Managing Accepted Vulnerabilities

GIAC GCCC Gold Certification

Author: Tracy Brockman, tbrockma@verizon.net
Advisor: Adam Kliarsky

Accepted: May 16, 2016

Abstract

The discovery of new vulnerabilities occurs every day and organizations that follow good security practices remediate these vulnerabilities as soon as possible. Good security practices could be using automated patching tools, making a configuration change, or by implementing other security controls to reduce the risk, these vulnerabilities pose. However, when an organization has a vulnerability they cannot remediate, they need to have a process for inventorying, tracking, reviewing, and reporting on open vulnerabilities until they are fully remediated.

1. Introduction

Every day a new vulnerability is discovered in a piece of code or software and shortly afterwards the news of a new virus, malware, or hack is being used to exploit the vulnerability. Deploying vulnerability scanners that receive automatic definition updates and performing daily scanning against all devices in the inventory system will notify of new vulnerabilities found and provide a recommended remediation solution. A remediation could be adjusting the configuration in the system, implementing an additional control, applying a missing patch to a device or application, or an upgrade to a new version is required to resolve the vulnerability (CIS, 2015).

1.1. Risk Process

Organizations with a mature security practice follow some form of an IT security risk process that helps them identify and manage IT security risk. The National Institute of Standards and Technology (NIST) risk management process includes four components: framing, monitoring, assessing, and responding to risk (United States, 2011).

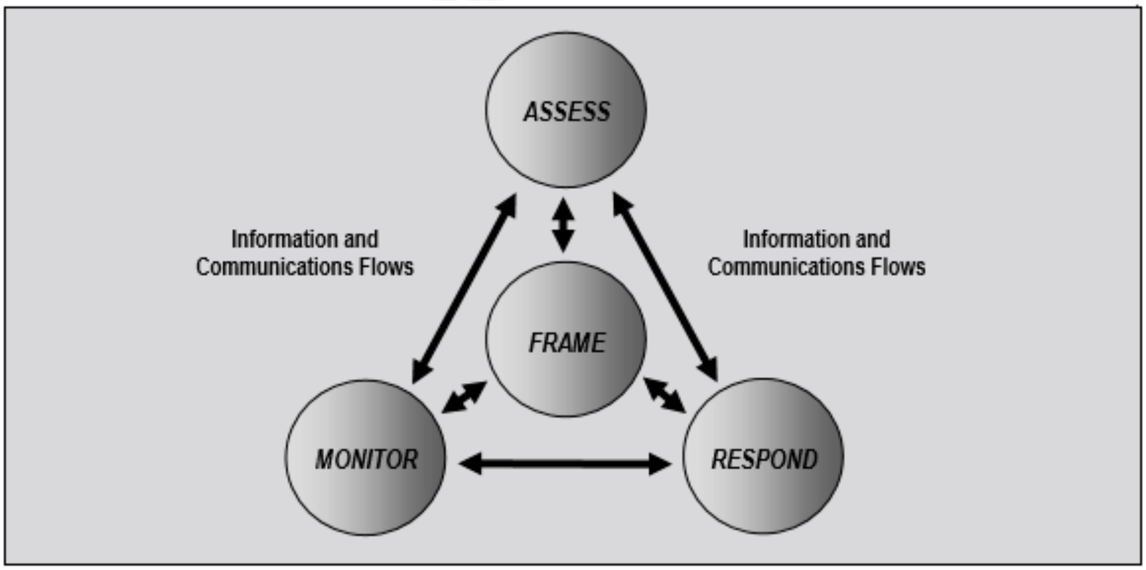
ASSESS
Information and Communications Flows
Information and Communications Flows
FRAME
MONITOR
RESPOND

Figure 1: Risk Assessment within the Risk Management Process

The first component is framing the risk, which an organization sets the context for making risk-based decisions that help in determining the organization's approach to assessing, responding and monitoring the risk. The next component is monitoring the risk, here an organization monitors the effectiveness of existing security controls, identify new emerging threats, and ensures the organization is compliant with its security policies and procedures, industry regulations, and federal, local, and international laws and directives (United States, 2011).

Assessing risk is the next component, which an organization identifies threats and vulnerabilities in the environment, and evaluates what the likelihood and the impact would be if the vulnerability were successfully exploited (United States, 2011). Organizations usually develop a formula for assessing the level of risk a security a vulnerability poses to the organization such as "*Risk = Criticality (Likelihood × Vulnerability Scoring [CVSS]) × Impact*" (Lee, 2014). The results of this formula help organizations with the final component, how to responding to the risk.

1.2. Responding to Risk

Organizations have four choices when responding to a risk assessment: transfer, avoid, mitigate, or accept the risk. A company can transfer the risk by purchase third party insurance to help reduce the financial impact in the event a breach occurs. Another option is to avoid the risk by decommissioning the application, system, or service that presents the risk (Harris, 2013 pp. 97-98).

The next option is to mitigate the risk by applying vendor patches, making configuration changes, or by implementing an alternative security control (Harris, 2013 pp. 97-98). In addition, "[p]*atch management is required by various security compliance frameworks, mandates, and other policies. … the Payment Card Industry (PCI) Data Security Standard (DSS)3, which requires that the latest patches be installed and sets a maximum timeframe for installing the most critical patches."* (Souppaya, 2013) Critical Control 4 states that an automated patch management system and software update tools should be deployed to keep all systems and applications on the current versions (CIS, 2015).

However, there are challenges with automated patching tools; security patches need to be prioritized based on criticality, deployed in phases, critical patches are installed first, or validating the patch file has not been compromised. Best practice is to test software updates on development systems to ensure there are no adverse effects, that no new services were installed that could introduce additional risk to the system or application, or if a firmware or bios update is needed, these cannot always be automated (Souppaya, 2013).

The final option is to accept the risk by "*Do nothing*" (Harris, 2013, pp. 99). If an organization has decided to do nothing, then they are willing to assuming any loss associated with the risk. However, there is more to accepting risk than just doing nothing. "*Successful vulnerability management programs have a systematic, accountable, and documented process to* [actively] *address vulnerabilities that exist within an organization.* [Actively] *managing vulnerabilities will reduce the potential for exploitation and will decrease the time and effort needed to respond after an exploitation has occurred.*" (SANS, 2013)

After the organization has discovered a vulnerability, the authorizing body reviews the recommended remediation options, the reason for not remediating, performs a risk analysis, and has decided to accept the risk by not remediating the vulnerability, what happens next?

As part of the overall organizational security program, the organization needs to track these vulnerabilities until they are remediated or the risk is eliminated by another method.

2. Inventory

Similar to the asset and software inventory, maintaining a complete inventory of accepted vulnerabilities is equally important. Company policy will outline how accepted vulnerabilities need to be documented and tracked. Creating, maintaining, and continuously reviewing a comprehensive inventory of known open vulnerabilities in the organization should be the goal.

In addition to the information already available in the asset and software inventory systems the inventory should include additional information about the vulnerability. These should include any recommended remediation, reasons why the organization has decided to accept the risk by not remediating the vulnerability, how often it will be reviewed, who the authorizing body was that accepted the risk on behalf of the organization, how the risk will be monitored for potential attacks or new threats, and what will be reported to whom.

The more information gathered in the inventory phase will be beneficial for audits, change in risk, or in the event, a security incident occurs. During security audits, auditors can review how vulnerabilities are being managed, the security posture of the system, ensure that by accepting the vulnerability the organization is not violating any regulatory or compliance standards, and to help evaluate the impact of the organization's risk exposure (SANS, 2013). During an incident, the incident responders would review this information to isolate what systems or applications could have been impacted, how a breach occurred, review the recommended remediation to prevent future exploits, and reporting.

Like other sensitive information, this data should be classified in accordance with the company's data classification policy and should be securely stored in a centrally located repository with restricted access. In addition, each acceptance should be assigned a unique identifying number. Possible solutions for tracking and storing risk acceptances could be a commercial application such as Rsam's Exceptions Management application, SharePoint, a database, or a spreadsheet.

2.1. System Information

Having a detailed inventory of systems and devices is necessary for tracking, auditing, and referencing accepted vulnerabilities. By knowing what is installed on a system, how it is configured, any services dependent on other systems, and its business criticality are significant if new threats are identified or when an acceptable remediation solution is found. This information should be reviewed if there is any change to the environment. The asset and software inventory systems should provide the information needed for documenting the system.

Critical Control 1 provides guidance in maintaining a complete inventory of all systems and devices that are connected to the organization's environment. It is recommended that the hardware inventory contains: the manufacturer, make and model number of the device, what firmware or bios version is currently running, the operating system and kernel version, is it a physical or virtual system, the location of the system, and all network mac addresses and their associated IP addresses (SANS, 2013).

In addition, the inventory should include information on how the system is configured: system name; system owner; purpose of system; what is the system's criticality, value, and classification; what TCP and UDP ports are open;

the services that are running; is the system accessibility from the Internet or Intranet; and any internally developed software.

And finally, the inventory should include details on the data and applications installed: the application versions that are installed, what type of data is hosted on system, who are the data owners, what is the criticality, value and classification of the data, and is the system subject to any legal or compliance mandates (United States, 2010). Also, identify any protective measures installed such as antivirus, host intrusion prevention systems, or application firewalls.

2.2. Vulnerability

Similar to the system inventory, having a detailed inventory of the vulnerability is essential for tracking, auditing, and referencing accepted vulnerabilities. When documenting the vulnerability it should include; the name, what systems and versions of the application in the environment are susceptible, the date the vulnerability was released, and what is the risk rating of the vulnerability. A description of how the vulnerability could be exploited, what type of attacks could be used to exploit the vulnerability such as cross-site scripting, denial of service, are there any reported active exploits, what the attack vector is it internally or via the Internet, and any supporting documentation or links (Security Tracker, 2015).

As an example, several vulnerabilities were found in Microsoft operating systems that could allow remote code execution:

Name: Multiple Windows Kernel vulnerabilities (Microsoft, 2015b)
Systems Affected: All workstations running Windows 7 with service pack 1, 32-bit and x64 based versions.
Date: November 10, 2015
Risk Rating: Critical
Type of Attacks: Email phishing attack or webpage redirect
Description: If an attacker can get an employee to open a specially crafted document, click on an embedded link within an email, or get them to go to a compromised website that has specially embedded fonts they could:

- "…*obtain potentially sensitive information from the kernel and bypass address space layout randomization (ASLR) controls on the target system [CVE-2015-6102, CVE-2015-6109]*."

- "…*can run a specially crafted program to execute arbitrary code on the target system with kernel-level privileges [CVE-2015-6100, CVE-2015-6101].*"
- "…*create a specially crafted font file that, when loaded by the target user, will trigger an error in the Adobe Type Manager Library and execute arbitrary code on the target user's system [CVE-2015-6103, CVE-2015-6104].*"
- "…*can run a specially crafted program at a low integrity level to exploit a permission validation flaw and modify files outside of the low integrity level restrictions [CVE-2015-6113].*" (Security Tracker, 2015)

Active Exploits: Currently there are no known active exploits (Microsoft, 2015a).

Supporting Documents/Links:

Microsoft Security Bulletin MS15-115 -
https://technet.microsoft.com/library/security/ms15-115
CVE-2015-6100 - http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6100
CVE-2015-6101 - http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6101
CVE-2015-6102 - http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6102
CVE-2015-6103 - http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6103
CVE-2015-6104 - http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6104
CVE-2015-6113 - http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6113

Fully capturing the components of open vulnerabilities in the environment will help for future reference. Some examples might be during an incident; the incident response team can reference this information to help identify how the attack occurred. They could also use this information to reference how many other systems were impacted. Another example, when a vendor provides a patch to fix the vulnerability the system administrators search for the vulnerability and cross-reference it to systems affected. Another example might be an auditor may review the information here to help understand the exposer of the data residing on the system and make recommendations accordingly.

2.3. Remediation / Mitigating Controls

Most vulnerability scanners or vendor sites will provide a recommended remediation to resolve the vulnerability unless it is a zero-day. Remediation could be applying a vendor patch, upgrading the version of software running, making a change to the configuration of the application or underlying operating system or uninstalling the application or services (Mell, et al., 2005). In cases where the vulnerability is a zero-day an alternate mitigating control may be used to reduce the risk of the vulnerability such as blocking a port or protocol with a host based or network based firewall, or by creating a signature on a host based or network based intrusion prevention system.

Information that should be captured when documenting the recommended remediation would be a description of the solution: name of patch to be applied or updated version of software, how the solution would be implemented, any system or application downtime, the level of effort to implement the solution, any costs to implement the solution, and any supporting documents or links. Continuing with the example, Microsoft provided a recommended remediation:

Solution: Microsoft has released a security update to remediate the vulnerabilities listed in Microsoft Security Bulletin MS15-115 (Microsoft, 2015b)
Patch or Version: 3097877
Delivery Method: The vendor patch will be applied via Microsoft System Center Configuration Manager (SCCM) or utilizing the windows update agent.
Downtime: System reboot may be required
Level of Effort: Low
Costs: No costs associated with applying vendor approved patch.
Supporting Documents/Links:
Microsoft Security Bulletin MS15-115 -
https://technet.microsoft.com/library/security/ms15-115

2.4. Accepting the Vulnerability

There are several reasons why an organization may choose to accept the vulnerability. Some technical reasons why a vulnerability cannot be remediated: it is an in-house developed application and will require a resource to develop a fix, the security patch has an adverse effect on the application or system, an update introduces a new vulnerability, or there is no security patch available. Business reasons why a system cannot be remediated: the application vendor is no longer in business, the software is no longer supported, the recommended remediation is not cost effective to implement, or there could be a legal hold and no changes are allowed to the system or application (Souppaya, 2013).

In addition, a timeline may be needed to align with the company's system development lifecycle, giving system administrators and system process owners' proper time to test new security patches being applied to an affected system or application. Another reason may be, there is a project to decommission the system that is affected and it will be removed from the environment. Again continuing with the example the Microsoft patch had an adverse effect on Windows 7 systems:

> **System(s) Impacted:** All workstations running Windows 7 with service pack 1, 32-bit and x64 based versions.
> **Patch or Version:** 3097877
> **Reason:** The security update released on November 10, 2015, for Windows 7 had multiple issues that had a major business impact. After applying the patch, it was reported that Microsoft Outlook, Microsoft PowerPoint, and Internet Explorer would crash during certain actions. Also after rebooting systems some users were unable to login to their systems (Microsoft, 2015a).
> **Research:** The vendor, Microsoft, has confirmed there are issues with security update 3097877 and recommend uninstalling it until an updated security patch can be released (Microsoft, 2015a).

2.5. Time

All open vulnerabilities should have a set expiration time and depending on the complexity of the vulnerability it could be a day, week, month, or a year before a remediation is available. The organization should have a policy stating what the max duration the vulnerability can be open before it needs to be reapproved. There also needs to be a mechanism for alerting system administrators, the security team, and the authorizing body when the vulnerability has passed its expiration time. Once the expiration has passed its approved time, it must be presented to the authorizing body for ongoing authorization (Dempsey et al., 2014). Whatever the reason for not remediating the vulnerability, the goal should be to have the vulnerability remediated in the shortest time possible.

2.6. Approvals

For reviews and during audits, it should be documented who accepted the vulnerability on behalf of the organization. The authorizing body should primarily contain senior business managers and process owners (Hardy, 2005). Some of the roles that should be a part of the authorizing body are; Chief Risk Officer (CRO), Chief Compliance Officer (CCO), Chief Financial Officer (CFO), Legal, and the department head that is accountable for the management, development, and business operations of the application and/or system affected. The Chief Information Officer (CIO) and/or Chief Information Security Officer (CISO) role should be to present the risk to the authorizing body (ISACA, 2009).

3. Tracking & Reviewing

3.1. Monitoring

Enhancing monitoring on systems and any interdependent systems that have open vulnerabilities should consist of both internal and external resources. Successful monitoring should be from multiple points throughout the network and on systems. By increasing the monitoring points, this improves the chances of detecting an attack while providing a more in-depth defense (ISACA, 2010).

3.1.1. Internal Monitoring

Internal monitoring should include: syslog's, activity of application usage, account usage, and data access to see if there is a deviation from normal traffic (it is important to note that you must have a known good baseline to compare against). By increasing the monitoring not just, for perimeter network traffic but also the internal traffic between systems can help detect if a system is compromised and is being used as a pivot point to gain access to other systems.

Some of the tools that can enhance the monitor of systems are; firewalls, vulnerability scanners, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), data loss prevention (DLP) systems, security information and event management (SIEM) systems.

It is important to continue to perform vulnerability scans against the network and systems, using both external and internal credential methods. Use vulnerability scanners with up-to-date databases to see if any new exploits are discovered since the original evaluation of the risk was reviewed. Use a commercial tool like Tenable Nessus, Rapid7 Nexpose, or Qualys Vulnerability Manger or an open source tool like (SANS, 2013).

IDSs can analyze network traffic and systems for any known attacks. Once an attack is detected, it can send alerts to IT system administrator and security team for review. Most IDSs have a subscription service that keeps them updated with new attack signatures. An experienced security analyst can create new signatures to monitor for new types attacks specific to their environment (ISACA, 2010).

Like IDSs, the use of IPSs can analyze network traffic and systems for any known attacks. Unlike IDSs, IPSs monitor the patterns in network traffic that have similar behavior to other attacks like malware, virus, or hacker activity to predict and stop attacks before they can take effect (ISACA, 2010).

A SIEM can help analyze the data from all the different log points. SIEM’s collects real-time data and logs from multiple sources then correlate and analyzes the data to detect and alert on security events. In the event of a security incident, the data collected in the SIEM can also assist in incident investigations or audits (SIEM, 2016).

3.1.2. External Monitoring

External monitoring would include monitoring vendor sites for new security patches or software version upgrades, security related sources such as threat intelligence feeds and news sites can provide information about new types of attacks, exploits, and vulnerabilities. Signing up for news email alerts or other subscription services can help automate this process (Mell, et al., 2005).

Monitoring other types of external resources can be equally important such as hacker blogs or even cybercriminal sites. These sites can provide information about zero-day exploits and attacks. Cybercriminal market sites can hold information that may not be found in other places such as user account and password information, company proprietary information, or other sensitive data. Some third party companies that can monitor and alert on this information are LookingGlass Cyber Solutions Inc. or Flashpoint.

3.1.3. Deviation

A deviation to the agreed acceptance of the vulnerability could be a related to; the system software, data stored on the system, change in an interdependent system, new legislation, or the vendor has released a new security patch or version upgrade. There could have been a change in the risk rating of the vulnerability, the logs indicate there is an active attempt to exploit or a successful exploit of the vulnerability, a new exploit has been released, or a new vulnerability is found that is related to the root cause of the existing vulnerability, (United States, 2011).

Any company information found on a cybercriminal site is a clear indicator that the company or at least its data has been breached. If an event is detected, then an alert should be sent to the system administrator and security team for review. If it is confirmed that there is a change of the parameters to the accepted vulnerability, an alert must be raised to the security team and the authorizing body immediately. The authorizing body should review the deviation and decided to continue to accept the vulnerability or take a new course of action based off the information presented.

3.2. New Remediation's / Mitigating Controls

Vendors work to create patches for their software to close these new vulnerabilities, and new technology solutions are being developed to mitigate against new and existing threats. Once a new solution is released either from the vendor or in-house developers created a new security patch for their software, new technology solutions are developed, or another mitigating control is recommended it should be thoroughly reviewed. Does the remediation resolve more than one vulnerability, what are the required steps to implement the fix, does it require system down, what are the costs, and will a new issue or vulnerability occur? (Mell et al., 2005)

To mitigate any production issues, changes should be tested in a development or test environment first. Once applied to the isolated environment, it should be given to business owners for testing to ensure that it has not broken any business functionality. A vulnerability scan should be performed to see if the vulnerability is still present or a new one is discovered. Finally, have a red team perform a pen test to validate the fix (Mell et al., 2005). Once the fix is validated, it should be deployed to the production environment, and the vulnerability should be closed. Concluding the example, Microsoft released an update patch to remediate the vulnerability and issue with the original security patch:

> **System(s) Impacted:** All workstations running Windows 7 with service pack 1, 32-bit and x64 based versions.
> **Patch or Version:** 3097877
> **Reason:** The vendor, Microsoft, released an update to the patch for Windows 7 to resolve the issues related to the original security patch.

- *"Resolves crashing that occurred in all supported versions of Microsoft Outlook when users were reading certain emails.*
- *Resolves crashing that occurred in supported versions of Microsoft PowerPoint when opening PowerPoint Presentations.*
- *Resolves crashing that occurred in supported versions of Internet Explorer when browsing certain web pages such as groupware web pages in Internet Explorer.*
- *Resolves problems that occurred while users were logging on to the system. For example, after a user restarted the computer and then pressed Ctrl+Alt+Delete at the logon screen, the screen flashed and then went black. The user was then unable to continue. There may be other, similar logon issues that are related to this issue."* (Microsoft, 2015c)

3.3. Closing Vulnerabilities

There are a few primary ways an accepted vulnerability can be remediated. The first is by decommissioning the affected system or application, or uninstalling the service. Note, disabling the service should not be accepted as a remediation solution even though it may resolve the risk it could be inadvertently or intentionally enabled again. The next way would be applying a vendor-supplied security patch, by upgrading the version of software or firmware, or by applying a fix by the application developer when dealing with the internally developed application. Another option is by making a configuration change or modifying privileges to a system or application, implementing or changing an IPS or firewall rule, or modifying an access control list on a router. Any of these changes should reduce or eliminate the attack vector presented by the vulnerability to the system (Scarfone, et al., 2008).

Before signing off and officially closing an accepted vulnerability, the remediation needs to be validated to ensure vulnerability is resolved and to verify the resolution did not open any new risk in the process. There are several ways the vulnerability can be validated. The first is to perform a vulnerability scan to see if the vulnerability is still present. Another option is to perform pen test using a tool like Metasploit. Metasploit has a large library of modules for various exploits that can be used to validate if a vulnerability is closed. The key point here is making sure the vulnerability is resolved before closed it (Mell et al., 2005).

Once the fix has been fully remediated and validated, the system build team needs to ensure that the security patch or configuration change is applied to the baseline image. This will ensure that all new systems built will already be fully patched and less likely to be deployed with open vulnerabilities. In addition, all changes should be fully documented and kept with the appropriate asset or software inventory (United States, 2010).

4. Reporting

Each organization has a different requirement on what needs to be reported. As part of any information security program defining metrics and reporting on them is critical to an organization. There are various metrics that measure how effective the security program is, and reporting on all open vulnerabilities is just a portion of it (Chew, et al., 2008).

An example of some of the measurements that are helpful for tracking open vulnerabilities by system are: the total number of open vulnerabilities per system, what the criticality is for each vulnerability, what the risk is to the application or data residing on the system, how many days the vulnerability has been open, what threats, if any, is the system susceptible to, what is the likelihood the vulnerability would be exploited, or has there been any change, new threats or vulnerabilities to the system, since the last review.

Another example would be reporting from an organizational structure. How many systems are susceptible to a single vulnerability, listing the number of vulnerabilities by responsible party, the impact to the overall risk tolerance to an organization, tracking the costs of remediating versus loss due to a compromise, or the number of system are out of compliance with policies, laws and directives (Chew, et al., 2008).

The frequency on how often the information is collected and reported on will be dictated by company policy. Severity or management levels may dictate the frequency of reporting. For example, at an executive level they may receive a report on a quarterly for critical and high risks, but only receive a full report of all open vulnerabilities on an annual basis. Subsequent reporting may have the authorizing body receiving a full report of all open vulnerabilities on a monthly basis. While at the system administration and IT security team level would be done on a daily or weekly basis (Chew, et al., 2008).

Finally, it is important to define the roles and responsibilities for those responsible for reviewing the reports. At each level in the organization, there is a stake in ensuring the success of the IT security program. The senior management's role is to ensure the risk being accepted aligns with the organization's acceptable risk tolerance, that there is adequate funding available for remediating vulnerabilities, and help drive support for the security program (Chew, et al., 2008).

Some of the responsibilities of the authorizing body are to; regularly review the status reports to ensure that the risk is still acceptable, re-evaluate the risk if there is a change to the original acceptance of the vulnerability, and reauthorizing expired vulnerabilities that cannot be remediated. In addition, they need to ensure compliance with policies, laws, and directives, and advise the organization on the overall risk posture of the organization (United States, 2010).

The IT security team and system administrators are responsible for; collecting the required data, setting up and automating continuous monitoring for new and existing vulnerabilities and threats using this information to help prioritize vulnerabilities to remediated, and applying security patch and software updates. They also provide guidance to IT managers, assets and system owners, and the authorizing body with up to date information on new and existing vulnerabilities and remediation options (Luu, 2015).

The metrics and information reported on are useful in evaluating the effectiveness of the security program for audits or any other requirement required by the organization. These quantitative and qualitative measurements reported are used by the organization in determining risk (Chew, et al., 2008), "Risk = Criticality (Likelihood × Vulnerability Scoring [CVSS]) × Impact" (Lee, 2014).

5. Conclusion

As part of an organization's overall security practice, there should be a process for managing open vulnerabilities that they are willing to accept. By creating a process for inventorying open vulnerabilities and the systems that are impacted by them; monitoring for attacks, indicators of compromise, or a new remediation; tracking for any changes to the agreed-upon acceptance and by reporting on them will help an organization to manage their risk. See appendix A for a sample workflow. There is more an organization can do than "Doing nothing" when accepting a vulnerability.

References

Center for Internet Security (CIS). (2015). *The CIS Critical Security Controls for Effective Cyber Defense Version 6.0.*

Chew, E., Swanson, M., Stine, K., Bartol, N., Brown, A., Robinson, W,. (2008). *Performance measurement guide for information security* (SP 800-55). Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

Dempsey, K., Ross, R., & Stine, K. (2014). *Supplemental Guidance on Ongoing Authorization Transitioning to Near Real-Time Risk Management*. Retrieved from National Institute of Standards and Technology website: http://csrc.nist.gov/publications/nistpubs/800-37-rev1/nist_oa_guidance.pdf.

Illuminate the Deep & Dark Web - Flashpoint. (n.d.). Retrieved April 15, 2016, from https://www.flashpoint-intel.com/

Hardy, G. (2005). *IT Governance Domains Practices and Competencies: Information Risks—Whose Business are They?*. Retrieved April 4, 2016 from http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/IT-Governance-Domains-Practices-and-Competencies-Information-Risks-Whose-Business-are-They-Whose-Business-are-They.aspx

Harris, S. (2013). *CISSP All-in-One Exam Guide* (6th ed., pp. 97-99). New York, NY: McGraw-Hill.

ISACA. (2009). *The Risk IT Framework*. Retrieved April 4, 2016, from http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/The-Risk-IT-Framework.aspx

ISACA. (2010, March 1). *IT Standards, Guidelines, and Tools and Techniques for Audit and Assurance and Control Professionals*. Retrieved April 4, 2016, from http://www.isaca.org/Knowledge-

Center/Standards/Documents/IT-Audit-Assurance-Guidance-1March2010.pdf

Lee, J. (2014). *An Enhanced Risk Formula for Software Security Vulnerabilities*. Retrieved from http://www.isaca.org/Journal/archives/2014/Volume-4/Pages/JOnline-An-Enhanced-Risk-Formula-for-Software-Security-Vulnerabilities.aspx

LookingGlass Cyber Solutions. (n.d.). Retrieved April 15, 2016, from https://www.lookingglasscyber.com/

Luu, T. (2015, January). Implementing an Information Security Continuous Monitoring Solution—A Case Study. *ISACA Journal*, *1*(2015), 22-28.

Mell, P., Bergeron, T., Henning, D. (2005). *Creating a patch and vulnerability management program: Recommendations of the National Institute of Standards and Technology (NIST)* (SP 800-40v2). Gaithersburg, MD: U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology.

Microsoft. (2015, November 10). *MS15-115: Description of the security updated for Windows: November 10, 2015*. Retrieved March 26, 2016, from https://support.microsoft.com/en-us/kb/3097877

Microsoft. (2015, November 10). *Microsoft Security Bulletin MS15-115 - Critical*. Retrieved March 26, 2016, from https://technet.microsoft.com/library/security/ms15-115

Microsoft. (2015, November 11). *MS15-115: Security update for Windows to address remote code execution: November 10, 2015*. Retrieved March 26, 2016, from https://support.microsoft.com/en-us/kb/3105864

Scarfone, K., Jansen, W., Tracy, M. (2008). *Guide to general server security: Recommendations of the National Institute of Standards* *and Technology* (SP 800-123). Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

Security Information and Event Management (SIEM) - Gartner IT Glossary. (n.d.). Retrieved April 10, 2016, from http://www.gartner.com/it-glossary/security-information-and-event-management-siem

Security Tracker. (2015, November 10). *Microsoft Windows Kernel Bugs Let Remote Users Execute Arbitrary Code and Local Users Bypass ASLR Restrictions and Gain Elevated Privileges* - SecurityTracker. Retrieved March 26, 2016, from http://securitytracker.com/id/1034114

Souppaya, M., Scarfone, K. (2013). *Guide to enterprise patch management technologies: Recommendations of the National Institute of Standards and Technology* (SP 800-40r3). Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

The SANS Institute. (2013). *Security 566: Implementing & Auditing the Critical Security Controls – In Depth (Books 1-5).*

United States. Joint Task Force Transformation Initiative. (2011). *Managing information security risk: Organization, mission, and information system view* (SP800-39). Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

United States. Joint Task Force Transformation Initiative. (2010). *Guide for applying the risk management framework to federal information systems: A security life cycle approach* (SP 800-37r1). Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

Appendix A

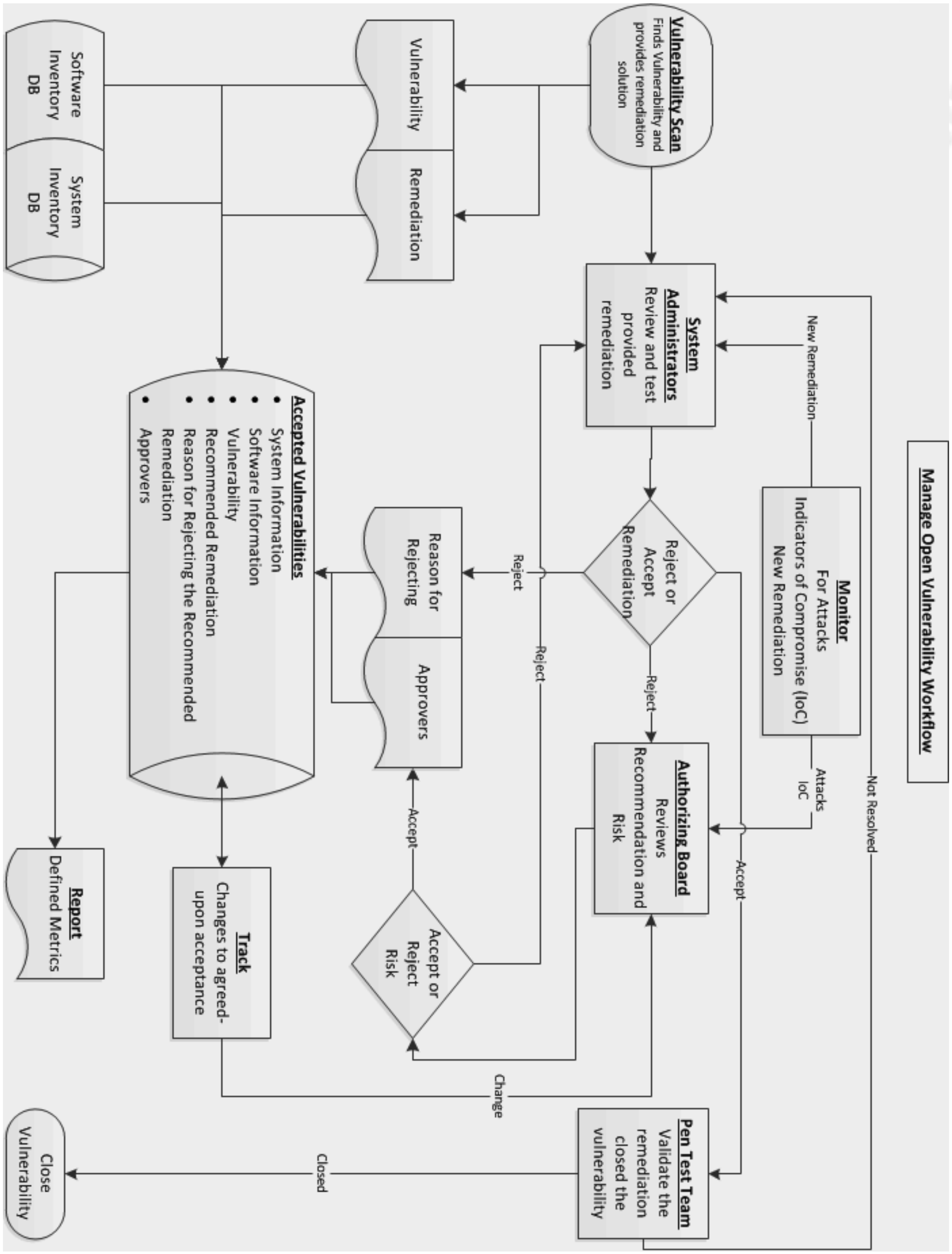

Figure 2 - Sample Workflow

Tracy Brockman, tbrockma@verizon.net