Security Policy Presentation transcript

[Sam Dayball 4/13/2023 Project Two: Security Policy Presentation](https://www.youtube.com/watch?v=T6jOZIYEZhc)

Slide 1: Intro

good afternoon everyone thank you for attending my presentation today my name is Sam Dayball I'll be going over our new proposed security policy presentation for today throughout this presentation I will be discussing our defense and depth tactics as well as our threats matrix coding principles standards and other additional policies as well as tests

Slide 2: Overview

defense in depth refers to protecting all of our assets via multiple levels of security measures this can be explained by you know the physical security aspect you know on premise on premises security systems like alarm systems closed circuit TV systems and access control then the next layer of defense we can take would be cloud security you know by securing our own our own cloud making sure that we go through supplier assurance policies then moving next layer down would be perimeter security this would involve you know having remote access security penetration tests firewalls patch management etc. then moving forward we have network security this would be done in methods of having network compliance vulnerability scanning again patch management is key and then next the next layer would be host security having some additional vulnerability scanning patch management again and then this is where encryption comes in into play moving forward we have endpoint security again more layers of encryption vulnerability scanning AI and AV malware protections MDM's and DLP's then finally we have app security so this really boils down to having again more patch management more encryption more vulnerability scanning along with SLA’s and SDLC’s all of these layers to help protect our critical assets and system data

Slide 3: Threats Matrix

Moving on to our next slide here discussing threats matrix so we have several different threats that we can present in this room of discussion this is where we'll start to introduce our our coding principles you'll see here i have a list of you know likely threats you know principles that need to take priority and then and we will move more in depth on this topic here in just a second.

Slide 4: 10 Principals

On this slide here you'll see that I have listed our 10 coding principles along with our corresponding coding standards of this list includes validating data input heating compiler warnings architecting and designing for security policies keeping it simple default deny policies adhering to the principle of least privilege sanitizing our data as it's sent to other systems practicing defense in-depth using effective quality assurance techniques and adopting a secure coding standard uh we'll get into a little bit more about all of these here in a short bit

Slide 5: Coding Standards

That brings us to our coding standards um so again data type that as our first coding standards we need to ensure that any unsigned integer operations do not wrap I've listed this as a very high priority coding standard as it does pose a serious risk to not adhere to this data value again another high priority item this excludes user input from format strings once again very important to adhere to this coding standards as it could potentially expose our data next we have string correctness do not attempt to create a standard string from a null pointer again this is listed as a high priority next we have sequel injections do not alternately input and output from a stream without intervening flush or positioning call again this is this is our first low priority coding standard next we have memory protection this meaning to detect and handle memory allocation errors again this is another high priority item as it could lead to operating issues with our code next we have assertions do not read uninitialized memory this is again high priority and I have this tagged as most probable so this is a has a has an extremely high likelihood of running into security issues next is exceptions this is very straightforward handle all exceptions I have this listed as a low priority essentially this is to keep all exceptions and warnings handled by our developers we need to not ignore exceptions just because they are not warnings or errors but we need to instead handle all of these as we're developing next we have object oriented programming meaning to honor replacement handle handler requirements another low priority item next we have thread safety preserving thread safety and liveness when using condition variables again another low priority one but it's still important for us to adhere to and then finally we have containers use valid references pointers and iterators to reference elements of a container and finally this is our another high priority coding standard

Slide 6: Encryption Policies

Next I want to talk about our encryption policies there's several places or several forms are data can be in or the first one I want to talk about is encryption in rest this means we're encrypting data even when it's not being used or being transferred centrally data is resting on a hard drive for example and it should be encrypted in or password protected or both in case there were to be stolen then whoever were to steal it or hack the device without a decryption key which would be extremely challenging. Next we have encryption at flight what I mean by this is any data that's moving over a network or our cloud based organization encrypting this data is extremely vital as these networks can potentially be hacked and monitored by any unknown attackers so non encrypted data that's moving through this network could easily be stolen and used against us or sold. And finally we have encryption in use this is essentially the policy that is ensuring that any data no matter what stage it's in whether it be in use or idle is encrypted so that it can't be stolen decrypted or at any point author accessed by an unauthorized individual

Slide 7: Triple-A Policies

Next this brings us to our AAA policies AAA policies refers to authentication authorization and accounting First off authentication refers to having users use a valid login their credentials being username and password preferably we could use even two factor authentication for an extra layer of protection so that just in case any user credentials stolen next we have authorization which is referring to maintaining authorized personnel who only have the permissions that are accurate to their needs in our systems this prevents any users from having too many permissions which could lead to a security risk in case their credentials are stolen and finally we have accounting this is referring to our need to continuously audit our systems and our list of user credentials should be updated and maintained frequently we should also audit our permissions to make sure no changes have been made to user credentials or user permissions that were unauthorized and finally to double check to make sure that any users that are current in our system are current employees and need access if they are no longer with our organization or no longer need access we should remove them from our list

Slide 7: Unit test 1

This brings me to my first unit test to display a unit test that verifies that the erase function with parameters begin to end erases the collection so as you can see in the code that I've provided a snippet of we created a test to add entries to our collection this added five entries into our collection then next we used the begin to start the parameters at the beginning of the collection then we set end to the end of the collection and then directed the erase function to erase the collection from beginning to end finally we use the assert true and assert EQ to verify that the collection is empty and verify the collection size remains as you can see from the snippet from the console output you can see that the test was successful

Slide 8: Unit test 2

Next we have this second unit test that was run the question here is can we insert the query and find data from the snippet you can see here that we created a collection assign the integer data type and we inserted values along the collection here you can see 4213 and seven then we used our assert true calls to find that data and the collection from the snippet you can see here that this test failed

Slide 9: Unit test 3

The next unit test poses the question does resizing decrease our collection so here you can see we created a test to verify that resizing does decrease the collection to start we added five entries to our collection then we saved the original collection size next we resize the collection to a smaller size by initiating new size equaling the original size minus two then we checked the collection to resize to the new size that we defined finally we verified using the expect EQ and expect LT calls to check the collection size with against the new size and against the original size from the console output you can see that this test was successful

Slide 10: Unit test 4

This brings us to our final unit test the question posed here is can we increase the reserved space without increasing the collection size so first we wanted to reserve space for 10 elements in the collection next we wanted to verify that the capacity is increased but the size hasn't so we used our expected GE collection to change to check the capacity making sure that the capacity remained at 10 then the expect EQ call we used to check that the size remained the same next we added five elements to the collection and then we verified that by doing so the size had increased but the capacity has not increased as you can see from the snippet we do verify that the that the test was successful

Slide 11: Automation Summary

Next I'll be discussing our automation summary and the tools that we can use to assist us in the automation of our operations so from the diagram you can see here we want to start with assessing and planning our production so this would include discussing threat landscapes regulatory changes at running change impact analysis prioritizing backlogs and responding to new threats so there's a lot in that first step next this takes us to the design element so especially for security purposes we want to do test driven designs and then apply our best practices next we would begin building this would be a you know building a secure build using trusted repositories and secure open source usage so any open source code that we use we need to validate that it is a secure source finally in our pre-production cycle we wanted to move into our verify and test stage we want to do vulnerability scanning use trusted source anything that's digitally signed and functional and then we want to check against compliant compliance and security testing and as you can see part of this diagram you'll see three different cogs here development security and operations these are the three gears that are going to be driving our success in automating our our procedures next we move into the production cycle here we want to transition and do a health check this just means two we need to configure and deploy our systems I'm going to check our security settings and run penetration testing once we finish with that then we're going to move into our monitor and detect stage we're going to log collections are going to run security analytics will use event alerting and intrusion detection next we move into the respond phase so this includes blocking attacks turning off services and rolling back so if any attacks were to occur we want to use these tactics to lock down any potential threats finally once we have the problem quarantined we want to maintain and stabilize so we want to assess our security baseline again if we need to we'll return back to our baseline and return our system back to a stable state after the attack or compromise and then again we will move back into our pre-production phase of assessing and planning to revise and rebuild our system if need.

Slide 13: Tools

We should begin implementing automation in this cycle beginning in pre-production all the way through our production cycles here we'll need to assess and plan effectively in order to ensure that the rest of the procedure moves fluidly we'll need an all hands on deck approach to plan and prepare for any new threats or security should be implemented or improved we'll also want to implement automation in the verification and testing we can do this through several of the automation tools listed in this policy document and as it moves through pre-production to production we should look to automate the transition and health check as we as well as monitoring and detecting stage manually monitoring and detecting can be cumbersome and utilize lots of manhours so automating this will save us a lot of additional work however response and maintenance should 100% be left to a manual process as proposed in the policy document astray and code sonar are two of the highest recommended automation tools that can help us address the biggest variety of our needs these apply to the widest variety of our coding standards and principles so using these tools would be highly recommended

Slide 14: Risks and Benefits

So what are the risks and benefits that we're facing the risk to not acting now as far as revising our security policy is the longer we wait the more prolonged exposure we have to external threats a lot of these external threats take time to initiate so the longer we wait the longer we're giving the attackers time to break into our databases and steal our data and of course the more exposure means we're at a higher risk and the higher risk we're at means the less secure we are and this and less secure our clients and customer data is so if we act now we can begin to protect data more effectively and if we do it now then it will be less work in the future meaning there will be less systems to revise and correct once we have our security policy in place

Slide 15: Recommendations

This brings me to my recommendations So what I would recommend is implementing automation tools that I have recommended for each of the standards we need to also utilize AAA policies and data encryption in every element where it's possible next we need to revisit this policy as frequently as possible to address new threats or weak points as we know cyber threats constantly change and evolve so this document and security policy needs to change and evolve as frequently as possible to stay up-to-date against new threats

Slide 16: Conclusions

This brings me to my final conclusions First off being encryption of our data in all forms is vital this means data in rest data in flight data in use all needs to be encrypted at every point next using our AAA policies like authentication authorization and accounting these practices must be used and implemented ASAP in order to keep us safe next conclusion automation is the key to providing our defense in-depth strategy the most effective structure and finally constant updates and maintenance will be needed perpetually