# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | “Hello and welcome to this security policy presentation at Green Pace! I am your speaker and humble developer, Ryan Mitchell. We wanted to give you, our beloved team, a short little rundown about some new policies that are being implemented company wide in an effort to keep up with modern security practices, where “security starts at the beginning”. |
| **2** | “So let’s talk about the idea of ‘Defense in Depth’ and what this means. As we can see from this graphic, there are many layers by which data can be passed. Each and every layer is (or could be) a form of defense in its own right, and it’s important that we start to consider this from the very moment a new idea for a product is created. Rather than follow a system in which we develop first, and patch it up later, we want to integrate basic defensive practices from the very start. The idea is that we want as many, strong layers of defense as we can reasonably manage, because after the first layer is penetrated, the remaining layers are sort of like our backup plan. |
| **3** | “Now, what kind of threats are we looking at? There are actually too many to fit on a powerpoint slide, but we can pick some of the major ones. Here we’ve divided some problems by priority, as well as likelihood. Some major threats such as SQL injection and improper data validation are what we would consider high priority, or ‘please do not let these happen at all costs’. Some threats aren’t as damaging, but they can be common such as forgetting to bounds check, or accessing freed memory (or allowing freed memory to be accessible). In addition to these two categories, some threats can even be found overlapping. Events such as parsing errors or issues with syntax are unavoidable, but still worthy enough that we should be conscious of the possibilities.” |
| **4** | “We now arrive at the 10 principles. \*Read the 10 principles\*  These are the major caveats that we want developers to emphasize. You do not have to memorize them, but hopefully with enough practice they will become second nature. They will also be referenced later on in this presentation” |
| **5** | “In addition to the principles, are some subcategories of coding standards. The former is a sort of philosophy or mindset, while the standards emphasize technique and practice. The most important practice that we would like to emphasize is to not hard core sensitive information. We don’t want black hats to have an easy time perusing our data. Some more important practices are excluding user input as often as possible, preventing freed memory from being accessible, and checking for integer wrapping. All very common things that I’m sure most developers think about. We also want to incorporate some new and perhaps familiar strategies, such as assertions and exception handling. And of course, ensuring that data behaves as predictably as possible by avoiding problems that our compiler will USUALLY (but not always) warn us about. |
| **6** | “So as many of us know, data moves between three states: at rest, in flight, and in use. Data at rest is as encrypted and hard to crack as can be without the correct cipher. Data in flight is when it travels through a network channel from one system to another, and data in use is when that send data is opened and used. Encryption is a huge part of cybersecurity, and our plan is to use several different forms across each stage of data transit. The precise form of encryption/decryption will be discussed later and will most likely use asymmetric keys.” |
| **7** | “Now for the triple A framework. This is a system that I’m sure many of us have intuited as standard, but we want to emphasize some aspects of it. First, authentication will continue in the traditional approved username/passcode format. These will be updated periodically for obvious reasons. Authorization is the clearance level of each account and will determine what system tasks each company employee will have on a principle of least privilege basis. If you require certain privileges, please feel free to contact any system administrator. The last implement of this system is accounting, and that’s mostly our job. We must be monitoring all activity that occurs on our system to ensure that no suspicious events take place. Session statistics and usage information WILL be logged, so browse the internet at your own discretion.” |
| **8** | “Part of our new security policy isn’t just forcing all company officials to memorize a bunch of fancy rules, we want to offer resources that will ease the workload for our developers. Automation is the future, and we want to use a piece of that in our coding practice with some unit tests. These are simple library additions and declarations that will be easily referenced, but to demonstrate-  \*Slide 9\* We can use simple unit tests to make assertions about our logic as programmed. This first test is the simple pointer test to ensure that it does not hold a null value.  \*Slide 10\* We can do things like test for limits about data structures like arrays. You can either assert that there is room, or assert that the size is a certain value.  \*Slide 11\* We can also check for clean deletion. With all the various languages, libraries, and processes all running simultaneously it can be quite a mess to keep track of everything. So for important things such as deletion, we can ensure that all potentially manipulatable objects are as they should be.  \*Slide 12\* Negative tests can also be used! If we expect a certain result to produce an exception, we can even test that this may be the case. Negative tests can be useful, but it’s up to the developer to really make the most of them.  \*Slide 13\* And if by chance you haven’t heard me say it yet, bounds checking is still a very important practice that we can actually use assertions for! A very helpful tool that can help detect any potential buffer overflows. |
| **9** | “We now transition to automation as a whole. To quickly summarize, we can see the pre-production loop which consists of planning, designing, building, and testing. This loop then transitions into the post production of health checking, monitoring, responding in the event of attacks, and maintaining stability. All these stages are wonderful, but you know what it could use?  \*Slide 15\* A little bit of automation. Automation is the use of tools that perform their tasks to a very high degree of accuracy and efficiency. So while our tool chain does work great, we want to apply tools and techniques to each area that we can. Some ideas presented here are adding security and risk assessment to our planning stage, automated testing tools for the design and testing stage, etc. For our post-production loop, we want to be as conscious as possible about following our new protocols and emphasize detection and prevention. We want our security and our development team to be as well equipped as possible for any eventualities.” |
| **10** | “People may be questioning whether “security starts at the beginning” is going to be the right direction to go, and we will be the first to admit that not everything is perfect. While waiting until later will allow strict focus from our development teams and probably create a much more expedient final deliverable, there will still be a disparity between our developers and our testers. Bugs are inevitable, but by collaborating immediately we can expect a more dynamic workplace atmosphere, the overall number of bugs created to decrease, and the number of bugs prevented at all to increase. Timelines will be affected, and for some of you these new tools will require learning, but we need you to trust us that being thorough is the best approach to ensuring that your time is not effectively wasted in the event of a security breach. |
| **11** | “The fact of the matter is, it will always be us against ‘them’, referring to black and gray hats of course. We are in an ever evolving war that will require us to be adaptable and willing to learn. Our security policy also can’t cover everything, because there are threats beyond our scope that haven’t even been unleashed yet. If you are a person with access to our data center, I highly recommend that you effectively manage all personal devices and accounts, and you validate the identity of any suspicious persons trying to contact you. We don’t want any instances of phishing to occur while we’re home for the day. We also can’t predict when real world events will take place, such as the pandemic or potential wars. As we’ve experienced, many of us had to work from home and this can pose a great risk if we all aren’t extremely careful. I highly recommend that if you are a remote employee that you use a reliable VPN or protective measure to encrypt your internet activity. Like I said, we just have to be flexible and communicate if we are to continue adapting and surviving in this day and age. |
| **12** | “Remember that we are all human, and those of us that are here together are on the same team. We don’t want to impose rules to be seen as restrictions, but reinforcements. As time progresses changes may be implemented, but by having a policy that we can all support, it will give us a fighting chance for longevity. Please don’t be afraid to talk to any of us if you have any questions, and don’t be afraid to talk amongst each other if you notice someone doing something that you believe may be compromising. Remember to work hard, be vigilant, and keep our secrets. Change is hard, and uncomfortable. But it is this discomfort that lets us know that we are growing. Thank you everyone.” |
| **13** | “All standards in this presentation were referenced from the SEI CERT C++ Coding Standard, a website which goes far beyond the principles mentioned. Feel free to investigate them for more secure coding practices.” |