding Standards Alignment:

1. Validate Input Data
Data Value [STD-002-CPP]

2. Heed Compiler Warnings
Data Type [STD-001-CPP]

3. Architect and Design for Security Policies
String Correctness [STD-003-CPP]

4. Keep It Simple
SQL Injection [STD-004-CPP]

5. Default Deny
Memory Protection [STD-005-CPP]

6. Adhere to the Principle of Least Privilege
Assertions [STD-006-CPP]

7. Sanitize Data Sent to Other Systems
Exceptions and Error Handling [STD-008-CPP]

8. Practice Defense in Depth
Exceptions [STD-007-CPP]

9. Use Effective Quality Assurance Techniques
Object Oriented Programming (OOP) [STD-009-CPP]

10. Adopt a Secure Coding Standard
Concurrency (CON) [STD-010-CPP]

Green Pace

CODING STANDARDS

1. Memory Protection
2. SQL Injection
3. Data Value
4. Assertions
5. Object Oriented Programming (OOP)
6. Exceptions
7. Exceptions and Error Handling (ERR)
8. Concurrency
9. Data Type
10. String Correctness

Prioritization System:

- Criticality: Address severe security risks first.
- Prevalence: Tackle common vulnerabilities early.
- Stability Impact: Ensure memory and error handling.
- Best Practices: Follow industry standards.
- Architecture Relevance: Tailor to system specifics.

Green Pace

ENCRYPTION POLICIES

- Encryption at rest - This means keeping data safe when it's stored, like on a computer or a server. We do this by turning the data into a secret code that only authorized people can understand. We use this policy to make sure that if someone tries to steal our data, they can't read it because it's all scrambled up.

- Encryption in flight - This is about protecting data while it's moving from one place to another, like when we send emails or browse the internet. We use special codes to keep the information safe while it's traveling through networks. This policy helps to keep our information private and secure, so nobody can spy on it while it's on its way.

- Encryption in use - This is about protecting data while it's being used by programs or stored in computer memory. We use encryption to hide the information so that even if someone tries to peek at it, they can't understand what it means. This policy is important to keep sensitive information safe while it's being processed by computers or applications.

Green Pace

TRIPLE-A POLICIES

- Authentication - Authentication verifies the identity of users before allowing them access to systems or data. This ensures that only legitimate users can log in, preventing unauthorized access to sensitive information. The policy applies by requiring secure login methods, such as passwords or biometrics, and monitoring user logins to detect any suspicious activities or unauthorized attempts to access the system.

- Authorization - Authorization determines what actions authenticated users can perform within the system based on their roles or privileges. This includes controlling access to various resources, such as files, databases, or applications. The policy applies by assigning appropriate access levels to users, ensuring they only have access to the information and functionalities necessary for their roles. Additionally, it monitors user activities to detect any unauthorized attempts to access or modify data.

- Accounting - Accounting involves tracking and recording user activities, including logins, changes to the database, addition of new users, user level of access, and files accessed by users. This helps maintain a comprehensive audit trail of system activities, facilitating accountability and compliance with regulatory requirements. The policy applies by implementing logging mechanisms to capture relevant user actions and regularly reviewing these logs to identify any anomalies or security breaches.

Green Pace

Unit Testing

Test 1: Does the system create an empty collection when initialized?

Test Procedure:

- Verify that the collection is empty upon creation.
- Check that the size of the collection is zero.

Result: Passed. Collection created as expected.

Green Pace

Unit Testing

Test 2: Is the maximum size of the collection greater than or equal to its current size?

Test Procedure:

- Test for 0, 1, 5, and 10 entries in the collection.
- Compare the maximum size with the current size.

Result: Passed. Maximum size is greater than or equal to current size.

Unit Testing

Test 3: Does the capacity of the collection accommodate its size appropriately?

Test Procedure:

- Test for 0, 1, 5, and 10 entries in the collection.
- Compare the capacity with the current size.

Result: Passed. Capacity is greater than or equal to current size in all cases.

Unit Testing

Test 4: Does resizing the collection increase its size?

Test Procedure:

- Resize the collection to increase its size by 5.
- Check if the new size matches the expected size.

Result: Passed. Collection size increased as expected.

Unit Testing

Test 5: Does resizing the collection decrease its size?

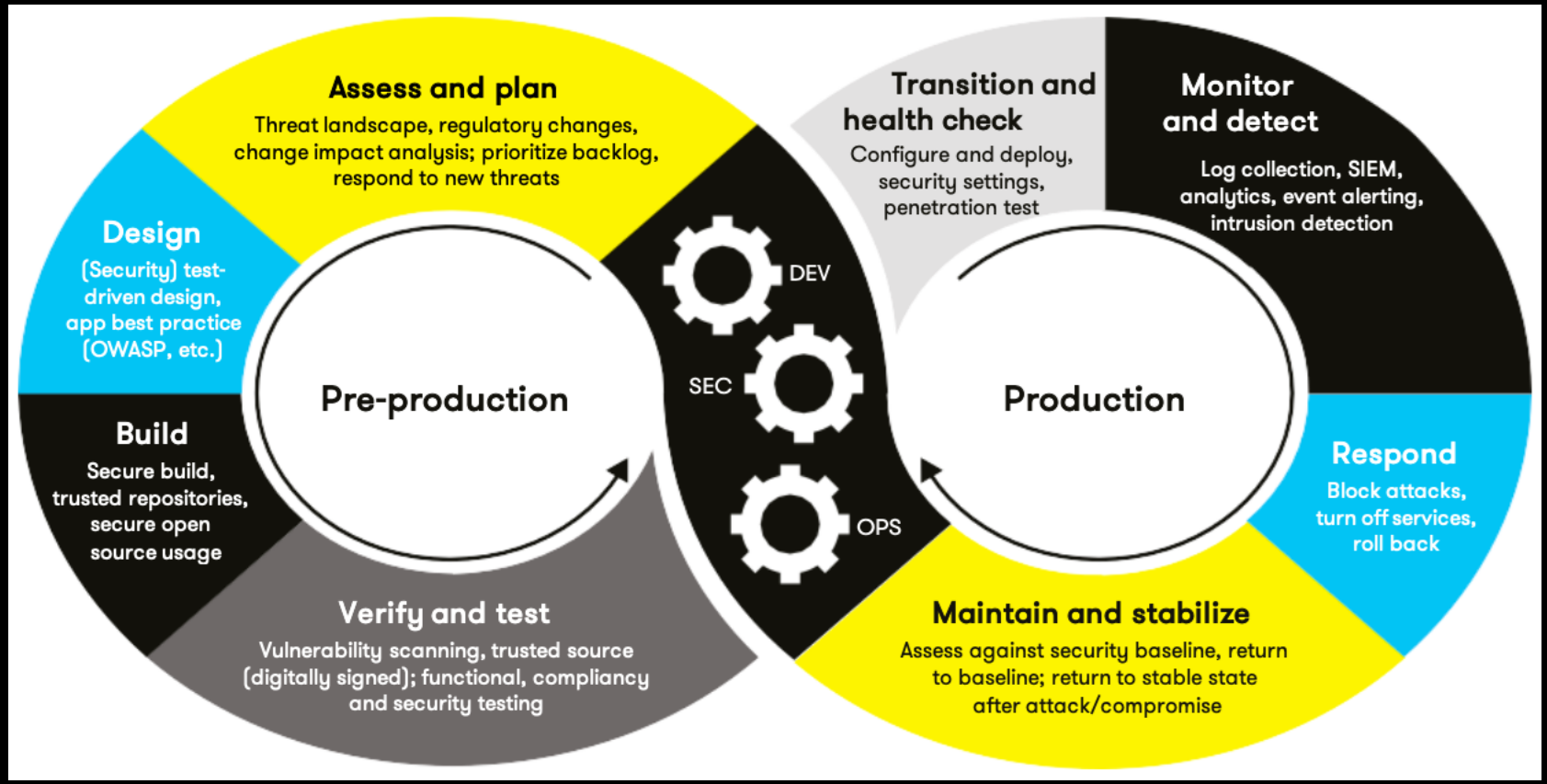
Test Procedure:

- Resize the collection to decrease its size by 5.
- Verify if the new size matches the expected size.

Result: Passed. Collection size decreased as expected.

Green Pace

AUTOMATION SUMMARY



Green Pace

TOOLS

Integrating automation into the DevOps process ensures that security standards are consistently applied without the need for manual oversight. Automated tools can scan for vulnerabilities during the design phase, verify the security of components in the build stage, and continuously test for weaknesses before deployment. In the live environment, automated monitoring systems vigilantly track activities to detect security threats. Should an issue arise, automated mechanisms are in place to swiftly address and resolve the problem, maintaining system stability and upholding security protocols efficiently.

RISKS AND BENEFITS

Problems:

- High severity and likelihood for STD-002-CPP, STD-004-CPP, STD-005-CPP, and STD-006-CPP.
- Low severity but high priority for STD-001-CPP and STD-010-CPP.

Solutions:

- Immediate action needed for rules with high severity and likelihood.
- Prioritize remediation based on severity, likelihood, and priority.

Benefits of Acting Now:

- Mitigation of high-risk vulnerabilities.
- Enhanced code quality and reliability.

Risks of Waiting:

- Increased likelihood of security breaches or software failures.
- Potential impact on system stability and performance.

Strategy Lacking:

- Lack of detailed plan for addressing specific vulnerabilities.
- Missing assessment of potential impact on system functionality.

Steps to Be Taken:

- Develop a prioritized remediation plan.
- Allocate resources for addressing high-risk vulnerabilities.
- Implement coding standards and best practices to prevent future issues.
- Regularly review and update risk assessments to ensure ongoing compliance and security.

Green Pace

RECOMMENDATIONS

Current Gap Analysis:

- Inadequate security monitoring and auditing procedures.
- Limited continuous monitoring leaves systems vulnerable.
- Risks of undetected security threats and breaches.

Real-World Example:

- Equifax breach (2017) highlights the importance of vigilant monitoring.
- Failure to detect unauthorized access led to massive data exposure.

Future Potential Gaps and Improvements:

- Enhance monitoring with intrusion detection systems.
- Implement regular security audits.
- Proactively learn from real-world incidents.

Green Pace

CONCLUSIONS

- To prevent future problems, adopting standards such as ISO/IEC 27001 for information security management is essential. This standard provides a systematic approach to managing sensitive company information, ensuring its confidentiality, integrity, and availability.

Green Pace

REFERENCES

- CON55-CPP. Preserve thread safety and liveness when using condition variables - SEI CERT C++ Coding Standard - Confluence. (n.d.). Wiki.sei.cmu.edu. Retrieved March 24, 2024, from https://wiki.sei.cmu.edu/confluence/display/cplusplus/CON55-CPP.+Preserve+thread+safety+and+liveness+when+using+condition+variables

- The ISO 27001 Certification Process | A-LIGN Compliance & Assessments. (n.d.). A-LIGN. https://www.a-lign.com/lp/iso27001-certification-process?utm_source=google&utm_medium=cpc&utm_campaign=GS-US-ISO27001-Tier2&utm_term=iso%2027001%20process&utm_content=engine:google

- What is a security breach? | Norton. (n.d.). Us.norton.com. https://us.norton.com/blog/privacy/security-breach

Green Pace