Global Information Assurance Certification Paper

Building a Vulnerability Management Program – A project management approach

GIAC (GCPM) Gold Certification

Author: Wylie Shanks, giac@infosecmatters.com
Advisor: Christopher Walker
Accepted: 5/21/2015

Abstract

This paper examines the critical role of project management in building a successful vulnerability management program. This paper outlines how organizational risk and regulatory compliance needs can be addressed through a "Plan-Do-Check-Act" approach to a vulnerability management program.

1. Introduction

IT security regulations increasingly are the norm demonstrating a standard of care in protecting sensitive data. To serve this standard, several regulatory bodies have mandated the creation of vulnerability management programs. Examples include: the Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act (HIPAA), and Federal Energy Regulation Committee (FERC).

Vulnerability management programs address the inherent security weakness created by software vulnerabilities. Known software vulnerabilities create opportunities for criminals and other adversaries to exploit these weaknesses. These vulnerabilities may result in unauthorized access to a system or network, access to or theft of confidential data, resulting in regulatory, financial or reputational impact for business (Vulnerability Management for Dummies, 2008).

Building a vulnerability management program requires management sponsorship to provide the necessary commitment, funding, resources (staff and equipment), and direction for the project to be successful. This paper examines the role of the "Plan-Do-Check-Act" approach to project management in building a successful vulnerability management program.

The next section of this paper examines the vulnerability management program.

2. Vulnerability Management Program

Unfortunately, new software vulnerabilities are discovered on a daily basis. Vulnerability management (VM) is the means of detecting, removing and controlling the inherent risk of vulnerabilities. The vulnerability management program utilizes specialized software and workflow to help eliminate detected risks (Vulnerability Management for Dummies, 2008).

Scaling the vulnerability management program is important to address company requirements, complexity, and its IT environment. Even the smallest companies can operate manual vulnerability management processes. However, the use of automation and workflow are recommended to ensure consistency, compliance (task completion

assurance), and to reduce cost. The vulnerability management maturity model shown below illustrates the scalability of the vulnerability management program.

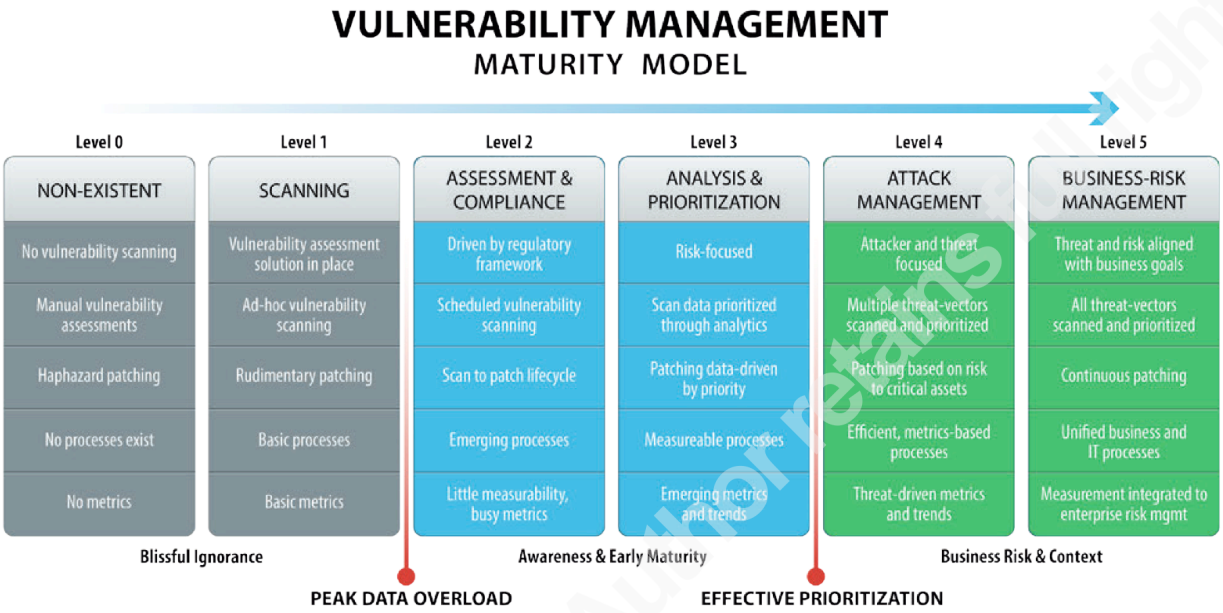

Figure 1 retrieved April 12, 2015 from http://www.coresecurity.com/system/files/attachments/energy-company-vulnerability-management-case-study.pdf

2.1. Components of Vulnerability Management

Vulnerability management is comprised of the following activities (Vulnerability management for dummies, 2008):

- Identifying / tracking assets (build asset inventory)
- Categorizing assets into groups
- Scanning assets for known vulnerabilities
- Ranking risks
- Patch management
    - Test patches
    - Apply patches
- Follow-up remediation scan – confirms vulnerability addressed.

The company selects a vulnerability assessment tool to automate many of these tasks. It is designed to collect asset inventory information, categorize assets, scan for vulnerabilities and rank discovered risks.

Wylie Shanks, giac@infosecmatters.com

2.1.1. Build an asset inventory

Without a current asset inventory, it is difficult to properly address the inherent environmental risk. In this step, the vulnerability assessment tool scans specified network subnets for assets. Systems discovered during the scan are added to the asset inventory. This process helps to ensure that all systems are identified and patched accordingly.

2.1.2. Categorizing assets

Asset categories or groups are created from the asset inventory. Asset groups are used to scan specific assets (rather than subnets). These categories also allow for vulnerability scan customization, addressing asset or business requirements, and assists with assigning risk rankings.

2.1.3. Vulnerability scanning

There are two aspects to vulnerability scanning – the scan and a report. The vulnerability scan is designed to test and analyze systems and services for known vulnerabilities. The scan comprises a list of scan options (ports, protocols, and network packet behavioral characteristics used for scanning) and assets. The report contains the prioritized list of vulnerabilities, vulnerability description, calculated risk, and remediation activities.

2.1.4. Risk ranking

Risks are prioritized by the calculated business risk. Assets and assets groups are assigned a business criticality rating. When vulnerability is discovered, the vulnerability assessment tool calculates the business risk of an asset. Remediation effort is subsequently prioritized on a risk basis. For example, an Internet web server susceptible to a vulnerability granting administrative level access should be remediated before an internal system requiring a low severity security patch.

2.1.5. Patch management

Patches are tested in a non-production environment to determine if there are system compatibility issues. The tested patches are migrated to production and approved for release. Automation of patch deployment ensures timely remediation and low

deployment cost. In addition, patches may be applied through the following methods (Souppaya & Scarfone, 2013):

- Software defined automatic update;
- Network access control; and
- User involvement (manual patch install or patch approval).

2.2. Regulatory requirements

Regulatory bodies require vulnerability management programs to contain a means of identify vulnerabilities and remediating them in a timely manner. In an effort to demonstrate compliance, documentation containing the identification and eradication of the vulnerability is kept. Maintaining data integrity is vitally important to the trust and verifiability of the report. The vulnerability assessment tool must ensure data integrity is maintained.

Regulatory requirements provide a means of obtaining support and funding for vulnerability management programs. As Snedaker explained, "[o]ne of the best ways to support an increase in IT spending for security is to clearly delineate the cost of preventing a security breach versus the cost of fixing a security breach" (Snedaker, 2006). Vulnerability management programs address the inherent risk within company environments through identification and remediation of vulnerabilities. It cost less to remove the risk through patching rather than implementing additional compensating controls in an attempt to mitigate risk to acceptable levels.

2.2.1. Governance

Most vulnerability management program regulatory requirements dictate, among other requirements, that companies must (The Best Damn IT Security Management Book Period, 2007):

- Appropriate program sponsorship for the vulnerability management program.
- Key stakeholder identification, representation and participation in the program.

- Documented security policies, practices, and standards.
- Documented roles and responsibilities.
- Documented communication and escalation plans.
- Asset identification (in-scope assets).

The next section of this paper examines the project management aspects of Plan-Do-Check-Act (PDCA) as related to the vulnerability management program.

3. Project alignment with PDCA

Every successful project begins with executive support. This integral support provides the approval and resources necessary to complete the project. As Snedaker described, "[i]f company executives do not understand or care about a project, they will not allocate the time, money, or resources needed to make it successful" (Snedaker, 2006).

A regulatory requirement to implement a vulnerability management program helps to ensure that executive approval and support of the project is available.

3.1. Plan-Do-Check-Act

Dr. W. Edwards Deming is credited with popularizing the Plan-Do-Check-Act (PDCA) model. However, his mentor Walter A. Shewhart, created this business process model as a means of analyzing and monitoring product deviations from customer requirements (Arveson, 1998). This method is utilized in the Project Management Body of Knowledge (PMBOK) as Initiating/Planning, Executing, and Monitoring & Controlling (Sliger, 2008) and is discussed further in section 4 of this paper.

The PDCA model is an iterative process as this illustration depicts:

Figure 2 retrieved April 12, 2015 from http://space4ict.com/aq/qc-starter-kit.html.

The vulnerability management program fits nicely into the PDCA model. Each process aligns with a component of the vulnerability management program as shown below:

Process	Description
Plan	• Risk assessment (vulnerability assessment) • Develop risk treatment plan (patching process) • Risk acceptance (what constitutes acceptance? High-risk vulnerabilities resolved / regulatory requirements met?)
Do	Implementation of risk treatment plan (patching / mitigation strategies)
Check	Continual monitoring / review of risks (on-going scanning)
Act	Maintain and improve the process (update scan profiles and mitigation strategies as new risks discovered)

Adapted from (*BS ISO-IEC 27005:2008, 2008)*

3.2. Plan

The planning phase of the vulnerability management program involves gathering company and regulatory vulnerability assessment requirements, detecting vulnerabilities, and rating and ranking their risk. Two regulatory requirement examples are listed below:

NERC CIP-005 ESP requirements – R4 Cyber Vulnerability Assessment (Parks, 2007)

- Documented vulnerability assessment process;
- Required ports and services are the only ones open and enabled;
- No default accounts, passwords or SNMP community strings exist;
- Assessment results are documented;
- Action plan created from documented results; and

- Remediate or mitigate identified vulnerabilities.

Payment Card Industry Data Security Standard (PCI DSS) – ("Payment Card Industry (PCI) Data Security Standard - Requirements and Security Assessment Procedures v3.1", 2015)

Requirement 11.2:

- Run internal and external network vulnerability scan at least quarterly and after any significant change in the network (firewall rule changes, system installations/upgrades, network topology changes):
    1. Utilize approved scanning vendor (ASV) tools only.
    2. Perform quarterly external and internal scans.
    3. Remediate high-risk vulnerabilities. (Highest priority items should be addressed first.)
    4. Repeat steps 2 and 3 until all high-risk vulnerabilities have been resolved.

Requirement 1.1.6:

- Document secure and insecure services, protocols and justify their business use.

The examples listed above are fairly similar and have overlapping requirements.

It is important to use exiting company standards when planning the vulnerability management program. Sneaker explained, "[w]henever possible, standardizing project infrastructure reduces the cost and time of the project. If you can reuse tools, processes, or methods from other projects, or … implement standardized tools or equipment … projects will typically generate better results by reducing the learning curve and ramp-up time (which often leads to errors and omissions)" (Snedaker, 2006).

With vulnerabilities identified creation of the risk treatment plan is now the focus. Vulnerabilities are rated based on risk to the company. Stakeholders are advised of the vulnerabilities detected on their systems or in their applications. The highest rated risks on the most critical systems should be addressed in priority. Tracking and measuring the remediation ensures regulatory compliance requirements are met.

Other planning considerations (scope) include:

- The number of systems included in the change.

- The time required implementing the change.
- The personnel and/or teams involved.
- Implementing the change when it is least impactful to the business.
- Confirmation that the change was successful and normal business operation is restored.

3.3. Do

The risk treatment plan is implemented in this phase. Risks are mitigated to the company's acceptable levels. The plan may include patching systems to acceptable levels, decommissioning systems (removing them from the environment), or applying compensating controls.

3.4. Check

Systems are monitored regularly to ensure vulnerability compliance requirements are met. Companies may choose to run the minimum number of scans required to meet compliance requirements. However, more frequent scanning (e.g. weekly) provides several benefits:

- The vulnerability database is current. (Systems missed due to patching or other downtimes are updated the following week.)
- The database provides more timely and accurate information for searches and reports. (It may be possible to determine the number of systems potentially affected by zero day vulnerabilities.)
- The timeliness of patching is recorded in the database. (A vulnerability is identified and subsequently confirmed patched in a follow-up scan. It existed in the environment for a period of time.)

Reporting is tailored to the company's objectives and reporting requirements. Reports provide evidence of vulnerabilities detected, remediation required and elimination of the risk (i.e. the vulnerability is patched). Additionally:

- Scorecards and dashboards provide a more current and accurate view of the company's inherent risk and security posture.
- Reports provide teams with actionable information required to remediate risk.

- Reports provide assurance that regulatory vulnerability assessment requirements have been met.

New security vulnerabilities are reported in security advisories and other vulnerability sources. Security staff should monitor these posts to determine their applicability to the company. This information is fed into the next phase, forming the basis of change to the vulnerability program.

3.5. Act

The data generated from previous phases is used to improve the vulnerability management program. Changes may apply to company security policies, practices and procedures. These changes may result in organizational risk reduction, increased process efficiency and improved regulatory compliance. Common areas of improvement, as outlined by Manzuik, are (Manzuik, 2007):

- Asset management;
- Configuration management; and
- Assessment management.

3.5.1. Asset management

Most organizations have difficulty maintaining a current asset inventory. The asset list is critical to an effective vulnerability management program. An organization's inherent security risk may include non-production or old (non-patched) assets that have been forgotten yet still reside on the company's network. Therefore, the scanning process must include all relevant assets and accommodate the introduction and removal of assets during their lifecycle.

3.5.2. Configuration management

Configuration management is part of a company's standard system build. Within the company, standard applications exist and must be maintained. The vulnerability assessment tool is used to identify applications for removal from the system landscape. This ensures that future vulnerabilities discovered in the application do not affect the company. Policy compliance (adherence to the company standard build) is not included in the scope for this paper.

3.5.3. Assessment management

The ability to identify vulnerabilities becomes less effective over time. New techniques are discovered and regulatory requirements changes to reflect new security risks. The assessment management process is updated to reflect these new and changing requirements.

The next section of this paper examines the project phases as defined in the Project Management Body of Knowledge (PMBOK).

4. Putting it all together

Every project begins with a plan. The plan serves as the roadmap or guideline for objective completion. The first four project phases are initiating, planning, executing, and controlling & monitoring (A Guide to the Project Management Body of Knowledge (PMBOK Guide), 2008).

4.1. Initiating phase

The first project phase is the initiating phase. Project deliverables in this phase include identifying project stakeholders and the project charter.

4.1.1. Identifying stakeholders

Common stakeholders in a vulnerability management program are:

- IT management and business unit representatives
- IT security, compliance or audit team
- IT infrastructure teams including:
    - Server team (Windows, Linux, Unix, Virtualization platform etc.)
    - Network team (internal and perimeter equipment)
    - Storage team
- IT application and operating system support teams

Only major stakeholders provide the necessary support for project success. They are identified and named in the project charter.

4.1.2. Project charter

The project charter authorizes the project and documents the initial project and stakeholder requirements. This includes regulatory requirements for a vulnerability management program and what the project hopes to achieve. In addition, the project charter contains the project objective (acceptance/success criteria), scope, assumptions, constraints, risks and the communication strategy.

4.1.2.1 Project objective

The project objective provides a high-level statement of the desired outcome or success criteria of the project. It answers the question – "What is deemed project success?" Success criteria are written as a specific and measurable way to quantify the delivery of project requirements. An example of a vulnerability program project objective is: deployment and implementation of the chosen vulnerability assessment tool on all five (5) servers by the end of the quarter.

4.1.2.2 Project scope

The charter's project scope provides the high-level details required to meet the project objectives. It provides project parameters – what is both inside and outside of the project's scope. Particular project details may be unknown when the project character is created. Known items are included in scope with a process to clarify scope as further information becomes available. Project charter scope provides the foundation for the evolving project plan and more detailed scope. The project is complete when all in-scope items have been satisfied.

4.1.2.3 Assumptions

Assumptions are documented in the project charter. Assumptions may exist throughout the project. However, most assumptions will be evaluated and adjusted during the more detailed planning and execution phases. Common examples of project assumptions may include:

- Management support for this project will remain constant.
- Funding for this project will not be affected by other organizational changes or priorities.
- Resources for this project will be available until project completion.

4.1.2.4 Constraints

Constraints may exist that cannot be changed. The project may be limited by organizational structure, company priorities and corporate culture. Any known constraints should be documented in the charter.

4.1.2.5 Change management

Changes occur in every project. A process is required in order to evaluate the change, assess its impact and ultimately approve requested changes. Stakeholders must review and approve all change requests as project time, cost and budget may be affected. Changes must not be implemented without written stakeholder approval.

4.1.2.6 Risks

Stakeholders identify high-level risks, which are subsequently documented in the project charter. Regulatory risk and competitive risks are included in this section, both examples of strategic risk. Other examples of risk types include (Snedaker, 2006):

- How much risk is acceptable?
- How is risk managed (avoid, reduce etc.)?
- What are the financial and legal consequences of project failure?

4.1.2.7 Communication strategy

Effective communication is critical to the success of every project. This section documents the frequency, type of communication (reports, status meetings etcetera) and participants (if known at time of project charter creation). Common communication items include ("Roffensian Consulting Inc Project Charter"):

- Stakeholder review and approval of project documents.
- Formal presentation and approval of milestones. (Approval facilitates proceeding to the next phase.)
- Status reports.
    - Tasks planned and completed during the reporting period.
    - Future tasks (next reporting period).
    - Current issues and risks.
    - Project status (e.g. traffic lights).
- Status meetings (e.g. weekly).

- Agenda sent to team prior to meeting.
- Team members provide task update.
- Meeting minutes distributed to invitees and stakeholders.

4.2. Planning phase

A detailed project scope and project plan is developed in this phase. This phase documents project "scope, time, costs, quality, communication, risk and procurements" (A Guide to the Project Management Body of Knowledge (PMBOK Guide), 2008). The output from this phase is the project plan. It documents in detail the plan to meet project objectives. The project is controlled and managed against this plan.

4.2.1. Scope

Common vulnerability management program in-scope items are:

- All relevant assets along with a means of identifying and tracking the assets;
- Identification of new vulnerabilities; and
- Vulnerability and compliance reports.

4.2.2. Project plan

The project plan incorporates and documents all sections of this phase. Stakeholder requirements are collected, scope is further defined and work is broken down into manageable tasks. (See Appendix A for a sample work breakdown structure.) Next, activities are defined, sequenced and resource activity is estimated. The schedule is created which includes a budget, estimated cost and how to achieve quality requirements. Risks are identified and analyzed along with developing plans to mitigate project risks. The final step is to procuring project products and services.

4.3. Executing phase

The project plan is executed in this phase. The project plan is directed and managed with quality assurance performed. Project resources are acquired, developed, and managed. Next, information is disseminated to project stakeholders and expectations are subsequently managed. The final step, procurement, is executed. Project products and services are acquired.

4.4. Monitoring & Controlling phase

There are several important activities conducted in this phase:

- Controlling and managing tasks;
- Controlling and verifying scope;
- Controlling costs and scheduling;
- Reporting on performance; and
- Monitoring and controlling risk.

4.4.1. Controlling and managing tasks

The work breakdown structure provides clear project task definition, specific deliverables and success criteria. Tasks may include regulatory documentation requirements (e.g. scan or vulnerability reports) to verify successful completion of a task. In addition, resource management (e.g. team members) is essential to ensuring successful project completion (Snedaker, 2006):

- Are resources available and ready to complete the tasks?
- Are tasks being completed on time?
- How are dependent tasks affected by slipping timelines?
- Do regulatory compliance timelines affect delivery of the task?

4.4.2. Scope is controlled and verified

The scope of a vulnerability management program is key to its success. For example, incomplete asset lists result in fewer assets scanned than required by regulatory requirements. As a result, non-compliance could result in fines or other damages enforced on the company. Asset lists must be maintained (asset addition and removal as required) and verified through scanning.

4.4.3. Cost and schedule are controlled

Compliance with regulatory requirements comes at a cost. After successful project completion there are on-going costs to operate and maintain supporting tools and processes. Vulnerability scans and automated patching should be scheduled to reduce staff cost and internal customer downtime and associated costs.

4.4.4. Performance reporting

Performance reports are used to document:

- project activities
- accomplishments
- project status
- issues

4.4.5. Monitoring and controlling risk

The vulnerability program is required to identify vulnerabilities in the company's computing environment. These vulnerabilities are identified, assessed and assigned a risk rating. Creation of a risk management (mitigation) plan serves to remediate the risk.

5. Conclusion

IT security regulations demonstrate a standard of care in protecting sensitive data. Vulnerability management programs are mandated in the Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act (HIPAA), and Federal Energy Regulation Committee (FERC) regulations to name a few.

Vulnerability management programs are designed to identify and address the inherent security weakness created by software vulnerabilities. Vulnerability assessments are conducted through asset scans. Identified risks are rated and ranked based on risk to the company. These results are shared with appropriate stakeholders so they can be remediated in a timely manner. Scan profiles and techniques are updated as new vulnerabilities and security weaknesses are discovered.

Building the program requires management sponsorship to provide the necessary resources and support for project success. The Plan-Do-Check-Act approach to project management aligns with building and operating a vulnerability management program.

5.1. Plan

The plan phase addresses the:

- Project sponsorship and approval.
- Risk assessment.

- Risk treatment plan.
- Risk acceptance.

5.2. Do

Execution of the risk treatment plan occurs in the “Do” phase. Software patching and mitigation strategies are applied. Compensating controls may be implemented when alternative mitigations are unavailable.

5.3. Check

Monitoring and controlling vulnerabilities is an iterative task. Risks are continuously identified and reviewed as part of on-going scanning efforts.

5.4. Act

Information obtained through previous phases is used to maintain and improve the process. Scan profiles and mitigation strategies are updated to reflect new security weaknesses.

References

Arveson, P. (1998). *The deming cycle*. Retrieved April 19, 2015, from http://balancedscorecard.org/Resources/Articles-White-Papers/The-Deming-Cycle

The best damn IT security management book period. (2007). Burlington, Mass.: Syngress.

Bradley, T. (2007). *PCI compliance: Implement effective PCI data security standard.* Rockland, Mass.: Syngress Pub.

BS ISO-IEC 27005:2008 - Information technology - security techniques - information security risk (1.st ed.). (2008). London: BSI Group.

A guide to the project management body of knowledge (PMBOK Guide) (4th ed.). (2008). Newtown Square, Pa.: Project Management Institute.

Manzuik, S., & Gold, A. (2007). *Network security assessment from vulnerability to patch*. Rockland, MA: Syngress Pub.

Parks, R. (2007). Guide to Critical Infrastructure Protection Cyber Vulnerability Assessment. Retrieved April 19, 2015, from http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/26-CIP_CyberAssessmentGuide.pdf

Payment Card Industry (PCI) Data Security Standard - Requirements and Security Assessment Procedures v3.1. (2015, April). Retrieved April 19, 2015, from https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-1.pdf

Pfleeger, C., & Pfleeger, S. (2012). *Analyzing computer security: A threat/vulnerability/countermeasure approach*. Upper Saddle River, NJ: Prentice Hall.

Roffensian Consulting Inc Project Charter. (n.d.). Retrieved April 20, 2015, from http://www.roffensian.com/Project Charter Skeleton.pdf

Sliger, M. (2008). Agile and the PMBOK(r) Guide. Retrieved April 19, 2015, from http://www.pmp-projects.org/Agile e PMBOK.pdf

Snedaker, S., & Rogers, R. (2006). *Syngress IT security project management handbook.* Rockland, MA: Syngress Pub.

Souppaya, M., & Scarfone, K. (2013). NIST Special Publication 800-40 revision 3 – Guide to patch management technologies. Retrieved April 12, 2015 from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf.

Vulnerability management for dummies (Qualys limited ed.). (2008). Chichester: John Wiley.

Appendix A – Vulnerability Management Program Work Breakdown Structure

Adapted from (Snedaker, 2006):

Assumptions:

1.1 A company standard exists and is applied to all devices (workstations, laptops, servers, routers etc.).
1.2 A patch management solution is in place.

Plan - Early stages:

1.1 Obtain company policies
1.2 Engage stakeholders and subject matter experts
1.3 Obtain requirements (including regulatory)
1.4 Obtain project approval & funding

Plan – Selection of Vulnerability Assessment (VA) tool:

1.1 Determine VA tool requirements:
    1.1.1 Meets regulatory compliance requirements
    1.1.2 Approved scanner
    1.1.3 Compliance reporting
1.2 Automated / scheduled scanning
1.3 Risk-based vulnerability scoring
1.4 Actionable reports – teams know what remediation is required

2.1 Build test cases and test plan. Review against vendor test cases
3.1 Build a scoring methodology used to determine winning VA tool
4.1 Select three (3) vulnerability assessment tools
5.1 Test VA tool against test plan
6.1 Score VA tool
7.1 Assess scoring and determine winning application
8.1 Negotiate pricing and contract terms and conditions.
9.1 Acquire VA tool

Do – Implement VA tool

10.1 Vulnerability management tool installed
11.1 Build policies / scan profiles to meet requirements and objectives. Consider testing for:
    11.2 Default vendor or account passwords
    11.3 Operating system vulnerabilities
    11.4 Application / database vulnerabilities
    11.5 Configuration vulnerabilities

Wylie Shanks, giac@infosecmatters.com

11.6 Protocol vulnerabilities
12.1 Test VA tool

Check - On-going:

13.1 Perform vulnerability scan
- 13.1.1 Build asset and application inventory
- 13.1.2 Prioritize remediation activities based on company security policy, risk (including regulatory requirements), and security posture
- 13.1.3 Determine cost / effort to fix
- 13.1.4 Patch identified vulnerabilities
- 13.1.5 Mitigate risk where necessary while vulnerabilities patched
- 13.1.6 Update IDS/IPS and content filtering system signatures with these vulnerabilities
- 13.1.7 Implement application whitelisting to provide virtual patching (where necessary)
- 13.1.8 Implement browser protection (e.g. Microsoft's Enhanced Mitigation Experience Toolkit (EMET) or sandboxing)
- 13.1.9 Rebuild / replace systems with non-vulnerable software
- 13.1.10 Take system or service offline

Act – Revise policies and scan profiles as new risks as identified

14.1 Revise policies / scan profiles as new vulnerabilities and security risk are identified