

Cyber Essentials

– Secure Configuration

Kevin Murphy

Aon Security

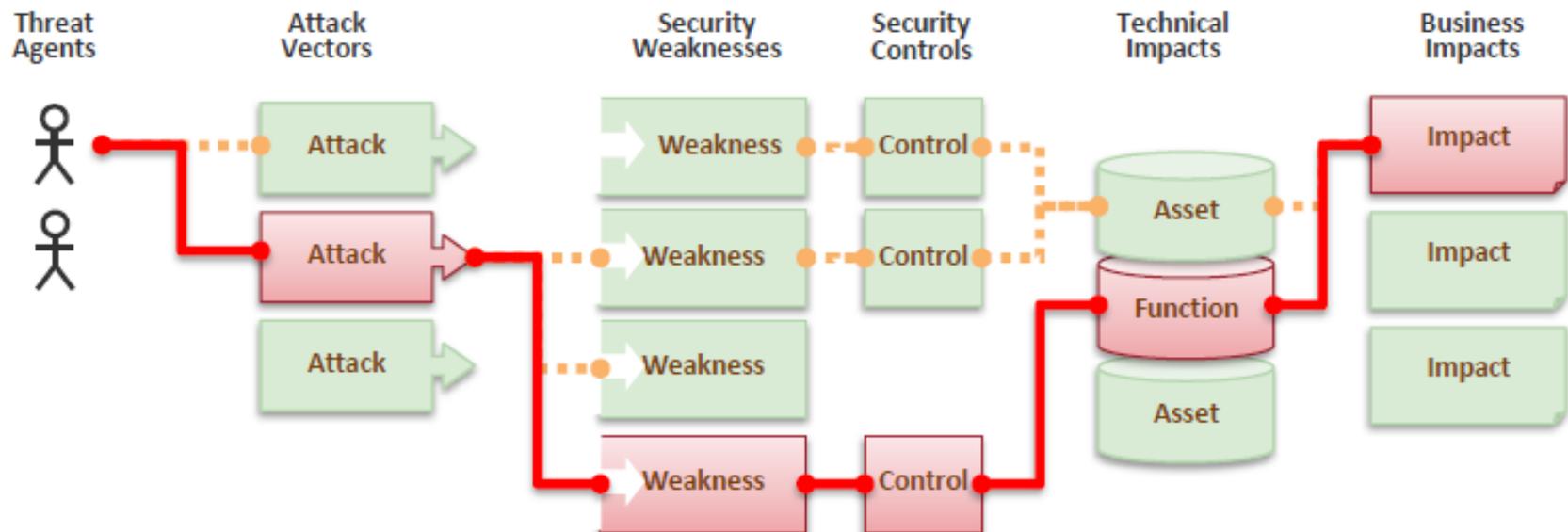
12 December 2016

Secure Configuration | Context

- Cyber Essentials defines a core set of controls which, when properly implemented, will provide organisations with basic protection from the most prevalent forms of threats coming from the Internet. In particular, it focuses on threats which require low levels of attacker skill, and which are widely available online.
- The Scheme focuses on Internet-originated attacks against an organisation's IT system.
 1. Phishing: malware infection through users clicking on malicious e-mail attachments or website links.
 2. Hacking: exploitation of known vulnerabilities in Internet connected servers and devices using widely available tools and techniques.

Secure Configuration | Objective

- Computers and network devices should be configured to reduce the level of inherent vulnerabilities and provide only the minimum services required.
 - Devices cannot be considered secure upon default installation. A standard, ‘out-of-the-box’ configuration can include administrative accounts with predetermined, publicly known default password; unnecessary accounts and applications enabled.
 - Default installations of computers and network devices can provide cyber attackers with a variety of opportunities to gain unauthorised access to an organisation’s sensitive information, often with ease.

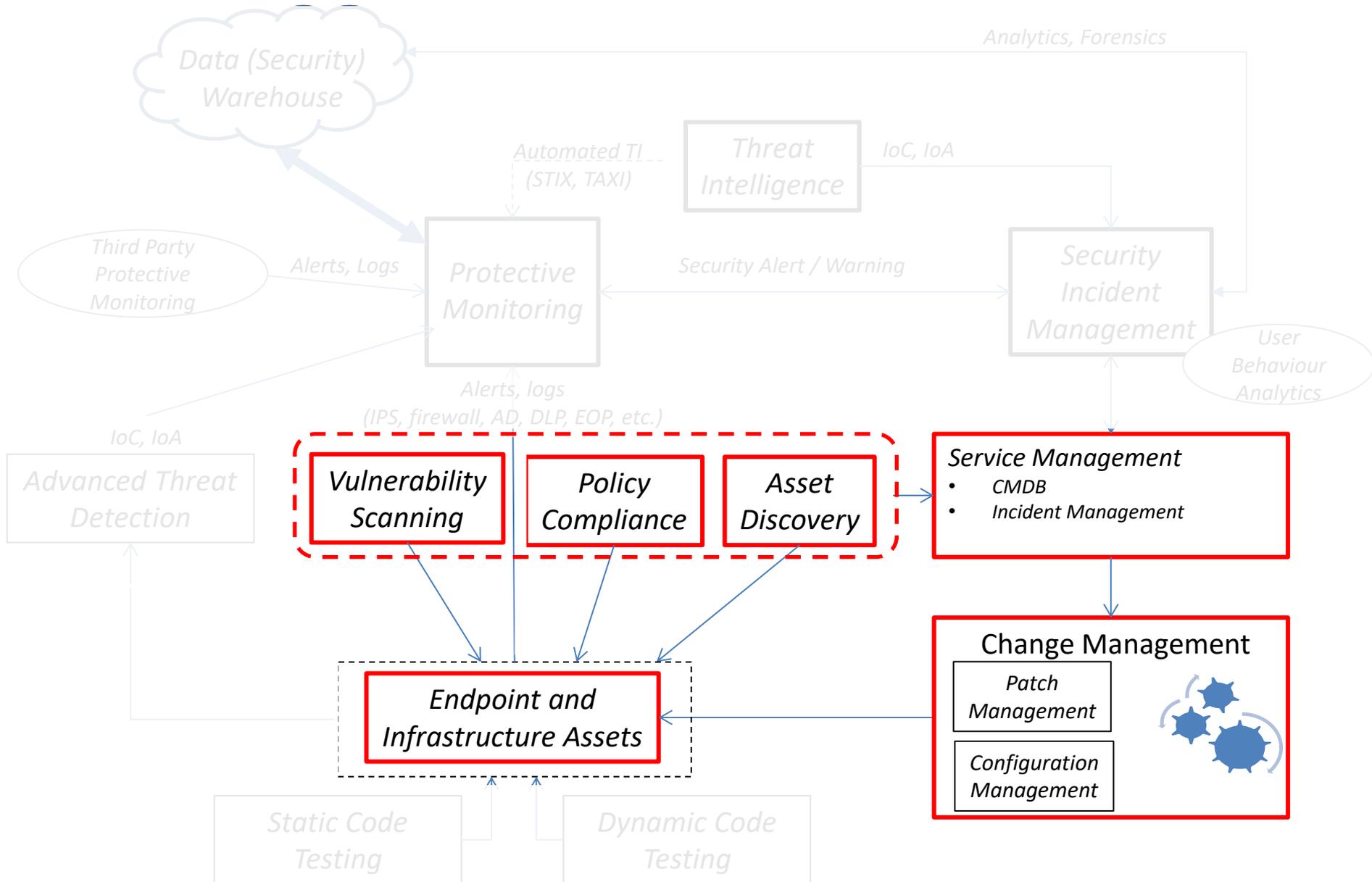


Secure Configuration | Requirements

Computers and network devices (including wireless access points) should be securely configured. As a minimum:

1. Unnecessary user accounts (e.g. Guest accounts and unnecessary administrative accounts) should be removed or disabled.
2. Any default password for a user account should be changed to an alternative, strong password. *Timeout; Lockout periods.*
3. Unnecessary software (including application, system utilities and network services) should be removed or disabled. *Lock down ability to install.*
4. The auto-run feature should be disabled (to prevent software programs running automatically when removable storage media is connected to a computer or when network folders are accessed). *Three clicks to run.*
5. A personal firewall (or equivalent) should be enabled on desktop PCs and laptops, and configured to disable (block) unapproved connections by default.
6. *Enable encryption for data at rest (storage)*

Secure Configuration | Operating Model



Minimum Baseline Security Standards

- Policy
- Standards
- MBSS
 - Applications
 - Databases
 - Infrastructure
 - Firewalls etc
 - Operating Systems
 - Windows
 - Linux
 - etc



Windows Operating System Server

Minimum Baseline Security Standard

Purpose

Windows Server is an enterprise operating system. This guide provides technical guidance intended to help system administrators and security officers improve the security of their servers. Using the information presented here, you can configure Windows Server systems to protect the integrity and confidentiality of the application environment.

This document is intended as the set of minimum baseline standards for securely configuring ("hardening") Windows Server, and for maintaining that minimum level of security. These standards must be implemented in concert with appropriate Aon Corporate policies and procedures.

Security Misconfiguration | OWASP

Open Web Application Security Project (www.owasp.org)
 - Top 10 Most Critical Web Application Security Risks (A5)

 Threat Agents	 Attack Vectors	 Security Weakness		 Technical Impacts	 Business Impacts
Application Specific	Exploitability EASY	Prevalence COMMON	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anonymous external attackers as well as users with their own accounts that may attempt to compromise the system. Also consider insiders wanting to disguise their actions.	Attacker accesses default accounts, unused pages, unpatched flaws, unprotected files and directories, etc. to gain unauthorized access to or knowledge of the system.	Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, database, framework, and custom code. Developers and system administrators need to work together to ensure that the entire stack is configured properly. Automated scanners are useful for detecting missing patches, misconfigurations, use of default accounts, unnecessary services, etc.		Such flaws frequently give attackers unauthorized access to some system data or functionality. Occasionally, such flaws result in a complete system compromise.	The system could be completely compromised without you knowing it. All of your data could be stolen or modified slowly over time. Recovery costs could be expensive.

Am I Vulnerable to Attack?

Is your application missing the proper security hardening across any part of the application stack? Including:

1. Is any of your software out of date? This includes the OS, Web/App Server, DBMS, applications, and all code libraries (see new A9).
2. Are any unnecessary features enabled or installed (e.g., ports, services, pages, accounts, privileges)?
3. Are default accounts and their passwords still enabled and unchanged?
4. Does your error handling reveal stack traces or other overly informative error messages to users?
5. Are the security settings in your development frameworks (e.g., Struts, Spring, ASP.NET) and libraries not set to secure values?

Without a concerted, repeatable application security configuration process, systems are at a higher risk.

How Do I Prevent This?

The primary recommendations are to establish all of the following:

1. A repeatable hardening process that makes it fast and easy to deploy another environment that is properly locked down. Development, QA, and production environments should all be configured identically (with different passwords used in each environment). This process should be automated to minimize the effort required to setup a new secure environment.
2. A process for keeping abreast of and deploying all new software updates and patches in a timely manner to each deployed environment. This needs to include all code libraries as well (see new A9).
3. A strong application architecture that provides effective, secure separation between components.
4. Consider running scans and doing audits periodically to help detect future misconfigurations or missing patches.

Example Attack Scenarios

Scenario #1: The app server admin console is automatically installed and not removed. Default accounts aren't changed. Attacker discovers the standard admin pages are on your server, logs in with default passwords, and takes over.

Scenario #2: Directory listing is not disabled on your server. Attacker discovers she can simply list directories to find any file. Attacker finds and downloads all your compiled Java classes, which she decompiles and reverse engineers to get all your custom code. She then finds a serious access control flaw in your application.

Scenario #3: App server configuration allows stack traces to be returned to users, potentially exposing underlying flaws. Attackers love the extra information error messages provide.

Scenario #4: App server comes with sample applications that are not removed from your production server. Said sample applications have well known security flaws attackers can use to compromise your server.

Secure Configuration



- A core security control
 - Protect from common, low capability threats.
 - Provides a strong foundation for building advanced security controls to address high capability attack

Useful References:

NCSC: <https://www.ncsc.gov.uk/guidance/10-steps-cyber-security>

NCSC: Security Design Principles for Digital Services; <https://www.ncsc.gov.uk/guidance/security-design-principles-digital-services-main>

NCSC: Common cyber attacks: reducing the impact;

https://www.ncsc.gov.uk/content/files/protected_files/guidance_files/common_cyber_attacks_ncsc.pdf

NCSC: Current and common threats and vulnerabilities; <https://www.ncsc.gov.uk/topics/cyber-threats>

Centre for Internet Security Benchmarks: <https://benchmarks.cisecurity.org/>

Open Web Application Security Project: www.owasp.org