System Security Context Vector (SSCV) Framework

Version 1.0 Specification

Executive Summary

The System Security Context Vector (SSCV) is a standardized method for describing the security posture and context of computing systems. When combined with CVSS (Common Vulnerability Scoring System) scores, SSCV enables organizations to calculate contextual risk scores that reflect the actual risk a vulnerability poses to a specific system, while ensuring that no vulnerability is ever reduced to zero risk.

SSCV v1.0 focuses on six core components that every security team can readily assess, making adoption straightforward and practical.

1. Introduction

1.1 Purpose

While CVSS provides a standardized way to assess vulnerability severity in isolation, it lacks context about the target system. A CVSS 10.0 vulnerability may pose reduced risk to an air-gapped, fully patched system with no sensitive data, while a CVSS 6.0 vulnerability could be critical on an internet-facing, legacy system processing payment data.

SSCV bridges this gap by providing a compact, standardized notation for system security context while maintaining the principle that no system is ever completely risk-free.

1.2 Vector Format

SSCV uses a format similar to CVSS for familiarity and ease of parsing:

```
SSCV:1.0/OS:C/NE:I/AC:F/EP:A/DL:M/PS:C
```

1.3 Design Philosophy

SSCV v1.0 intentionally focuses on six fundamental security components that: - Are universally applicable to IT systems - Can be assessed without specialized tools - Provide meaningful risk differentiation - Support automated collection where possible

Future versions will add components for specialized use cases and mature security programs.

2. SSCV Components

2.1 OS - Operating System Currency

Measures how current and supported the operating system is.

Value	Name	Description	Risk Weight
L	Legacy	EOL/Unsupported OS (e.g., Windows XP, CentOS 6)	3.0
O	Old	Supported but 2+ major versions behind	2.0
C	Current	Latest stable version or one version behind	1.0
H	Hardened	Security-focused variant (e.g., SELinux enforcing, Windows Server Core)	0.8
X	Not Evaluated	This component was not included in the assessment	N/A

Note: Even hardened systems retain 0.8 weight as hardening reduces but cannot eliminate all OS-level vulnerabilities.

2.2 NE - Network Exposure

Describes the system's network accessibility.

Value	Name	Description	Risk Weight
A	Air-gapped	No network connectivity	0.3
I	Internal	Internal network only, no direct internet routes	1.0
P	Perimeter	DMZ or controlled external access (behind WAF/proxy)	2.0
E	External	Direct internet exposure	3.0
X	Not Evaluated	This component was not included in the assessment	N/A

Note: Air-gapped systems retain 0.3 weight because air gaps can be bridged through physical access, supply chain attacks, or insider threats.

2.3 AC - Access Control

Indicates the authentication and authorization mechanisms.

Value	Name	Description	Risk Weight
N	None	No authentication required	3.0
B	Basic	Single-factor authentication	2.0
F	Full	MFA + role-based access control	1.0
Z	Zero-trust	Continuous verification, micro-segmentation	0.7
X	Not Evaluated	This component was not included in the assessment	N/A

Note: Zero-trust architectures retain 0.7 weight as they reduce but cannot eliminate risks from sophisticated attacks or insider threats.

2.4 EP - Endpoint Protection

Describes the level of endpoint security tooling.

Value	Name	Description	Risk Weight
N	None	No AV/EDR solution	3.0
B	Basic	Traditional signature-based AV	2.0
A	Advanced	Modern EDR/XDR with behavioral analysis	1.0
M	Managed	24/7 SOC monitoring with active threat hunting	0.8
X	Not Evaluated	This component was not included in the assessment	N/A

Note: Even 24/7 managed protection retains 0.8 weight due to dwell time and zero-day vulnerabilities.

2.5 DL - Data Sensitivity Level

Classifies the sensitivity of data processed/stored.

Value	Name	Description	Risk Weight
P	Public	No sensitive data	0.6
I	Internal	Business confidential	1.5

Value	Name	Description	Risk Weight
M	Mixed	Contains PII/PHI/PCI data	2.5
C	Critical	State secrets, critical infrastructure, financial systems	3.0
X	Not Evaluated	This component was not included in the assessment	N/A

Note: Public data retains 0.6 weight because integrity and availability concerns still apply.

2.6 PS - Patch Status

Reflects how current the system's patches are.

Value	Name	Description	Risk Weight
C	Current	Fully patched (< 30 days)	1.0
D	Delayed	30-90 days behind	1.5
B	Behind	> 90 days behind	2.5
U	Unknown	Patch status unclear	3.0
X	Not Evaluated	This component was not included in the assessment	N/A

3. Calculating Contextual Risk Score

3.1 Core Formula

The Contextual Risk Score (CRS) uses a streamlined approach that ensures meaningful minimum risk levels:

```
Step 1: Calculate base components
- Context_Multiplier = Average of (OS + AC + EP + PS)
- Exposure_Factor = (NE × DL) / 8
 (Division by 8 balances exposure impact while maintaining differentiation)

Step 2: Calculate adjusted score
- Adjusted_Score = CVSS_Base × Context_Multiplier × Exposure_Factor

Step 3: Apply minimum threshold
- Minimum_Score = CVSS_Base × 0.2
  (Ensures no system is considered risk-free)
```

```
Step 4: Calculate final CRS
- If Context_Multiplier × Exposure_Factor ≤ 1:
    CRS = max(Adjusted_Score, Minimum_Score)
- If Context_Multiplier × Exposure_Factor > 1:
    Raw_Score = CVSS_Base × Context_Multiplier × Exposure_Factor
    CRS = min(Raw_Score, CVSS_Base × 2)
    (Poor security amplifies risk, but capped at 2× original)
```

Important Notes:

- The division by 8 for exposure factor creates a range from ~0.04 to 1.125, allowing good security to reduce risk while high exposure still amplifies it
- The 0.2 minimum (20%) is the enforced floor - no vulnerability can be reduced below 20% of its CVSS score*
- Maximum amplification is capped at 2× the original CVSS score
- Components marked as 'X' (Not Evaluated) are excluded from calculations

*Organizations may use different minimum thresholds but must mark scores with ~ suffix

3.2 Rationale for Minimum Threshold

The framework enforces a minimum risk threshold (default 20%) because:

- **No Perfect Security**: Even the most secure systems can be compromised through zero-days, supply chain attacks, or insider threats
- **Defense in Depth**: Security measures reduce but never eliminate risk
- **Unknown Unknowns**: New attack vectors constantly emerge

- **Historical Data**: Shows that even "secure" systems still get breached

This minimum ensures organizations maintain appropriate vigilance even for well-protected systems.

3.3 Modified Threshold Notation

When organizations choose to use a minimum threshold other than the standard 20%, the resulting CRS must be marked with a tilde (~) suffix:

- **Standard calculation (20% minimum)**: `5.2`
- **Modified threshold**: `5.2~`

This notation ensures transparency when scores are shared. The SSCV vector string remains unchanged - only the final CRS score receives the tilde suffix. Organizations using modified thresholds should document their chosen percentage for audit and compliance purposes.

Examples:

- `CRS: 3.4` - Standard SSCV calculation with 20% minimum

- `CRS: 3.4~` - Modified calculation (could be 0%, 30%, or any non-standard threshold)
- `CRS: 0.8~` - Likely using 0% threshold (raw calculation)

3.4 Severity Mapping

CRS Range	Severity	Action Required
0.0 - 3.9	Low	Patch during normal maintenance
4.0 - 6.9	Medium	Patch within 30 days
7.0 - 8.9	High	Patch within 7 days
9.0 - 10.0	Critical	Immediate action required

4. Examples

Example 1: Internet-Facing Web Server

System: Public e-commerce web server
SSCV: `SSCV:1.0/OS:C/NE:E/AC:F/EP:A/DL:M/PS:C`
Vulnerability: Apache Struts RCE (CVE-2017-5638)
CVSS: 10.0

Calculation:

- Context_Multiplier = (1.0 + 1.0 + 1.0 + 1.0) / 4 = 1.0
- Exposure_Factor = (3.0 × 2.5) / 8 = 0.9375
- Combined Factor = 1.0 × 0.9375 = 0.9375
- Since combined factor ≤ 1:
  - Adjusted_Score = 10.0 × 0.9375 = 9.375
  - Minimum_Score = 10.0 × 0.2 = 2.0
  - CRS = max(9.375, 2.0) = **9.375 (Critical)**

Note: Despite good security controls, the high exposure (internet-facing with sensitive data) keeps this as a critical priority, just slightly reduced from the original CVSS 10.0.

Example 2: Air-Gapped Industrial System

System: Air-gapped industrial control system
SSCV: `SSCV:1.0/OS:L/NE:A/AC:N/EP:N/DL:I/PS:U`
Vulnerability: Buffer overflow requiring local access
CVSS: 7.0

Calculation:

- Context_Multiplier = (3.0 + 3.0 + 3.0 + 3.0) / 4 = 3.0
- Exposure_Factor = (0.3 × 1.5) / 8 = 0.05625
- Combined Factor = 3.0 × 0.05625 = 0.169
- Since combined factor ≤ 1:

  - Adjusted_Score = 7.0 × 0.169 = 1.183
  - Minimum_Score = 7.0 × 0.2 = 1.4
  - CRS = max(1.183, 1.4) = **1.4 (Low)**

Note: Even with air-gapping, the minimum threshold ensures meaningful risk. If an organization used 0% threshold instead of the standard 20%, the score would be **1.183~** indicating a modified calculation.

Example 3: Well-Secured Database Server

System: Internal database with strong security
SSCV: `SSCV:1.0/OS:H/NE:I/AC:Z/EP:M/DL:C/PS:C`
Vulnerability: Privilege escalation (CVE-2024-xxxxx)
CVSS: 7.8

Calculation:

- Context_Multiplier = (0.8 + 0.7 + 0.8 + 1.0) / 4 = 0.825
- Exposure_Factor = (1.0 × 3.0) / 8 = 0.375
- Combined Factor = 0.825 × 0.375 = 0.309
- Since combined factor ≤ 1:
  - Adjusted_Score = 7.8 × 0.309 = 2.41
  - Minimum_Score = 7.8 × 0.2 = 1.56
  - CRS = max(2.41, 1.56) = **2.41 (Low)**

Example 4: Poorly Secured Public Server

System: Legacy public-facing server
SSCV: `SSCV:1.0/OS:L/NE:E/AC:B/EP:B/DL:M/PS:B`
Vulnerability: Remote code execution
CVSS: 8.8

Calculation:

- Context_Multiplier = (3.0 + 2.0 + 2.0 + 2.5) / 4 = 2.375
- Exposure_Factor = (3.0 × 2.5) / 8 = 0.9375
- Combined Factor = 2.375 × 0.9375 = 2.227
- Since combined factor > 1 (poor security amplifies risk):
  - Raw_Score = 8.8 × 2.227 = 19.6
  - CRS = min(19.6, 8.8 × 2) = **17.6** → Capped at **10.0 (Critical)**

Example 5: Typical Corporate Workstation

System: Developer workstation
SSCV: `SSCV:1.0/OS:C/NE:E/AC:F/EP:A/DL:I/PS:D`
Vulnerability: Browser vulnerability
CVSS: 6.5

Calculation:

- Context_Multiplier = (1.0 + 1.0 + 1.0 + 1.5) / 4 = 1.125
- Exposure_Factor = (3.0 × 1.5) / 8 = 0.5625
- Combined Factor = 1.125 × 0.5625 = 0.633
- Since combined factor ≤ 1:
  - Adjusted_Score = 6.5 × 0.633 = 4.11
  - Minimum_Score = 6.5 × 0.2 = 1.3
  - CRS = max(4.11, 1.3) = **4.11 (Medium)**

5. Implementation Guidelines

5.1 Getting Started

1. **Assess What You Know**: Start with systems where you have clear visibility
2. **Use Automation**: Leverage agents for OS, EP, and PS detection
3. **Document Business Context**: Manually classify NE and DL for each system
4. **Iterate Weekly**: Update patch status as systems change
5. **Consider Thresholds**: Default to 20% minimum; document if using different threshold with ~ suffix

5.2 Data Collection Methods

Automated (via agents):

- OS: Version detection, EOL database comparison
- EP: Running process detection for AV/EDR
- PS: Last update timestamp, pending patches

Manual/Configuration Management:

- NE: Network topology classification
- DL: Data classification per compliance requirements
- AC: Authentication mechanism audit

5.3 When to Override CRS

- If CRS differs from CVSS severity by 2+ levels, use CRS
- If CRS and CVSS differ by 1 level, consider:
  - Active exploitation in the wild
  - Availability of exploit code
  - Specific threat intelligence
  - Compensating controls not captured in SSCV

Modified Threshold Guidelines Organizations may adjust the 20% minimum threshold in specific circumstances:

- **0% threshold** (marked with ~): For academic analysis or understanding raw risk calculations

- **Higher thresholds** (25-50%, marked with ~): For highly regulated industries or conservative risk appetites
- **Always document**: Record the threshold used and business justification

Remember: The standard 20% threshold reflects security industry consensus that no system is ever completely secure. Deviations should be intentional and documented.

5.4 Integration Points

1. **Vulnerability Scanning**: Enrich scan results with SSCV data
2. **Patch Management**: Sort patches by CRS instead of raw CVSS
3. **Risk Assessments**: Include SSCV in system risk profiles
4. **Compliance**: Document SSCV as part of risk-based patching justification
5. **Asset Management**: Store SSCV vectors with system inventory

6. Quick Reference

6.1 Component Summary

Component	Values	Best → Worst	Min Weight
OS	H C O L	0.8 → 3.0	0.8
NE	A I P E	0.3 → 3.0	0.3
AC	Z F B N	0.7 → 3.0	0.7
EP	M A B N	0.8 → 3.0	0.8
DL	P I M C	0.6 → 3.0	0.6
PS	C D B U	1.0 → 3.0	1.0

6.2 Score Notation

- **Standard CRS**: `5.2` (using 20% minimum threshold)
- **Modified CRS**: `5.2`~ (using non-standard threshold)
- Document any threshold other than 20% for transparency

6.4 Common System Profiles

- **Secure Cloud Workload**: `OS:C/NE:P/AC:F/EP:A/DL:I/PS:C`
- **Legacy Database**: `OS:O/NE:I/AC:B/EP:B/DL:C/PS:D`
- **DMZ Web Server**: `OS:H/NE:P/AC:F/EP:M/DL:M/PS:C`
- **Developer Workstation**: `OS:C/NE:E/AC:F/EP:A/DL:I/PS:D`
- **IoT Device**: `OS:L/NE:E/AC:B/EP:N/DL:P/PS:U`

6.5 Calculation Cheat Sheet

For good security (Context × Exposure ≤ 1):

1. Average (OS + AC + EP + PS) = Context_Multiplier

2. (NE × DL) ÷ 8 = Exposure_Factor
3. CVSS × Context × Exposure = Adjusted_Score
4. CVSS × 0.2 = Minimum_Score
5. Final CRS = max(Adjusted_Score, Minimum_Score)

For poor security (Context × Exposure > 1):

1. Calculate Context_Multiplier and Exposure_Factor (same as above)
2. CVSS × Context × Exposure = Raw_Score
3. Final CRS = min(Raw_Score, CVSS × 2)

Always: Add ~ suffix if using non-standard minimum threshold

7. Future Versions

SSCV v2.0 (planned) will add components for mature security programs:

- Business Criticality (BC) - When data sensitivity isn't enough
- Update Mechanism (UM) - For detailed patch management
- Supply Chain Security (SC) - For SBOM-aware organizations
- Safety Requirements (ST) - For ICS/OT environments
- Physical Security (PH) - For data center considerations
- Availability Requirements (AV) - For uptime-critical systems

Organizations can adopt v2.0 components selectively as their programs mature.

8. Conclusion

SSCV provides a practical framework for contextualizing vulnerability risk while maintaining the fundamental security principle that no system is ever completely secure. By focusing on six core components that every security team can assess, SSCV v1.0 enables immediate adoption without requiring specialized tools or extensive security maturity.

The framework acknowledges that:

- Security measures reduce but never eliminate risk
- Context matters enormously in real-world risk
- Perfect security is an impossible goal
- Simplicity drives adoption and value
- Organizations may adjust thresholds with proper documentation

Organizations implementing SSCV should remember it's a tool for prioritization, not a replacement for comprehensive security practices and defense-in-depth strategies.

License

Contributing

We welcome contributions to the SSCV framework! Whether you're interested in:

- Reporting bugs or requesting features
- Improving documentation or translations
- Enhancing the web calculator
- Sharing implementation use cases
- Contributing code improvements

Please see our CONTRIBUTING.md file for detailed guidelines on how to contribute.

Version History

- v1.0 - Initial release with 6 core components focused on adoption