Vulnerability Management in a nutshell

Intro:

Vulnerability Management plays an important role in an organization's line of defense. However, setting up a Vulnerability Management process can be very time consuming. This blogpost will briefly cover the core principles of Vulnerability Management and how it can help protect your organization against threats.

What is Vulnerability Management

To better understand Vulnerability Management, it is important to know what it stands for. On the internet, Vulnerability Management has several definitions. Sometimes these can be confusing and misinterpreted because different wording is used across several platforms. Several products exist that can assist an organization in creating a Vulnerability Management Process. Some of the current market leaders include but are not limited to: CrowdStrike, Tenable.IO and Rapid7.

According to Tenable, Vulnerability Management is *an ongoing process that includes proactive asset discovery, continuous monitoring, mitigation, remediation and defense tactics to protect your organization's modern IT attack surface from Cyber Exposure.*[1]

According to Rapid7, Vulnerability Management is the process of identifying, evaluating, treating, and reporting on security vulnerabilities in systems and the software that runs on them. This, implemented alongside with other security tactics, is vital for organizations to prioritize possible threats and minimizing their attack surface.[2]

According to CrowdStrike, Vulnerability Management means the ongoing, regular process of identifying, assessing, reporting on, managing and remediating security vulnerabilities across endpoints, workloads, and systems. Typically, a security team will leverage a Vulnerability Management tool to detect vulnerabilities and utilize different processes to patch or remediate them.[3]

Why Vulnerability Management

A well-defined Vulnerability Management process can be leveraged to decrease the cyber exposure of an organization. This ranges from identifying open RDP ports on internet-facing Shadow IT to outdated third-party software installed on the domain controller. In case vulnerabilities are abused by attackers, they could obtain access to the internal network, distribute malware such as Ransomware, obtain sensitive information, … And the list goes on. Decreasing your exposure and increasing patch management can reduce the likelihood of an attack happening on the organization's infrastructure.

Vulnerability Management core principles

If we take a look at the definitions above, several terms are being used over and over again. We can summarize Vulnerability Management in 6 steps. As Vulnerability Management is a continuous process, each individual step provides input for subsequent steps. It is important to note that this is a simplified version of Vulnerability Management. The following image illustrates what a Vulnerability Management process can look like:

---

[1] https://www.tenable.com/source/vulnerability-management
[2] https://www.rapid7.com/fundamentals/vulnerability-management-and-scanning/
[3] https://www.crowdstrike.com/cybersecurity-101/vulnerability-management/

Identify

Identification of the scope is the first part of the Vulnerability Management cycle. This is an important phase, as you can't protect what you don't know. If we take a look at the CIS Critical Security Controls[4], the first step to stop today's most pervasive and dangerous attacks is to "Actively manage (inventory, track, and correct) all enterprise assets" – meaning that it is really important for an organization to know what infrastructure they have. The first step in the Vulnerability Management program is to identify all known and unknown assets and start prioritizing them. This can include but is not limited to the following information:

- Which assets are most critical to the business?
- Which assets are externally exposed?
- Which assets have confidential information?

The process of identifying assets can be automized with a combination of discovery scans on the internal network and identification of known and unknown external assets through attack surface management platforms. This phase is a crucial part, as all next steps are based on the scope defined during the identification phase.

Assess

Assessing the infrastructure for weaknesses can be automated through vulnerability scanning with known scanners such as Tenable.IO and Rapid7 However, manual verification might be needed to determine the actual exploitability of vulnerabilities as vulnerability scanners do not cover all security controls in place such as specific workarounds that were implemented to limit the likelihood of exploitation. By using a combination of automated scanners and manual verification of the issues, a comprehensive view on what vulnerabilities are currently affecting your organization can be established .

[4] https://www.cisecurity.org/controls

Prioritize

Some organizations might not prioritize their vulnerabilities obtained by automatic scanners or penetration tests. However, as Seth Godin said: “Data is not useful until it becomes information”. It is the task of the Vulnerability Management team to prioritize the vulnerabilities not only on their actual technical impact but also to keep in mind the business impact. For example, a critical Log4J vulnerability on an externally available and well-known website should be remediated sooner that the same Log4J vulnerability on a lunch-serving testing server that is only accessible from the internal network.

Report

After all issues have been prioritized, an actionable report should be given to the teams that will actually perform the patching/resolving of the issues. It is important for the Vulnerability Management team to keep in mind that they should create actionable tickets or remediating actions for the operations team. A bad example of a ticket can be as follows:

Title:Log4J identified

Description: Log4J was identified on your server

Resolution: Please fix this as soon as possible

A good example of a ticket can be something like this[5]:

Title: Apache Log4j Remote Code Execution (Log4Shell)

Severity: Critical

Estimated Time to Fix: 1 hour

Description: Apache Log4j is an open source Java-based logging framework leveraged within numerous Java applications. Apache Log4j versions 2.0-beta9 to 2.15.0 suffer from insufficient protections on message lookup substitutions when dealing with user controlled input. By crafting a malicious string, an attacker could leverage this issue to achieve a remote code execution on the Log4j instance used by the target application.

Solution: Upgrade Apache to version 2.16.0 or later.

Affected devices: 10.0.0.3, 10.0.9.3

CVE’s: CVE-2021-44228

References: https://logging.apache.org/log4j/2.x/security.html

https://www.lunasec.io/docs/blog/log4j-zero-day/

https://www.lunasec.io/docs/blog/log4j-zero-day-severity-of-cve-2021-45046-increased/

[5] https://www.tenable.com/plugins/was/113075

Remediate

Resolving vulnerabilities should be the goal of the entire Vulnerability Management process, as this will decrease the exposure of your organization. Remediation is a process on its own and might consist of automatic patching, process updates, Group Policy updates, …. With the actionable ticketing performed by the Vulnerability Management team in the previous phase, it should be easy for the operations teams to identify what actions need to be done and how long it will take. After successful remediation, a validation of the remediation should be performed by the Vulnerability Management team. If the issue is resolved, the issue can be closed.

Improve

As Vulnerability Management is a continuous process, it should be reviewed all the time. A Vulnerability Management program is – like Rome – not built in one day. However, over time a robust and reliable Vulnerability Management process will be in place if the processes are well defined and known within the organization.