Ripple Software Consulting

Pentest +
Vulnerability Scanning

Author: Joseph Lee
Email: joseph@ripplesoftware.ca
Mobile: 778-725-3206

Vulnerability Management Program

Vulnerability Management Program Overview

- **Identify, prioritize** and **remediate** vulnerabilities
- **Organized approach** to scanning with defined workflow
- Scans must be **interpreted and verified** by **trained analyst**
- **Continuous assessment / monitoring**
  - Includes data from **agent-based** approaches to vulnerability detection
  - Reports security related information to the vulnerability management platform
  - Post remediation activities include **re-scanning** infrastructure
- **Risk Appetite**
  - Willingness to tolerate risk
  - Determines the actions taken such as if remediation actions are taken
  - Sometimes costs of remediation outweigh the risks
- **ITSM – IT Service Management**
  - Tracking system for IT issues and vulnerabilities
  - Integration between vulnerability scanners and ITSM can improve workflow
  - ITSM feeds into the remediation workflow
- **IT Governance and Change Management Process**
  - May create bureaucratic hurdles to making remediation / patching vulnerabilities
- **Regulatory environments** can affect vulnerability scanning
  - **PCI-DSS / PCI-SSC (Security Standards Council)**
    - Requires both **internal** and **external** testing
    - Must schedule vulnerability scans **every quarter (3 months)** or after **significant changes to network infrastructure**
    - Internal scans must be conducted by **qualified personnel** (certified and / or experienced)
    - **Must remediate** high-risk vulnerabilities, if found, and repeat scans iteratively
    - External scans must be conducted by **Approved Scanning Vendor (ASV)** authorized by **PCI Security Standards Council (PCI-SSC)**
  - **Federal Information Security Management Act of 2002 (FISMA)** and **Federal Information Security Modernization Act of 2014 (FISMA)**
    - Requires **government agencies** to comply with security standards
    - Systems are categorized into
      - Low impact
      - Moderate impact

        - High impact
    - Further guidance found in:
        - **FIPS-199 Standards for Security Categorization of Federal Information and Information System**
        - **NIST SP 800-53**
- **Corporate Policy**
    - Many organizations implement corporate policy for pen-testing outside legal requirements
    - **Fiduciary responsibility** to shareholders is a semi-legal grey area that motivates continuous monitoring and defence in depth
    - **Cyber-insurance** is becoming more popular choice for corporations to mitigate risk
        - **Policy premium** is affected by **corporate security posture**
        - Insurance may mitigate financial risk to the company, but **may not be able** to **replace destroyed data** or **prevent loss of brand reputation** resulting from data breach

Vulnerability Scanning

- Scanning tools can be automated / scheduled for **continuous monitoring**
- Reports can be delivered automatically over secure channels
- Reports should include critical details such as
    - Name of the vulnerability
    - Overall severity
    - Detailed description
        - Ports/hosts
        - Risk information (CVSS score and vector)
        - Plugin that detected the vulnerability
    - Solution / remediation
    - References to more information from vendors / security researchers
- **Remote vulnerability scans** may result in **high number of false positives** or **low confidence** findings and so should be supplemented with more detailed scans or intrusion attempts (Check SOW beforehand)
- Vulnerability scans are used by both internal cybersecurity teams (blue team) and pen-testers (red team)
- **White box vulnerability scanning**
    - **Credentialed scans** can allow **remote access to the host** to supplement information and provide more detailed info than external service / port scans
    - **Agent based scanning** uses a **software agent on the host** to determine server configurations / service scans
    - Consideration for **virtual-machines** and **containerization**
        - May result in **false negatives** (not finding the services) when using traditional network based vulnerability scanning

        - Agent based scanning can work better in these circumstances
- **Scheduling**
    - Required timeframes for scans can depend on regulatory requirements, compliance, or business operations / policy
    - Determined by the SOW / contract
    - Risk appetite can also determine how often to conduct scans
    - Technical constraints may limit frequency of scanning
    - Business constraints can prevent resource intensive scans during business hours
    - Budget / resource constraints can limit scanning frequency
    - Scanning agent licensing limitations can limit scanner or number of scans that can be conducted per day / month, etc
    - Start slow and increase to prevent overwhelming resources / bandwidth
    - Customized scans designed specifically for the organization or resources
- **Service Focused Scans**
    - **IoT**
        - Devices maybe using bluetooth / bluetooth mesh / or other **fixed** / **mobile** radio frequency spectrum
        - Data in transit maybe unencrypted
        - Interrupting / jamming the device's frequency range may disable devices
    - **Applications**
        - **Vendor / distribution and version detection** can be externally scanned with **Nmap** / other scanning software and mapped to **known vulnerabilities**
        - **Source code analysis** with agents or manually can provide more assurance
        - Vendors may provide **security bulletins** regarding newly discovered / patched vulnerabilities
        - **Credentialed scans** can provide more details about service configuration and reduce **false positives** or find **false negatives**
        - Various scanning agents may detect different vulnerabilities so **vendor diversity** is important
        - Often remediation involves a **reconfiguration** or **update** to service version
        - Non-critical services should be disabled / uninstalled
        - Source code analysis may uncover **unneeded modules** which should be removed
        - **Common Criteria** reports may be available to provide details on security guidelines / configuration for the service application with an **evaluation assurance level (EAL)**
    - **Operating systems**
        - **Vendor / distribution and version detection** can be externally scanned with **Nmap** / other scanning software and mapped to **known vulnerabilities**
        - Vendors may provide **security bulletins** regarding newly discovered / patched vulnerabilities
        - **Credentialed scans** can provide more details about service configuration and reduce **false positives** or find **false negatives**

        - Remove unneeded user accounts / software packages
        - Scanning for file permissions can detect mis-configured permissions that can be remediated with **least privilege**
        - **Common Criteria** reports may be available to provide details on security guidelines / configuration for the service application with an **evaluation assurance level (EAL)**
    - **Scan perspective**
        - Conducting scans from different locations on the network (internal / external)
        - Conducting **credentialed scans** or **black-box scans** provided different levels of assurance / information
- **Identify scan targets**
    - Approach to building an asset inventory be require a complete asset catalog or limited depending on requirements of contract / SOW
    - **Vulnerability scanner plugins** should be updated regularly
    - Targets should include
        - **Data in storage**
            - Local hard-drives
            - Network attached storage
            - Cloud storage
        - **Data in use**
            - Data in RAM
        - **Data in transit**
            - Data sniffed on the wire
            - Data sniffed in Wifi spectrum
    - Other endpoints to consider
        - Public facing IP(s)
        - Private WAN / private leased network
        - Traceroute of **intermediary appliances** such as switches / routers / hubs / IDS / IPS / Firewalls / VPN concentrator / Remote desktop concentrators
        - **MX records** can provide details about **mail-servers**
        - **TXT records** can provide details about 3rd party cloud services
        - **Sublist3r** can provide **DNS records** for **other sub-domains**
    - QualysGuard / Nmap / other scanners provide **asset inventory functionality**
    - **Data classification** and valuation of assets can determine **remediation prioritization**
- **Scoping**
    - Use **network segmentation** to limit the scope of the network that needs to be scanned
        - Using **subnet CIDR** or **VLAN** to limit scope
    - Endpoints on large networks can be **categorized** and tested categorically instead of testing each system
    - Example: **PCI-DSS** has requirements on network segmentation

    - Configure scanning software to specific needs of the assessment
    - Create templates / workflow for various types of scans
- **Critical / Fragile systems**
    - Systems critical to business operations should be considered for time to schedule scans as to **not interrupt critical business operations**
    - Systems can be classified into production / test / development systems
    - ICS, IoT, medical equipment should be tested in testing environment first rather than production
    - **Customer Commitments**
        - MOU (Memorandum of understanding) and SLA (Service level agreements) create expectations related to uptime, performance, and security and should be considered when planning a pen-test
        - Scanning may negatively impact uptime availability so customers should be notified of these risks
- **Stealth**
    - Use stealth settings to avoid detection especially if red-teaming / organization's employees are not aware of pen-test activity
    - Better approximates the activity of real-world attacks
    - If not red-teaming or deadlines required then skip stealth modes
- **Documented Exceptions**
    - Organizations may decide to not remediate a vulnerability for some reason
        - Reliance on legacy systems maybe required for operations
        - Cost / risk analysis may prove to costly
        - Exceptions can be documented so the results don't show up in scans to save time
        - Be aware that creating an exception **may violate legal or industry standard compliance** or go against best practices

Vulnerability Scan Analysis

- Scanners produce reports that need to be interpreted by trained analyst
- Validating Scan results
    - **False positives**
        - Analyst should verify all results found buy automated scanner
        - Testing the vulnerability by exploitation if possible (check SOW)
- Scanner that use the **CVSS standards** allow faster **mapping of vulnerabilities to risk**
- Reconcile scanner results with other data sources
    - **Logs** from servers, applications, network devices, etc.
    - **SIMS / SEMS / SIEMS**
        - Correlated log entries from networks / systems
    - **Configuration Management Systems**

        - Provide information on the operating system, applications, etc.
- **Trend analysis**
    - **Industry reports on attack trends** can help calculate risk / point to new vectors that should be scanned
        - **OWASP top 10**
            - https://owasp.org/www-project-top-ten/
        - **IBM X-Force Threat Intelligence**
            - https://www.ibm.com/security/xforce
        - **TrendMicro Threat Reports**
            - https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threat-reports
    - The age of existing vulnerabilities can determine **accessibility to exploit code**
        - Older vulnerabilities are more likely to have more sophisticated and readily available exploits
    - Trend analysis can help stay ahead of the attackers and provide defence in depth
    - Trend analysis reports are available built-into some vulnerability scanning software
    - Analysts should stay in touch with most common vulnerabilities and categories

NIST SP 800-53 Security Privacy Controls for Federal Information Systems and Organizations

- All **federal systems** must conform to **NIST SP-800 53** irregardless of their categorization
- **FIPS 199** provides standards for categorization of federal systems
    - Risk categories are applied to each **C** (confidentiality), **I** (integrity), **A** (availability)
    - **LOW**
    - **MODERATE**
    - **HIGH**
- **NIST SP 800-53 Control Description**
    - **a.** Scans for vulnerabilities in the information system and hosted applications and when new vulnerabilities are reported
    - **b.** Employs vulnerability scanning tools and techniques that facilitate interoperability among tools and automate parts of the vulnerability scanning process by using standards for:
        - **1.** Enumerating platforms, software flaws, and improper configurations
        - **2.** Formatting checklists and test procedures
        - **3.** Measuring vulnerability impact
    - **c.** Analyzes vulnerability scan reports and results from security control assessments
    - **d.** Remediate legitimate vulnerabilities in accordance with an organizational assessment of risk
    - **e.** Shares information obtained from the vulnerability scanning process and security control assessments to help eliminate similar vulnerabilities in other information

systems (i.e. systematic weakness or deficiencies)

SCAP – Security Content Automation Protocol

- Led by **NIST** to create standardized approach for communicating security-related information
- **NIST SP 800-117 Guide to Adopting and Using Security Content Automation Protocol**
  - https://csrc.nist.gov/projects/security-content-automation-protocol/
- **CCE – Common Configuration Enumeration**
  - Standard language for system configuration issues
- **CPE – Common Platform Enumeration**
  - Standard language for describing product names and versions
- **CVE – Common Vulnerability Enumeration**
  - Standard language for describing security-related software flaws
- **CVSS – Common Vulnerability Scoring System**
  - Standard language for describing severity of security-related software flaws
- **XCCDF – Extensible Configuration Checklist Description Format**
  - Language for specifying checklists and reporting checklist results
- **OVAL – Open Vulnerability and Assessment Language**
  - Language for specifying low-level testing procedures used by checklists
- **Newly added since 2011**
  - **OCIL – Open Checklist Interactive Language**
    - Defines a framework for expressing a set of questions to be presented to a user and corresponding procedures to interpret responses to these questions
  - **AID – Asset Identification**
    - Provides the necessary constructs to uniquely identify assets based on known identifiers and/or known information about the assets
  - **ARF – Asset Reporting Format**
    - Data model to express the **transport format of information about assets**, and the relationships between assets and reports
  - **CCSS – Common Configuration Scoring System**
    - Set of measures of the severity of software **security configuration issues** derived from CVSS
    - Specifically applies to **configuration** as opposed to CVSS which applies to **vulnerabilities**
  - **TMSAD – Trust Model for Security Automation Data**
    - Describes a common trust model that can be applied to specifications within the security automation domain, such as Security Content Automation Protocol (SCAP)
  - **SWID – Software Identification tags**
    - Files containing descriptive information about a specific release of a software product
    - Defines a lifecycle where a **SWID Tag** is **added to an endpoint** as part of the

software product's **installation process** and **deleted** by the **product's uninstall process**
        - Designed to ensure that **all deployed software** assets are configured according to their organizations' security policies

CVSS – Common Vulnerability Scoring System

- Rating the vulnerability on **6 different measures**:
    - **Access vector –** How an attacker will exploit the vulnerability
        - **L – Local (+ 0.395)** Must have physical access or logical access to the affected system
        - **A – Adjacent Network (+ 0.646)** Must have access to the local network that the affected system is connected to
        - **N – Network (+ 1)** The attacker can exploit the vulnerability remotely over a network
    - **Access complexity –** Difficulty level in exploiting the vulnerability
        - **H – High (+ 0.350)** Requires specialized conditions/skills that are difficult to find
        - **M – Medium (+ 0.610)** Requires somewhat specialized conditions/skills
        - **L – Low (+ 0.710)** Does not require any special conditions/skills
    - **Authentication –** Describes the authentication required to exploit the vulnerability
        - **M – Multiple (+ 0.450)** Two or more authentications required
        - **S – Single (+ 0.560)** One authentication required
        - **N – None (+ 0.704)** No authentication required
    - **Confidentiality –** Describes the type of information disclosure that might occur
        - **N – None (+ 0)** There is no information disclosure
        - **P – Partial (+ 0.275)** Access to some information but the attacker does not have complete freedom over what information is disclosed
        - **C – Complete (+0.660)** All information on the system is compromised
    - **Integrity –** Whether or not information or system configuration can be altered
        - **N – None (+ 0)** No information or system config can be altered
        - **P – Partial (+ 0.275)** Modification of some information is possible, but attacker does not have complete control over what information is modified
        - **C – Complete (+ 0.660)** The entire system integrity is compromised, and the attacker can change any information
    - **Availability –** The type of disruption possible
        - **N – None (+ 0)** No availability impact
        - **P – Partial (+ 0.275)** System performance is degraded
        - **C – Complete (+ 0.660)** Complete system shutdown / unavailable
- **CVSS Vector –** Uses a single line format to convey the ratings of a vulnerability on all six metrics
    - Example: **CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N**
- **Summarizing CVSS Score**

    - Exploitability = 20 X Access Vector X Access Complexity X Authentication
    - Exploitability for above vector: 20 X 1 X 0.610 X 0.704
    - Exploitability = 8.589
- **Impact Score**
    - Impact Score = 10.41 X (1 – (1 – Confidentiality) X (1 – Integrity) X (1 – Availability))
    - For the above vector: 10.41 X (1 – (0.725) X (1) X (1))
    - Impact = 10.41 X 0.275
    - Impact = 2.863
- **Impact Function**
    - If impact score is 0, impact function = 0
    - Else impact function = 1.176
- **CVSS Base Score**
    - Base Score = ((0.6 X Impact) + (0.4 X Exploitability) – 1.5) X Impact Function
    - Base Score for above example = 4.297
        - **Nessus Risk categories:**
            - CVSS < 4.0 – Low
            - CVSS > 4.0 and < 6.0 – Medium
            - CVSS > 6.0 and < 10 – High
            - CVSS = 10 – Critical

Software Security Testing

- **Static Code Analysis / Source Code Analysis**
    - Considered **white-box** testing
    - Decompilation required for compiled proprietary software
    - OWASP provides static code analysis tools:
        - .NET, Java, PHP, C, JSP, and others
        - https://owasp.org/www-community/controls/Static_Code_Analysis
- **Dynamic Code Analysis**
    - **Fuzzing / monkey fuzzing / fault injection**
        - Noisy will attract attention from blue team / cybersecurity dept.
        - Fuzzing process can be time intensive
        - Fuzzed data input can be remediated by strict data input handling
        - **Monkey fuzzing**
            - Sending random data to check behaviour of an application / service
            - May result in DOS or trigger other vulnerability
- **Software Vulnerability Scanners**
    - Full descriptions available in **Exploit_Tools.pdf**
    - **Web-Application Scanners**
        - Acunetix WVS

        - Arachni
        - IBM AppScan
        - HP WebInspect
        - Netsparker
        - QualysGuard Web-Application Scanner
        - W3AF
        - Nikto / Nikto2
        - Nessus
        - Nexpose
    - **Interception Proxies**
        - TamperData
        - Fiddler
        - Burp-Suite
    - **Database Vulnerabilities**
        - SQLMap
        - SQLNinja

Remediation Workflow Cycle

- **Testing**
    - Testing can happen in testing / sandbox environment
    - Since the process is a cycle, any remediations should be tested
- **Detection**
    - Service degradation is concern when testing production environments
    - Query throttling and scheduling can alleviate service degradation
- **Remediation**
    - Prioritization of vulnerabilities
    - Criticality of the systems and information affected by the vulnerability
    - Difficulty in remediating the vulnerability
    - Severity of the vulnerability
    - Exposure of the vulnerability
    - Document all steps taken in remediation process

Common IT Vulnerabilities

- **Server and Endpoints**
    - Remotely available endpoints are easy to attack
- **Missing Patches / Updates**
    - Core element of any information security program / management system
- **Mobile devices**
    - Often require separate individual scanning since they may not be on the network at

all times

- **Unsupported OS and Applications / Legacy**
  - Limit access / Air-gap these devices as much as possible
  - Increase monitoring of legacy devices
  - Strict firewall rules
  - Apply IDS / IPS in the network
  - Uninstall unneeded applications (i.e. browsers)
- **Buffer Overflows**
  - Inserting more data into memory than is allocated to the application
  - Overwrites other information in memory
  - If writing into executable memory then the code may be executed
  - Caused by programming errors / bad exception handling / bad error handling
- **Privilege Escalation**
  - Increase the level of access that attacker has to target system
  - Highest level is **root, admin** or **superuser**
  - Example: Dirty COW
    - https://dirtycow.ninja/
- **Arbitrary Code Execution**
  - Allows attacker to run software code of their choice on the system
  - Catastrophic for security if run with root or admin privileges
  - **Remote code execution**
    - More dangerous subset of code execution vulnerabilities
    - Attacker can run code from a network connection
- **Hardware**
  - **Meltdown** and **Spectre**
  - Microcode in hardware
  - Shimmed drivers
- **Firmware Vulnerabilities**
  - Code may contain vulnerabilities
  - Often lack auto-updating mechanism
  - Often remain un-patched
  - The firmware update vector can be attacked by attackers and shimmed with malicious firmware
  - BIOS attacks are very low level attack on system
- **Embedded Systems**
  - Often have full operating systems with network access on them
  - Can be good initial entry point to a network
  - Credentials are often left as default
- **Insecure Protocols**
  - **Telnet / FTP** and any unencrypted protocols
    - Credentials can be sniffed
    - Data can be injected / altered in transit

        - Switch to more secure protocols instead
    - **Authentication protocols** that have been broken
        - Kerber-ROAST
    - **Encryption protocols** that have been broken
        - **SSL – Secure Socket Layer**
    - **Encryption ciphers / cryptographic algorithms** that are weak
        - Can be sniffed and decrypted
        - DES / RC4
    - **Certificate problems**
        - Are these being checked and authenticated properly?
        - Mismatch between name on cert and name on server
        - Expiration of the digital certificate
        - Unknown certificate authority
    - **DNS Domain Name System**
        - DNS amplification attacks
    - Internal IP disclosure
        - Application packet headers
        - VPN packets
    - **Virtualization**
        - VM escape
        - Management Interface Access
        - VM Guests
            - Contain all the vulnerabilities that a regular host would and so they must be patched
            - Provide an attacker with full network access same as regular host on the network
- **Other network devices**
    - **IoT**
    - **SCADA**
    - **ICS**
    - **Embedded systems**
    - **RTOS**
- **Web application vulnerabilities**
    - **Injection attacks**
    - **Cross-site scripting (XSS)**
- **Debug Mode**
    - If server left in debug mode critical data can be transferred over the network
    - Users who have been given debug permissions may have admin privileged access

Vulnerability Information Sources

- **Mitre CVE**

    - https://cve.mitre.org/cve/
    - A list of publicly disclosed cybersecurity vulnerabilities that is free to search, use, and incorporate into products and services
- **NVD – National Vulnerability Database**
    - https://nvd.nist.gov/
    - Launched by the National Institute of Standards and Technology (NIST) in 2005
    - Uses the Security Content Automation Protocol (SCAP)
    - Provides API, bulk-downloads, and web-interface
- **CVE Details**
    - https://www.cvedetails.com/
    - https://www.itsecdb.com/oval/
    - Provides a web interface to all IT security related items including patches, vulnerabilities and compliance checklists
    - Collects OVAL (Open Vulnerability and Assessment Language) definitions from several sources
        - Mitre
        - Red Hat
        - Suse
        - NVD
        - Apache
- **Bugtraq ID (BID)**
    - https://www.securityfocus.com/bid/
    - CVE to BugTraq ID concordance
        - https://cve.mitre.org/data/refs/refmap/source-BID.html
- **VulnDB**
    - https://vulndb.cyberriskanalytics.com/
    - Proprietary paid product
    - Based on now depreciated OSVDB Open Source Vulnerability Database
- **Veracode (2017)**
    - https://info.veracode.com/report-state-of-software-security.html
- **OWASP Top 10 Security Issues**
    - https://owasp.org/www-project-top-ten/