Prioritizing Vulnerability Response with a Stakeholder-specific Vulnerability Categorization (SSVC)

Jonathan M Spring (jspring@cert.org)

Thanks to Eric Hatleback; Allen Householder; Art Manion; Madison Oliver; Vijay Sarvapalli; Deana Shick; Laurie Tyzenhaus; Chuck Yarbrough

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Motivation

We propose a Stakeholder-Specific Vulnerability Categorization (SSVC) as an improvement

- Focus is on *decisions*, not technical severity
- Transparent, role-specific recommendations
- Experiment design to test process consistency

Thanks to co-authors, conference attendees, and GitHub contributors who have helped improve SSVC so far

- Communication between analysts and risk managers
  - Analysts know what risk manager chooses
  - Risk managers know analysts will decide on vuls consistently

Motivation

We propose a Stakeholder-Specific Vulnerability Categorization (SSVC) as an improvement

- Focus is on *decisions*, not technical severity
- Transparent, role-specific recommendations
- Experiment design to test process consistency

Thanks to co-authors, conference attendees, and GitHub contributors who have helped improve SSVC so far

- Communication between analysts and risk managers
  - Analysts know what risk manager chooses
  - Risk managers know analysts will decide on vuls consistently

SSVC contributions

1. Decision process and descriptions that could be used to make vulnerability management decisions
2. Method for how a justifiable decision process and descriptions can be constructed, adapted, and tested

(1) is valuable, and though we're on version 2, it is always improving

(2) is perhaps more important because it lets you adapt

Development and improvement are ongoing.

If you have suggestions, tell us! https://github.com/CERTCC/SSVC

SSVC contributions

1. Decision process and descriptions that could be used to make vulnerability management decisions
2. Method for how a justifiable decision process and descriptions can be constructed, adapted, and tested

(1) is valuable, and though we're on version 2, it is always improving

(2) is perhaps more important because it lets you adapt

Development and improvement are ongoing.

If you have suggestions, tell us! https://github.com/CERTCC/SSVC

SSVC contributions

1. Decision process and descriptions that could be used to make vulnerability management decisions
2. Method for how a justifiable decision process and descriptions can be constructed, adapted, and tested

(1) is valuable, and though we're on version 2, it is always improving

(2) is perhaps more important because it lets you adapt

Development and improvement are ongoing.

If you have suggestions, tell us! https://github.com/CERTCC/SSVC

SSVC priorities

An SSVC decision is the priority of action on a work item

Propose 4 levels of priority for most stakeholders:

- Defer (lowest)
- Scheduled
- Out-of-cycle
- Immediate (highest)

We do not tell you what to do, just what vuls are most important for you to act on.

- These vuls carry the highest risk to you if you do nothing
- Even so, you might decide to accept the risk of the vul for systems that are too expensive to change or patch
  - E.g., EOL, regulated software, or high downtime costs

SSVC priorities

An SSVC decision is the priority of action on a work item

Propose 4 levels of priority for most stakeholders:

- Defer (lowest)
- Scheduled
- Out-of-cycle
- Immediate (highest)

We do not tell you what to do, just what vuls are most important for you to act on.

- These vuls carry the highest risk to you if you do nothing
- Even so, you might decide to accept the risk of the vul for systems that are too expensive to change or patch
  - E.g., EOL, regulated software, or high downtime costs

SSVC priorities

An SSVC decision is the priority of action on a work item

Propose 4 levels of priority for most stakeholders:

- Defer (lowest)
- Scheduled
- Out-of-cycle
- Immediate (highest)

We do not tell you what to do, just what vuls are most important for you to act on.

- These vuls carry the highest risk to you if you do nothing
- Even so, you might decide to accept the risk of the vul for systems that are too expensive to change or patch
  - E.g., EOL, regulated software, or high downtime costs

SSVC roles

Propose different decision-making for different roles:

- Supplier
- Deployer
- Coordinator

These track the roles in coordinated vulnerability disclosure

This contrasts with CVSS, which is often used as a one-size-fits-all

For each role, a decision is the priority of action on a work item

- A work item is may be to develop or deploy a patch, decommission the vulnerable system, develop or deploy remediations that reduce threat, or accept the risk. Need not be tied to a CVE ID.

SSVC roles

Propose different decision-making for different roles:

- Supplier
- Deployer
- Coordinator

These track the roles in coordinated vulnerability disclosure

This contrasts with CVSS, which is often used as a one-size-fits-all

For each role, a decision is the priority of action on a work item

- A work item is may be to develop or deploy a patch, decommission the vulnerable system, develop or deploy remediations that reduce threat, or accept the risk. Need not be tied to a CVE ID.

SSVC roles

Propose different decision-making for different roles:

- Supplier
- Deployer
- Coordinator

These track the roles in coordinated vulnerability disclosure

This contrasts with CVSS, which is often used as a one-size-fits-all

For each role, a decision is the priority of action on a work item

- A work item is may be to develop or deploy a patch, decommission the vulnerable system, develop or deploy remediations that reduce threat, or accept the risk. Need not be tied to a CVE ID.

Credible cases, NOT worst cases

SSVC includes evaluating the impact on safety, well-being, and the mission of the deployer organization

- So there are explicit criteria for different situations

For example, for physical well-being (AKA, safety):

Minor / none	Physical discomfort for users of the system.
Major	Physical distress and injuries for users of the system.
Hazardous	Serious or fatal injuries, where fatalities are plausibly preventable via emergency services or other measures.
Catastrophic	Multiple immediate fatalities. Emergency response probably cannot save the victims.

Credible cases

SSVC includes evaluating the impact on safety, well-being, and the mission of the deployer organization

- So there are explicit criteria for different situations

For example, for physical well-being (AKA, safety):

Minor / none	Physical discomfort for users of the system.
Major	Physical distress and injuries for users of the system.
Hazardous	Serious or fatal injuries, where fatalities are plausibly preventable via emergency services or other measures.
Catastrophic	Multiple immediate fatalities. Emergency response probably cannot save the victims.

Credible cases

SSVC includes evaluating the impact on safety, well-being, and the mission of the deployer organization

- So there are explicit criteria for different situations

For example, for physical well-being (AKA, safety):

Minor / none	Physical discomfort for users of the system.
Major	Physical distress and injuries for users of the system.
Hazardous	Serious or fatal injuries, where fatalities are plausibly preventable via emergency services or other measures.
Catastrophic	Multiple immediate fatalities. Emergency response probably cannot save the victims.

Example danger of “worst possible” thinking

“Pain Rating” CC 2.5 BY-NC Randal Monroe. https://xkcd.com/883/

Reaching a priority decision

Every decision is the result of a logical combination of 4-6 simple decision points.

For example, leadership might decide they want:

Exploitation	==	Proof of Concept	AND
Utility	==	Laborious	AND
Technical Impact	==	Total	AND
Public Safety Impact	==	Minor / None	THEN
Scheduled			

Reaching a priority decision

Every decision is the result of a logical combination of 4-6 simple decision points.

For example, leadership might decide they want:

Exploitation	==	Proof of Concept	AND
Utility	==	Laborious	AND
Technical Impact	==	Total	AND
Public Safety Impact	==	Minor / None	THEN
Scheduled			

Reaching a priority decision

Every decision is the result of a logical combination of 4-6 simple decision points.

For example, leadership might decide they want:

Exploitation	==	Proof of Concept	AND
Utility	==	Laborious	AND
Technical Impact	==	Total	AND
Public Safety Impact	==	Minor / None	THEN
Scheduled			

Reaching a priority decision

Every decision is the result of a logical combination of 4-6 simple decision points.

For example, leadership might decide they want:

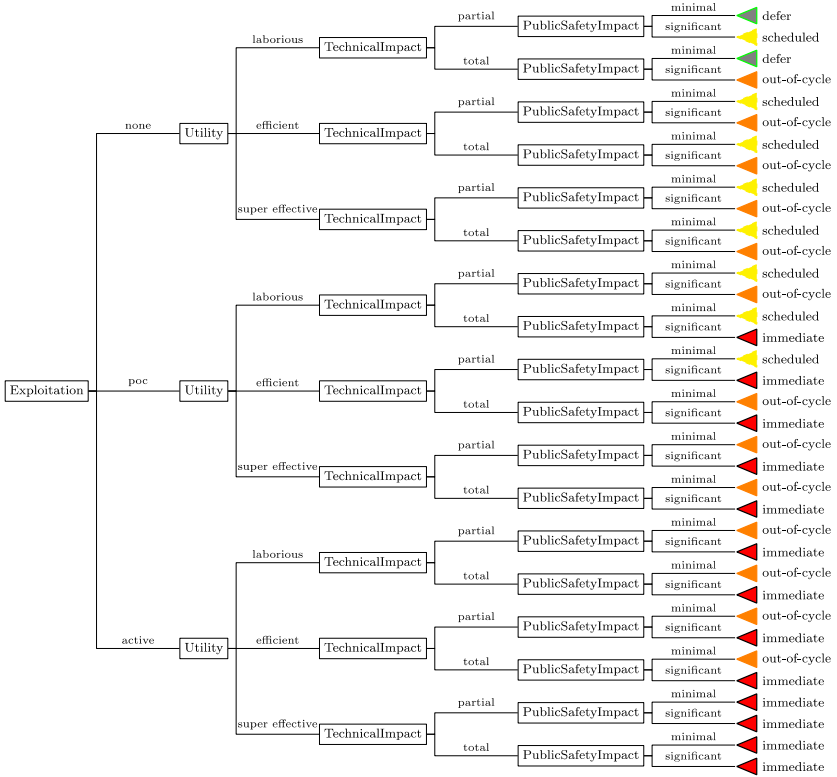
Exploitation	==	Proof of Concept	AND
Utility	==	Laborious	AND
Technical Impact	==	Total	AND
Public Safety Impact	==	Minor / None	THEN
Scheduled			

Organizing decisions into trees

For Risk Managers, we visually compress these logical statements into a tree.

For example, the suggested supplier tree for version 2:

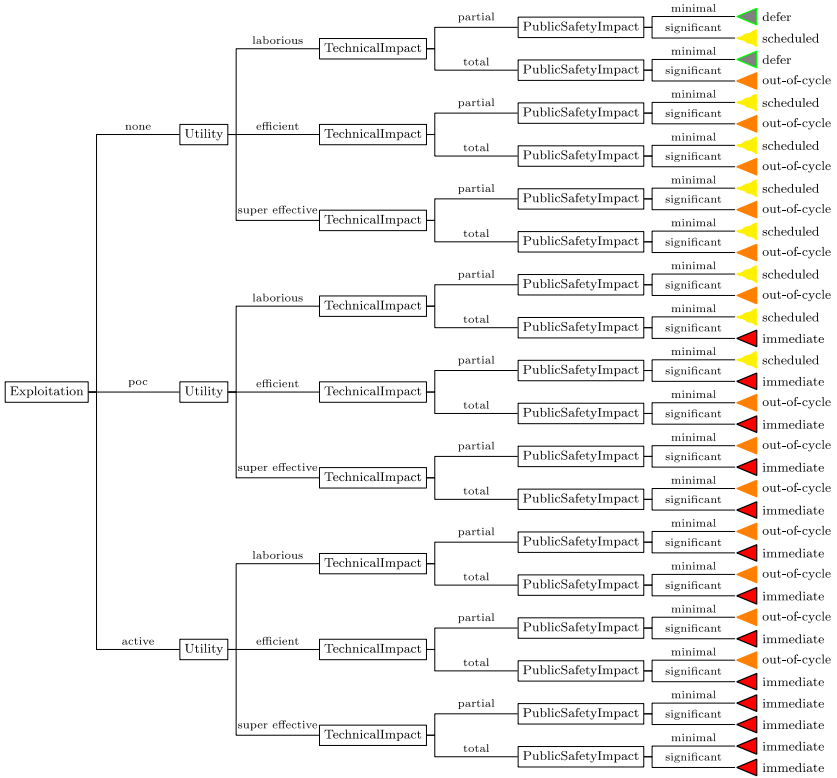
Organizing decisions into trees



For Risk Managers, we visually compress these logical statements into a tree.

For example, the suggested supplier tree for version 2

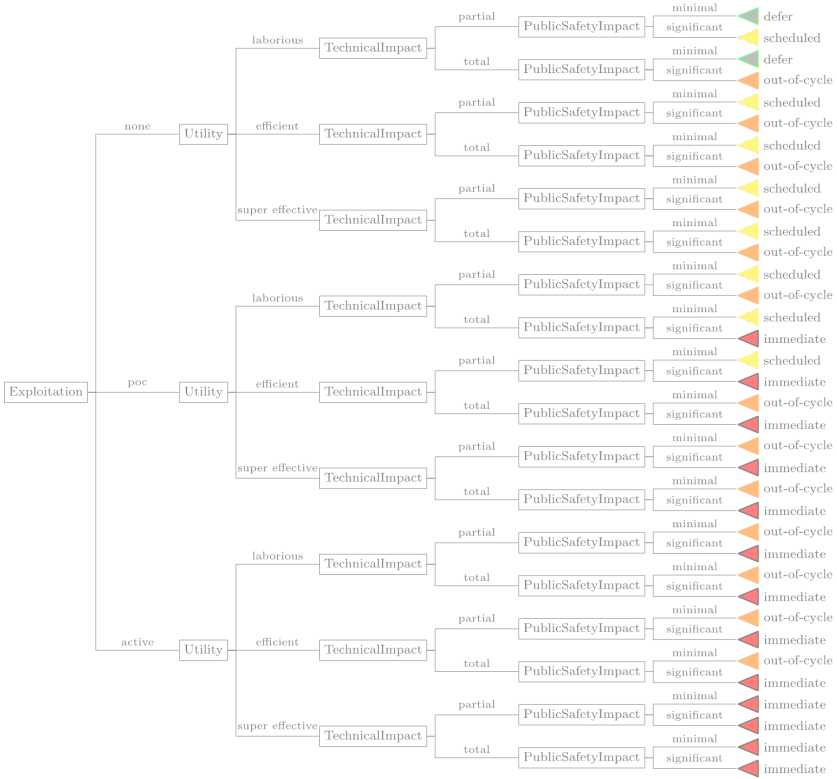
Trees as risk posture



This captures how leadership wants the org to act in all combinations of situations

An analyst's job is to answer a few questions to determine the situation each vul is in

Trees as risk posture



This captures how leadership wants the org to act in all combinations of situations

An analyst's job is to answer a few questions to determine the situation each vul is in

Decision points

The different roles can use different decision points (that is, they use different information)

Supplier

- Exploitation
- Utility
- Technical Impact
- Public Safety Impact

Deployer

- Exploitation
- Utility
- Exposure
- Mission Impact
- Situated Safety Impact

Decision points

The different roles can use different decision points (that is, they use different information)

Supplier

- Exploitation
- Utility
- Technical Impact
- Public Safety Impact

Deployer

- Exploitation
- Utility
- Exposure
- Mission Impact
- Situated Safety Impact

Decision points

The different roles can use different decision points (that is, they use different information)

Supplier

- Exploitation
- Utility
- Technical Impact
- Public Safety Impact

Deployer

- Exploitation
- Utility
- Exposure
- Mission Impact
- Situated Safety Impact

Shared Decision Points

Common decision points are things that make sense to put in vul reports and communications

Supplier

- Exploitation
- Utility
- Technical Impact
- Public Safety Impact

Deployer

- Exploitation
- Utility
- Exposure
- Mission Impact
- Situated Safety Impact

Unique Decision Points

Unique decision points are things that an organization may need to figure out for itself

Supplier

- Exploitation
- Utility
- Technical Impact
- Public Safety Impact

Deployer

- Exploitation
- Utility
- Exposure
- Mission Impact
- Situated Safety Impact

Related Decision Points

Related decision points are things that one stakeholder knows more about than others

Supplier

- Exploitation
- Utility
- Technical Impact
- Public Safety Impact

Deployer

- Exploitation
- Utility
- Exposure
- Mission Impact
- Situated Safety Impact

If you'd like to explore this more at your own speed:

https://democert.org/ssvc/

Example decision point description

Exploitation (Evidence of Exploitation of a Vulnerability)

Measures the present state of exploitation of the vulnerability. Our intent is not to predict future exploitation but only to acknowledge the current state of affairs. Predictive systems, such as EPSS, could be used to augment this decision or to notify stakeholders of likely changes

None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vulnerability.
PoC (Public Proof of Concept)	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. Some examples of condition (3) are open-source web proxies serve as the PoC code for how to exploit any vulnerability in the vein of improper validation of TLS certificates. As another example, Wireshark serves as a PoC for packet re-play attacks on ethernet or WiFi networks.
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

Example decision point description

Exploitation (Evidence of Exploitation of a Vulnerability)

Measures the present state of exploitation of the vulnerability. Our intent is not to predict future exploitation but only to acknowledge the current state of affairs. Predictive systems, such as EPSS, could be used to augment this decision or to notify stakeholders of likely changes

None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vulnerability.
PoC (Public Proof of Concept)	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. Some examples of condition (3) are open-source web proxies serve as the PoC code for how to exploit any vulnerability in the vein of improper validation of TLS certificates. As another example, Wireshark serves as a PoC for packet re-play attacks on ethernet or WiFi networks.
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

Example decision point description

Exploitation (Evidence of Exploitation of a Vulnerability)

Measures the present state of exploitation of the vulnerability. Our intent is not to predict future exploitation but only to acknowledge the current state of affairs. Predictive systems, such as EPSS, could be used to augment this decision or to notify stakeholders of likely changes

None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vulnerability.
PoC (Public Proof of Concept)	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. Some examples of condition (3) are open-source web proxies serve as the PoC code for how to exploit any vulnerability in the vein of improper validation of TLS certificates. As another example, Wireshark serves as a PoC for packet re-play attacks on ethernet or WiFi networks.
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

Example decision point description

Exploitation (Evidence of Exploitation of a Vulnerability)

Measures the present state of exploitation of the vulnerability. Our intent is not to predict future exploitation but only to acknowledge the current state of affairs. Predictive systems, such as EPSS, could be used to augment this decision or to notify stakeholders of likely changes

None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vulnerability.
PoC (Public Proof of Concept)	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. Some examples of condition (3) are open-source web proxies serve as the PoC code for how to exploit any vulnerability in the vein of improper validation of TLS certificates. As another example, Wireshark serves as a PoC for packet re-play attacks on ethernet or WiFi networks.
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

Example decision point description

Exploitation (Evidence of Exploitation of a Vulnerability)

Measures the present state of exploitation of the vulnerability. Our intent is not to predict future exploitation but only to acknowledge the current state of affairs. Predictive systems, such as EPSS, could be used to augment this decision or to notify stakeholders of likely changes

None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vulnerability.
PoC (Public Proof of Concept)	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. Some examples of condition (3) are open-source web proxies serve as the PoC code for how to exploit any vulnerability in the vein of improper validation of TLS certificates. As another example, Wireshark serves as a PoC for packet re-play attacks on ethernet or WiFi networks.
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

Example decision point description

Exploitation (Evidence of Exploitation of a Vulnerability)

Measures the present state of exploitation of the vulnerability. Our intent is not to predict future exploitation but only to acknowledge the current state of affairs. Predictive systems, such as EPSS, could be used to augment this decision or to notify stakeholders of likely changes

None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vulnerability.
PoC (Public Proof of Concept)	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. Some examples of condition (3) are open-source web proxies serve as the PoC code for how to exploit any vulnerability in the vein of improper validation of TLS certificates. As another example, Wireshark serves as a PoC for packet re-play attacks on ethernet or WiFi networks.
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

Data Management

A decision tree is easy to automate, given the data inputs (even if this looks scary)

We assume humans will evaluate the decision points.

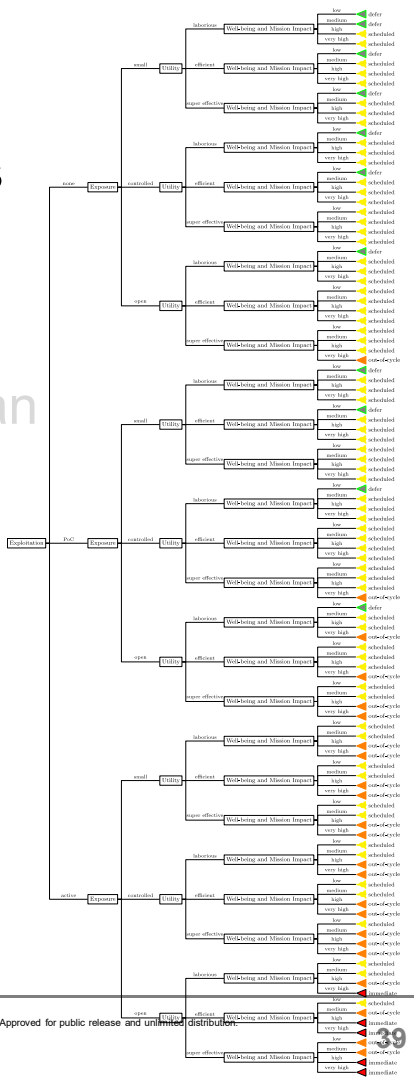
- SSVC is designed to be modular and reduce this human effort.

Many decisions are reusable, for example:

- Exposure applies to the system, not the vulnerability
- Tree customization and risk posture set up

Some are actually automatically collectable

- Exploitation = PoC, such as in Historical Analysis of Exploit Availability Timelines



Data Management

A decision tree is easy to automate, given the data inputs (even if this looks scary)

We assume humans will evaluate the decision points.

- SSVC is designed to be modular and reduce this human effort.

Many decisions are reusable, for example:

- Exposure applies to the system, not the vulnerability
- Tree customization and risk posture set up

Some are actually automatically collectable

- Exploitation = PoC, such as in Historical Analysis of Exploit Availability Timelines

Data Management

A decision tree is easy to automate, given the data inputs (even if this looks scary)

We assume humans will evaluate the decision points.

- SSVC is designed to be modular and reduce this human effort.

Many decisions are reusable, for example:

- Exposure applies to the system, not the vulnerability
- Tree customization and risk posture set up

Some are actually automatically collectable

- Exploitation = PoC, such as in Historical Analysis of Exploit Availability Timelines

Data Management

A decision tree is easy to automate, given the data inputs (even if this looks scary)

We assume humans will evaluate the decision points.

- SSVC is designed to be modular and reduce this human effort.

Many decisions are reusable, for example:

- Exposure applies to the system, not the vulnerability
- Tree customization and risk posture set up

Some are actually automatically collectable

- Exploitation = PoC, such as in Historical Analysis of Exploit Availability Timelines

Scoring example

CVE-2020-6967

In Rockwell Automation all versions of FactoryTalk Diagnostics software, a subsystem of the FactoryTalk Services Platform, FactoryTalk Diagnostics exposes a .NET Remoting endpoint via RNADiagnosticsSrv.exe at TCPtcp/8082, which can insecurely deserialize untrusted data.

CVSS v3.1 base score: 9.8

https://nvd.nist.gov/vuln/detail/CVE-2020-6967

https://us-cert.cisa.gov/ics/advisories/icsa-20-051-02

SSVC for CVE-2020-6967

Technical Impact	Description
Partial	The exploit gives the adversary *limited* control over, or information exposure about, the behavior of the software that contains the vulnerability. Or the exploit gives the adversary an importantly low stochastic opportunity for total control. …
Total	The exploit gives the adversary *total* control over the behavior of the software, or it gives total disclosure of all information on the system that contains the vulnerability

CVE-2020-6967 suggested response

Technical Impact	Description
Partial	The exploit gives the adversary *limited* control over, or information exposure about, the behavior of the software that contains the vulnerability. Or the exploit gives the adversary an importantly low stochastic opportunity for total control. …
Total	The exploit gives the adversary *total* control over the behavior of the software, or it gives total disclosure of all information on the system that contains the vulnerability

SSVC for CVE-2020-6967

State of Exploitation	Description
None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vul.
Proof of Concept	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. …
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

CVE-2020-6967 suggested response

State of Exploitation	Description
None	There is no evidence of active exploitation and no public proof of concept (PoC) of how to exploit the vul.
Proof of Concept	One of the following cases is true: (1) exploit code sold or traded on underground or restricted fora; (2) typical public PoC in places such as Metasploit or ExploitDB; or (3) the vulnerability has a well-known method of exploitation. …
Active	Shared, observable, reliable evidence that the exploit is being used in the wild by real attackers; there is credible public reporting.

This value can change over time.

Updates can be at least partially automated

- https://github.com/CERTCC/git_vul_driller

SSVC for CVE-2020-6967

Utility	Description
Laborious	No to *automatable* and diffuse *value density*
Efficient	{Yes to *automatable* and diffuse *value density*} OR {No to *automatable* and concentrated *value density*}
Super Effective	Yes to *automatable* and concentrated *value density*

Utility has two sub-parts

SSVC for CVE-2020-6967 (Utility part 1)

Automatable	Description
No	Steps 1-4 of the kill chain (Hutchins et al. 2011) cannot be reliably automated by the attacker for this vulnerability. These steps are reconnaissance, weaponization, delivery, and exploitation. Example [barriers] include (1) the vulnerable component is not searchable…, (2) weaponization requires human direction, (3) delivery over channels [widely blocked], and (4) exploitation frustrated by, e.g., ASLR.
Yes	The attacker can reliably automate steps 1-4 of the of the kill chain.

CVE-2020-6967 (Utility part 1) suggested

Automatable	Description
No	Steps 1-4 of the kill chain (Hutchins et al. 2011) cannot be reliably automated by the attacker for this vulnerability. These steps are reconnaissance, weaponization, delivery, and exploitation. Example [barriers] include (1) the vulnerable component is not searchable…, (2) weaponization requires human direction, (3) delivery over channels [widely blocked], and (4) exploitation frustrated by, e.g., ASLR.
Yes	The attacker can reliably automate steps 1-4 of the of the kill chain.

SSVC for CVE-2020-6967 (Utility part 2)

Value Density	Description
Diffuse	The system that contains the vulnerable component has limited resources. That is, the resources that the adversary will gain control over with a single exploitation event are relatively small.
Concentrated	The system that contains the vulnerable component is rich in resources. … Examples of concentrated value are database systems, Kerberos servers, web servers hosting login pages, and cloud service providers. However, usefulness and uniqueness of the resources on the vulnerable system also inform value density.

CVE-2020-6967 (Utility part 2) suggested

Value Density	Description
Diffuse	The system that contains the vulnerable component has limited resources. That is, the resources that the adversary will gain control over with a single exploitation event are relatively small.
Concentrated	The system that contains the vulnerable component is rich in resources. … Examples of concentrated value are database systems, Kerberos servers, web servers hosting login pages, and cloud service providers. However, usefulness and uniqueness of the resources on the vulnerable system also inform value density.

CVE-2020-6967 suggested response

Utility	Description
Laborious	No to *automatable* and diffuse *value density*
Efficient	{Yes to *automatable* and diffuse *value density*} OR {No to *automatable* and concentrated *value density*}
Super Effective	Yes to *automatable* and concentrated *value density*

SSVC for CVE-2020-6967

System Exposure	Description
Small	Local service or program; highly controlled network
Controlled	Networked service with some access restrictions or mitigations already in place (whether locally or on the network). A successful mitigation must reliably interrupt the adversary's attack, which requires the attack is detectable both reliably and quickly enough to respond. *Controlled* covers the situation in which a vulnerability can be exploited through chaining it with other vulnerabilities.
Open	Service on the Internet or another widely accessible network.

CVE-2020-6967 suggested response

System Exposure	Description
Small	Local service or program; highly controlled network
Controlled	Networked service with some access restrictions or mitigations already in place (whether locally or on the network). A successful mitigation must reliably interrupt the adversary's attack, which requires the attack is detectable both reliably and quickly enough to respond. *Controlled* covers the situation in which a vulnerability can be exploited through chaining it with other vulnerabilities.
Open	Service on the Internet or another widely accessible network.

Sometimes, a response is going to be deployment-specific, so we can only constrain the response – to know it we need a specific deployment context.

SSVC for CVE-2020-6967

Situated Safety Impact	Description
Minor / None	Each of these has a meaning in several areas of well-being: physical harm, environmental, financial, and psychological.
Major	
Hazardous	
Catastrophic	

CVE-2020-6967 suggested response

Situated Safety Impact	Description
Minor / None	I’m not an expert in how this diagnostic system is used, but as best I can tell it’s not going to kill anyone, so I think it’s one of these two options?
Major	
Hazardous	
Catastrophic	

SSVC for CVE-2020-6967

Mission Impact	Description
Non-essential degraded	Organization functions harmed, but can sustain a while
MEF support crippled	Organization functions harmed, can sustain briefly
MEF Failure	Any one MEF fails, but the whole mission can sustain
Mission Failure	Multiple or all MEFs fail leading to whole org failure

A **mission essential function (MEF)** is a function "directly related to accomplishing the organization's mission as set forth in its statutory or executive charter"

CVE-2020-6967 suggested response

Mission Impact	Description
Non-essential degraded	Organization functions harmed, but can sustain a while
MEF support crippled	Organization functions harmed, can sustain briefly
MEF Failure	Any one MEF fails, but the whole mission can sustain
Mission Failure	Multiple or all MEFs fail leading to whole org failure

This is highly context dependent, but usually a plant or facility operating is probably a MEF for the organization, and the affected systems support the facility. Right?

That was easy!

- Technical Impact: Partial
- Evidence of Exploitation: None
- Utility: Efficient
- System Exposure: Small/controlled
- Safety Impact: Minor/Major
- Mission Impact: MEF support crippled

Would lead to a recommended (version 2) priority of:

- (Deployer) defer
  - Deployer should act once there is a PoC
- (Supplier) scheduled

That was easy!

- Technical Impact: Partial
- Evidence of Exploitation: None
- Utility: Efficient
- System Exposure: Small/controlled
- Safety Impact: Minor/Major
- Mission Impact: MEF support crippled

Would lead to a recommended (version 2) priority of:

- (Deployer) defer
  - Deployer should act once there is a PoC
- (Supplier) scheduled

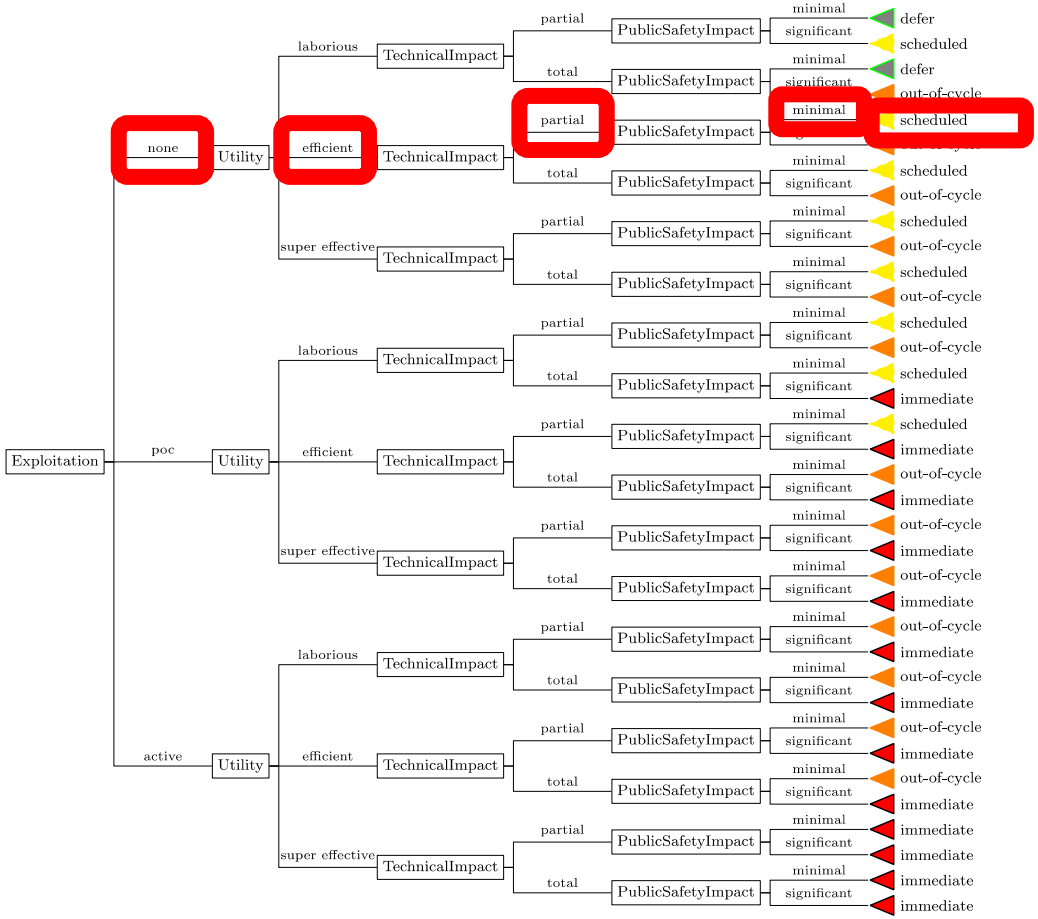
That was easy!

- Technical Impact: Partial
- Evidence of Exploitation: None
- Utility: Efficient
- System Exposure: Small/controlled
- Safety Impact: Minor/Major
- Mission Impact: MEF support crippled

Would lead to a recommended (version 2) priority of:

- (Deployer) defer
  - Deployer should act once there is a PoC
- (Supplier) scheduled

Example Supplier Decision: CVE-2020-6967



We want your input!

Version 2 has ingested the community input on v1

- Available here:
  - https://resources.sei.cmu.edu/asset_files/WhitePaper/2021_019_001_653461.pdf

We welcome public comments and suggestions:

- https://github.com/CERTCC/SSVC

Our next task is an inter-rater agreement study with more diverse participants

- If you'd like to volunteer, please email me (jspring AT cert org)

We want your input!

Version 2 has ingested the community input on v1

- Available here:

We welcome public comments and suggestions:

- https://github.com/CERTCC/SSVC

Our next task is an inter-rater agreement study with more diverse participants

- If you'd like to volunteer, please email me (jspring AT cert org)

We want your input!

Version 2 has ingested the community input on v1

- Available here:

We welcome public comments and suggestions:

- https://github.com/CERTCC/SSVC

Our next task is an inter-rater agreement study with more diverse participants

- If you'd like to volunteer, please email me (jspring AT cert org)

Summary

1. SSVC structures evidence-based decisions about the priority of vulnerabilities.
2. Decision trees document and structure how to combine responses to decision points (such as exposure, exploitation, or utility) to reach priority decisions.
3. Decisions are qualitative and remain qualitative in order to be transparent, reliable, and explainable.
4. SSVC supports multiple stakeholder groups and we encourage people to tailor the details (with appropriate testing) to meet their needs or situation.
5. SSVC provides recommended decision trees as starting points.

Open Questions? Want to chat?

Come to the SSVC BoF later today

17:00-18:00

Liffey Meeting Room 5

Thanks! Questions?

The SSVC paper is available:

https://github.com/CERTCC/SSVC/tree/main/doc

https://resources.sei.cmu.edu/asset_files/WhitePaper/2021_019_001_653461.pdf

Searching for “SSVC prioritizing” should work, too

jspring AT cert org