CS 405 Project One Guidelines and Rubric

Competencies

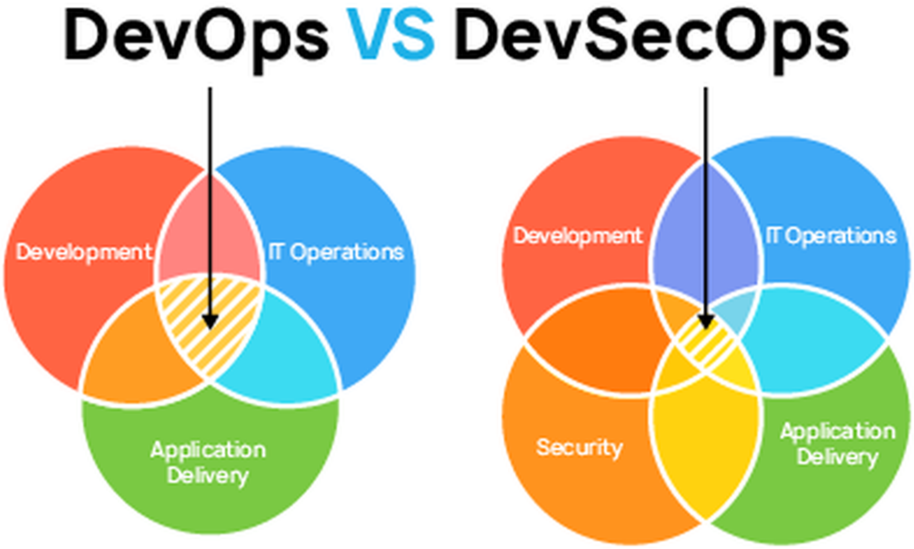
In this project, you will demonstrate your mastery of the following competencies:

- Determine potential vulnerabilities and weaknesses using best practices
- Develop secure code to counteract threats

Green Pace

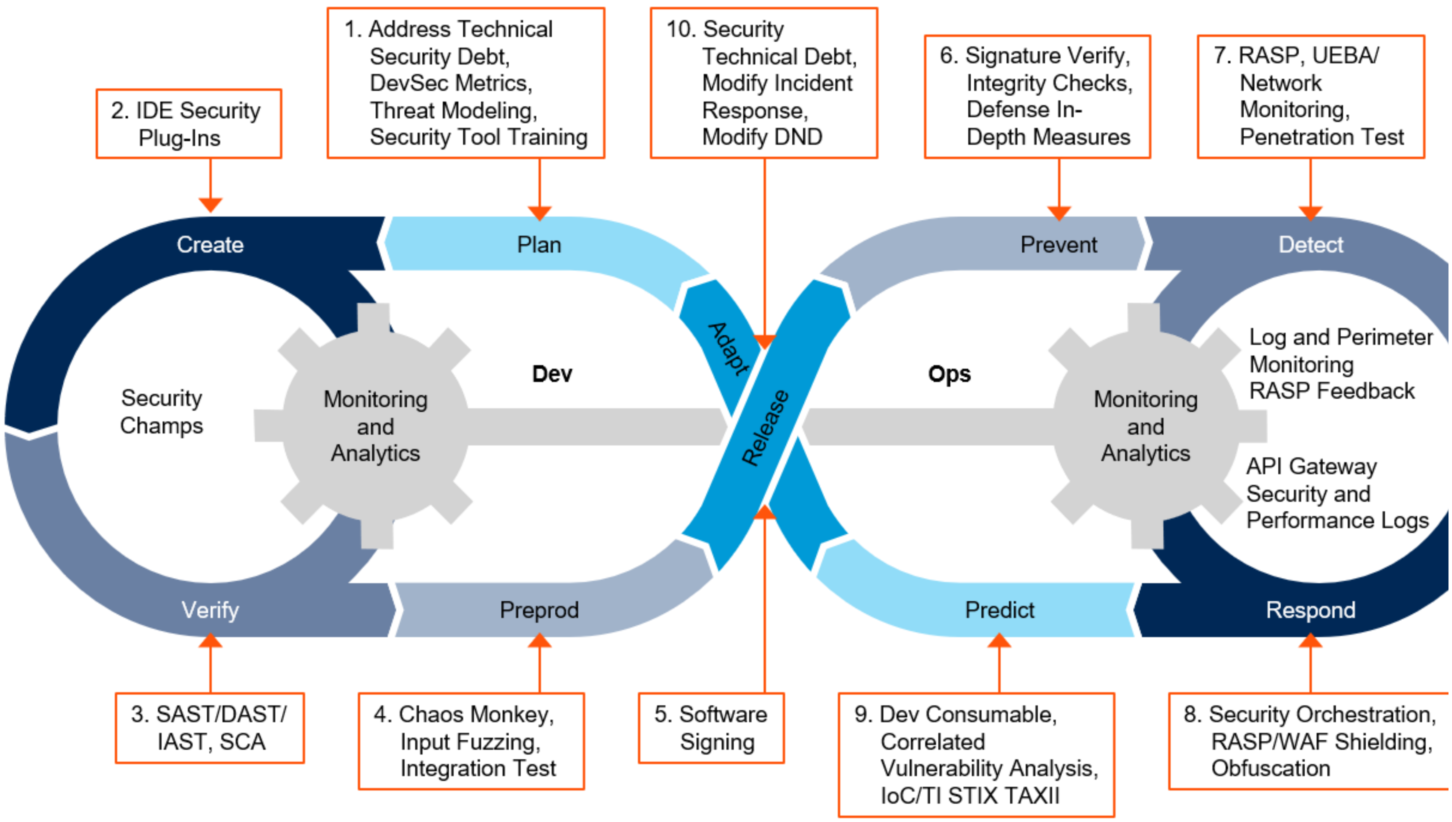
Scenario

You are working as a developer for a software company called Green Pace. It is an engineering company that specializes in custom software design and development for environmentally responsible entrepreneurs worldwide. At Green Pace, the security mission is defense in depth. In order to ensure that all applications comply with the same security policies, you have been tasked with documenting, categorizing, and providing examples for coding and architectural vulnerabilities. Green Pace wants to maximize automation to ensure compliance and keep costs down. Essentially, the company is moving its DevOps practice to a DevSecOps to make it more secure, and the company wants to be well prepared for the security audit. The image below shows how, in DevSecOps, security is a separate and equally veiled function supporting development, operations, and application delivery. In order to complete this project, you need to understand potential vulnerabilities and weaknesses in code and coding best practices. In addition, you will need to be able to understand how to develop secure code to counteract threats.



An auditor will be performing a policy compliance audit to ensure the DevSecOps teams can implement automation, best practices, and continuous testing. You have been tasked with preparing the Green Pace security policy for the auditor. Using what you know about Green Pace practices, you will take the implicit best practices and standardize them in a new security policy. In addition, you will use principles, best practices, and industry standards to support the overarching Triple-A security framework that uses a defense-in-depth best practice as its foundation. Once this is complete, you will be prepared to make recommendations regarding implementation and how to maintain and update the policy in the future. The image below shows the developers' security pipeline. You will be using this diagram to illustrate where and how automation fits into the development process.

The DevSecOps Toolchain


2. IDE Security Plug-Ins
1. Address Technical Security Debt, DevSec Metrics, Threat Modeling, Security Tool Training
10. Security Technical Debt, Modify Incident Response, Modify DND
6. Signature Verify, Integrity Checks, Defense In-Depth Measures
7. RASP, UEBA/ Network Monitoring, Penetration Test
Create
Plan
Adapt
Release
Prevent
Detect
Security Champs
Monitoring and Analytics
Dev
Ops
Monitoring and Analytics
Log and Perimeter Monitoring RASP Feedback
API Gateway Security and Performance Logs
Verify
Preprod
Predict
Respond
3. SAST/DAST/ IAST, SCA
4. Chaos Monkey, Input Fuzzing, Integration Test
5. Software Signing
9. Dev Consumable, Correlated Vulnerability Analysis, IoC/TI STIX TAXII
8. Security Orchestration, RASP/WAF Shielding, Obfuscation

Source: Gartner
ID: 377293

Directions

You have been tasked with standardizing security vulnerabilities in code and policy in systems architecture. Specifically, you will open the security policy template and use the instructions outlined below to complete the coding standards based on SEI CERT. The completed security policy will be used to ensure compliance in DevSecOps as part of your defense-in-depth strategy and Triple-A framework.

Coding Standards:

This section of the security policy is used to recognize coding vulnerabilities, create standards, and ensure policy compliance for coding within your organization. You will use the same security policy template you used in the Module Three milestone to complete each of the standards templates by adding principles, threat level, and tools. At the completion of this project, you will have a finished security policy. Your security policy should have 10 standards. Several of the vulnerabilities listed may have more than one standard.

- Data type
- Data value
- String correctness
- SQL injection
- Memory protection
- Assertions
- Exceptions

1. **C/C++ Standards in the Coding Standards**
   Use the SEI CERT C++ Coding Standard resource in Supporting Materials to collect the information needed to complete your standards. There are 49 rules and 500 coding standards. You will need to narrow down the rules that apply. You will improve your coding policy standard rationale and examples based on instructor feedback. If you had nothing to improve, then focus on the new information that should be added to complete each of the 10 standards by continuing to task two.
2. **Risk Assessment**
   Complete this for each of the coding standards. You will fill in the columns of the **Risk Assessment table** that read *Severity, Likelihood, Remediation Cost, Priority,* and *Level.*
3. **Automated Detection**
   Complete this for each of the coding standards. You will complete the Automation section by determining which tool or tools to use for each of the coding standards. You may choose tools from the list found in the **appendix**, or you may propose alternative tools that will detect issues in each of the standards. You may list one or more tools that can automatically detect an issue. Include the name with a version number, the name of the rule or check (preferably with a link), and any relevant comments or description.

4. **Automation**

   Provide a written explanation of where in the process the automation should take place. This section of the template is not part of the coding standards tables. It is a separate section to allow for a paragraph or two of writing. Here you will write and summarize, in general, how automation (tools) will be used for the enforcement of and compliance to the standards defined in this policy. Green Pace already has a well-established DevOps process and infrastructure. Define guidance on where and how to modify the existing DevOps process to automate enforcement of the standards in this policy. Use the DevSecOps diagram provided in the template for your context. You may use the SonarSource resource in Supporting Materials to help locate and justify automation in the DevSecOps pipeline.
5. **Summary of Risk Assessments**

   In task two, you added a threat assessment to each of the 10 standards. Now you will consolidate all risk assessments into one table to have a comprehensive list, including all coding standards, ordered by standard number. The table in the security policy appears as shown below.

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
STD-001-CPP	High	Unlikely	Medium	High	2

   Next you will write policies for encryption and Triple A.
6. **Create policies for the three types of encryption (in flight, at rest, and in use) and each of the three elements of the Triple-A framework using the tables provided**.
   a. Explain each type of encryption, how it is used, and why and when the policy applies.
   b. Explain each type of Triple-A framework strategy, how it is used, and why and when the policy applies.

   Note: Look for and complete this section in the template. (The security policy template contains the complete list.)

Policy Names	Explain what it is and how and why the policy applies.
Encryption in rest	

7. **Map the principles** to each of the standards and provide a justification for the connection between the two. In the Module Three milestone, you added definitions for each of the 10 principles provided. Now it's time to connect the standards to principles to show how they are supported by principles. You may have more than one principle for each standard, and the principles may be used more than once. Principles are numbered 1 through 10. You will list the number or numbers that apply to each standard, then explain how each of these principles supports the standard. This exercise demonstrates that you have based your security policy on widely accepted principles. Linking principles to standards is a best practice.

What to Submit

To complete this project, you must submit the following:

Security Policy for Green Pace

Use the template provided and submit one comprehensive guide.

Supporting Materials

The following resource(s) may help support your work on the project:

Template: Security Policy Template Word Document

Complete and submit this template for Project One.

Website: SonarSource Code Analyzers Rules Explorer

You may use this resource to help locate and justify automation in the DevSecOps pipeline.

Website: STR53-CPP Range Check Element Access

Use this resource to collect the information needed to complete your standards.

Project One Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Data Type Correctness Policy	N/A	Defines the policies and standards and applies best practices; maps the principle to the standard and supplies an explanation (100%)	Shows progress toward proficiency, but might have left out a data type; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (85%)	Does not attempt criterion (0%)	4
Data Value Correctness Policy	N/A	Defines the policies and standards and applies best practices; maps the principle to the standard and supplies an explanation (100%)	Shows progress toward proficiency, but might have left out a data type; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (85%)	Does not attempt criterion (0%)	4
(Data) String Correctness Policy	N/A	Defines the policies and standards and applies best practices; maps	Shows progress toward proficiency, but with errors or	Does not attempt criterion (0%)	4

		the principle to the standard and supplies an explanation (100%)	omissions; areas for improvement may include normal strings and internationalization; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (85%)		
SQL Injection Policy	N/A	Defines the policies and standards and applies best practices to stop SQL injection; maps the principle to the standard and supplies an explanation (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include explanation of which technique was used and how it stops SQL injection; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (85%)	Does not attempt criterion (0%)	4
Memory Protection Policy	N/A	Defines the policies and standards and applies best practices; maps the principle to the standard and supplies an explanation (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include memory that is allocated but has not been cleaned or cleared and mention of best practices, libraries, use of STL smart templates, smart pointers, various types, and SPO; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (85%)	Does not attempt criterion (0%)	4
Assertions Policy	N/A	Defines the policies and standards and applies best practices; maps the principle to the standard and supplies an explanation (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include 10 principle definitions or specific examples; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (85%)	Does not attempt criterion (0%)	4
Exceptions Policy	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner. *The catch-all scenario is covered, known as error hiding* (100%)	Defines the best practices for exceptions (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include 10 principle definitions or specific examples; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	8
Encryption Policy In Flight	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Describes how to apply an encryption policy in flight (85%)	Shows progress toward proficiency, but with errors or omissions; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	10
Encryption Policy At Rest	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Describes how to apply an encryption policy at rest (85%)	Shows progress toward proficiency, but with errors or omissions; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	10
Encryption Policy In Use	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Describes how to apply an encryption policy in use (85%)	Shows progress toward proficiency, but with errors or omissions; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	4

Triple-A Framework Authorization Policy	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Defines the authorization policy, where it needs to be used, and best practices (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include clarity and specific examples of authorization in practice; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	11
Triple-A Framework Authentication Policy	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Defines the authentication policy, where it needs to be used, and best practices (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include clarity and specific examples of authentication in practice; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	11
Triple-A Framework Accounting Policy	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Defines the accounting policy, where it needs to be used, and best practices (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include clarity and specific examples that demonstrate how accounting is used to track threats; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	8
Automated Detection Policy	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Summarizes top-level strategy, the key principles, and the goal of automation; shows how it supports defense in depth; shows a generalized policy explaining when and how LDAP, and AD are used; maps policy to a standard (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include clarity and specific examples of how automation detection should be used as threat identification or prevention; needs to complete each column in the table according to instructions and explain when, where, and how to apply the policy (55%)	Does not attempt criterion (0%)	10
Articulation of Response	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose (85%)	Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability (55%)	Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas (0%)	4
				Total:	100%