Dale Ayers

SNHU

12/18/2022

[dale.ayers@snhu.edu](mailto:dale.ayers@snhu.edu)

**Project Two Script**

*Video:* [*https://www.youtube.com/watch?v=XouhzODFgEI*](https://www.youtube.com/watch?v=XouhzODFgEI)

**Introduction**

Hello, my name is Dale Ayers, and today I am going to explain cybersecurity concepts to you through the use of… CASTLES, and other medieval iconography. In this project, you will assume I am presenting a cybersecurity policy to a global software company specializing in software for companies interested in green energy.

**Defense in Depth**

* According to Ray Dunham, contact auditor for a leading CPA firm, there are specific goals that a security policy should strive towards. "The goal when writing an organizational information security policy is to provide relevant direction and value to the individuals within an organization with regard to security" (Dunham, 2021)
* Security Policies are but one piece of the defense pie
* Without a company-wide buy-in to a philosophy of security, our technology becomes useless

Defense in depth reminds me a lot of a castle. When a castle is constructed, its not just one wall it's a moat, and a wall, an inner wall with battlements, arrow slits, stores of food, secret passageways, the bailey, and the draw bridge. Broadly the idea behind defense in depth and a castle are the same; each part of defense has its own job to perform. Not one tool can accomplish the whole job, so we build one thing on top of each other, covering for eaches strengths and weaknesses.

**Threats Matrix**

When thinking about a threats matrix, think that you are a lord in medieval Europe, and you need to identify possible attacks from the surrounding lords. You might anticipate an attack from a rival lord but deem one from a different lord unlikely or not a threat. The same can be applied to cybersecurity, some threats are anticipated, depending on the type of organization you are in, and some are deemed unlikely or not a threat. The four main threats that we are worried about as a global *green* software company are ransomware, social engineering, remote working risks, leaks through poor communication, and encryption standards.

Everyone should be worried about ransomware and social engineering. Ransomware is as it sounds, a malicious actor steals your data, doesn't allow you access to it then demands a ransom. Social engineering comes in many different forms, but phishing is a big one; the main idea is that people can be manipulated into giving away secure information. The other two play into each other. Because we will be providing software globally, there is a lot of remote work and communication risk. This means employees should have to follow specific security standards when communicating and accessing a remote working environment.

**10 Principles**

These ten principles will be like our chivalry standards for our knights. Depending on which department you work for a few of these rules will apply more to you than others, but everyone should know and understand each.

1. Validate Input Data

* Whitelisting data is the practice of only allowing certain types of input data. For example, it must be ten characters long and only consist of letters.

1. Heed Compiler Warnings

* Listen to the compiler; just because the code "works" doesn't mean that the code works.

1. Architect and Design for Security Policies

* The security architecture is a set of security principles, methods, and models designed to align with the enterprise's objectives to keep the organization safe from cyber threats. Because of the nature of Green Paces clients this standard becomes more nuanced and complex because each organization will have its own specific set of security needs.

1. Keep it Simple

* If a system is too complex, it's hard for everyone on the team to understand it, and worse security is the result.

1. Default Deny

* Think of a door that automatically locks when shut, this would mean you always need a key to open it, we should operate the same way in our coding standards. By default, you will always need a key to access any part of the system.

1. Adhere to the Principle of Least Privilege

* Just because I can log in to my bank account doesn't mean that I can log in to yours. Access to systems should be limited to the minimum amount required to accomplish the job.

1. Sanitize Data Sent to Other Systems

* Just because you delete something doesn't mean you *deleted it.* Information can stay hidden on computers for a long time; any company equipment containing sensitive information must be dealt with appropriately. This could mean destroying it in some cases.

1. Practice Defense in Depth

* Remember the castle

1. Use Effective Quality Assurance Techniques

* Rigorous testing for known vulnerabilities could include unit or beta testing, debugging, etc. Because of the constant changing of they threat landscape it's impossible to lay out the most concrete examples of QA, but rigorous testing should be at the forefront of any QA policy.

1. Adopt a Secure Coding Standard

* A set of rules and standards that help prevent, detect and eliminate known vulnerabilities. Each software dev will have their own personal style but it's important that they know some of the standards and practices that help us work better together as a team.

**Coding Standards**

Each of these coding standards follows one or more of our 10 Principles. For example, number one is data validation; number six is using effective QA techniques; numbers five and ten are utilizing the principle of least privilege. I ranked each one on how easily I believed it could be hacked. Each organization will have its own system for ranking the risk of each coding standard.

1. Do not cast to an out-of-range enumeration value.

2. Do not pass a nonstandard-layout type object across execution boundaries.

3. Use valid references, pointers, and iterators to reference elements of a basic\_string

4. Guarantee that storage for strings has sufficient space for character data and the null terminator

5. Do not access freed memory

6. Use a static assertion to test the value of a constant expression

7. Catch exceptions by lvalue reference

8. Range check element access

9. Do not store an already-owned pointer value in an unrelated smart pointer

10. Close files when they are no longer needed

**Encryption Policies**

Encryption should be used at all stages of data use, that is

At rest

In flight

In Use

To illustrate this concept I've employed the use of the battle of the seven harrys from Harry Potter. In this scene they must take Harry from a safe house where they know hes safe, transport him to a different safehouse (in flight) where he can perform his wizardly duties (In use). Encryption will be used at all stages at Green Pace.

**AAA**

Authentication

* Who Goes there? This can be understood by logging in or two factor authentication.

Authorization

* Where am I allowed to go?
* What am I allowed to do?
* If I was allowed into the castle to be a cook, that doesn't mean im allowed to freely move about the castle going into whatever room I want. I'm default denied entry to only the rooms I'm expressly permitted to be in.

Accounting

* Who was here?
* What did they do?
* All activities inside the castle are logged and monitored

**Unit Testing**

Isolating a small piece of code so that it can be logically tested

\*show and talk about examples

**Automation and Tools**

"Security automation is the automation of security tasks, including both administrative duties and incident detection and response. Security automation provides numerous benefits to the organization by enabling security teams to scale to handle growing workloads" (Chkadmin, 2022).

According to Tomasz Nidecki a Cyber security writer with 25 years of IT experience, "DevSecOps is a set of practices that combines software development (Dev), security (Sec) and IT operations (Ops). It is an extension of DevOps. In practice, it means testing the security of software as part of continuous integration (CI) or continuous deployment (CD)" (Nidecki, 2022).

Security tools should be used continuously

Security should not be "left to the end"

Everyone should be thinking about security in every stage of development.

**Risk & Benefits**

Risk Management is about prioritizing risk

Identifying Risk

Assess Risk

Control Risk

Review Controls

"According to a 2017 survey conducted by the Ponemon Institute, companies waste an average of 425 hours a week responding to and investigating false positives, costing them an average of $1.37 million, annually" (Vidal, 2020).

If we were in medieval Europe an example of a risk like this would be keeping the town fed. If you're the only one eating, then the town peasants may be outside your castle with torches and pitchforks.

**Recommendations**

Perform Cyber Security Gap analysis- how can we improve?

Identify a Specific Industry Framework-what industry are we in?

What are the specific industry risks?

Evaluation of People and Processes- why aren't the people being fed?

Are we doing the right things day in and day out?

Data Gathering and Analysis

Evaluate our practices against a cybersecurity framework to best maximize our effectiveness

At Green Pace this could mean anything from better communication standards, holding people accountable for not being compliant, and increased unit testing. Once these frameworks are understood, its easier to know how anyone can improve their security policies.

**Conclusion**

Don't *"leave security to the end."*

Foster a Culture of taking security seriously. We can have the latest state of the are tools and frameworks but if our culture is lacking, then those tools become useless. (if a knight doesn't take his job of vetting who comes through the gate seriously, then what use is a drawbridge and a moat?)

Implement the latest state-of-the-art tools and frameworks

References

Chkadmin. (2022, May 11). What is security automation? Check Point Software. https://www.checkpoint.com/cyber-hub/cyber-security/security-automation/#:~:text=Security%20automation%20is%20the%20automation,scale%20to%20handle%20growing%20workloads

Dunham, R. (2021, October 26). The importance of a company information security policy. Linford & Company LLP. https://linfordco.com/blog/information-security-policies/

Nidecki, T. (2022, February 28). DevSecOps vs. SecDevOps. Acunetix. https://www.acunetix.com/blog/web-security-zone/devsecops-vs-secdevops/

Vidal, M. (2020, March 4). Low-priority threats: The Achilles heel of security teams. Security Boulevard. https://securityboulevard.com/2018/03/low-priority-threats-the-achilles-heel-of-security-teams/