A Guide to Automating Vulnerability Prioritization Using SSVC Decision Trees

Author: Patrick Garrity, Nucleus Security
Status: Call for Feedback

Automating Vulnerability Prioritization Using Decision Trees

The purpose of this guide is to propose an approachable framework for helping vulnerability management teams automate vulnerability prioritization using decision trees that are aligned with Stakeholder-Specific Vulnerability Categorization. By providing decision outcomes that can be automated, organizations can scale the broader use and adoption of the SSVC decision tree-based logic for vulnerability prioritization. This document presents a real-world example of decision criteria that could be used through automation and built into rule sets using an enterprise vulnerability management tool like Nucleus Security. This guide focuses specifically on helping vulnerability management teams automate vulnerability prioritization using decision trees for CVE based vulnerabilities.

An Introduction to Stakeholder-Specific Vulnerability Categorization (SSVC)

Before diving into automation vulnerability prioritization using decision trees, it is important to provide a high-level overview of Carnegie Mellon's Stakeholder-Specific Vulnerability Categorization (SSVC) framework. The SSVC paper, authored by Jonathan Spring, Eric Hatleback, Allen D. Householder, Art Manion, and Deana Shick, introduces a systematic approach to categorizing vulnerabilities based on stakeholder perspectives. It emphasizes the need to align vulnerability prioritization with organizational risk tolerance and strategic objectives.

Before SSVC, people primarily had to use the Common Vulnerability Scoring System (CVSS) for vulnerability prioritization with their vulnerability management program. But by implementing the SSVC framework, which streamlines vulnerability triage and decision making, they can now align themselves on key criteria of what vulnerabilities are prioritized and why based on the decision criteria.

Dependencies for Automating Vulnerability Prioritization Using Decision Trees

Automating vulnerability prioritization using decision trees at scale requires several dependencies to be in place. These dependencies are crucial to achieve organizational alignment and for effectively aggregating, normalizing, and analyzing data to facilitate automated decision-making. It’s worth mentioning that not all dependencies are required to implement SSVC decision trees, and we recommend considering taking a reiterative approach to refining the decision tree criteria.

The following dependencies should be considered:

1. **Organizational alignment:** Achieving organizational alignment is crucial in the process of automating vulnerability prioritization using decision trees. It involves ensuring that all stakeholders understand and support the automation initiative. This alignment requires clear communication, collaboration, and coordination among departments, along with defined roles, governance structures, and necessary resources. With organizational alignment, automation efforts can be seamlessly integrated into existing workflows, leading to streamlined and efficient vulnerability management practices.

2. **Vulnerability and Asset Inventory:** To automate vulnerability prioritization, you need to have a comprehensive and up-to-date vulnerability and asset inventory, which includes information about the assets within your organization's infrastructure. This provides valuable data on the asset's vulnerabilities present in your environment. Aggregating and normalizing this data is essential for the automation of effective decision-making.

3. **Reliable Sources of Vulnerability Intelligence:** Access to reliable sources of vulnerability intelligence is critical for accurate prioritization. These sources provide information about vulnerabilities, their severity, exploitability, and potential impact. Examples of such sources include the National Vulnerability Database (NVD), EPSS (Exploit Prediction Scoring System), CVSS (Common Vulnerability Scoring System), CISA's Known Exploited Vulnerabilities (KEV) list, Mandiant, GreyNoise, Intel471, Recorded Future, and other relevant intelligence sources. Correlating data from multiple sources can greatly enhance the accuracy of decision-making

4. **Correlation of Asset Metadata:** For the automation of more advanced decision criteria, correlating asset metadata across different sources can significantly improve the ability to automate decision-making at scale. Asset metadata includes information such as asset criticality, data sensitivity, compliance scope, asset exposure (internal or external), asset ownership, and other relevant attributes. By considering this metadata, decision trees can incorporate additional context and tailor prioritization decisions based on

specific asset characteristics.

5. **Automation Capabilities:** To achieve automated vulnerability prioritization, you need suitable automation capabilities that enable the decision-making process. These capabilities may involve leveraging vulnerability management tools, security orchestration and automation platforms, or custom-built solutions. Automation facilitates the efficient processing of data, applying decision rules, and generating actionable insights.

By addressing these dependencies, organizations can establish a robust foundation for automating vulnerability prioritization using decision trees. It allows for more efficient and effective allocation of resources towards mitigating the most critical vulnerabilities based on the specific context and risk profile of their assets.

Outlining Possible Decision Outcomes for Vulnerability Prioritization

To effectively use decision trees for vulnerability prioritization, it is crucial to define the decisions based on specific decision criteria. In this example, we have identified four distinct decisions taking a similar approach as CISA: Track, Scheduled, Out-of-Cycle, and Act. These decisions are defined as follows:

Track - The vulnerability does not require immediate action. It should be continuously monitored, and reassessment can be conducted if additional information becomes available.
Scheduled - The vulnerability exhibits characteristics suggesting that remediation should be scheduled during normal patch cycles.
Out-of-Cycle - The vulnerability demands attention from internal supervisory-level individuals. Actions include seeking assistance or further information regarding the vulnerability. Remediation should be completed earlier than standard update timelines.
Act - The vulnerability requires immediate attention from internal supervisory-level and leadership-level individuals. Prompt stakeholder meetings are typically necessary to determine the overall response, followed by the execution of agreed-upon actions as soon as possible. Remediation should be prioritized for immediate completion.

Outlining Decision Criteria for Vulnerability Prioritization

In the next step, we outline the decision criteria used to prioritize vulnerabilities and determine appropriate actions. When determining criteria and thresholds for automated decision making in vulnerability prioritization, several factors should be considered:

1. **Start with a Best Effort Basis:** It is vital to define decision criteria based on readily available vulnerability intelligence and asset attributes within your organization.

2. **Continuously Reiterate Decision Criteria:** Regularly review and refine decision criteria as your organization matures, keeping up with emerging threats and changes in the risk landscape.
3. **Incorporate Stakeholder Feedback:** Engage key stakeholders to gather input and ensure the decision criteria align with your organization's risk tolerance and strategic objectives.
4. **Monitor and Evaluate Effectiveness:** Establish mechanisms to assess the effectiveness of the chosen criteria and thresholds, monitoring key metrics and gathering feedback for improvement.

For this guide, the example provided is a relatively basic example of decision criteria. We believe that decision criteria can evolve and become more sophisticated.

In the example we provide, we employ three decision criteria that that be automated using vulnerability intelligence and asset context/metadata: Exploitation Status, Asset Exposure, and Asset Criticality.

Decision Criteria 1: Exploitation Status

Active – There is known active exploitation or an extremely high probability in the next 30 days.
POC (Proof of Concept) – There is a known proof of concept exploit or a high probability of exploitation in the next 30-days
None – There is no evidence of active exploitation, no known exploit, and a low probability of exploitation in the next 30-days

Decision Criteria 2: Asset Exposure

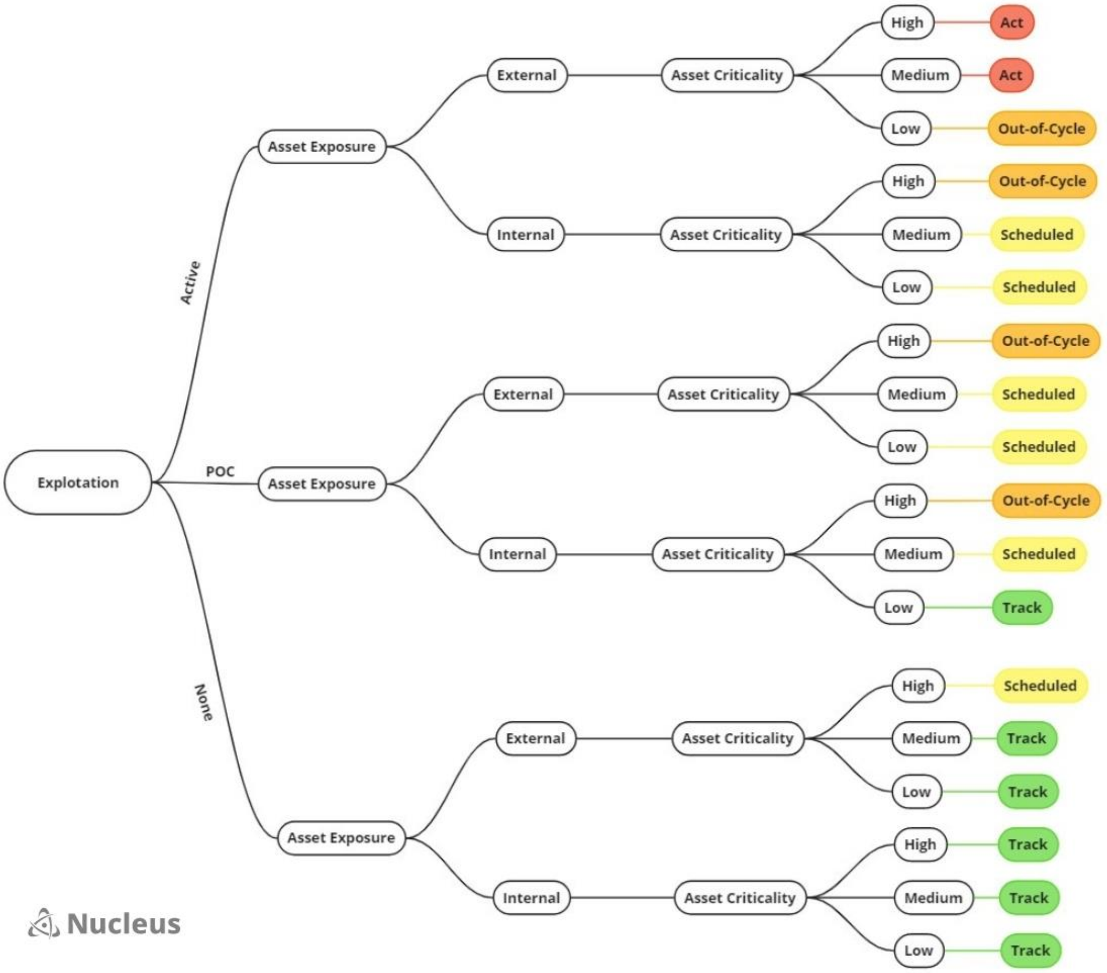
External – The asset is exposed to the internet
Internal – The asset has no expose to the internet

Decision Criteria 3: Asset Criticality*

High – The vulnerable asset is essential to the organization's critical operations and mission.
Moderate – The vulnerable asset supports the organizations critical operations and mission but is not essential.
Low – The vulnerable asset does not support any critical systems.

*Asset Criticality is dependent on having asset context and it's worth noting that if you don't have this readily available, you might consider sensible defaults or removing asset criticality as a decision until you have the right asset context to accomplish this.

Visualizing the Vulnerability Prioritization Decision Tree



To enhance understanding and navigation of the decision-making process for vulnerability prioritization, it is essential to visualize the decision tree. The decision tree captures the relationships between the decision criteria (Exploitation Status, Asset Exposure, and Asset Criticality) and the corresponding actions (Track, Scheduled, Out-of-Cycle, and Act). Each node in the decision tree represents a decision criterion, and the branches represent different possible values or outcomes for that criterion.

As the decision tree branches out, it reflects the different combinations and paths that lead to specific actions. It's worth noting that broader and more granular decision criteria could expand the decision tree choices and the potential outcomes resulting in a seemingly infinite number of possibilities.

Building Decision Tree Rules for Automation

Now that we've built and modeled out our vulnerability decision tree criteria, we can build out automation rules to be used within your vulnerability management tool. We use a table to outline the criteria that can be used to build the automation rules using the decision criteria above. The attributes can easily be replaced with the vulnerability intelligence and asset metrics defined in the next step. Here is a table outlining the criteria we will use for implementing rules that model the decisions outlined above.

Exploitation	Asset Exposure	Asset Criticality	Decision
None	Internal/External	Medium/Low	Track
None	Internal	High	Track
None	External	High	Scheduled
POC	Internal	Low	Track
POC	External	Low	Scheduled
POC	Internal/External	Medium	Scheduled
POC	Internal/External	High	Out-of-Cycle
Active	Internal	Low/Medium	Scheduled
Active	Internal	High	Out-of-Cycle
Active	External	Low	Out-of-Cycle
Active	External	High/Medium	Act

Implementing these rules would look like this:

Example Automation Rules

Example Rule 1: IF Exploitation is None AND Asset Criticality is Medium or Low, THEN Decision is Track.

Example Rule 2: IF Exploitation is POC AND Asset Criticality is High, THEN Decision is Out-of-Cycle.

Example Rule 3: IF Exploitation is Active AND Asset Exposure is (High OR Medium) AND Asset Criticality is High/Medium, THEN Decision is Act.

Defining Vulnerability Intelligence & Asset Metrics for Automation

Defining vulnerability intelligence and asset metrics is crucial for automating key decision criteria. It is important to consider different threat intelligence sources and asset context to effectively assess vulnerabilities.

In this example, we demonstrate the use of multiple criteria to determine a vulnerability's exploitation status. These criteria include CISA's Known Exploited Vulnerabilities list, Mandiant's Commercial Vulnerability Intelligence, and whether a Metasploit module exists. However, it is essential to note that the provided thresholds are for illustrative purposes only.

To optimize risk management within your organization, it is highly recommended to conduct a thorough evaluation to identify the most suitable vulnerability exploitation and asset criteria aligned with your risk tolerance. When adopting decision trees for the first time, it is essential to leverage the available information to make informed decisions on the criteria and continuously refine them as you gain insights into what is both appropriate and achievable for your organization.

Exploitation Status Criteria

Active - CISA KEV **OR** Mandiant Exploit Rating: Wide, Confirmed
POC - Mandiant Exploit Rating: Confirmed, Available **OR** Metasploit Module: Exists
None - Mandiant Exploit Rating: No Known **OR** Metasploit Module: Does Not Exist

Asset Exposure Criteria

External - External IP Address **OR** Asset Inventory Attribute – External
Internal - Internal IP Address **OR** Asset Inventory Attribute – Internal

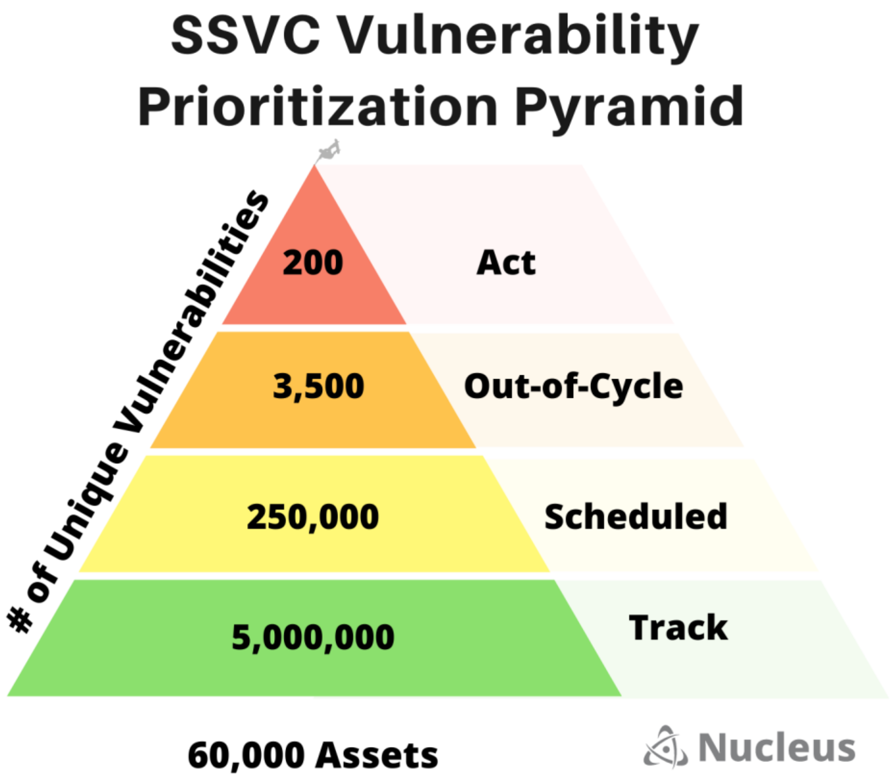
Asset Criticality Criteria

High - Asset Inventory Attribute – Criticality High **OR** Asset Metadata – Criticality High
Moderate - Asset Inventory Attribute – Criticality Medium **OR** Asset Metadata – Criticality Medium
Low - Asset Inventory Attribute – Criticality Low **OR** Asset Metadata – Criticality Low

Modeling Vulnerability Decision Tree Outcomes w/ SSVC

Once you have defined your decision criteria and outcomes, it is important to measure the vulnerabilities that categorize into each of these outcomes. This can help provide high level visibility into the scale at which your organization requires resources.

To help quantify the level of impact, we created the SSVC vulnerability prioritization pyramid which can be used to measure the number of unique vulnerabilities that map to each decision outcome. Please consider that this visual representation is a starting point and further analysis is required to better understand what technologies are impacting the work effort required to remediate these vulnerabilities and many other factors need to be considered.



This is a modification of a real-world example with normalized vulnerabilities of an enterprise environment with 60,000+ assets. The numbers represent the number of unique vulnerabilities based on the decision criteria outlined. This is only for example purposes and no assumptions should be made from the data being presented in the SSVC Vulnerability Prioritization Pyramid.

Summary

Automating vulnerability prioritization using decision trees provides a scalable and systematic approach for organizations to effectively manage vulnerabilities. By clearly defining decision criteria, visualizing decision trees, building out automation rules, and closely measuring outcomes, organizations can streamline the vulnerability prioritization process. The example provided in this paper serves as a starting point for implementing decision tree-based logic for vulnerability prioritization, incorporating the principles of Stakeholder Specific Vulnerability Categorization (SSVC) to align with organizational risk tolerance and strategic objectives.

Appendix: Examples of Vulnerability Intelligence Criteria for Consideration

Source	Exploitation Status	Availability
CISA KEV	Exploitation	Free
Google Project Zero	Exploitation	Free
Metasploit	POC	Free
ExploitDB	POC	Free
GreyNoise	Exploitation	Free/Commerical
Mandiant	Exploitation, POC	Commercial
Intel471	Exploitation, POC, Chatter	Commercial
EPSS	Exploitation Predictability	Free

Appendix: Frequently Asked Questions

How can EPSS be leveraged within the Exploitation node and beyond?
EPSS can be utilized as a consideration in determining probable active exploitation, as a high EPSS score indicates that it is likely to experience active exploitation (associated with the CVE ID) in the next 30 days. If what you are left with after conducting an SSVC exercise with your tree are large buckets of *Act* and *Out-of-Cycle* vulnerabilities, the utilization of EPSS scores to determine prevalence in exploitation can order the buckets by way of their likelihood to be exploited.

How does this approach differ from CVSS?
CVSS metrics or scoring could be used as decision criteria in an SSVC decision tree. However, we did not incorporate CVSS in our example criteria as we felt it would add to increased

complexity. We most often see decisions being made on CVSS metrics to determine the severity of a vulnerability.

Why does this guide differ in terminology and criteria from Carnegie Melon's version of SSVC?
We find the Carnegie Melon version of SSVC as the foundation to this guide but have chosen to make changes to the guide for a few reasons. First the original creation of SSVC was designed for uses cases in triaging vulnerabilities and aligned with federal use cases. Because of this, some of the terminology doesn't translate well to the commercial sectors. We also believe that the foundation to using decisions trees in SSVC is very useful and that organizations should adjust criteria based on their own unique circumstances and information they have available. This could be impacted by size, resources, visibility, access to threat intelligence and many other factors.

How can I learn more about SSVC?
Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization (Version 2.0)
https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=653459

Stakeholder-Specific Vulnerability Categorization (SSVC) | Cybersecurity and Infrastructure Security Agency
https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

What is SSVC (Stakeholder-Specific Vulnerability Categorization)? | Nucleus Security
https://www.youtube.com/watch?v=LV6PclEQ3QA

PrioritizedRiskRemediation SSVC examples| Chris Madden
https://github.com/theparanoids/PrioritizedRiskRemediation/blob/main/README.md

Flipping the Vulnerability Management Model from CVSS to SSVC | Stephen Shaffer
https://stephenshaffer.io/flipping-the-vulnerability-management-model-cvss-ssvc-aaa78f1426e1

Acknowledgements
In writing this guide, several people provided insightful feedback that helped incorporate different perspectives into this paper. This guide is not necessarily a representation of their opinion or views. A special thanks for providing feedback goes to Chris Madden, Jonathon Spring, Stephen Shaffer, Ben Edwards, Scott Kuffer, Stephen Carter, Nick Berrie, Ryan Cribelar, Rilee Smith, Jay Jacobs, and some other smart people I might have overlooked.