James Soto

8-4-2023

CS-405 Secure Coding

Project Two: Security Policy Presentation

<https://youtu.be/gePwfZUalPw>

Hi, my name is James Soto, and I will be going over a security policy presentation.

Defense in depth is a security strategy that involves using multiple layers of security controls to protect a system or network. The goal of defense in depth is to make it as difficult as possible for an attacker to gain access to sensitive data, information or any aspect of the system.

In the threats matrix, we encountered four likely threats that were high risk threats that needed attention A S A P. We had three low priority risks that were probable in their likelihood of an attack and three unlikely threats to occur.

The 10 principles are to validate input data from unverified sources, which includes to treat all the data from any external or unverified source as untrusted until the information has been validated and sanitized.

Number two is limit the scope of functions and variables. The smaller the scope of the system, the less chance of unauthorized access.

Number three, use smart pointers.

Number four, prefer the standard library functions. C plus plus and C offers one of the most extensive libraries available that are tested, optimized, are extremely stable, covering a wide range of implementations for practically any code that is regularly patched and updated.

Number 5, regularly patch and update the compilers, libraries and tools.

Number 6 Use constants by correctly declaring the pointers and the variables as constants whenever possible. This ensures that they can't be modified or manipulated,

Number 7 sanitize all inputs by validating and sanitizing any input before the data is used.

Number 8 limit the use of dynamic memory.

Number 9 Minimize the use of global variables so they cannot be accessed or modified, and finally.

Number 10 try to avoid buffer overflows because C or C++ does not perform automatic bond checking on arrays, which could lead to a buffer overflow.

Here we have the coding standards.

This slide contains the encryption policies.

Encryption in rest is designed to prevent the attacker from accessing the unencrypted data by ensuring the data is encrypted onto a hard drive, a flash drive, or cloud storage, which is a very popular solution. If the storage system or system is attacked or compromised, encryption at rest ensures that the data remains inaccessible.

Encryption at flight is the process of encrypting data that is being moved or transmitted between a system or over a network. Using encryption while the data is in transit ensures that the data cannot be intercepted or read without the encryption keys.

Encryption in use is data that is accessed or used by a user or any application. That is when the data is the most vulnerable.

This slide explains what Triple-A policies are.

Authentication is the process of verifying the identity of the user, the device that is being used or the system to authenticate a user and allow them to access the system. The authentication process will involve checking the credentials of the users, which is the correct username and password.

Authorization occurs after the user is authenticated and it determines what the user has access to inside of the system. The system should provide access based on permissions and rights to ensure that the users don't access something that they're not supposed to.

Accounting is the process of keeping track and monitoring of the resources the user accessed, and what they did. This provides a record that could be studied by the organization to ensure that an attack did not occur from an authorized user.

The following slides are Unit testing.

The first Unit test is capacity, greater or equal to size. This unit test checked the capacity is greater than or equal to the sizes of 0, 1, 5, or 10 entries.

The second unit test is can add to an empty vector. This test verifies adding a single value to an empty collection.

The following test is max size, greater or equal to size and this tests checks that the max size is greater than or equal to the entries of 0 1 5 and 10.

This is a negative unit test called access out of bounds throws an exception, and the test tries to access an index at five that does not exist in the collection, expecting an out of range exception. if no exception is thrown, the test fails.

Finally, the last unit test is erase from empty vector does not decrease the size. This test checks if the collection size decreases when attempting to erase an element from an empty collection.

In this slide we have the steps of an automation summary of Development, Security and operations also known as DevSecOps.

The tools that I recommend we use are static application tests and scans, to scan the source code for potential security vulnerabilities. Dynamic application security testing, that test the running application for vulnerabilities. Infrastructure as code scans the entire infrastructure for any security vulnerabilities. Security orchestration, automation and response automates the security incidents response process. Lightweight directory access protocols is used to query information rapidly. Active directory is used to store sensitive information.

In this slide I will discuss the risks and benefits of finding a resolution to a security threat immediately or delayed.

A resolution that is immediately implemented will identify the issue and the vulnerability which allows for the team to create the patch, test the patch, and deploy the patch across the entirety of the system. The risks are that the deployment of the patch could create system instability or incompatibility with the software or the system that is already in place. A benefit of fixing the security issue immediately is that it prevents potential exploits and mitigates the risks for the company.

A delayed resolution will allow for the team to thoroughly test the patch and study how the attack happened. The biggest risk of a delayed resolution is the damage that could occur. The company could also lose the customer's trust and they could be legal implications with government's worldwide depending on the scope of the attack.

My recommendation would be to implement a rock solid security policy in the organization to protect the data, the infrastructure and the system which will be based on the threats and the risks that the company faces. I recommend that the organization encrypt every piece of data, create a triple-A framework, perform unit testing on the code as often as possible, and use external tools or companies to run vulnerability checks to ensure that the code is safe. Training needs to occur for all employees to ensure that they understand the policies and management needs to enforce and monitor the compliance of these policies.

This is important because of the real world example of how Yahoo was attacked in 2013 and 2014, where the security breach was not disclosed until 2016 and resulted in over three billion users having their identities stolen. Yahoo did not have the proper security measures in place, the employees were not properly trained, which was the root cause of this security breach. An untrained employee fell for a spear-phishing email that allowed the hackers to gain access to the user information for years. This attack ultimately damaged yahoo’s reputation and the user’s trust in the company.

The conclusion of this presentation is that the team should follow secure coding standards while following the security policies put in place. During the development cycle the team should start testing the code early into development, using unit testing, integration testing, black box testing, system testing, and penetration testing. Vulnerability scanner should be used to scan the software/code for security vulnerabilities which could have been missed by the other types of testing.