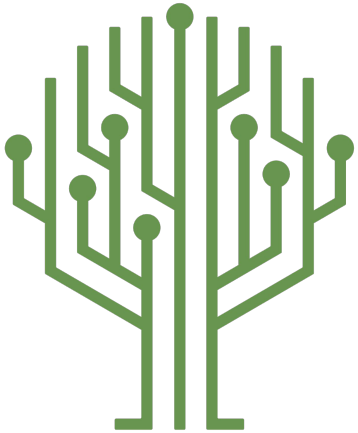


Green Pace Security Policy

Jasmine Zeng - CS 405

Overview - Introduction

- Software development at Green Pace (“the company”) involves continuous, consistent application of secure principles to all applications developed and deployed by the company.
- They must be maintained throughout all policies by uniformly defining, implementing, governing, and maintaining them over time.
- This security policy will define the core security principles:
  - C/C++ coding standards
  - Authorization
  - Authentication
  - Auditing
  - Data encryption

Overview - Threats Matrix

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
STD-001-CPP	Low	Unlikely	Medium	2	3
STD-002-CPP	High	Likely	Medium	18	1
STD-003-CPP	Low	Likely	Low	9	2
STD-004-CPP	High	Likely	Medium	18	1
STD-005-CPP	High	Likely	Medium	18	1
STD-006-CPP	Low	Unlikely	High	1	3
STD-007-CPP	Low	Likely	Low	9	2
STD-008-CPP	Medium	Unlikely	Medium	4	3
STD-009-CPP	Medium	Unlikely	Medium	4	3
STD-010-CPP	Low	Unlikely	Medium	2	3

The threats matrix provides a guide to the Green Pace rules and their level of importance, likelihood, remediation cost, priority, and level when implementing them in accordance to the security policy. Automation practices such as unit testing can help Green Pace find these risks and areas of improvement faster.

Principles

1. **Validate input data** (Standards 3, 9)
2. **Heed compiler warnings** (Standard 7)
3. **Architect and design for security policies** (Standard 8)
4. **Keep it simple** (Standards 1, 6, 10)
5. **Default deny**
6. **Adhere to the principle of least privilege**
7. **Sanitize data set to other systems** (Standard 5)
8. **Practice defense in depth** (Standard 4)
9. **Use effective quality assurance techniques** (Standard 2)
10. **Adopt a secure coding standard**

Coding Standards

1. Data type
2. Data value
3. String correctness
4. SQL injection
5. Memory protection
6. Assertions
7. Exceptions
8. Object oriented programming (OOP)
9. Input output
10. Declarations and initialization

Coding standards have been listed from highest importance to least importance. As Green Pace goes down the list, the latter cannot be executed without proper initialization of the former standards. This is essential when handling massive amounts of data in the system. This is even more pertinent when implementing security practices in Green Pace applications. Data must be established in the proper parameters before the system can proceed.

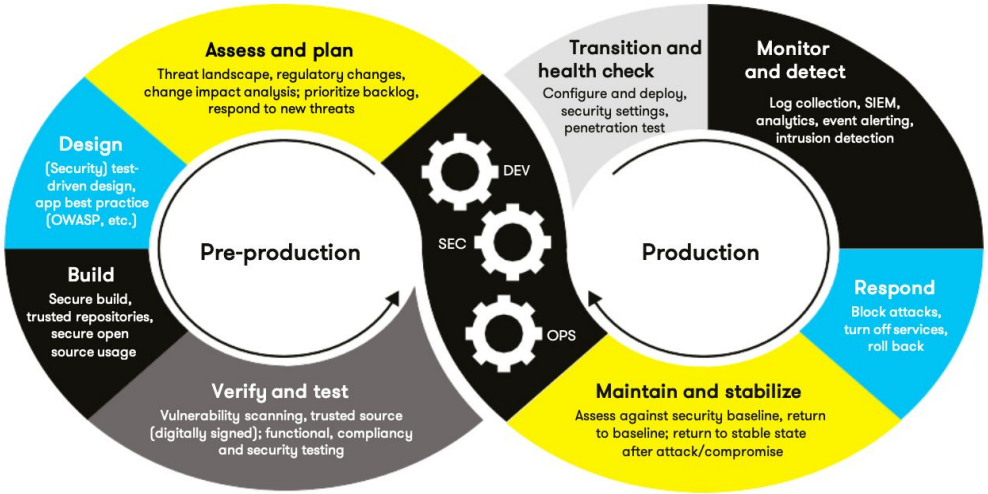
Encryption Strategy

a. Encryption	Description
Encryption in rest	Secures data at rest. Produces minimal effect on input/output latency and throughput. The policy applies to protect data and make it transparent only to relevant users, applications, and services. • Only users with keys are permitted to access data at rest • Data is not stored for longer than needed
Encryption at flight	Secures data in transit. Produces minimal effect on input/output latency and throughput. The policy applies to protect data and make it transparent only to relevant users, applications, and services. • Only users with keys are permitted to access data in transit • Data is not in transit unless needed for purposes of storage or use
Encryption in use	Secures data in use. Produces minimal effect on input/output latency and throughput. The policy applies to protect data and make it transparent only to relevant users, applications, and services. • Only users with keys are permitted to access data in use • Data is not used or accessed unless needed and by parties with approved access

Triple-A Framework

b. Triple-A Framework*	Description
Authentication	Authentication is the verification of a user, process, or device's identity. It is a prerequisite before the system grants them access to resources in a system. • Authentication must be implemented at any access/entry point • Multi factor authentication must be used • Keys cannot be hard coded into the program • Implemented in user logins
Authorization	Authorization is when a server determines if the client or relevant party has permission to use or access data. • Account permissions must be implemented and provide various levels of access, and various levels of files accessed by users • All accounts must possess the minimal permissions • Permissions cannot be granted by anyone except for administrator accounts • Only administrators can make changes to the database
Accounting	Accounting tracks and records user activities on a network or system. • An audit log of all user activity must be made • The audit log can only be accessed by administrator accounts • Nobody is allowed to alter the audit log • New users can be self-initiated

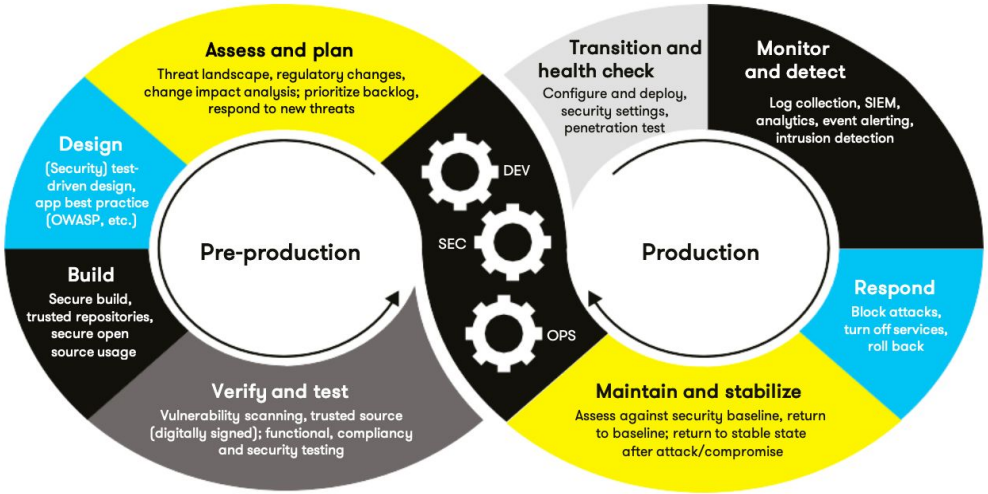
Unit Testing



Green Pace (2023). *Green Pace Secure Development Policy*.

Unit testing is essential to Green Pace security practicies, policies, and procedures. Per the Green Pace Secure Development Policy, multiple dependencies can be used for each of the coding standards. Many of the security policies utilize Astree, Axivion Bauhaus Suite, and CodeSonar dependencies for unit testing. Pictured above is Green Pace's automation practices.

Automation Summary



Green Pace (2023). *Green Pace Secure Development Policy*.

The graphic above represents Green Pace's automation practices. Unit testing is one of many pratices by the company when looking for security vulnerabilities in programs being developed or maintained. Security tools lie in the "Verify and test", "Monitor and detect", and "Maintain and stabilize" phases. This helps Green Pace closely monitor anything that needs to be updated to the company's standards, industry standards, and government policies where needed.

Risks and Benefits

At Green Pace, waiting is not an option. The best defense is the best offense. By staying up-to-date on policies, industry standards, and government policies, Green Pace will keep their programs secure. Strong foundations with cybersecurity in mind will only strengthen programs in the future. Solutions to maintain these programs include unit testing, dependencies, and patches.

When programs are built with security as an afterthought, they will eventually face consequences down the line. Such programs will cost more to maintain and fix than to develop. Additionally, our clients, stakeholders, and users would face risks of privacy and data breaches.

Recommendations

- Green Pace should continue to update its practices, policies, and procedures as technology and its threats continue to evolve.
- Technical and non-technical requirements should remain pertinent to Green Pace's security practices.
- Longstanding motives for attacks are generally categorized as political, social, and/or economic.
- Social engineering is one way breaches have continued in programs and their systems (CloudFlare, n.d.) throughout the history of technology. Green Pace should consider this non-technical factor with high priority during development.
- Multi-factor authentication and encryption techniques should be used to further secure the system. Tokens and codes are generally kept private by all parties including users from the public with emerging technological literacy.

Conclusion

The existing Green Pace security policy will continue to be revised as current and emerging trends in technology continue to evolve. Green Pace has always developed applications with known cybersecurity threats in mind, incorporating top of the line dependencies, and prioritizing security throughout all stages of the software development lifecycle.

To strengthen the security policy, non-technical requirements like monitoring social engineering attempts can be implemented in the future. Security and maintenance should also not be left to full automation. Green Pace should expand the cybersecurity department and have quarterly inspections to ensure everything is up to date on current and emerging cybersecurity practices and trends.

References

Green Pace (2023). *Green Pace Secure Development Policy*.

CloudFlare (n.d.). *What is social engineering?*. Retrieved from https://www.cloudflare.com/learning/security/threats/social-engineering-attack/.