Common Vulnerability Scoring System (CVSS) for CAN

Guideline on using CVSS to score vulnerabilities of products and systems using CAN

White Paper EmSA-WP-103

Version 1.00
27 August 2025

Jointly authored by

Embedded Systems Academy GmbH	**Embedded Systems Academy, Inc.**
Bahnhofstraße 17	**84 W. Santa Clara St., Suite 700**
30890 Barsinghausen	**San Jose, CA 95113**
Germany	**United States**

www.em-sa.com

Contents

1 Scope and Motivation

This guideline covers using CVSS V4.0 to calculate specific vulnerability scores for vulnerabilities in devices or systems using CAN communication.

https://www.first.org/cvss/calculator/4-0

The EU Cyber Resilience Act (CRA) requires manufacturers to

- handle vulnerabilities systematically and to
- report them with a common severity score.

Since CVSS is the standard scoring system expected by regulators and databases such as EUVD and NVD, everyone involved in CAN products, from designing and manufacturing components, via integrating them into larger systems to operating them, must be able to work with CVSS.

CAN technology does not map cleanly to IT-oriented scoring categories, which creates uncertainty and inconsistent results. This document provides practical guidance on applying CVSS v4.0 to CAN devices and submodules, so that scoring becomes consistent, risks are communicated clearly, and compliance with CRA obligations can be demonstrated.

This approach also complements the risk assessment framework of IEC 62443.

2 Introductory Notes

2.1 How the CVSS works

CVSS produces a numeric score between 0.0 and 10.0 that represents the severity of a vulnerability to score. The score is calculated from a set of metrics grouped into categories as addressed below. Each metric is expressed as a selection in a vector string including every metric (for example: CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:N/VI:H/VA:H). The vector makes the scoring transparent and reproducible, since anyone can see with the abbreviated metrics which choices were made.

2.2 Who calculates what?

Manufacturer and suppliers of CAN devices reporting a vulnerability fill in the sections Base Metrics and optionally Supplemental Metrics. This provides a CVSS score for the vulnerability of the affected component.

Integrators that build larger system using an affected component copy the values reported from the manufacturer and add their own re-score for Environmental (Modified Base Metrics) considering how that affected component is integrated into the larger system. This adapts the CVSS score as to how it affects that system.

End-users and operators of such systems copy both value sets from manufacturer and integrator and add their own Environmental (Security Requirements) and Threat Metrics. This gives them an individual CVSS for their specific use case.

2.3 Vulnerable system and subsequent system

In CVSS V4.0 the following set of base parameters are used twice.

- Loss of Confidentiality (VC, SC)
- Loss of Integrity (VI, SI)
- Loss of Availability (VA, SA)

The first set is used for the ***Vulnerable System Impact Metrics (Vx)***, so this is the initial/direct impact of an attack on a single node or location where the attack happens.

The second set is used for the ***Subsequent System Impact Metrics (Sx)***, so this includes the effect on other connected nodes or systems.

If an attack is directly to a single CAN bus and cannot propagate beyond that CAN bus, then the values for the subsequent system may be left to the setting None.

If the attack can propagate beyond a single CAN bus, rate the subsequent system(s).

If the primary attack is to a single CAN node, like a gateway, then this node is the first vulnerable system and everything connected to the gateway is the subsequent system.

2.4 Interpreting Scores and Consequences

A CVSS score is not only a technical indicator but also a trigger for action. The scale from 0.0 to 10.0 is divided into ranges that signal how urgent a response should be.

- **0.1 – 3.9 (Low)**: Vulnerabilities in this range are not expected to cause major disruption. They should be documented and addressed during normal maintenance cycles. It might be acceptable to only document the issue without further mitigation.

- **4.0 – 6.9 (Medium)**: These may be exploited with moderate impact. Operators should plan remediation in a timely manner and track whether mitigations exist.

- **7.0 – 8.9 (High)**: These vulnerabilities can cause serious disruption or compromise system integrity. Operators should treat them as priority issues, apply patches as soon as feasible, or deploy compensating controls until a fix is available.

- **9.0 – 10.0 (Critical)**: These demand immediate attention. Operators should assume active exploitation is possible and implement emergency measures such as disabling services, applying workarounds or restricting access until a permanent fix is in place.

System operators are expected to integrate these scores into their vulnerability handling process. A High or Critical score should trigger a documented risk assessment and vulnerability reporting to ENISA or single CRA reporting platform.

3 Metric Groups of the Calculator

The CVSS calculator divides the metrics to report into the following groups: Base Metrics, Supplemental Metrics, Environmental Modified Base Metrics, Environmental Security Requirements and Threat Metrics.

3.1 Base Metrics

Provided by manufacturer.

These are mandatory parameters required to determine the base score.

Attack Vector (AV)

Describes where the attacker must be located to exploit the vulnerability.

- **Network (N)**: choose if the vulnerability can be exploited remotely over the Internet or other routable networks.
- **Adjacent (A)**: choose if the vulnerability can be exploited via another local, limited range network (e.g. adjacent CAN or fieldbus, Bluetooth).
- **Local (L)**: This applies if a CAN node has a local user interface (like a permanent connected diagnostic unit) and attack is via that user interface.
- **Physical (P)**: choose if the vulnerability requires direct access to the CAN bus or a hardware, possibly PCB level, debug port.

Attack Complexity (AC)

Captures how difficult the attack is to carry out once the attacker is in the right place.

- **Low (L)**: choose if exploitation works reliably with simple crafted CAN frames or replay traffic.
- **High (H)**: choose if exploitation depends on precise timing, hardware conditions, or specialized techniques such as controlled error frame generation.

Attack Requirements (AT)

Describes whether specific conditions must already be present in the target environment for an exploit to succeed.

- **None (N)**: choose if the vulnerability can be exploited under normal operating conditions, without any special state or prerequisite. See Example at end.
- **Present (P)**: choose if exploitation requires the system to be in a particular mode or configuration, such as a node being in diagnostic session, bootloader mode, or other non-regular operation state.

Privileges Required (PR)

Specifies what level of access the attacker must already have on the target.

- **None (N)**: choose if no prior access is needed (no barrier requiring password or key).
- **Low (L)**: choose if the attacker needs limited privileges, such as diagnostic access or firmware update rights.
- **High (H)**: choose if the attacker must already control the ECU with full admin/debug access.

User Interaction (UI)

Describes whether a human user must participate for the vulnerability to be exploited.

- **None (N)**: choose if no user involvement is needed.
- **Passive (P)**: choose if a user must be present but does not need to act, such as being logged into a diagnostic tool while the attack runs.
- **Active (A)**: choose if the exploit only succeeds when a user performs an action, such as initiating a firmware update or enabling a configuration, bootloader or test mode.

Loss of Confidentiality (VC, SC)

Assesses how much sensitive information could be disclosed.

- **None (N)**: choose if no meaningful data (personal data, company intellectual property) is exposed.
- **Low (L)**: choose if exposed data (e.g. telemetry) can be read but has little value.
- **High (H)**: choose if sensitive personal, diagnostic or intellectual property data, cryptographic material, or critical state information is exposed.

Loss of Integrity (VI, SI)

Assesses the extent to which data or functions can be altered.

- **None (N)**: choose if no data or commands can be changed.
- **Low (L)**: choose if small unauthorized changes are possible but have limited impact.
- **High (H)**: choose if injected frames can spoof sensor data or alter control commands (this is the case with full access to classical CAN and no authenticity protection).

Loss of Availability (VA, SA)

Assesses how much the vulnerability can disrupt operations.

Note that flooding the CAN network with high-priority traffic is not considered a security vulnerability. Such brute-force disruption does not exploit a security weakness in the system, it is physical sabotage, comparable to cutting the network wires. Therefore, we only distinguish between availability vulnerabilities caused by carefully selected, temporary replay or sequence injections. In any case, as part of the availability metric take into account how nodes or the entire system recovers from such attacks, including temporal flooding.

- **None (N)**: choose if operation is not disrupted by the attack.
- **Low (L)**: choose if attacker can temporarily reset non-critical nodes, but system self-recovers after power-cycle.
- **High (H)**: choose if attacker can re-configure nodes so disruptions are persitent, also continue after power-cycle, requires operator action to fix.

3.2 Supplemental Metrics

Provided by manufacturer.

These do not change the calculated score, they provide additional information useful for affected users and operators.

Note that the value Not Defined may result in worst case scoring,

Safety (S)

Define only when exploitation could directly or indirectly cause harm to human life, health, property or environment.

- **Negligible (N)**: choose if exploitation has no realistic safety impact.
- **Present (P)**: choose if exploitation could plausibly cause injury, fatalities, or physical damage.

Automatable (AU)

Indicates whether the attack can be easily repeated or scaled.

- **Yes (Y)**: choose if it can be scripted and repeated (typical for CAN injection/re-play).
- **No (N)**: choose if it always requires specialized lab effort (system specific injec-tions, side-channel).

Recovery (R)

How difficult it is for the affected system to recover once the vulnerability is exploited.

- **Automatic (A)**: choose if normal system processes restore operation (e.g. ECU reboot clears the fault).
- **User (U)**: choose if recovery needs user action like re-flashing firmware or re-starting services.
- **Irrecoverable (I)**: choose if exploitation leaves permanent damage or requires hardware replacement.

Value Density (V)

Describes how much value an attacker gains from a single successful exploitation.

- **Diffuse (D)**: choose if one exploited instance only yields access to limited data or control of a single device, such as compromising a single CAN node.
- **Concentrated (C)**: choose if one exploited instance yields access to many assets at once, such as compromising a fleet gateway or backend server.

Vulnerability Response Effort (VRE)

How much effort is needed by the manufacturer to respond (patch, mitigate, deploy fixes).

- **Low (L)**: choose if a software patch or configuration change is sufficient.
- **Moderate (M)**: choose if multiple nodes or subsystems need coordinated fixes.
- **High (H)**: choose if response requires major redesign, cryptographic replacement, or hardware recall.

Provider Urgency (PU)

How urgent the manufacturer believes remediation is relative to other issues.

- **Red (R)**: choose to flag this as requiring immediate attention and remediation.
- **Amber (A)**: choose to flag this as important but not requiring immediate handling.
- **Green (G)**: choose to flag this as to be handled in routine maintenance.

3.3 Environmental (Modified Base Metrics)

Provided by system integrator, taking affected product from Base Metrics and integrating it into a larger system.

This set of parameters are a repetition and allow you to adapt the base score to a specific deployment or integration context. You only fill in Environmental metrics when the deployment or integration changes the assumptions made in the Base score.

For example, a vulnerability that is theoretically exploitable on a CAN node or bus may have different significance depending on where and how the node or bus is integrated into a larger system. Is it just part of environmental controls in an operator cabin or does it affect operator controls?

The key difference from the Base set is that you don’t evaluate the vulnerability, you now choose the values to reflect your specific system’s context.

3.4 Environmental (Security Requirements)

Provided by end-user / operator. An operator affected by the vulnerability can now add their personalized security requirements.

This set of parameters reflects the values required by the end application or user. For confidentiality, integrity and availability you specify how important these are to the application. For example, if there is a confidentiality breach, but to the end-user and his application this is not important, then it may be set to Low here.

3.5 Threat Metrics

The Threat Metrics parameter captures the current state of exploitation.

Exploit Maturity (E)

Is the attack currently exploited?

- **Attacked (A):** choose if exploitations are already happening.
- **PoC (P):** choose if proof-of-concept code exists but not seen actively exploited.
- **Unreported (U):** choose if there is no public information about exploitation.

4 Applying CVSS to a Classical CAN Node

4.1 Generic scoring of classical CAN nodes

A known vulnerability of classical CAN nodes and systems is, that attackers with full physical access to the CAN wiring can read, record, inject, replay frames or flood the bus with high priority frames. Therefore, confidentiality, integrity and availability are all breached. Based on the assumption that data confidentiality is not required, the CVSS score and vector is as follows:

- **CVSS v4.0 Score 5.2 / Medium**
- CVSS:4.0/AV:P/AC:L/AT:N/PR:N/UI:N/VC:N/VI:H/VA:H/SC:N/SI:N/SA:N

If confidentiality is required, then the CVSS score and vector becomes:

- **CVSS v4.0 Score 7.0 / High**
- CVSS:4.0/AV:P/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

Even without a specific security breach, such classical CAN nodes already have a relatively high score.

This can be eliminated by taking care of the physical attack vector. If the system is enclosed and locked in a way that no attacker gets access, then there is no attack vector and no vulnerability.

4.2 Adding attack requirements

Where that is not possible, another option to lower the score is to carefully re-evaluate the Attack Requirements. Exploitation requires not only physical access but also that the CAN system is powered and active, possibly in a special state. Depending on system this might involve certain initial sequences outside the machine the attacker cannot fully control.

If you can claim that some attack requirements must be present, then the new value is:

- **CVSS v4.0 Score 5.4 / Medium**
- CVSS:4.0/AV:P/AC:L/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

Or without confidentiality required:

- **CVSS v4.0 Score 4.4 / Medium**
- CVSS:4.0/AV:P/AC:L/AT:P/PR:N/UI:N/VC:N/VI:H/VA:H/SC:N/SI:N/SA:N

On the integration level we can do even more to reduce the score. In the Modified Base Metrics we again have Confidentiality, Integrity and Availability.

If Confidentiality is not required, because no personal or important data is communicated, that can be set to None.

4.3 Adding security event monitoring

If the integrated CAN system features a CAN security event monitor that can detect and report issues like injections, malicious re-configurations or manipulative resets, you can reduce the Integrity and Availability metrics to Low.

That gives us:

- **CVSS v4.0 Score 2.4 / Low**
- CVSS:4.0/AV:P/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/MVC:N/MVI:L/MVA:L

5 Impact of Node CAN Security Functions

In the CiA (CAN in Automation) working groups, there are currently several security functions in discussion and development. In this section we summarize their impact on vulnerability scoring using CVSS.

5.1 Node authentication and injection detection

Here injection detection means that a node who uses a CAN ID for transmission can detect if anyone else is transmitting this CAN ID – which would only happen if it were a malicious injection.

If a node offers an authenticated ID and status, then on a system level a security task can cyclically poll nodes for their ID and status – injections would be reported.

Such a setup would reduce the **Integrity** metric to **Low** in the Modified section, as injections are detected and eventually reported. It would not allow a setting of None, as the injected frames would still go through, they are only detected after the fact.

5.2 Configuration lock and reset protection

If a node offers an option to lock its configuration and to only accept authenticated reset commands, then attacks on the availability are greatly reduced. An attacker could not easily reconfigure a node or send it commands to reset.

Therefore, such a setup would reduce the **Availability** metric to **Low** or **None**, depending on further attack options available.

5.3 Authentication and encryption

Security sublayer implementations like CANcrypt and SPsec add authenticity and confidentiality to selected or all CAN frames communicated. This immediately protects all three confidentiality, integrity and availability as malicious frames are rejected by the receivers and not processed.

When used for all essential communication, such nodes or systems have a **Confidentiality**, **Integrity** and **Availability** metric of **None** – unless a vulnerability explicitly is a successful attack on the security sublayer.