## **Title Slide**

Hello, everyone. My name is Kevin Thompson, and today I will be presenting the new security policy for Green Pace Development. This policy is designed to help protect our code and systems from security vulnerabilities while ensuring compliance with industry standards.

## **Overview Slide 1: Introduction to the Security Policy**

Our security policy aims to safeguard Green Pace's software and systems by implementing clear, repeatable, and actionable guidelines. This policy is rooted in the defense-in-depth principle, which means layering security measures to minimize vulnerabilities. By following these policies, we will ensure secure development practices across all projects.

## **Overview Slide 2: Threat Matrix**

Here is our Threat Matrix, which identifies coding vulnerabilities such as SQL injection, buffer overflows, and unsafe memory operations. Each threat is categorized by severity, likelihood, remediation cost, and priority level. Additionally, we demonstrate how automated tools like SonarQube, Veracode, and Checkmarx can detect and prevent these vulnerabilities effectively.

## **Principles Slide**

The foundation of our security policy includes ten core principles: Validate Input Data, Heed Compiler Warnings, Architect for Security, Keep It Simple, Default Deny, Least Privilege, Sanitize Data, Practice Defense in Depth, Use Quality Assurance, and Adopt Secure Coding Standards. Each principle directly maps to one or more coding standards to ensure consistency and reliability.

## **Coding Standards Slide**

Here are our ten coding standards, prioritized by their impact on security. For example, the top priorities include preventing SQL injection, validating input data, and safeguarding against buffer overflows. Standards are ranked based on severity, likelihood of exploitation, and remediation costs.

## **Encryption Strategy Slide**

Our encryption policies cover data in flight, at rest, and in use. Encryption in flight protects data during transmission to prevent interception. Encryption at rest safeguards stored data, even if physical devices are compromised. Lastly, encryption in use ensures data remains secure while actively processed, especially in shared environments.

## **Triple-A Framework Slide**

The Triple-A framework supports Authentication, Authorization, and Accounting. Authentication ensures only authorized users can access systems. Authorization grants role-based permissions to users, and Accounting tracks all activities to maintain an audit trail. Together, these elements enhance security and accountability.

## **Unit Testing Slides**

Unit testing is critical to identifying vulnerabilities early. For instance, to validate input data, we use automated frameworks like JUnit or Google Test to ensure only safe inputs are processed. Taking it further, we integrate static analysis tools to continuously monitor for compliance.

## **Automation Summary Slide**

Security automation is embedded within our DevSecOps pipeline. Tools like SonarQube and Veracode run during the build and testing stages, ensuring that vulnerabilities are detected early. Automated enforcement of coding standards ensures compliance without manual intervention.

## **Risks and Benefits Slide**

Implementing this policy now allows us to mitigate risks like data breaches and compliance violations. The benefits include improved system resilience and reduced remediation costs. Delaying implementation increases the likelihood of vulnerabilities being exploited, potentially harming our reputation and finances.

## **Recommendations and Conclusion Slides**

Moving forward, we must address gaps such as inconsistent application of coding standards and limited automation in legacy systems. I recommend adopting advanced tools like AI-driven threat detection and increasing staff training on secure coding practices. This concludes my presentation. I am happy to answer any questions.

## **References Slide**

CISO Council, Federal Chief Data Officer (CDO) Council, Chief Information Security Officer (CISO) Council, Federal Chief Information Officer (CIO) Council, & Cybersecurity and Infrastructure Security Agency (CISA). (2024). *Federal Zero Trust Data Security Guide*. https://www.ciwo.gov/assets/files/Zero-Trust-Data-Security-Guide\_Oct24-Final.pdf

Department of Defense (DoD), Defense Information Systems Agency (DISA), National Security Agency (NSA) Zero Trust Engineering Team, & Freter, R. (2022). Zero Trust Reference Architecture. In *Department of Defense (DoD)*. https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT\_RA\_v2.0(U)\_Sep22.pdf

*OWASP Top Ten | OWASP Foundation*. (n.d.). https://owasp.org/www-project-top-ten/