# CS 405 Project Two Script Template

Name: Robert Bogart

Date: 8/15/24

Assignment Name: Project Two: Security Policy Presentation

Link: <https://youtu.be/mhEvRSgQNY4>

Complete this template by replacing the bracketed text with the relevant information.

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hey everyone, today I'm walking you through our security policy. It’s not just about keeping things secure—it's about staying one step ahead of the bad guys. |
| **2** | We’re diving into why this security policy matters. Cyber threats are getting sneakier, so we need this policy as our first line of defense against anything nasty that comes our way. |
| **3** | Here’s a breakdown of the threats we face. Some are more likely than others, but we're using automation to catch any issues before they blow up into bigger problems. |
| **4** | We’ve got ten solid principles to keep our code secure. Things like validating input and keeping it simple go a long way in making sure our systems are tough to crack |
| **5** | These coding standards are prioritized to keep our code clean and secure. We’re focusing on things like memory protection and SQL injection prevention to avoid any major issues. |
| **6** | When it comes to encryption, we’re covered from all angles—whether data’s at rest, in flight, or in use, we’ve got a lock on it to keep it safe from prying eyes. |
| **7** | Our Triple-A policies are all about who can access what and making sure we know what’s going on. Authentication, Authorization, and Accounting work together to keep everything in check. |
| **8** | During our unit testing, most things ran smoothly. We checked if collections were empty and if vectors could be filled up—and yeah, everything worked out just fine. |
| **9** | We’re using automation to enforce our coding standards. By embedding security checks throughout our DevSecOps pipeline, we’ll catch vulnerabilities early and keep our code secure. |
| **10** | If we don’t integrate our standards now, we could be leaving the door open for trouble. But by acting fast, we’ll be ahead of the game, catching issues before they become big headaches. |
| **11** | We’ve got some gaps to fill, especially around automation and training. Let’s tighten up by specifying tools and making sure everyone knows the ropes. |
| **12** | Remember the Equifax breach? They missed some key standards, and it cost them big time. This shows why it’s crucial to stay on top of our security game. |
| **13** | To keep our systems secure, we need continuous testing, secure code reviews, and routine updates. Let’s stick to these standards to avoid future issues. |
| **14** | Here are the references we used to pull all this together—just in case you want to dive deeper into the details. |