Security First Principle Minimization README

General Information

- Author: Mahesh Krishnan Kalappattil
- Date: 11/27/2022
- Description: This presentation covers the importance of security first principle Minimization and how it improves security.

Why You Should Care

Minimization is one of the ten security first principles. Minimization defines to keep the functionality to minimum required by the program or service. The cyber threats on applications are ever increasing. The threats could be either leveraging an unpatched vulnerability or a zero day. The higher the network footprint, the larger the attack surface will be. Leaving the unwanted ports open enables an attacker more choices to initiate an attack and exploit the vulnerabilities if any. Hence it is important to close all unwanted ports that are not in use.

Minimization also applies to privilege. Running the program with elevated privileges increases the risk of severe damage when the system is compromised. Instead of holding the elevated privilege through the lifespan of program, the privileged mode should be requested only when needed and should be relinquished right after. This minimizes the risk of executing privileged commands or abuse of super user privileges when the system is compromised due to another vulnerabilty exploit.

Three Main Ideas

1. Minimization improves the security posture of application by reducing the attack surface.
2. Keeping unwanted ports closed helps reduce the network footprint
3. Using the elevated privileges only on demand basis reduces the risk of super user privilege abuse upon another vulnerability exploit.

Example

A program should implement multiple minimization techniques. It should not rely on just one minimization. For example, if an application binds to a port which is actually not needed, it may solicit attacks on the port. Shutting down the port in network firewall alone is an example of bad implementation of minimization principle. Instead the application should be modified to eliminate the binding to unused ports as well. This helps in multiple layers of defense. For example, even if the network policy changes in the future, the application does not end up exposing the unused port to the outside world.

Additional Resources

- Using setuid() to elevate privileges on demand
- Using iptables to close ports
- Port scanning using nmap