# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello this is Braxton Morrow and this is my presentation |
| **2** | The policy I chose was two-factor authentication. Multi-Factor Authentication is the ability to use an outside application (an authenticator), to authorize access to your application via a 6-digit code that changes typically every 30-45 seconds. This applies to Defense-In-Depth as it adds another layer of security to your system |
| **3** | This is a chart to better allow developers and other team members to understand the severity and measure the impact that something may have. |
| **4** | Here are the ten security principles my favorite by far is Keep it Simple as K.I.S.S is one of my favorite acronyms. |
| **5** | Here are the 10 coding standards |
| **6** | Encryption at rest is essentially data that is encrypted when not moving, encryption at flight is data that is encrypted as it moves, and encryption in use is data that is encrypted as it’s being used. |
| **7** | **Authentication –** This is where the user is confirmed as a non-attacker and a User who should have some amount of access to your system. Typically done with a Username as Password, also as I said before utilizing Two-Factor Authentication as well sometimes.  **Authorization –** This is the level of access any given user has access to your system. Level of access includes, read, write, delete, or modify files within a given system. The final level of access is Admin.  **Accounting –** The process of monitoring what a user is doing with their level of access to the system. Typically used in some kind of log, it will keep track of what, when, and how something was accessed within your system. |
| **8** | Two-Factor Authentication, used early and often to ensure quality product this is part of the concept of don’t leave security until the end. |
| **9** | Here is the automation summary diagram |
| **10** | The DevSecOps pipeline is a secure coding practice that allows the developers to utilize a full circle approach that is built around keeping code as secure as efficiently as possible. A traditional pipeline has several distinct phases: Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor. |
| **11** | Well typically there is no benefit to waiting, the faster you act in terms of security (ideally the day code hits an IDE) the better and more secure your system will be. A simple solution that solves your problem eloquently is ideal, also one that you can readily and rapidly use and adapt to a few different situations as the code and the program changes there will be other loopholes that need fixing and patching. |
| **12** | As with every security policy, no system is perfect and there will be gaps. In this case with two-factor authentication the User and the System are both vulnerable to phone cloning. I have a perfect real-life example of this. I have a friend let’s call him “Rabbit” (obviously not his real name similar to a nickname we all call him however), he was a HUUUGGGEEE Old School Runescape player and had a Hardcore Ironman he spent thousands of hours on. He was also known to do some nefarious things such as Duel Arena Gambling duel for in game on his non-ironman main account, and purchasing in game gold with real USD. He did have a two-factor authenticator setup for his account however, both of them. He apparently won a duel against some less than reputable person, and overnight while he was sleeping somehow that salty duel loser cloned his phone, got ahold of his authenticator and stole both of his accounts. |
| **13** | Essentially no system is perfect, however with a solid amount of effort and thought, your system can be hard to get into and potentially not worth the effort of a would-be attacker. Making your system as hard to get into as you can help prevent some things but not everything. |