Marks Distribution of Course

- Assignments 8%
- Quizzes 8%
- Paper write-up 14%
- Mid-term Exam 25%
- Final 45%

Paper Write-up Assignment

1. Select any Area/Topic of your interest
2. Perform Literature Review of the related
   a) Research Papers
   b) Patents
   c) Standards
   d) Open Source Solutions
   e) Products
3. Identify the gap and formulate
   a) Research Gap/Problem Statement
   b) Research Question(s)
4. Prepare Document and Presentation, submit both document and presentation

What Is Vulnerability Management?

- Practice of staying aware of known vulnerabilities in an environment
- Resolving or mitigating these vulnerabilities to improve the environment's overall security posture
- It entails a number of interdependent activities

What Is Vulnerability Management?

- Practice of staying aware of known vulnerabilities in an environment
- Resolving or mitigating these vulnerabilities to improve the environment's overall security posture
- It entails a number of interdependent activities

Vulnerability Management Life cycle

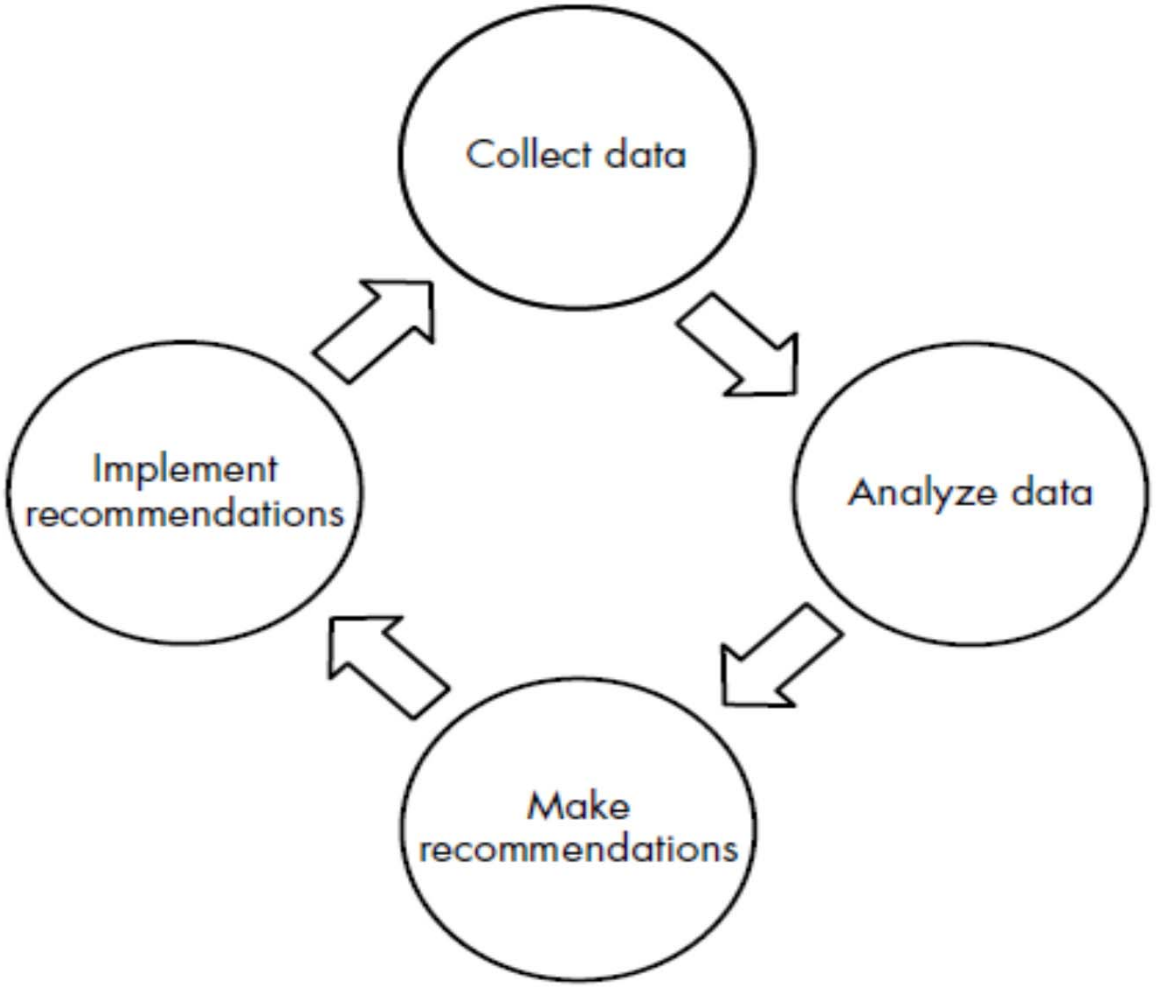


Figure 1-1: The vulnerability management life cycle

Vulnerability Management Life cycle

- Collect data about your systems to determine the vulnerabilities that exist on them
- Analyze the collected data as well as security-related data from other sources
- Data analysis results will help in making recommendations about the actions needed to improve security posture
- Recommendations might include installing patches or applying mitigations, such as firewall rules or system-hardening techniques
- Implement the recommendations

Vulnerability Management Life cycle

- The cycle begins again
- Collect another round of systems data and the vulnerabilities that remain after analysis and mitigation
- As well as new vulnerabilities that weren't apparent in the previous cycle
- The management process is neither short nor simple
- Finding vulnerabilities can be easy, but dealing with them and improving your security baseline will be ongoing
- The process will also involve many different roles and business processes throughout the organization

Collecting Data

- Gathering information about your organizational environment
- Includes information about the hosts on your network
- —endpoints and network devices—and vulnerability information about each host
- Host information can come from an exploratory scan using a network-mapping tool (like Nmap)
- Vulnerability data comes from one source: vulnerability scanners
- These tools discover vulnerabilities by interacting with devices, either through network-based scans or host-based agents

Collecting Data

- Network scanners reach out to every IP address within a range, or a specific list of IPs
- To determine which ports are open, which services are running on those ports
- The operating system (OS) versions and relevant configurations
- Software packages running on each device
- Host-based scan-less agents query the system directly to determine running services and version information

Collecting Data

- The internal collected data quickly becomes stale
- Must gather it regularly
- Vulnerability information changes daily:
  - people install new software packages or
  - perform updates, and new vulnerabilities are discovered and publicly disclosed
- Regular scanning and routine scanner updates to incorporate new vulnerability information ensure to have accurate and complete data about the current environment
- Regular scanning might have negative effects

Collecting Data

- Must balance this risk against the importance of having accurate vulnerability data
- External data collection encompasses the data sources that come from outside your organization
- This information includes public vulnerability details, embodied by the constantly growing mass of common vulnerabilities and exposures (CVE) data
- NIST provides; public exploit information from the Exploit Database and Metasploit; additional vulnerability, mitigation,
- Exploit detail from open sources like CVE Details (https://cvedetails.com/)

Collecting Data

- Any number of proprietary data sources, such as threat intelligence feeds
- Querying online sources directly or keeping local data repositories
- collecting data from third-party sources is as easy as reaching out and getting it
- keep a live connection in the case of threat intelligence feeds

Analyzing Data

- Need to analyze this data to gain useful vulnerability intelligence about your environment
- Scanners will find many vulnerabilities on nearly every device
- Separating important vulnerabilities from the unimportant ones can be difficult
- reduce the list of vulnerabilities to a more manageable length, known as culling
- Culling is straightforward: it's a binary yes-or-no decision you make on every vulnerability
- The criterion for accepting a vulnerability might be, for example, the vulnerability is newer or Zero day

Analyzing Data

- Ranking requires a criterion using some sort of scale
- For instance, you could rank a set of vulnerabilities based on their effects on confidentiality, integrity, or availability
- Use the Common Vulnerability Scoring System (CVSS),
- A 1-to-10 scale that takes into account a vulnerability's severity along all three of the CIA triad's axes
- A strong understanding of your organization's risk landscape
- Your own scoring system that focuses on internally developed risk metrics

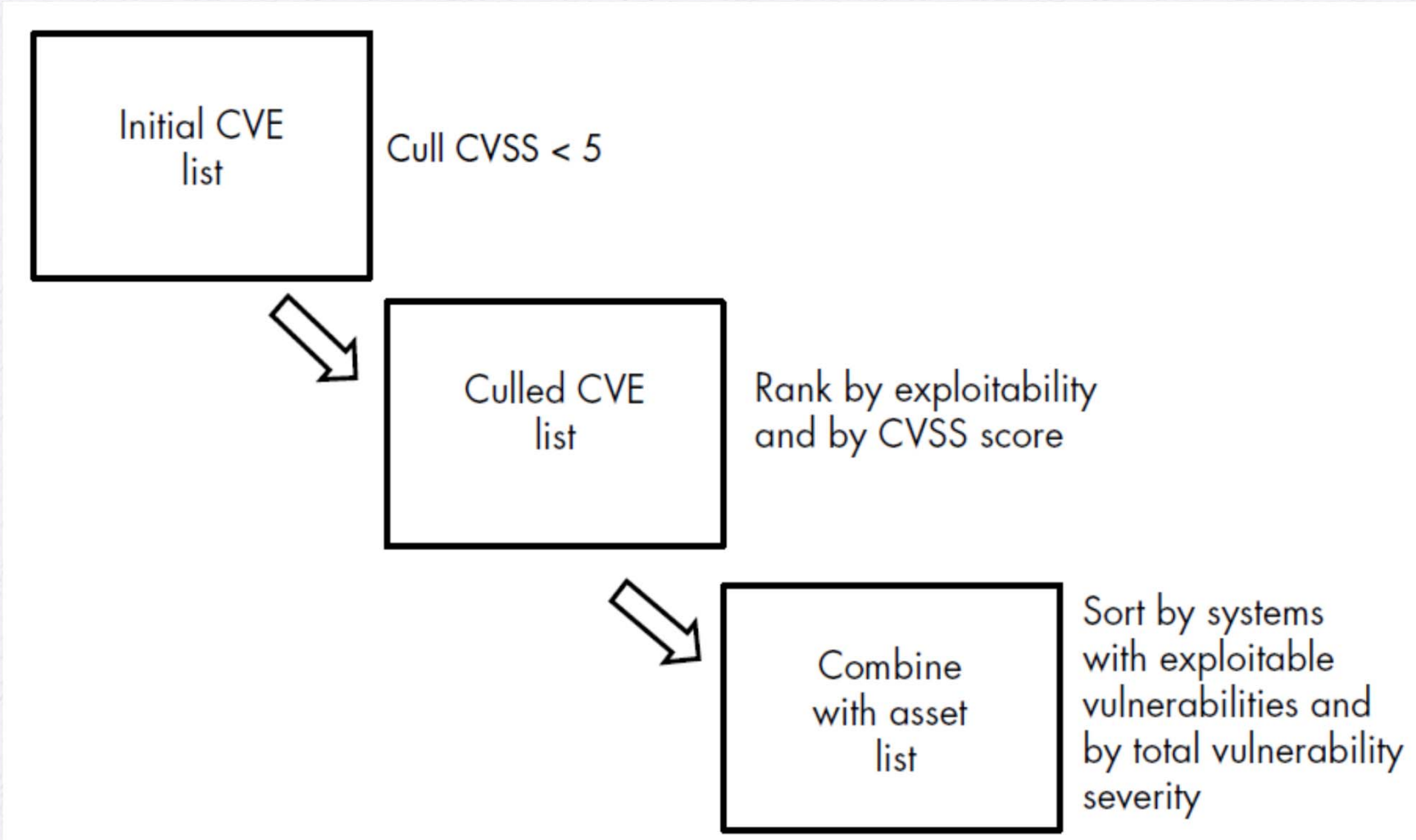
Analyzing Data

- Culling results in a smaller dataset to analyze, whereas ranking is an analysis method in itself, consider using both
- By first culling the vulnerability set, limit subsequent analysis to vulnerabilities making analysis faster and more relevant
- Once you identify the most critical vulnerabilities
- You can rank the remaining vulnerabilities to more easily
- Determine their relative significance

Analyzing Data

- Cull vulnerabilities with a low CVSS score
- Rank the remaining vulnerabilities by exploitability and then by CVSS score, from high to low
  Combine this list with the asset list
- Rank the resulting list first by the number of exploitable vulnerabilities per system and then by the total severity of vulnerabilities found on the system
- The resulting list shows the systems with the highest risk at the top

Analyzing Data



Applying Cull-Rank to a Real-World Example

- Ran a vulnerability scan
- Scan result shows a list of approximately 2,000 total vulnerabilities spread across 84 devices
- Cull vulnerabilities with a CVSS score less than 5
- Cutting your list to about 500 vulnerabilities on 63 devices
- 38 unique vulnerabilities—as most of the vulnerabilities exist on multiple hosts
- Find out whether any of these 38 unique vulnerabilities have publicly known exploits

Applying Cull-Rank to a Real-World Example

- If they do, you need to address those vulnerabilities first
- Establish what the CVSS severity of each vulnerability is
- Higher severity means greater consequences of compromise
- Focus on the more severe vulnerabilities
- 3 have known exploits, and the remaining 35 have been sorted in order of CVSS severity
- Combine the list of vulnerabilities with the actual vulnerable hosts
- For each host, determine how many vulnerabilities it has and the severity of those vulnerabilities

Applying Cull-Rank to a Real-World Example

- Clear picture of where need to focus your remediation efforts
- Among those 63 hosts with vulnerabilities, 48 have one to two vulnerabilities of severity no higher than 7
- Whereas 11 have up to 15 vulnerabilities with one or two in the critical range (CVSS of 9 and higher)
- The last four contain all the rest of those 500 total vulnerabilities among them—an average of 125 on each host, including all three exploitable vulnerabilities!
- Clearly these systems need heavy remediation, and you have a good argument for addressing the situation immediately

Making Recommendations

- Now that you have a list of hosts and vulnerabilities that is sorted by risk to your organization
- Recommend actions to remediate the vulnerabilities
- Start with the highest risk and work way down the list
- Involves working with system and application owners as well as other stakeholders
- Major types of remediation are patching and mitigation
- Patching is simple: you apply the patch that resolves the vulnerability in question

Making Recommendations

- Mitigation is more complex and is context dependent
- If a patch isn't available or if it's infeasible to apply one
- Look at other ways to address the risk
- Perhaps changing a configuration will prevent a specific vulnerability from being exploited
- Perhaps the vulnerable service isn't needed outside specific IP ranges so you can protect it with firewall rules or router access control lists (ACLs), reducing the exposure

Making Recommendations

- Perhaps an existing intrusion detection system (IDS) or intrusion prevention system (IPS) needs additional rules to detect whether someone is attempting to exploit that specific vulnerability and block it
- All of these are examples of vulnerability mitigation, and the correct response will depend on organization environment

Implementing Recommendations

- Approach the system and application owners to suggest they implement the proposed remediation actions
- If they were involved in the recommendation process, this step should be straightforward
- If the recommendations are unexpected, explain the security risks and the reasons for the recommendations
- All agree on a timeframe for the implementation
- Once those responsible have implemented the recommendations—via patching or mitigation—the final step is to verify that the changes have been made and are effective