CIS-481: Introduction to Information Security

InfoSec Chapter Exercise #7

Team: 3
Participants: Marvin Weaver, Brandon Crosier, Aaron Gordon, Sam Steele, Kitrina Justus

Logistics

A. Get together with other students on your assigned team in person and virtually.
B. Discuss and complete this assignment in a collaborative manner. Don't just assign different problems to each teammate as that defeats the purpose of team-based learning.
C. Choose a scribe to prepare a final document to submit via Blackboard for grading, changing the file name provided to denote the number of your assigned **Team**.

Problem 1
Consider the logical access control needs for joint software development teams using a typical Linux environment. Roles must include Developers (that can commit changes made in the code), Testers, and Code Reviewers. The technical access control mechanisms that you design must reflect these organizational roles. Your access control solution must:

1. Protect the software being developed from outsiders stealing it
2. Protect against unauthorized changes (including from internal actors)
3. Ensure that we can trace *who* made each change

Situation 1: A small team on a single machine *(5 points) - discretionary*

Discretionary access controls would be the best access control for this situation. Discretionary access control allows users to share resources in a peer-to-peer environment and allow users to control and grant access to other users as needed. Since the team is so small, it doesn't need all of the features of the nondiscretionary access controls. They would know every person who needs access to data and what access they would need.

They wouldn't need version control software as much as bigger companies as they would likely check the changes to source code with each other before implementing the change. They also wouldn't need accountability software as much as a big company as they would be assigning tasks to team members and checking their code periodically. Since they don't need as much protection as a big company, we'd recommend using a website like Azure DevOps to assign tasks and allow the users to submit code reviews. That way, team members can get their code checked before implementing and users will have a list of who made what changes.

An intrusion detection and prevention system (IDPS) is also recommended as it will notify the team if there was an attempt to breach the data. While this may not seem necessary for a smaller team, it will keep the team updated on the happenings of its code.

Situation 2: A medium-to-large team on a LAN [Hint: Use of a version control system like Subversion is highly recommended] *(10 points) - non discretionary mandatory*

A medium-to-large team should use a non discretionary mandatory access control approach because it allows them to give certain users specific access to specific information without having to need a middle manager to watch over it constantly. We would also recommend them to use a version control system such as git because they offer data issuance which allows an organization to track all changes made to any file or project. This is made possible because you can't change any data in a Git repository without changing all the ID's of everything after it. This means that you can track who was the last to alter a file or project.

An IDPS, like Tripwire, is also recommended for a medium-to-large company as they would want to see if an attack was attempted on their data. Since they can't know everyone on the team, this would easily protect the data against unfamiliar agents who do not have access. A vulnerability scanner, like Core Impact, would also be recommended since it would show where their data is unprotected. Since this is a medium-to-large team, they would also have the resources to fix the identified vulnerabilities.

Situation 3: A large, distributed team, including outsourced contractors *(10 points) - non-discretionary role based*

Access controls for a large, distributed team should be nondiscretionary and role-based. Non discretionary access controls would allow for all access to be controlled by a central authority. In conjunction with non discretionary access control, task-based controls should also be implemented. Task-controls allow for access to be given to employees for job-necessary tasks. With a large company, working with outsourced contractors, not every employee needs to have access to all functions within the company. Task-based controls allow for a central authority to restrict employees functions to particular assignments. These types of controls would be perfect especially with outsiders having access to some company information. Positions and jobs get turned over so administrators can simply assign access to certain projects or revoke access once an employee is no longer on the project.

Similar to situation 2, an IDPS like Tripwire is also recommended for a large distributed team because they would be able to see any attempted attacks on any of their systems. In this situation, they still do not know everyone on the team and this recommendation would easily protect the data against unauthorized persons attempting to gain access. A vulnerability scanner would also be recommended because it would expose any unprotected data for the person or persons who have the security responsibility. Since this is a large team, there is no doubt that they would have all the resources ready and available to fix any uncovered security issues.

[Inspired by https://www.cs.columbia.edu/~smb/classes/f09/l08.pdf - many thanks to Columbia University for providing under Creative Commons!]