CS 405 Project Two Script

Slide Number	Narrative
1	Hello and thanks for joining for our presentation today, my name is Bradley Jackson and I will be presenting the new Security Policy for Green Pace.
2	Apologies for the size of the image here, if anyone has issues viewing the images during the presentation the slides will be available afterward. So, here we have a visual representation of defense in depth (DiD). You can see layers of security such as physical, cloud, perimeter, network, and so on. So why are these layers important? Each of these layers helps to protect the business-critical data on which we operate. Currently, Green Pace has a very minimal implementation of DiD. We hope to change that with the presentation of uniformly defined, implemented, governed, and maintained security principles.
3	This slide shows our threat matrix. From top left to bottom right, we have threats which would likely appear sooner rather than later. And conversely, in the bottom right we have threats unlikely to appear which we plan to address when there are lulls in daily activity. We can see where each of our code standards falls within this matrix.
4	Here we have our 10 principles and each of our code standards falls into at least one. We have standards for the validation of input data, such as validation of array length in C++. We also have a number of code standards which would potentially appear as warnings when compiling C++. The majority of our standards fall into the category of effective quality assurance techniques. I want to highlight this because it is very important in practice. TEST YOUR CODE, PEOPLE. It is not hard. You can pretty easily test it against the standards listed in our policy doc. In fact, there are even examples spelled out! By the way, our policy document will be released along with the slides and like I said, there are more way details about each principle
5	When reading through the new policy, you may notice that there are P's after the standards. These P's are priority levels assigned to each code standard based on their severity, likelihood, and remediation cost. Also, as I mentioned before, the standards are mostly in C++ but apply to any language built on top of C or languages that have similar mechanisms.
6	Next, let's talk about encryption policies. Encryption in rest protects stored data which takes many forms, which are probably known to you, such as encrypted hard drives, USBs, etc. There are also encrypted storage options in the cloud for any cloud people in here. Encryption at flight, or as I call it in transit, are things like SSL or TLS which use certificates to trust websites and encrypt data as it is sent over the wire. Encryption in use involves things like encrypting a database.

Slide Number	Narrative
	Note that encrypted databases can be a little slower but obviously if the data is precious to the company, it may be a necessity. This is one of those places where you have to weigh the cost vs risk but those first two types of encryption, at rest and at flight are fairly low cost and performant.
7	Ok, Triple-A, this is one of my favorites. In a previous job I was heavily embedded in the identity space. The simplest form of authentication isa username and password. Hopefully most of you have set up MFA for your userids here at Green Pace. We all have to do our part. Authorization is when I delegate access roles to your account after you have authenticated. We want to keep access per user to a minimum, they do not need access to things they do not directly work with on a day-to-day basis. Some companies even do time-based access to certain sensitive resources. Accounting is something we all have to do our part in as well. If there is a huge breach we want our leadership to know we covered our rear ends and theirs.
8	Here are some examples of positive and negative unit tests for element acces here. I won’t go into too much detail here other than the fact that these are combating index errors and a few different ways. Remember to use emojis to make your code more fun and readable, it really works!
9	Automation should be a part of the build and verify phases of pre-production. For the purpose of this document we will assume no SAST or unit testing were done by developers. The design process should heavily involve security and require approval to move to production. It is possible to incorporate parts of the security testing and approval process with automation tools. For example, static code scans should be required as well as validation from the development team that testing has been completed. In terms of automation within the CI/CD pipeline, GitOps can be utilized to implement automated code scanning tools when commits are made to internal repositories. This will add time to the build process; however, it is a fundamental way to assure compliance with common coding standards. These tools would scan the source code of an application and report any known malpractice and vulnerabilities. Automated gatekeeping processes should prevent applications with known vulnerabilities from being deployed in production until the findings are remediated. Once an application is in production, maintainers should be alerted if new findings are revealed. For example, the log4j vulnerability was discovered after many affected applications were deployed within production. In the case of such a post-mortem finding, teams should be given notice and a period within which to remediate the finding. If application architecture plans to be changed, teams should need to submit these changes in an automated fashion to information security for further review.

Slide Number	Narrative
10	The tools I have here for automation are various SAST tools. There are seemingly limitless amounts of these out there, guys. And there are tons which are super well-known within the industry. Checkmarx, for example, is one you could hook into your CI/CD pipeline to do some static code scans. Cpp check is another well-known SAST specifically for C++. I put clang here because it is, at a very low level, a part of the CI/CD. I mean your local development environment is as low as it gets, right? What was that joke– Localhost– no, 127.0.0.1 is where the heart is? Ha. Anyway, there are a number of gates you could add to your CI/CD in order to automate away the majority of malpractice findings.
11	Alright, risks and benefits. We have talked about this a little already but it is very important to consider the financial and reputational risks of ignoring security policies. One mis-step on Green Pace's part and the entire company could be known for being the "those guys" of the year. No one wants to be in the news for a breach especially if peoples' private data is leaked, that is a big no-no. SO to prevent the majority of these we have to weigh the cost versus risk. The benefit of doing this is we are compliant, we continue to do business, and we are effective in our SOC/SIRT ops. One example of malpractice is Sony. Back in 2012 the Europe division of Sony had a private information leak for over a million users just due to an SQL injection attack. That is something we outlined in the new security policy for Green Pace since it is timeless but, trust me, it is completely avoidable.
12	Recommendations for the policy going forward are that the policy remain a living document. Even though we have those timeless things like SQL injection, etc. we can't just expect everything to stay the same. We are working in one of the most fast-paced industries of all times and have to adapt constantly. Green Pace could continue to hire third-parties for audit and review of the security policy. That is a good one. More eyes and transparency is great for ensuring operational excellence. Also, we are working on a roadmap to implement everything I have talked about thus far. Another thing which is not in the slide that I would like to mention is that feedback from employees is welcome. If you have a great idea or a finding we may have overlooked please let us know and we will work with you to get that all fixed up and documented for future devs.
13	Don't leave security to the end, guys. Systems need to revolve around security these days. It is so easy to overlook the simplest things and you can put your company in a bad position for the future. At the lowest level, preventing threats begins at securing business-critical data. And without the data, we cannot operate.

Slide Number	Narrative
14	Here are the references if you want them, like I said they will be included in the slides we send out. Thanks for your time and hopefully we can all work together on this. See you later! (smile and wave at zoom meeting participants 🙂).