Preliminary Report

Andree Salvo
Southern New Hampshire University
CYB 240-13711
3-2 Project Two Stepping Stone
Instructor: Brian Remson

The first proactive control is **Injection**. This occurs when an attacker can inject harmful code into an application, such as through a form or input box, and the app executes it. This can lead to numerous problems, including data leaks, unauthorized access, or even complete control of the app. SQL injections are one of the most commonly known types of attacks. To prevent this, we must validate and sanitize all user input, with no exceptions. Using **parameterized queries** is a must, it separates user input from the actual code, so the app doesn't just run whatever gets typed in. Additionally, setting up input validation rules and avoiding dangerous characters is also helpful. These are simple controls, but if we had added them in an earlier stage, we wouldn't have had to panic about fixing it later.

The second control is **broken access control**. Broken access control occurs when an unauthorized user can access resources that they shouldn't be able to, such as admin pages or other users' data. This is a very big problem. To prevent this, we must always adhere to the principle of least privilege, which means granting people only the necessary access to what they need, nothing more, and nothing less. We deny access by default and only allow it when required. We should also implement access control checks on every request, not just on the front end, and maintain a clear access control list to define who can perform which actions that need to be done.

Another proactive control that's also just as important is **Security Misconfiguration**. This occurs when default password settings, unnecessary features, or open ports are left in place, making it easier for an attacker to gain access. For example, leaving the admin console open or failing to turn off debug mode on a live app. These small things can create significant vulnerabilities if they're not handled properly. To avoid this, I recommend ensuring that our servers, databases, and applications are configured adequately before processing. That means using **secure settings from the start**, removing anything we don’t need, and keeping everything up to date. We should also **automate security checks** where possible, such as scanning for outdated software or misconfigured permissions.

In conclusion, incorporating these protections from the beginning makes everything easier later and reduces the chance of major issues down the road.