Vulnerability Management

Vulnerability Management is something we have been talking about for years. We have, as a community, built tools and created enterprise systems for scanning and classifying vulnerabilities. However, many organizations struggle to reduce risk and prioritize remediation efforts. There are several obstacles to effective vulnerability management, which this paper will identify and provide strategies for overcoming.

Let's start by looking at three examples of vulnerability management programs.

Example #1: Vulnerability Identification

Company A had a large network and a well-established program for scanning and tracking vulnerabilities. Over the course of a few years, the program grew from quarterly scans of just critical assets to monthly scans of the entire enterprise. Overall, the company did an excellent job of identifying vulnerabilities. They maintained a database of vulnerability information that grew to over 6TB! Unfortunately, they struggled to remediate the millions of vulnerabilities identified. They lacked accountability from the patching team, did not have a clear policy and strategy in place to drive actions and they did not have enough buy-in from senior leadership to get teams from other departments to work on remediation actions.

Example #2: What Management?

Company B had a very minimal process in place. They conducted quarterly external vulnerability scans, but had no program in place for internal scans. They did not track remediation or maintain any easily accessible database of discovered vulnerabilities.

Example #3: Metrics

Company C wanted to use the vulnerability management program to drive metrics for the security department. They produced monthly metrics on numbers of vulnerabilities discovered, systems scanned, percentages of enterprise scanned, etc. The numbers produced did a decent job of measuring how much work the vulnerability scanning team did, however the metrics were not very useful in driving organizational risk reduction. Because the metrics did not measure real risk, the organization had a mistaken picture of how exposed the company was to attack.

What is Vulnerability Management?

In general, vulnerability management is the process of identifying and remediating vulnerabilities. The emphasis of an effective program should be on risk. Although there may be good reasons to use the vulnerability management program to generate metrics, using metrics that focus on tasks (scans performed, systems scanned, vulnerabilities identified, etc) do not provide a good picture of real risk. Generating metrics that truly reflect risk is very difficult. Part of the difficulty is that it is very hard to directly connect any given vulnerability to the risk that vulnerability will be exploited.

Risk Rating

Understanding real risk is hard in the current state of vulnerability management. One attempt to address this problem is the CVSS score. Unfortunately, the raw CVSS score alone has not been a good predictor of actual exploitability. For example, a recent set of patches from Microsoft included MS15-

067, a patch for Remote Desktop. The patch is rated as Critical, and is naturally a good target for an attacker. It exploits a vulnerability in RDP that can allow remote code execution by sending malicious packets to a machine running RDP. Despite these factors and risk ratings, after more than 90 days there is still no known exploit for this vulnerability.

Some factors to consider when trying to determine real risk:

1. Deployment – how widely deployed is the software/OS with the vulnerability? How accessible are the systems?
2. Mitigating / Compensating Controls – what other controls in the environment help protect against attacks for this vulnerability (AV, IPS, Firewalls, etc)
3. Ease of Exploitation – can it exploited remotely, or within the system? What technical barriers exist?
4. Availability of exploits – vulnerabilities with published exploits in either ExploitDB or Metasploit are much more likely to be exploited. There is approximately a 5% chance that any given vulnerability with a CVSS score of 7.5 or higher will be exploited. That number jumps to 25% if an exploit exists in ExploitDB and more than 35% if the exploit is a module in Metasploit.
5. Age of vulnerability – older vulnerabilities are more likely to have exploits.

Factors to Consider when Choosing a Scan Engine

For a company just getting started in with vulnerability management, an early decision is to pick a scan engine. There are several considerations when choosing a scan engine:

1. On-Premises vs Cloud – how comfortable is the organization with storing vulnerability data off-site with a vendor? How much money does the organization want to spend on infrastructure?
2. Vendors – Rapid7 (Nexpose), Tenable (Nessus/Security Center), Retina E-Eye, Intel/McAfee (Foundstone), Saint, Qualys
3. Scope/Scale – how big is the network to scan? How much of the network should be scanned? How segmented is the network?
4. Frequency of Scans – quarterly, monthly, daily?
5. Length of data retention – old scan data is generally not relevant to current risk, but maintaining older information might be important if the organization wants to use the data to show trends or for other reasons, such as identification of network nodes.
6. Multi-use Potential – should the scan engine also identify rogue devices on the network? Will the scan engine be used to build an asset management system?

How to get Buy-in

1. Identify stakeholders

   A successful program requires the support of senior leadership. C-level executives must be the sponsors of the program, in order to get all of the various groups who must do remediation work to prioritize the tasks. IT teams will also be stakeholders, as they will be doing the majority of the work. Business teams will be impacted as well. Often, the business owns the data or the business process and the IT teams must coordinate with them to perform maintenance work.

2. Identify objections / obstacles

In order to get the various stakeholders to really buy-in to a remediation program, the obstacles and objections they have must be identified and overcome. For business teams it is often a lack of understanding of risk that causes them to prioritize business over remediation. The owners of the vulnerability management program must be able to effectively articulate risk to the business of not remediating so decisions can be made with good information.

For IT teams, it is often a lack of resources that keeps remediation from being a priority. There are a few good strategies for helping the IT teams with resource constraints. One is to make sure the C-Level support is there to give prioritization. Another is to try to minimize the work load for the team by investing in automation and by understanding how to properly remediate. More information on this subject is in the "Quick Wins" section below.

3. Make your case for:
    a. Maintenance windows – if the company doesn't already have maintenance windows, establishing them can greatly ease not only patching, but many kinds of system maintenance issues.
    b. Routine Patching – having a patch schedule and a patch process makes remediation of common vulnerabilities much easier and can lead to automation for better and faster remediation.
    c. Assigning Ownership – having an owner for a remediation task add accountability and increases the odds that the remediation will be completed.

Speaking of Automation…

There are many areas within vulnerability management that lend themselves to automation. There are some that are simple and obvious and some that real areas for improvement within the vulnerability management space. Scanning as a process can be largely automated. The scans can be scheduled and reporting can be automatically sent out to remediation teams. This is a good example of simple automation.

Generation of reports and the delivery of reports and dashboards for the various stakeholders are also good candidates for automation. The stakeholders should not have to go looking for information on what needs to be remediated or what the remediation tasks are. GRC tools like RSA's Archer can take scan information and generate tasks and workflows for the remediation owners.

Finally, one of the big areas for improvement is automation in the areas of patch deployment. It should be possible, using automated patch deployment tools, to schedule the installation of missing patches directly from scan data. The use of automation for simple remediation tasks like this can ease the workload on remediation owners and make it easier to work on the bigger, more complex tasks.

Root Cause Analysis and Quick Wins

One way to improve the effectiveness of a vulnerability management program is to analyze the data. How long does it take to patch? What are the pain points in the remediation process? What is the real organizational risk, based on known vulnerabilities?

Ease the workload on remediation teams by combining remediation actions. For example, if the company is running an older version of Java, many known vulnerabilities can be addressed by installing the latest version of Java (or even better, removing Java from systems that don't need it). Most scan engines will report vulnerabilities based on plugin. So, there may be five vulnerabilities in Java 1.6.30,

and ten in 1.6.31 and ten more in 1.6.32. Installing 1.6.33 would then fix 25 known vulnerabilities. Rolling up remediation actions is one important way to ease workload and to assist the remediation teams with reducing real risk.

Add accountability by assigning ownership of tasks to specific individuals. Then track the progress of remediation tasks.

Conclusion

Good vulnerability management can be challenging. Most of the current roadblocks to effective risk reduction are not in technical problems like good scanning, but rather in process problems.