Dani Yosopov  
  
02/19/2023  
  
Project Two: Security Policy Presentation  
  
<https://www.youtube.com/watch?v=JOwkoiSGLYM&ab_channel=DaniYosopov>

**First slide**:  
Hello everyone, my name is Dani Yosopov and today I will talk about the security policy in Green Pace.

**Second slide**:  
In this slide, we can see the Defense-in-depth, which is a multi-layered defense in order to protect from cyber-attacks this gives better protection because if one defense layer is failed another defense layer will encounter the attack until the attack is stopped.

To illustrate it better in physical form we have the same implantation when it comes to military defense in the airspace borders against missile attacks of my home country Israel.

Using different anti-missile systems for different missile ranges.

This creates a multi-layered defense.

**Third slide**:  
As we can see in this slide in each secure coding best practice there are the likelihood and the impact assigned.

Using this chart allows us the determine the threat levels that can be used by an attacker to target weak areas of your application.

I would use DevSecOps automation to detect these coding vulnerabilities.

DevSecOps is automation that simplifies the integration of security approaches into continuous integration and deployment pipelines, reducing the number of errors that occur when security analysis is performed manually like unit testing.

we will talk more about DevSecOps in a deeper manner in the ninth slide.

**Fourth slide**:  
Those are the ten principles we will follow in this project.

1. Validate Input Data.

2. Heed Compiler Warnings.

3. Architect and Design for Security Policies.

4. Keep It Simple.

5. Default Deny.

6. Adhere to the Principle of Least Privilege.

7. Sanitize Data Sent to Other Systems.

8. Practice Defense in Depth.

9. Use Effective Quality Assurance Techniques.

10. Adopt a Secure Coding Standard.

**Fifth slide**:  
Those are the ten coding standards.

1. Data Type - Do not cast to an out-of-range enumeration value.

2. Data Value - Ensure that division and remainder operations do not result in divide-by-zero errors.

3. String Correctness - Do not attempt to modify string literals.

4. SQL Injection - Sanitize data passed to complex subsystems.

5. Memory Protection - Do not access freed memory.

6. Assertions - Incorporate diagnostic tests using assertions.

7. Exceptions - Honor exception specifications.

8. Object Oriented Programming (OOP) - Do not invoke virtual functions from constructors or destructors.

9. Exceptions and Error Handling (ERR) - Guarantee exception safety.

10. Declarations and Initialization (DCL) - Do not define a C-style variadic function.

**Sixth slide**:  
In this slide, we will talk about the different types of encryptions.

1. Encryption in rest  
encrypting data at rest, you’re essentially converting data into another form of data. This usually happens through an algorithm that can’t be understood by a user who does not have an encryption key to decode it.

This will be applied to any data in the system.

2. Encryption at flight

encryption in flight is the encryption of data that moves over a network. This is especially important for those using the open internet for transporting data, which is part of most public cloud implementations.

This will be applied every time the system is using the network to send or receive data.

3. Encryption in use

Encryption in use ensures that sensitive data is never left unsecured, regardless of the life cycle stage.

This will be applied throughout the entire life cycle stages.

**Seventh slide**:   
In this slide, we will talk about Triple-A Framework and what it means.  
1. Authentication

Is the process of verifying a user or device before allowing access to a system or resources.

This will be applied throughout the entire life cycle stages.

2. Authorization

The policy of specifying access or privileges to resources to a specific type of user.

This will be applied throughout the entire life cycle stages.

3. Accounting

This policy monitors the user activity by the amount of system time or the amount of data a user has sent or received during an online session. This will be applied throughout the entire life cycle stages.

**Eight slide**:  
In this slide, we can see an example of unit testing that was used with the Google unit testing framework.

A unit test is a way of testing a unit which is the smallest piece of code that can be logically isolated in a system.

With unit testing, developers can have more control over their individual code block quality before integrating different components and then sent for regression testing. Also, it is easier to identify and rectify mistakes or defects at the code level which can be detected as a security vulnerability.

**Ninth slide**:  
DevSecOps is a trending practice in application security that involves introducing security earlier in the software development life cycle.

These are the tools used in DevSecOps:

1. Static application security testing

These are tools scanning proprietary or custom code for coding errors and design flaws that could lead to exploitable weaknesses.

Used primarily during the code, build, and development phases of the SDLC.

2. Software composition analysis

These are tools that scan source code and binaries to identify known vulnerabilities in open-source and third-party components.

They also provide insight into security and license risks to accelerate prioritization and remediation efforts.

Used during the build integration to preproduction release.

3. Interactive application security testing

These tools work in the background during manual or automated functional tests to analyze web application runtime behavior.

the tool is providing detailed insights to developers down to the line of code where they occur during development.

4. Dynamic application security testing.

is an automated opaque box testing technology that mimics how a hacker would interact with your web application or API.

It tests applications over a network connection and by examining the client-side rendering of the application, much like a pen tester would.

**Tenth slide**:  
Cybersecurity has Risks:

1. Malware attack.

Attacks use many methods to get malware into a user’s device.

2. Software supply chain attacks.

A software supply chain attack is a cyber-attack against an organization that targets weak links in its trusted software update and supply chain.

3. Password attacks.

A hacker can gain access to the password information of an individual by ‘sniffing’ the connection to the network, using social engineering, guessing, or gaining access to a password database.

The solution for those risks is to apply all the principles learned in this presentation.

This will give us the opportunity to enjoy the Cybersecurity Benefits.

Cybersecurity Benefits:

1. Protects Personal Data.

Personal data is the most valuable commodity for businesses or individual users. However, digital applications have blurred the lines of privacy.

2. Protects Business Reputation.

Customer retention is an essential business factor that can be done by strengthening brand loyalty.

3. Prevents Websites Crashes.

Small businesses often host their website. Hence, infected systems will lead to a website crash.

**Eleventh slide**:  
I would recommend following those tips.

1. Security policies can contain some gaps, which is why it's important to keep up with updates and apply them to the project.

2. Start the project with security in mind at the earlier stages throughout the final product. “Don't Leave Security to the End”

**twelfth slide**:  
In conclusion, based on the principles and standards described in this presentation I believe that all possible security attacks are covered if we keep up with the new updates and follow the principles and standards.