Hardware CWE™ Special Interest Group (SIG)

Gananand Kini, Bob Heinemann, Luke Malinowski, Gage Hackford, Chris Lathrop, Steve Christey Coley, Alec Summers

MITRE

January 13, 2023

CWE and CAPEC are sponsored by U.S. Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency (CISA).

Agenda

REMINDER: This meeting is being recorded.

- **Housekeeping**
- **Announcements**
- **Working Items for this meeting:**

1	Scope exclusions (updates and changes): Will be made public by this meeting.	Steve C.	20 min
2	Transient Execution Weaknesses Update: Progress report and call for participants.	Gananand K.	10 min
3	CWE-319 rephrase to include HW: Content of CWE-1324 as a JTAG Demonstrative example.	Luke M.	30 min

Housekeeping

- **Schedule:**
  - **Next Meeting:**
    - **Rescheduled for February 10**
    - **12:30 – 1:30 PM EST (16:30 – 17:30 UTC)**
    - **Microsoft Teams**
- **Contact: cwe@mitre.org**
- **Mailing List:** *hw-cwe-special-interest-group-sig-list@mitre.org*
- **Minutes from previous meetings available on our Github site:**
  - **https://github.com/CWE-CAPEC/hw-cwe-sig**

Mailing List Items

Mailing List: *hw-cwe-special-interest-group-sig-list@mitre.org*

- *NOTE: All mailing list items are archived publicly at:*
  - *https://www.mail-archive.com/hw-cwe-special-interest-group-sig-list@mitre.org/*

Announcements

- **CAPEC targeted for January 24, 2023.**
- **CWE 4.10 targeted for January 31, 2023.**
- **New ICS/OT SIG Working Groups launched. “Boosting CWE Content” group is impacting HW CWE (updates and additions to HW CWE).**
- **Mailing List Discussion:**
  - CWE Mappings to potential categories “Hardware Trojans”, “Reverse Engineers”, and “Untrusted Manufacturing”

Scope Exclusions Updates and Changes

Steve C.

tl;dr what was “updated” since we last covered scope exclusions for more than 10 minutes?

Scope Exclusions – Status

- **10 scope exclusions (so far)**
- **These were internally identified, reviewed, and refined for much of 2023**
- **Primary influencers: HW SIG, ICS/OT SIG, and external submitters**
- **Integrated into the external submission process mid-2023**
  - The "Initial Review" phase considers scope exclusions
  - Few hardware submissions since mid-2022
- **"Scope exclusions" are only a subset of "submission problems" (like not describing a weakness clearly, being too low-level abstraction, etc.)**
  - All "submission problems" could apply to existing CWEs, too (ouch)
- **Will be published to cwe.mitre.org ASAP, Real Soon Now, etc.**
- **For this presentation, only a few scope exclusions will be highlighted**
  - See backup slides for others
- **Nothing has been decided; community-wide discussion is important**

Scope Exclusions – Summary (as of Jan 13, 2023)

- **Exclusions of most interest to HW SIG are:**
  - SCOPE.HUMANPROC - Human/organizational process
  - SCOPE.NOMITS - No actionable mitigations
  - SCOPE.CUSTREL - Not customer-relevant
  - SCOPE.CONFLICT - Conflict/contradiction with other weaknesses
- **Most of these have been influenced by HW SIG:**
  - SCOPE.NOTREAL - Not a real-world issue
  - SCOPE.MOTIVE - Motivation instead of the mistake
  - SCOPE.SITUATIONS - Focus on situations in which weaknesses may appear
  - SCOPE.GROUPING - Grouping of issues without a common behavior
  - SCOPE.NOSEC - Not security-related
  - (New) SCOPE.ADMINERR - Admin/user error

SCOPE.HUMANPROC - Human/organizational process

Info	Details
Description	The submission focuses on a problem in a human or organizational process or policy (e.g., insufficient developer training) that is not measurable and does not produce concrete artifacts that identify weaknesses.
Rationale	Weaknesses can emerge as a result of these activities. Other efforts cover this area, e.g., BSIMM, OWASP SAMM, and NIST Secure Software Development Framework. Note that this exclusion is similar to SCOPE.ADMINERR in that it reflects human actions.
Examples	Lack of developer training, insufficient testing, not following secure coding standards, and corporate policy gaps. **Maybe CWE-1297: Unprotected Confidential Information on Device is Accessible by OSAT Vendors.**
Resolution	As of January 2023, such submissions will be rejected (if framed as "weaknesses") but could help modify existing CWEs for elements such as modes of introduction.
Debate	No active debate, but manufacturing processes (of concern to hardware, ICS/OT, supply chain, etc.) may fall under this exclusion.
Status	Proposed

Side bar: this is not defined clearly enough. A human sysadmin performing an insecure configuration is not “HUMANPROC” because they are directly causing the insecure behavior in the product.

SCOPE.NOMITS - No actionable mitigations

Info	Details
Description	There are no actionable mitigations available to the developer/designer/manufacturer to prevent or reduce the weakness.
Rationale	If a product has a weakness type with no known fixes or mitigations, then a CWE entry would not be helpful to the developer / designer / manufacturer, i.e., it is not actionable.
Examples	-
Resolution	Submissions will be reviewed on a case-by-case basis, delayed, and possibly cited as examples until this exclusion is finalized after extensive community feedback.
Debate	As of January 2023, there is some community disagreement about this scope exclusion: (1) Developers could **avoid the affected functionality altogether**. (2) Weaknesses without mitigations could serve as **topics for academic study**.
Status	Proposed

SCOPE.CUSTREL - Not customer-relevant

Info	Details
Description	The issue is not relevant to the threat model or security concerns of the product's owner/operator (i.e., the customer).
Rationale	Traditionally, the focus on publicly-disclosed vulnerabilities has been on products that affect customers. Widening the scope to include non-customer interests would risk creating weaknesses with no relevance to customers who want to acquire secure products.
Examples	**CWE-1278 is about avoiding reverse engineering using certain techniques, which does not affect the customer directly.**
Resolution	Submissions will be reviewed on a case-by-case basis, delayed, and possibly cited as examples until this exclusion is finalized after extensive community feedback.
Debate	As of January 2023, there is some community disagreement about this scope exclusion. **Customers are not the only participants within the ecosystem**, and concerns such as Intellectual Property violations have clear financial implications for vendors if compromised. If not handled carefully, allowing CWEs in this area could cause conflicts; e.g., if a CWE is created that a product doesn't obscure its code enough, that CWE would apply to all open source products. Note that there may be a logical inconsistency between SCOPE.CUSTREL and SCOPE.CONFLICT, since there may be differences in threat models across different industries, and each participant might have different priorities. For example, in healthcare, availability is a priority that may conflict with password requirements, since locking out a medical device user may cause security issues. Similarly, medical device users might be considered a threat actor against the security of their device if they can modify the device's behavior in ways that are not allowed by the doctor; the Do-It-Yourself (DIY) community of insulin-pump hackers demonstrates this conflict with device manufacturers.
Status	Proposed

SCOPE.CONFLICT - Conflict/contradiction with other weaknesses

Info	Details
Description	The issue directly conflicts with or contradicts other CWE entries, e.g., "X is bad" in one CWE, and "Y is bad so do X instead" in another CWE.
Rationale	Directly-conflicting CWE entries can cause user confusion and/or suggest some lack of agreement as to what constitutes "secure“ products.
Examples	-
Resolution	The submission might be rejected unless it reveals some other problems with existing CWE content, which might prevent it from progressing to later stages until the existing CWE issues can be addressed satisfactorily.
Debate	As of January 2023, one area of active debate involves the desire of some community members for CWE to **include weaknesses for code/logic that can be easily reverse-engineered** (note that this is also affected by SCOPE.CUSTREL). If CWE covers a concern that effectively promotes code obfuscation (i.e., says that it's too easy to extract the real code/logic), this **could be seen to directly conflict with CWE-656: Reliance on Security Through Obscurity**, which effectively follows Kerckhoff's principle, summarized as: "a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge." As another example, in 2021, members of the CWE-Research list discussed older CWE entries related to password aging that directly implied that password aging was an important capability, but the modern belief is that password aging should be unnecessary. Yet, creation of a new entry such as "Reliance on Password Aging" would conflict with the original CWE entries and any products that still rely on passwords. Note that there may be a logical inconsistency between SCOPE.CUSTREL - and SCOPE.CONFLICT… [see SCOPE.CUSTREL]
Status	Proposed

Microarchitectural weaknesses update

Gananand K.

Ongoing work …

- We have a great partitioning of the issues currently.
- Titles seem to be describing functional behavior. Need them to better describe the weaknesses and conditions that lead to the weakness instead.
- Also need better descriptions and extended descriptions.
- Going to distribute (soon!) to the team of Scott C., Nicole F., David K., and Jason O. for some additional feedback in Word document format using Box. (Silently praying this works for everyone)

Microarchitectural Weaknesses thus far

- **SUB-CWE-B: Transient Data Forwarding from an Operation that Triggers a Processor Event**
  - Weakness appears to be related to incorrectly forwarding data when a processor event is triggered.
- **SUB-CWE-C: Transient Execution Influenced by Shared Microarchitectural Predictor State**
  - Weakness here appears to be related to sharing the microarchitectural predictor state between domains incorrectly when domain transition occurs.
- **SUB-CWE-D: Microarchitectural Predictor causes Transient Execution**
  - Weakness here appears to be related to the misuse of the microarchitectural predictor itself such that sensitive data could potentially be inferred from the behavior and state changes observed after predictor affects transient execution.

*SUB – indicates a submission for consideration

Microarchitectural Weaknesses thus far

- **SUB-CWE-A: Processor Event Causes Transient Execution**
  - Weakness still unclear.
  - Transient execution typically is normal and expected behavior. The title seems to imply a non-issue here.
  - Initial motivation was for this to be a a category and a catch all for transient execution issues that do not fall under the previous CWEs.

Want to make your voice heard on this topic?

- Contact cwe@mitre.org and let us know.
- Attend our discussion group.
- Discuss on the HW CWE Mailing List.

CWE-319 and CWE-1324 Discussion

Luke M.

Summary

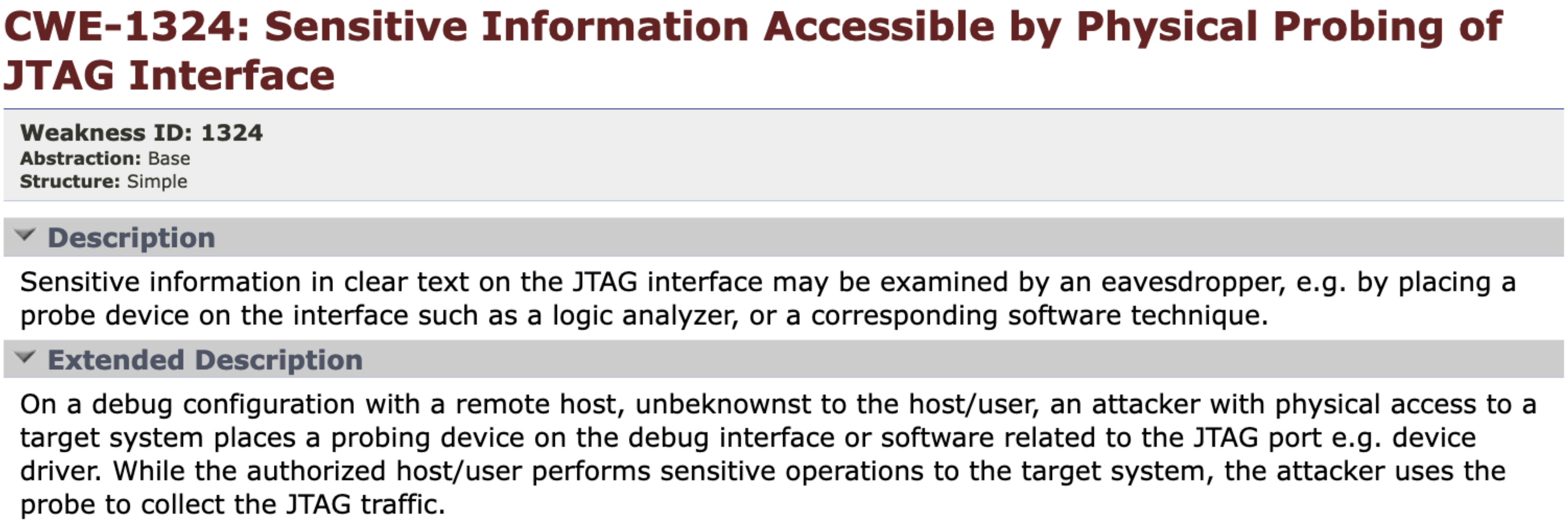
- CWE-319: Cleartext Transmission of Sensitive Information
  - In CWE 4.10 has its scope expanded to include hardware weaknesses
  - "The ~~software~~ product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors."
- CWE-1324: Sensitive Information Accessible by Physical Probing of JTAG Interface
  - Through discussions with author, the underlying weakness primarily deals with "sensitive information sent in cleartext on the JTAG interface"
  - Up until now, CWE-1324 addressed unencrypted hardware communications
  - In addition:
    - This entry is at a lower level of abstraction than preferred
    - This entry suggests multiple weaknesses through its mitigations that are covered by other CWE's (Access Control, "Active Debug Code")

MITRE Proposal

- With that we think it reasonable to now rely on CWE-319 to address the weakness of transmitting cleartext for both hardware and software.
- We think it would make a good addition to CWE-319 as a demonstrative example
  - CWE-1324 successfully illustrates CWE-319 in JTAG enabled platforms
  - This will centralize all concerns about unencrypted communications in one spot and improve usability for the underlying weakness
- Access Control and "Active Debug Code" aspects of CWE-1324 – are captured elsewhere in the hardware view

Discussion!

CWE-1324 Reference



CWE-1324: Sensitive Information Accessible by Physical Probing of JTAG Interface

Weakness ID: 1324
Abstraction: Base
Structure: Simple

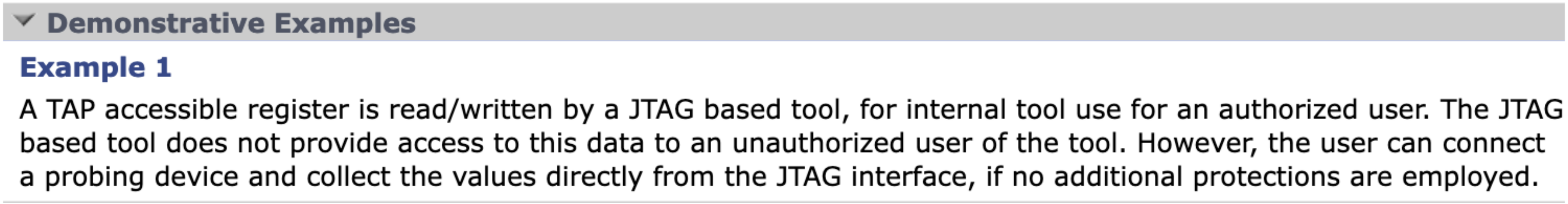
Description

Sensitive information in clear text on the JTAG interface may be examined by an eavesdropper, e.g. by placing a probe device on the interface such as a logic analyzer, or a corresponding software technique.

Extended Description

On a debug configuration with a remote host, unbeknownst to the host/user, an attacker with physical access to a target system places a probing device on the debug interface or software related to the JTAG port e.g. device driver. While the authorized host/user performs sensitive operations to the target system, the attacker uses the probe to collect the JTAG traffic.

CWE-1324 Reference

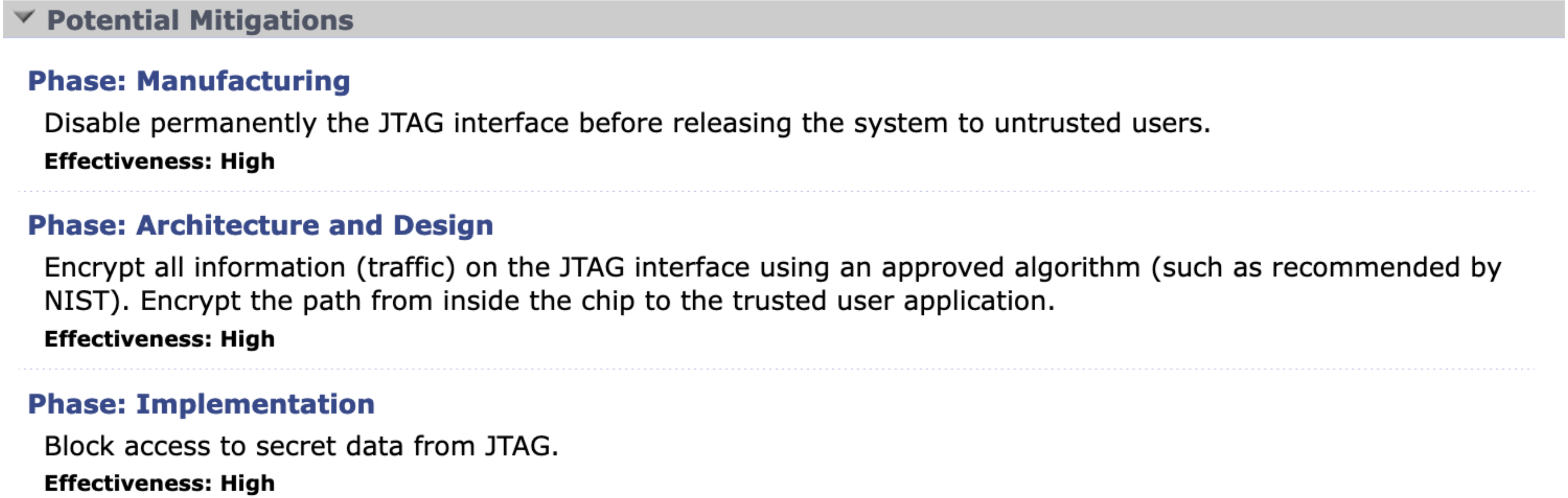


Demonstrative Examples

Example 1

A TAP accessible register is read/written by a JTAG based tool, for internal tool use for an authorized user. The JTAG based tool does not provide access to this data to an unauthorized user of the tool. However, the user can connect a probing device and collect the values directly from the JTAG interface, if no additional protections are employed.

CWE-1324 Reference



Potential Mitigations

Phase: Manufacturing

Disable permanently the JTAG interface before releasing the system to untrusted users.

Effectiveness: High

Phase: Architecture and Design

Encrypt all information (traffic) on the JTAG interface using an approved algorithm (such as recommended by NIST). Encrypt the path from inside the chip to the trusted user application.

Effectiveness: High

Phase: Implementation

Block access to secret data from JTAG.

Effectiveness: High

Next Meeting (February 10)

CWE@MITRE.ORG

- **Mailing List:** *hw-cwe-special-interest-group-sig-list@mitre.org*
  - ***NOTE: All mailing list items are archived publicly at:***
    - *https://www.mail-archive.com/hw-cwe-special-interest-group-sig-list@mitre.org/*
- **What would members of this body like to see for the next HW SIG agenda?**
- **Questions, Requests to present? Please let us know.**

Scope Exclusions – Others Not Covered in Main Presentation

SCOPE.NOTREAL - Not a real-world issue

Info	Details
Description	The issue has not happened and cannot occur in real-world software or other electronic logic, and/or has no actual security implications.
Rationale	-
Examples	Due to a misinterpretation of source material, CWE-365 originally described an insecure behavior that did not exist in any real-world compilers, so it was deprecated.
Resolution	As of January 2023, the submission will be rejected or delayed pending close review, possibly including community engagement.
Debate	No active debate.
Status	Proposed

SCOPE.MOTIVE - Motivation instead of the mistake

Info	Details
Description	Any characterization of motivation (e.g., "malicious") that does not focus on the actual weakness, whether intentionally or accidentally introduced.
Rationale	A weakness is based on the behavior of the product, and malicious actors can introduce the same weaknesses as non-malicious developers.
Examples	-
Resolution	As of January 2023, submissions are likely to be rejected unless they can be converted to actual weaknesses. Any provided references or examples could be used to modify demonstrative or observed examples of existing CWEs.
Debate	No active disagreement as of January 2023.
Status	Proposed

SCOPE.SITUATIONS - Focus on situations in which weaknesses may appear

Info	Details
Description	The submission focuses on conditions or situations in which weaknesses are more likely to appear but are outside of the direct control of the product.
Rationale	CWE is focused on the flaws/defects that can occur within a product. While various situations can contribute to the introduction of weaknesses, these are better captured as modes of introduction, which are recorded in a separate field in CWE entries.
Examples	"The growing connectivity between IT and OT systems can introduce new vulnerabilities.“ (SEI ETF)
Resolution	Submissions are likely to be rejected unless they can be recast as categories or can influence modes of introduction of existing CWEs, i.e., modify entries.
Debate	No active disagreement as of January 2023.
Status	Proposed

SCOPE.GROUPING - Grouping of issues without a common behavior

Info	Details
Description	The submission is a grouping of issues or concerns related to the same feature such as technology, language, development lifecycle, etc., but there is not a common behavior between them all.
Rationale	Weaknesses are based on specific behavior and other shared criteria that can be hierarchically classified under a common ancestor in the research view 1000. If no such ancestor is possible, then the submission is not likely a weakness.
Examples	"If the developer doesn't handle files well, attackers can steal or delete data.“ With respect to hardware, weaknesses that have physical characteristics do not share the same hierarchy in research view 1000. However, the hardware community asked for them to be grouped into a category (CWE-1388).
Resolution	The submission might be acceptable as a category, or some of its content could modify existing entries. In some cases, the submission might influence subtree organization.
Debate	No active disagreement as of January 2023.
Status	Proposed

SCOPE.NOSEC - Not security-related

Info	Details
Description	The issue is solely concerned with safety or reliability and is not related to security. Clarification on privacy: if an issue is related to desires for access control or preservation of confidentiality, it is within scope of "security".
Rationale	Many aspects of industrial safety, such as correct electric shielding and insulation, do not affect security. There are many safety-focused standards throughout different industries, and expanding CWE's scope to include safety-only issues would limits utility to most current CWE users.
Examples	-
Resolution	Submissions will be reviewed on a case-by-case basis, delayed, and possibly cited as examples until this exclusion is finalized after extensive community feedback.
Debate	As of January 2023, it is suspected that there will be some community disagreement about this scope exclusion.
Status	Proposed

SCOPE.ADMINERR - Admin/user error

Info	Details
Description	The issue focuses on security errors that are made by an admin or user of the product/service, not the developer or maintainer of the product/service.
Rationale	CWE covers defects or flaws within the product itself. This is separate from how administrators or users may misuse or abuse the products in ways that were not intended, typically through insecure configuration, so there is little that product developers can do. It seems likely that such issues will not be relevant to most CWE users. Note that this exclusion is similar to SCOPE.HUMANPROC in that it reflects human actions.
Examples	-
Resolution	Submissions will be reviewed on a case-by-case basis, delayed, and possibly cited as examples until this exclusion is finalized after extensive community feedback.
Debate	As of January 2023, this is a new exclusion. It is suspected that there will be some community disagreement about this scope exclusion, since many services deploy and manage their code for customers in ways that obscure the distinction between system administrators and software developers ("DevOps").
Status	New