The EMB3D™ Threat Model for Embedded Devices

Adam Hahn[1]
Jack Cyprus[1]
Dave Keppler[1]
Marie Collins[1]
Chris Harvey[1]
Niyo Little Thunder Pearson
Wyatt Ford[2]
Ang Cui[2]
Michael Locasto[3]

September 2024

The views, opinions, and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

McLean, VA

[1] The MITRE Corporation, [2] Red Balloon Security, [3] Narf Industries

Table of Contents

1 Introduction 1

2 EMB3D™ Overview 2

3 EMB3D™ Structure 3

3.1 Device Properties 3

3.2 Threat Properties 4

3.3 Mitigations 5

3.3.1 Focus on Mitigations 5

3.3.2 Mitigation Tiers 5

3.3.3 ISA/IEC 62443-4-2 Mappings 7

4 Users and Workflow 8

4.1 Threat Modeling Workflow 8

4.2 EMB3D™ Use Cases for Vendors, Asset Owners, Security Researchers, and Testing Organizations 11

4.2.1 User 1: Vendor Product Team 11

4.2.1.1 Prioritization of mitigation adoption into a product line: 11

4.2.1.2 Strategic planning 12

4.2.2 User 2: Asset Owner Security Architect 12

4.2.3 User 3: Security Researcher/Testing Organization 13

5 Conclusion 13

Appendix A Terminology, Challenges, Scope, and Related Efforts 14

A.1 Terminology 14

A.2 Threat Maturity 15

A.3 Key Challenges 16

A.4 Scoping 17

A.5 Related Work 18

1 Introduction

The security of our Nation's critical infrastructure depends on embedded devices that frequently lack adequate security controls or have not undergone sufficient testing for vulnerabilities. The prevalence of these issues is evident through the CISA ICS Advisories,[1] which to date have released 2,459 alerts for ICS devices and software, of which 1,243 have been given a Common Vulnerability Scoring System (CVSS) severity of at least Medium.[2] Further, initiatives such as the White House Industrial Control System Cybersecurity (ICS) memorandum and CISAs Secure-by-Design and -Default focus[3,4] identify the need to improve the security critical infrastructure and associated devices. Despite these efforts, there remains an inconsistent understanding about what threats are posed to embedded devices and what security mechanisms or capabilities mitigate them.

This document introduces the EMB3D™ threat model for critical infrastructure embedded devices. Critical infrastructure embedded devices include a broad set of unique technologies used in, but not limited to, industries such as oil, natural gas, water/wastewater management, automotive, medical, satellite, autonomous, and UAS, along with dependencies on more diverse technologies used across general computing platforms/environments. The threat model is intended to be used by vendors, asset owners/operators, test organizations, and security researchers as a resource to improve the overall security of embedded devices' hardware and software. The objective of this threat model is to provide a single repository of information defining known threats to embedded devices, which align to the unique device features/properties that enable specific threat actions. By mapping the threats to the associated device features/properties, the threat model allows the user to easily enumerate threat exposure based on the known device features. Vendors, asset owners/operators, and security researchers will then be able to identify relevant threats to devices more consistently and comprehensively and ensure those devices include the necessary mitigations to protect them from those threats. In this way, the threat model can also serve as a uniform method for organizations to track and communicate threats and associated security mechanisms in a device. It also provides a common language to communicate security requirements and protections, helps asset owners better evaluate the security claims of a device, guide evidence-based testing, and inform acquisition decisions for a device.

EMB3D™ should be considered a living framework, where new threats and mitigations are added and updated over time as threat actors and security researchers discover new categories of vulnerabilities, threats, and security defenses. Further, EMB3D™ is intended to be a public community resource, where all information is openly available, and the security community can submit additions and revisions.

[1] Cybersecurity Alerts & Advisories | CISA URL: https://www.cisa.gov/news-events/cybersecurity-advisories

[2] ICS Advisory Project. URL: https://www.icsadvisoryproject.com

[3] National Security Memorandum on Improving Cybersecurity for Critical Infrastructure Control Systems. The White House. July 28, 2021. URL:https://www.whitehouse.gov/briefing-room/statements-releases/2021/07/28/national-security-memorandum-on-improving-cybersecurity-for-critical-infrastructure-control-systems/

[4] Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security-by-Design and -Default. April 13, 2023. URL: https://www.cisa.gov/sites/default/files/2023-06/principles_approaches_for_security-by-design-default_508c.pdf

This paper presents the EMB3D™ threat model structure, content, and use cases associated with expected users. Additional discussion on scope, terminology, challenges, related efforts, terminology can be found in Appendix A.

2 EMB3D™ Overview

Figure 1 provides an overview of EMB3D™ and its alignment with other key security frameworks and knowledge bases. The threats defined within EMB3D™ are based on observation of use by threat actors, proof-of-concept and theoretical/conceptual security research publications, and device vulnerability and weakness reports. Mitigation requirements are defined for each threat and are focused on technical mechanisms that device vendors should implement to protect against the threats. The goal is to build in security vice the common practice of pushing responsibility for mitigations to the asset owner, which may lack the technical knowledge or financial capacity to address these issues.

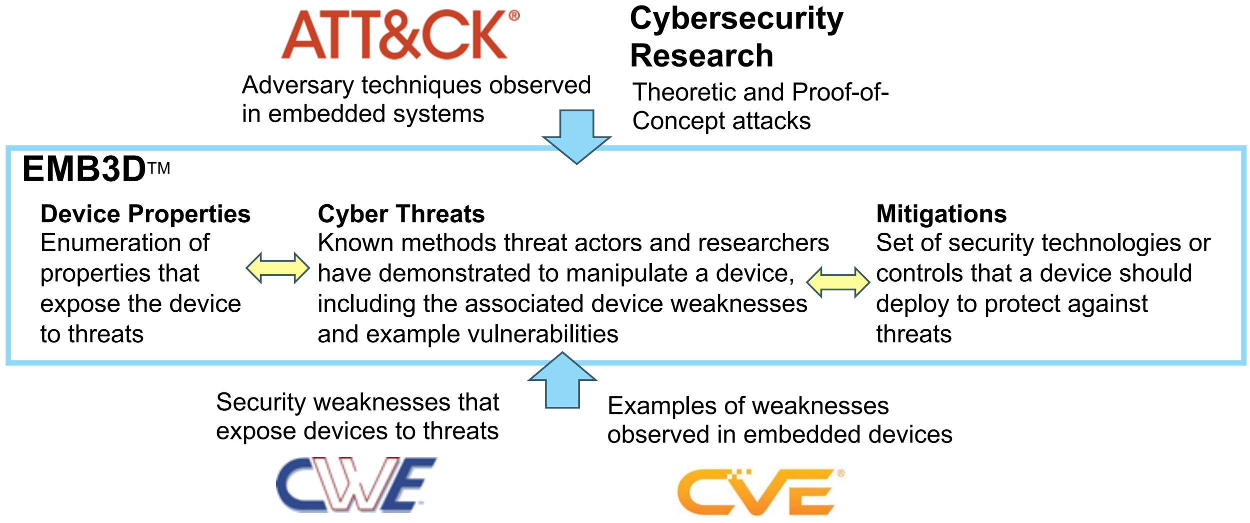

ATT&CK®
Adversary techniques observed in embedded systems
Cybersecurity Research
Theoretic and Proof-of-Concept attacks
EMB3D™
Device Properties
Enumeration of properties that expose the device to threats
Cyber Threats
Known methods threat actors and researchers have demonstrated to manipulate a device, including the associated device weaknesses and example vulnerabilities
Mitigations
Set of security technologies or controls that a device should deploy to protect against threats
Security weaknesses that expose devices to threats
CWE
Examples of weaknesses observed in embedded devices
CVE®

Figure 1. Overview of EMB3D™ and Alignment with Associated Knowledge Bases

A key contribution of EMB3D^{TM} is the integration and cultivation of information on threats to embedded devices across various diverse sources. It integrates information about known adversary behaviors with references to MITRE ATT&CK®[5] techniques. In addition to the integration of information on known adversary behaviors, it also provides references to a wide range of theoretic and proof-of-concept vulnerability research efforts that have been demonstrated by security researchers and are documented through various security conferences, papers, blogs, and other sources. Each threat defined within EMB3D^{TM} includes both a Threat Evidence reference and a Threat Maturity categorization:

- Threat Evidence is a reference to a reputable report describing the threat and documenting its feasibility; this can include a reference to ATT&CK techniques, or other documented reports or research papers/presentations. This provides the users with an easy

[5] MITRE ATT&CK: URL: https://attack.mitre.org

and direct way to understand and access a continually evolving set of information and research about device threats.

- Threat Maturity categorizes how mature that threat is based on whether it (i) has been observed in the wild targeting an embedded device, (ii) is associated with a known weakness (e.g., CWE) that has been exploited, (iii) has a proof-of-concept exploit, or (iv) has a purely theoretic demonstration or description. Threat maturity is a continually evolving property. Many threats continue to mature over time, especially as security researchers make new discoveries and as more threats are seen in real environments. Therefore, the threat maturity field is intended to evolve as new threat evidence is identified.

The threat information is also mapped to a specific CWE, which defines the specific weakness in the device that enables that threat. It also provides references to CVEs, specifically those defined within CISA ICS Advisories,[6] as examples of those weaknesses within actual ICS devices and software platforms. This information is intended to further provide credibility about device weaknesses that need to be mitigated to protect against each threat.

3 EMB3D™ Structure

The content of EMB3D™ is structured into three main categories: (i) device properties, (ii) threat properties, and (iii) mitigations. Information on each of these categories is provided below.

3.1 Device Properties

The device properties enumerate and describe various hardware and software components and capabilities of a device. These range from physical hardware, network services and protocols, software, and firmware. The following list identifies the top-level device properties categories and provides a short description of the types of properties within each category.

- **Hardware Architecture**: The processors, memory, storage, Field Programmable Gate Arrays (FPGA), and other circuit-board level components, along with physical interfaces used for normal device usages (e.g., consoles, serial, USB), or those used for debugging purposes (e.g., Joint Test Action Group [JTAG], Universal Asynchronous Receiver-Transmitter [UART]).
- **System Software**: The underlying system-level software that controls the device (e.g., operating systems and firmware), its capabilities, and procedures to update the software (e.g., over-the-air update).
- **Application Software:** Software platforms (e.g., hosted web servers, web-server interaction, runtime environment) and programmability features of devices that implement its role-specific services and functionality.
- **Networking:** Device-based networking hardware (e.g., Ethernet interfaces, Bluetooth receiver/transmitters) and associated protocols supported by these interfaces (e.g., Open

[6] Cybersecurity Alerts & Advisories. URL: https://www.cisa.gov/news-events/cybersecurity-advisories

Platform Communications United Architecture [OPC UA], Controller Area Network [CAN bus]).

Each of these high-level categories is then further divided into sub-properties that are then mapped to a set of threats. The properties mapping informs users which threats are associated with a given device property. The properties mapping is not intended to enumerate every relevant precondition or requirement that determines the credibility of the threat, but rather to provide sufficient distinction between which threats are most relevant.

3.2 Threat Properties

The threats, which are mapped to the previously identified device properties, identify how a threat actor can achieve some objective or effect on a system or device. Each threat will include the following information:

- **Threat ID:** This is a unique identifier for the threat, expressed in the format TID-###.
- **Threat Overview:** This is a short description of the threat.
- **Threat Description:** This is a more complete description of the threat. The description includes (i) information about the technical mechanisms/features that are targeted by the threat; (ii) the actions that must be performed by the threat actor to cause the threat's effect, including the impact or effect the threat will have on the device; and (iii) the vulnerabilities or weaknesses within that mechanism that enable the threat actions.
- **Threat Maturity and Evidence:** This provides background information on the threat, including its current maturity level and evidence supporting its feasibility:
  - *Threat Maturity:* This defines the maturity of the threat, including whether it is an Observed Adversarial Technique, Known Exploitable Weakness (KEV), Proof of Concept, or Theoretic threat.
  - *Threat Evidence:* This provides references to the specific threat event, including ATT&CK TTPs, technical reporting, and research papers/presentations defining the threat. The exact contents of this field depend on the defined Threat Maturity level, as defined in the following table.

Threat Maturity	Threat Evidence
Observed Adversarial Technique	ATT&CK technique or documented report
Known Exploitable Weakness	Documentation of known weakness exploitation, such as a CWE associated with a KEV catalog entry
Proof of Concept	Reference to research paper/report
Theoretic	

- **Supporting Weaknesses and Vulnerabilities:** This provides information about weaknesses that correlate with the threat, and specific examples of these weaknesses being observed within embedded devices.

- *Weakness:* The CWE best associated with this threat. This mapping is to a CWE with the lowest possible Abstraction level (e.g., Variant, Base), but will map to a higher-Abstraction level CWE (e.g., Pillar, Class) if a relevant lower-level one is not available.
- *Vulnerabilities:* Where available CVE mappings are provided as an example of that weakness being identified on an embedded device.

3.3 Mitigations

A set of mitigations are documented for each threat. They are primarily intended to be used by device vendors to reduce the risk of a threat but can also be used by end users to validate that devices include recommended mitigations. Mitigations are discussed at a high level to define what mechanisms or technologies provide protection from that threat while retaining flexibility regarding how mitigations can be implemented within the unique constraints of each device. They are not intended to be prescriptive. Where possible, EMB3D™ includes information about inadequacies, weaknesses, and implementation challenges with implementing different mitigations to provide more context regarding their adoption.

Each Mitigation includes the following fields:

- **Description**: This provides a description of how the mitigation is implemented and why it provides mitigation to a threat.
- **References**: This includes references to published artifacts that includes additional technical description of the mitigations and associated implementation guidance.
- **Tier**: This specifies the level of maturity of the mitigation and the associated implementation difficulty. There are three tiers; Foundational, Intermediate, and Leading. Each is defined in more detail below.

3.3.1 Focus on Mitigations

A mitigation focuses specifically on mechanisms that can be built into the device to reduce the impact of the associated threat. This is opposed to environment-specific device deployment guidance that may reduce the device's overall exposure to the threat, but in some deployments may not be practical or effective. For example, vendor guidance suggesting that either (i) the device should be isolated from other devices on a network or (ii) additional network monitoring capabilities are needed to detect the threat is not considered an adequate mitigation for EMB3D™.

3.3.2 Mitigation Tiers

While each threat includes mitigation guidance, these often have varying efficacies and challenges with their implementations. Mitigation tiers are intended to help device vendors/OEMs better understand how to assess the challenge of deploying mitigations and better strategize and prioritize efforts to add additional mitigations or technologies to address threats. EMB3D™ mitigations are mapped to one of three tiers, *Foundational*, *Intermediate*, and *Leading*. The alignment of EMB3D mitigations to specific Tiers acknowledges constraints with the adoption and integration of new technologies within specific devices or environments. For example, many standards or best-practice guidelines exist stating how to update firmware securely, deploy encryption on communications, or harden systems against memory corruption attacks. More novel emerging threats may not have mitigations that are as mature and well understood. There may be different methods that could be adopted to mitigate a threat, but implementing these may present different costs or challenges. For example, embedded devices often faced with size, weight, and power consumption constraints that may be limited in the type of hardware deployed. Further, devices may reside within an architecture in a way in which a certain mitigation may be challenging (such as automated software updates). Further, the tiering of mitigations is intended to evolve over time as security research expands, technology improves, and industry norms change.

Table 1 provides an overview of the key mitigation tiers and associated criteria for the placement of a mitigation in a tier. The following section will discuss the mitigation tiers in more detail.

Table 1 Mitigation Tiers and Associated Categorization Criteria

	FOUNDATIONAL	**INTERMEDIATE**	**LEADING**
Demonstrated feasibility	The mitigation is already in use on comparable embedded devices	Has been demonstrated on devices within other sectors (e.g., mobile), but may not be prevalent within comparable embedded devices	Viable proof of concept exists in any domain but may not be publicly available
Open design	Public documentation, practices, or reference architectures exist on how it can be implemented on comparable devices	Insufficient well-documented artifacts discussing implementation on embedded systems	Research demonstrating the concept of a design
Technology dependencies/ complexity	Should not require additional hardware or dependencies on integration of proprietary / commercial technology	May require additional or more capable hardware, increased software complexity, and the integration of publicly available and/or proprietary technologies	No dependency restrictions

Tier 0: Foundational - Foundation (0) tier mitigations are the minimal capability deployed to a device that provides mitigation from an associated threat. These mitigations are technologies or capabilities that have been deployed across embedded devices, demonstrating that their implementation and operations are feasible. Foundational mitigations must have well defined implementation guidelines to ensure they can be broadly adopted by devices and should include a reference to an artifact that defines an associated guidance document, reference architecture, or best-practice on how it should be implemented. These mitigations do not include additional/dedicated hardware or dependences on other proprietary or commercial technologies.

Tier 1: Intermediate - Intermediate (1) tier mitigations have been commercially adopted by systems in other domains (e.g., IT, mobile), thereby demonstrating their significant value. There may be supporting documentation and reference architectures showing how these have been adopted/deployed within these other domains, however, there lacks specific artifacts focused on embedded systems implementations. Intermediate mitigations may require a hardware or design change that requires more long-term planning, design, and testing efforts. This may also include the integration of specific proprietary technologies necessary to support the mitigation.

Tier 2: Leading - Leading (2) tier mitigations are those that are the most robust mitigation for the identified threat and include the state of novel research in that area. Leading mitigations may include a viable proof-of-concept, known implementation in a test device, or even limited deployment; however, since these mitigations have not been broadly deployed, robust implementation and technical guidance may not be available. The intent is to track these mitigations as they mature over time. Implementing this mitigation may require that the organization performs additional significant research, development, and testing.

3.3.3 ISA/IEC 62443-4-2 Mappings

ISA/IEC 62443-4-2 *Security for Industrial Automation and Control Systems: Technical Security Requirements for IACS Components*, is a heavily used industry standard that defines security controls necessary for components developed for use in ICS environments. Due to industry's heavy use of this standard, EMB3D has mapped each mitigation to the security controls defined in this standard. The intended use of this mappings is to help organizations using 62443-4-2 identify which EMB3D mitigations are necessary to fulfill the intent of the controls.

When an EMB3D mitigation maps to a 62443 control, it suggests that EMB3D mitigation is likely necessary to meeting the intent of that control (assuming the associated EMB3D threat is relevant). In many cases, a single 4-2 control is mapped to multiple EMB3D mitigations. This is because many 4-2 controls do not define specific technical security mechanisms, but instead emphasize general outcomes from the implementation of security controls. For example, the table below provides an example of where many EMB3D mitigations map to a single 62443 controls*, 3.2 Protection of Malicious Code.* This is because there are many different methods and locations that can be used to inject malicious code into a device, therefore, numerous mitigations are necessary to protect against each.

EMB3D Mitigations	62443-4-2 Controls
MID-004 Memory Hardening Against Code Injection	3.2 Protection of Malicious Code
MID-005 Memory Safe Programming Languages	
MID-007 Driver Memory Isolation	
MID-006 Control Flow Manipulation Protections	
MID-015 Process and Thread Memory Segmentation	
MID-020 ROP Gadget Minimization	
MID-021 Pointer Authentication	

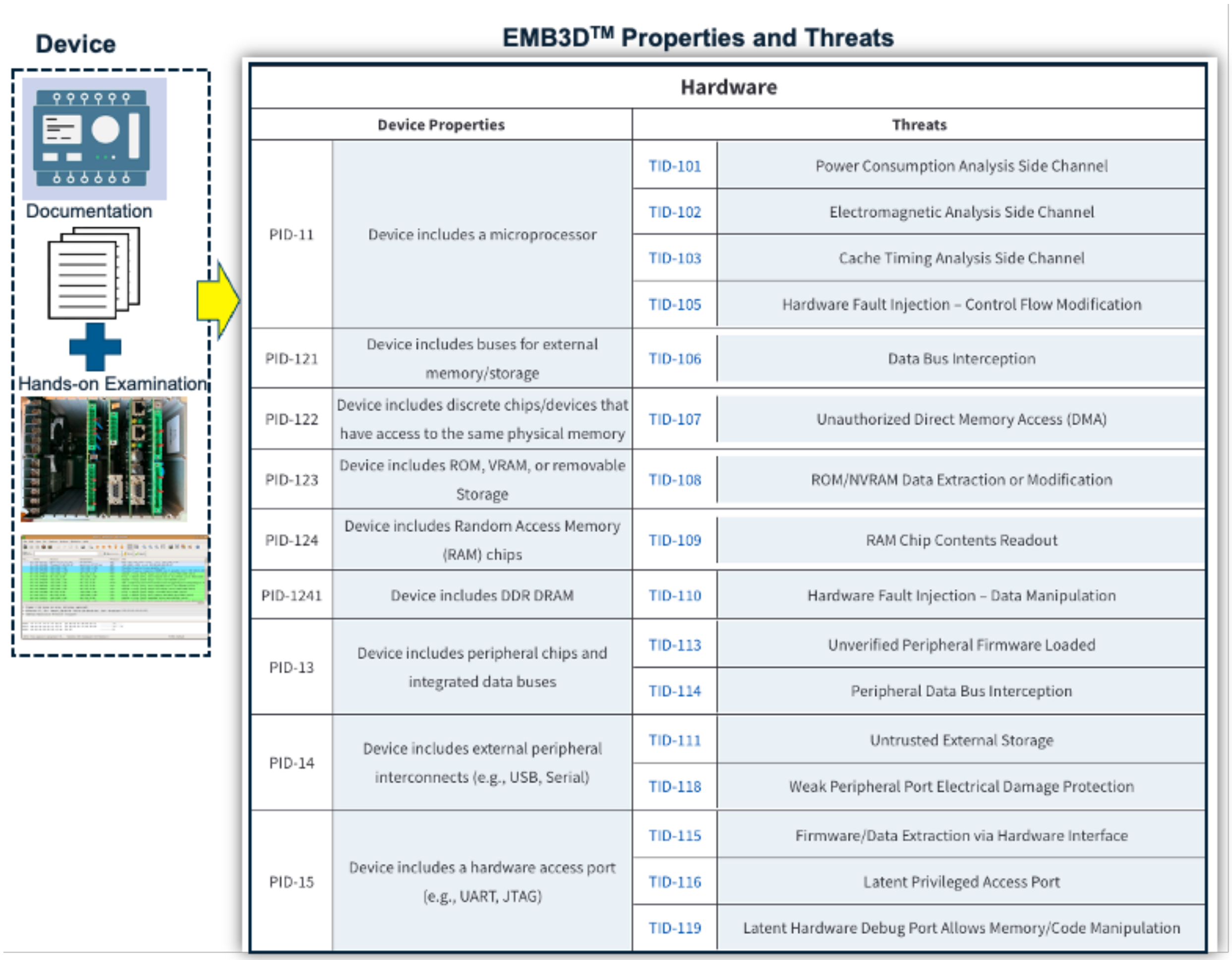
These mappings are not references to potentially relevant 62443 controls, that is, they do not suggest security controls that might need to be considered when an EMB3D mitigation is adopted. For example, many EMB3D mitigations require the deployment of different cryptographic mechanisms, while the 62443 control *4.3 - Use of Cryptography* recommends using various best practice cryptographic mechanism, this control is not mapped to EMB3D mitigations purely because they require the adoption of cryptographic mechanisms.

4 Users and Workflow

4.1 Threat Modeling Workflow

The recommended workflow for applying EMB3D™ incorporates three steps: (1) enumerate device properties and map to threats, (2) enumerate threats and evaluate their relevance/risk, and (3) map to mitigations. Each of these are described in more detail below.

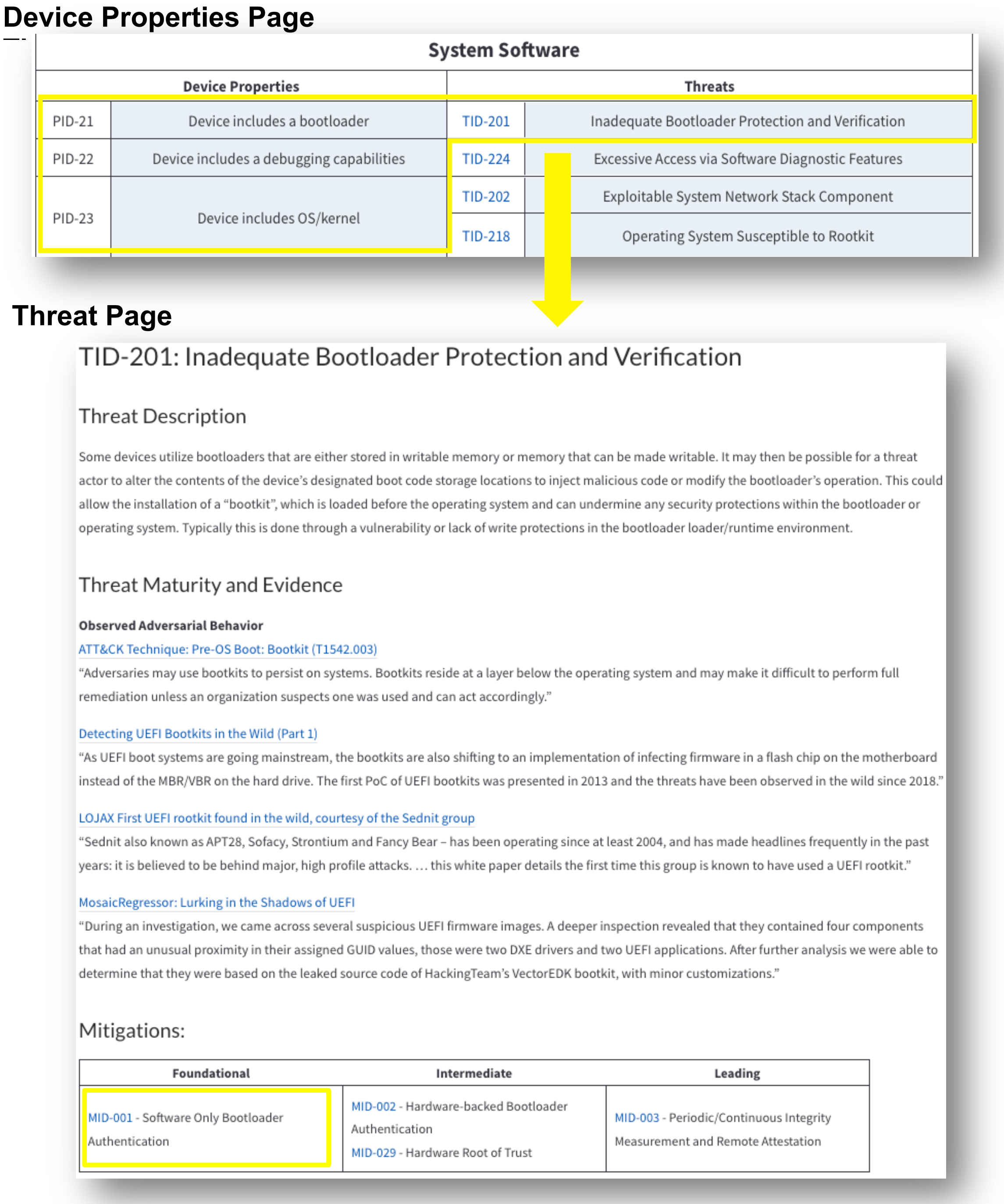
Step 1. Enumerate device properties and map to threats: The user first enumerates a set of relevant device properties based on any relevant information about the device. While a vendor may be able to fully enumerate all properties, an asset owner or security researcher may need to review available documentation or perform initial device testing or decomposition to fully enumerate the relevant properties. Figure 2 demonstrates the device properties and sub-properties associated with the device's firmware, and their mapping to specific threats. The yellow boxes are intended to demonstrate the identification of a device's relevant properties and their mapping to specific threats.



Hardware			
Device Properties		Threats	
PID-11	Device includes a microprocessor	TID-101	Power Consumption Analysis Side Channel
		TID-102	Electromagnetic Analysis Side Channel
		TID-103	Cache Timing Analysis Side Channel
		TID-105	Hardware Fault Injection – Control Flow Modification
PID-121	Device includes buses for external memory/storage	TID-106	Data Bus Interception
PID-122	Device includes discrete chips/devices that have access to the same physical memory	TID-107	Unauthorized Direct Memory Access (DMA)
PID-123	Device includes ROM, VRAM, or removable Storage	TID-108	ROM/NVRAM Data Extraction or Modification
PID-124	Device includes Random Access Memory (RAM) chips	TID-109	RAM Chip Contents Readout
PID-1241	Device includes DDR DRAM	TID-110	Hardware Fault Injection – Data Manipulation
PID-13	Device includes peripheral chips and integrated data buses	TID-113	Unverified Peripheral Firmware Loaded
		TID-114	Peripheral Data Bus Interception
PID-14	Device includes external peripheral interconnects (e.g., USB, Serial)	TID-111	Untrusted External Storage
		TID-118	Weak Peripheral Port Electrical Damage Protection
PID-15	Device includes a hardware access port (e.g., UART, JTAG)	TID-115	Firmware/Data Extraction via Hardware Interface
		TID-116	Latent Privileged Access Port
		TID-119	Latent Hardware Debug Port Allows Memory/Code Manipulation

Figure 2. Enumerating Device Properties Using EMB3D™

Step 2. Enumerate threats and evaluate their relevance/risk: After the user has defined a set of device properties, they should review each threat mapped to those properties (Figure 3). The threat description provides additional information about the threats, including the maturity level and documented threat evidence. Using this information, the user can better understand the risk of that threat and determine whether additional mitigations are warranted. Further, references to relevant CVEs and CWEs provide the user with information about a device's specific weaknesses that enable the threat.



System Software

Device Properties		Threats	
PID-21	Device includes a bootloader	TID-201	Inadequate Bootloader Protection and Verification
PID-22	Device includes a debugging capabilities	TID-224	Excessive Access via Software Diagnostic Features
PID-23	Device includes OS/kernel	TID-202	Exploitable System Network Stack Component
		TID-218	Operating System Susceptible to Rootkit

TID-201: Inadequate Bootloader Protection and Verification

Threat Description

Some devices utilize bootloaders that are either stored in writable memory or memory that can be made writable. It may then be possible for a threat actor to alter the contents of the device's designated boot code storage locations to inject malicious code or modify the bootloader's operation. This could allow the installation of a "bootkit", which is loaded before the operating system and can undermine any security protections within the bootloader or operating system. Typically this is done through a vulnerability or lack of write protections in the bootloader loader/runtime environment.

Threat Maturity and Evidence

Observed Adversarial Behavior

ATT&CK Technique: Pre-OS Boot: Bootkit (T1542.003)

"Adversaries may use bootkits to persist on systems. Bootkits reside at a layer below the operating system and may make it difficult to perform full remediation unless an organization suspects one was used and can act accordingly."

Detecting UEFI Bootkits in the Wild (Part 1)

"As UEFI boot systems are going mainstream, the bootkits are also shifting to an implementation of infecting firmware in a flash chip on the motherboard instead of the MBR/VBR on the hard drive. The first PoC of UEFI bootkits was presented in 2013 and the threats have been observed in the wild since 2018."

LOJAX First UEFI rootkit found in the wild, courtesy of the Sednit group

"Sednit also known as APT28, Sofacy, Strontium and Fancy Bear – has been operating since at least 2004, and has made headlines frequently in the past years: it is believed to be behind major, high profile attacks. … this white paper details the first time this group is known to have used a UEFI rootkit."

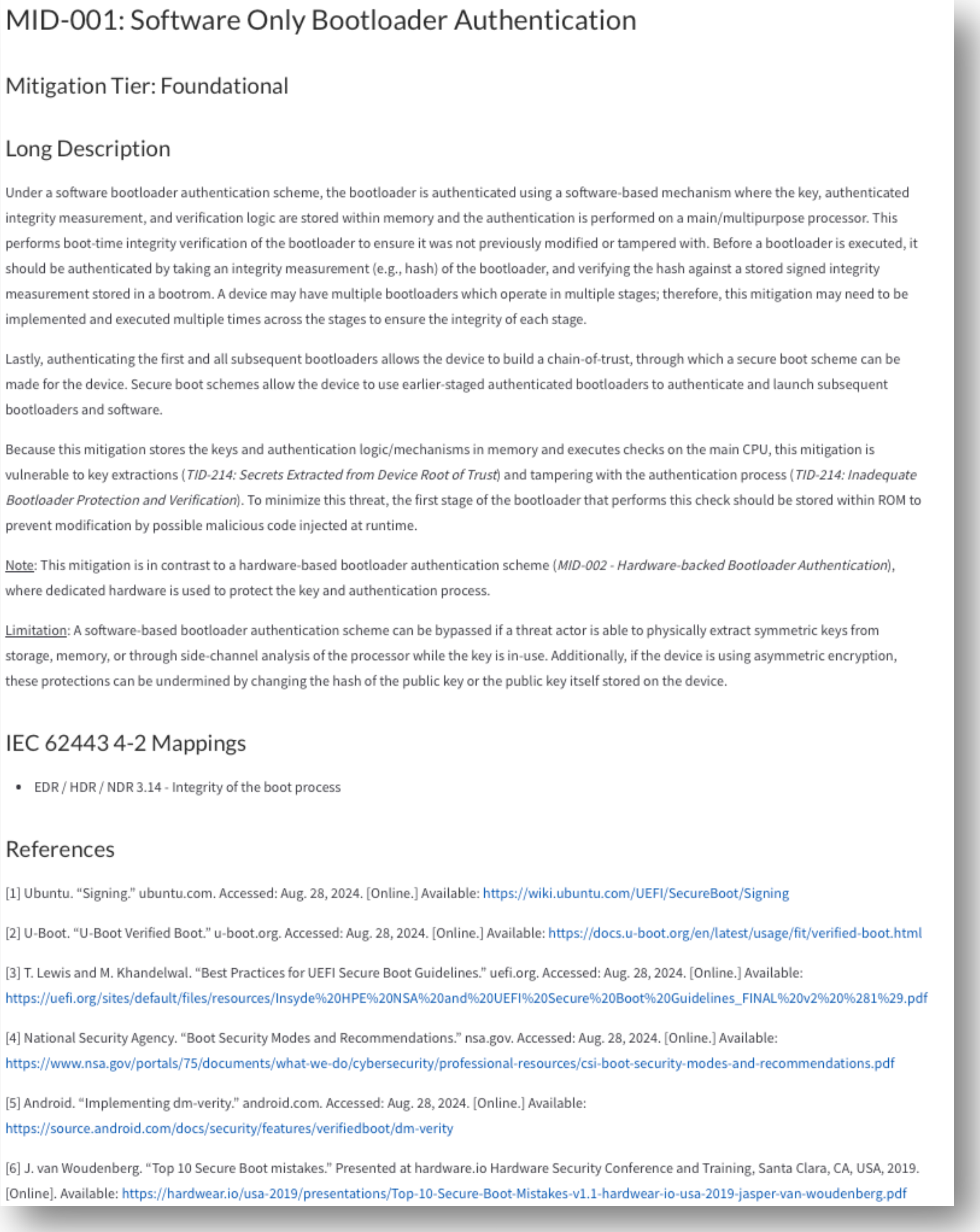
MosaicRegressor: Lurking in the Shadows of UEFI

"During an investigation, we came across several suspicious UEFI firmware images. A deeper inspection revealed that they contained four components that had an unusual proximity in their assigned GUID values, those were two DXE drivers and two UEFI applications. After further analysis we were able to determine that they were based on the leaked source code of HackingTeam's VectorEDK bootkit, with minor customizations."

Mitigations:

Foundational	Intermediate	Leading
MID-001 - Software Only Bootloader Authentication	MID-002 - Hardware-backed Bootloader Authentication MID-029 - Hardware Root of Trust	MID-003 - Periodic/Continuous Integrity Measurement and Remote Attestation

Figure 3. Mapping of Device Properties to Threats Pages



MID-001: Software Only Bootloader Authentication

Mitigation Tier: Foundational

Long Description

Under a software bootloader authentication scheme, the bootloader is authenticated using a software-based mechanism where the key, authenticated integrity measurement, and verification logic are stored within memory and the authentication is performed on a main/multipurpose processor. This performs boot-time integrity verification of the bootloader to ensure it was not previously modified or tampered with. Before a bootloader is executed, it should be authenticated by taking an integrity measurement (e.g., hash) of the bootloader, and verifying the hash against a stored signed integrity measurement stored in a bootrom. A device may have multiple bootloaders which operate in multiple stages; therefore, this mitigation may need to be implemented and executed multiple times across the stages to ensure the integrity of each stage.

Lastly, authenticating the first and all subsequent bootloaders allows the device to build a chain-of-trust, through which a secure boot scheme can be made for the device. Secure boot schemes allow the device to use earlier-staged authenticated bootloaders to authenticate and launch subsequent bootloaders and software.

Because this mitigation stores the keys and authentication logic/mechanisms in memory and executes checks on the main CPU, this mitigation is vulnerable to key extractions (*TID-214: Secrets Extracted from Device Root of Trust*) and tampering with the authentication process (*TID-214: Inadequate Bootloader Protection and Verification*). To minimize this threat, the first stage of the bootloader that performs this check should be stored within ROM to prevent modification by possible malicious code injected at runtime.

Note: This mitigation is in contrast to a hardware-based bootloader authentication scheme (*MID-002 - Hardware-backed Bootloader Authentication*), where dedicated hardware is used to protect the key and authentication process.

Limitation: A software-based bootloader authentication scheme can be bypassed if a threat actor is able to physically extract symmetric keys from storage, memory, or through side-channel analysis of the processor while the key is in-use. Additionally, if the device is using asymmetric encryption, these protections can be undermined by changing the hash of the public key or the public key itself stored on the device.

IEC 62443 4-2 Mappings

- EDR / HDR / NDR 3.14 - Integrity of the boot process

References

[1] Ubuntu. "Signing." ubuntu.com. Accessed: Aug. 28, 2024. [Online.] Available: https://wiki.ubuntu.com/UEFI/SecureBoot/Signing

[2] U-Boot. "U-Boot Verified Boot." u-boot.org. Accessed: Aug. 28, 2024. [Online.] Available: https://docs.u-boot.org/en/latest/usage/fit/verified-boot.html

[3] T. Lewis and M. Khandelwal. "Best Practices for UEFI Secure Boot Guidelines." uefi.org. Accessed: Aug. 28, 2024. [Online.] Available: https://uefi.org/sites/default/files/resources/Insyde%20HPE%20NSA%20and%20UEFI%20Secure%20Boot%20Guidelines_FINAL%20v2%20%281%29.pdf

[4] National Security Agency. "Boot Security Modes and Recommendations." nsa.gov. Accessed: Aug. 28, 2024. [Online.] Available: https://www.nsa.gov/portals/75/documents/what-we-do/cybersecurity/professional-resources/csi-boot-security-modes-and-recommendations.pdf

[5] Android. "Implementing dm-verity." android.com. Accessed: Aug. 28, 2024. [Online.] Available: https://source.android.com/docs/security/features/verifiedboot/dm-verity

[6] J. van Woudenberg. "Top 10 Secure Boot mistakes." Presented at hardware.io Hardware Security Conference and Training, Santa Clara, CA, USA, 2019. [Online]. Available: https://hardwear.io/usa-2019/presentations/Top-10-Secure-Boot-Mistakes-v1.1-hardwear-io-usa-2019-jasper-van-woudenberg.pdf

Figure 4. Mitigation page example defining the mitigation, its tier (Foundation), 62443-4-2 mapping, and associate references.

Step 3. Map to mitigations: If a threat is considered a viable risk to the device, the associated mitigations provide guidance on what technical mechanisms can best prevent or reduce the risk of that threat. As shown in Fig. 4, the mitigations include references to guidance documents and best practices, along with information about potential limitations/challenges when deploying each mitigation and associated 62443-4-2 control mappings.

4.2 EMB3D™ Use Cases for Vendors, Asset Owners, Security Researchers, and Testing Organizations

EMB3D™ was developed to support distinct roles and organizations within the security community, specifically (i) vendors product development teams, (ii) asset owner security architects, and (iii) security researchers/testing organizations. Because each of these roles may have slightly different use cases, this section highlights how EMB3D™ can be used to support each role.

4.2.1 User 1: Vendor Product Team

The design and development of a device requires numerous decisions regarding the cost, reliability, and consumer demand of various features and capabilities. EMB3D™ can be used by product security teams to better communicate to product development teams the need for certain product security investments by providing a consistent and standardized view about what threats exist and what mechanisms are needed to address them. By comparing device properties and currently implemented security mechanisms, EMB3D™ enables security and product teams to efficiently identify the strengths and potential weaknesses in a device's security posture. This can have two beneficial outcomes. First, it can help product managers, architects, and security teams work together to prioritize security-focused development to obtain more robust mitigation coverage against the most likely threats. Additionally, the model's mitigations content helps guide developers towards implementing more effective defenses and avoiding common pitfalls, altogether leading to better, more secure products. These processes are discussed more below.

4.2.1.1 Prioritization of mitigation adoption into a product line:

As discussed in Section 3.3.2, Mitigations are tiered so that lower tiers include mitigations that can be easier to integrate within a device, therefore product teams should initially prioritize the adoption of Foundational mitigations before expanding to adopt higher tiered mitigations. For many threats, the mitigations in higher tiers (Intermediate/Leading) provide additional or complementary protections to those defined in the Foundational tier, while in other cases, the higher tiered mitigations should replace those defined in lower tiers.

Complementary Mitigations Across Tiers: If the higher-tiered mitigation is complementary, the Intermediate/Leading mitigations should be deployed together with the Foundational tiered mitigations as they provide protection against different portions of the threat. Mitigations can be complementary because the mitigations address a different portion of a threat or because they mitigate the threat in a different but complementary way. Two examples complementary mitigations across Foundation and Intermediate tiers of a threat are provided below.

- *Example #1: TID-106: Data Bus Interception* - The Foundational tier (*Physically Protect Circuit Board Traces and Chip Pins [MID-069]*) is focused on restricting the threat actor's ability to gain physical access, while the Intermediate tier (*Use Highly Integrated Processors to Avoid Physical Attacks [MID-074]*) focuses on the difficult of accessing/manipulating data if physical access is somehow obtained. Since the Foundational mitigation focuses on restricting physical access and the Intermediate mitigation focus on protecting the data, these two mitigations provide complementary protection.

- *Example #2: Threat: TID-202: Exploitable System Network Stack Component* - The Foundational tier (*Memory Safe Programming Languages [MID-005]*) focuses on restricting unauthorized access and execution of memory, while the Intermediate tier (*Control Flow Manipulation Protections [MID-006]*) focuses on preventing the threat actor's ability to influence the sequence of existing memory instruction execution. Since the Foundational and Intermediate mitigations prevent against memory-based exploitation in different ways, they should be deployed in a complementary way.

Higher Tier Mitigations Replace Lower Tiers: In many other cases, higher-tiered mitigations (Intermediate/Leading) provide the same functions or threat coverage as a lower-tiered mitigation and therefore can be deployed as a substitute for the Foundational one. An example is provided below:

- *Example #1: TID-201: Inadequate Bootloader Protection and Verification* – The Foundational tier (*Software Bootloader Authentication [MID-001]*) focuses on software mechanisms that performs the authentication of the bootloader, while the Intermediate tier (*Hardware-backed Bootloader Authentication [MID-002]*) performs a similar function but leveraging a more secure hardware-based root-of-trust. If a device can integrate a hardware root-of-trust to achieve MID-002, there is no value in maintaining the software-based mechanisms (MID-001). Therefore MID-002 supersedes, rather than complements, MID-001.

Mitigations do not directly state whether they should be used in a complementary way with others, or as a replacement. Instead, it is assumed the user reviews and assesses the value of higher-tiered mitigations in relations to existing lower-tiered mitigations. However, this also suggests that organizations should avoid immediately adopting Intermediate or Leading tier mitigations without ensuring that the Foundation tier is also adopted, or fully superseded, as it may expose the device to portions of the risk not addressed by the higher tiered mitigation.

4.2.1.2 Strategic planning

EMB3D provides a roadmap for how organizations should broaden their capabilities and expertise into various technologies necessary to adopt more advanced Intermediate and Leading capabilities. Organizations should prioritize the investment and time needed to design, integrate, and deploy pertinent mitigations. Mitigations at these higher tiers will often require hardware changes and updates to product hardware roadmaps. They may require internal programs and teams to develop and test prototypes of mitigation implementations before the can be integrated into specific product lines.

4.2.2 User 2: Asset Owner Security Architect

Asset owners often struggle to assess the risks associated with using a device and are thereby challenged to specify the expected device security capabilities during an acquisition. When a device lacks adequate security, the asset owner inherits the responsibility of understanding the resulting risk and designing compensating controls into their broader architecture. It is also a challenge for an asset owner to identify what security testing should be performed on the device to ensure security capabilities have been completely and correctly implemented.

Asset owners can leverage EMB3D™ to hold vendors more accountable for risk by requiring (i) detailed mappings of device properties, and (ii) documentation of the mitigations that have been built into the device to address the threats associated with those device properties. Requirements for vendors to provide this device data can be included in procurement language to ensure asset owners can more directly evaluate threats associated with those properties. Further, asset owners can use EMB3D™ to scope security testing efforts to better assess device risks and protections. EMB3D™ provides a common language and framework for asset owners to communicate to vendors regarding device security requirements, what threats need to be addressed, and the necessary mitigations for those threats.

4.2.3 User 3: Security Researcher/Testing Organization

Security researchers and testing organizations can use EMB3D™ to develop a more standard framework and methodology to assess devices based on a common language to describe device properties, threats, and mitigations. The framework can aid in the development of test plans to help provide clear guidelines on what threats and mitigation testing activities will focus on, and which are out of scope for a specific effort. Further, by mapping test outcomes or findings to EMB3D™ threats, a finding's severity and credibility can be more clearly communicated.

5 Conclusion

This paper introduces the EMB3D™ threat model for critical infrastructure embedded devices. The threat model provides a cultivated knowledge base of known cyber threats to devices, including those observed in the wild and demonstrated through proof-of-concept and theoretic research. The threats are mapped to device properties to help users develop and tailor accurate threat models for specific embedded devices. Each threat is also associated with a set of mitigations that should be adopted to reduce the risk of these threats to a device. EMB3D™ is intended to provide a common language that can be adopted by device vendors, asset owners, security researchers, and testing organizations to (i) consistently assess, understand, and prioritize threats to devices; (ii) scope device assessments and testing activities; and (iii) communicate findings and associated risks.

Appendix A Terminology, Challenges, Scope, and Related Efforts

This appendix provides additional information on the scope and methodology associated with building the EMB3D™ threat model as well as documenting some of the challenges associated with its development.

A.1 Terminology

While terms such as threat, threat model, weakness, and vulnerability are used heavily within the cybersecurity field, they often have imprecise or conflicting definitions. This section will define how these terms are used within EMB3D™.

- **Threats:** An action or set of actions that a threat actor may use to achieve some objective or effect on a system or device.
- **Threat Model:** A set of threats that have been identified to be relevant to a specific system or device.
- **Weakness:** "A condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities."[7]
- **Vulnerability:** "A flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components."[8]

Threats are closely related to weaknesses and vulnerabilities in that weaknesses/vulnerabilities do not present any risk in the absence of a threat. While threats describe the actions necessary for the actor to achieve an effect on a device, weaknesses and vulnerabilities define the technical limitations and means that enable that threat behavior. While weakness enumerations (e.g., CWE) provide significant value to the identification and mitigation of weaknesses in systems, understanding whether and how a threat can potentially target a weakness is important to understanding that weakness and determining its priority for mitigation. Further, while threats target documented weaknesses, such as CWEs, there are also numerous examples where threat actors target normal system functions to achieve their objective, including mechanisms to discover devices[9] or to leverage existing services to gain access.[10] While these threats may not always have practical or effective mitigations, understanding how a threat actor may target and interact with a device is still an important factor in the overall understanding and assessment of risks.

[7] CWE Glossary. URL: https://cwe.mitre.org/documents/glossary/index.html

[8] CWE Glossary. URL: https://cwe.mitre.org/documents/glossary/index.html

[9] Remote System Information Discovery. MITRE ATT&CK. URL: https://attack.mitre.org/techniques/T0888/

[10] Remote Services. MITRE ATT&CK. URL: https://attack.mitre.org/techniques/T0886/

A.2 Threat Maturity

Understanding the credibility and maturity of a threat is critical to determining whether it should be included in a threat model. While knowledge bases such as ATT&CK document observed adversarial techniques, many other threats have been observed only in laboratory/testbed environments or have only theoretic explanations. Due to the long lifespan of embedded devices and the shifting threat landscape, devices must be designed to be secure against both currently observed and expected future threats. To help distinguish the creditability of a threat, each is assigned one of four categories: (i) Observed Adversarial Technique, (ii) Known Exploitable Weakness, (iii) Proof of Concept, and (iv) Theoretic. Each category is defined in more detail below.

Observed Adversarial Technique: This type of threat has been observed in use by cyber adversaries to target embedded devices or critical infrastructure environments. This requires a reference to observed adversarial use of the threat, such as an ATT&CK Procedure Example, or reference to a reputable CTI report documenting the observed use. It can also include information about the vulnerabilities in devices (e.g., CVEs) or general categories of vulnerabilities (e.g., CWEs).

Known Exploitable Weakness: This includes well-defined weaknesses, specifically those defined by CWEs, that have been demonstrated to be exploitable by threat actors in other non-embedded environments (e.g., Enterprise, Mobile). This category is used to ensure threats that have been broadly observed targeting weaknesses within other environments are equally prioritized. Meeting this maturity level assumes the threat is known to target a technology-specific CWE (either at the Base or Variant abstraction), and that there is evidence that CWE has shown to be exploitable. One potential example for evidence of exploitation is a mapping to a CVE that is known to be exploited by its inclusion within the DHS Known Exploited Vulnerabilities (KEV) Catalog.[11]

Proof of Concept: A proof-of-concept threat requires evidence that it has been demonstrated under sufficiently realistic assumptions to validate its effectiveness. For a threat to be considered a viable proof of concept, it requires the following:

1. Demonstration that the threat can have its intended effect/impact on a realistic target (e.g., software, device, network),
2. Demonstration that the information required for the threat's execution can be obtained,
3. Demonstration that necessary environmental assumptions, human interactions, or corresponding events are reasonable,
4. Availability of defensible materials to further prove the concept (e.g., video recording, Github publishing, full documentation of the work completed and its effects).

Theoretic: This type of threat has not yet been sufficiently demonstrated to verify its effectiveness against real-world environments. The threat may be dependent on insufficiently validated assumptions regarding the reliability of a technical exploitation mechanism, or a threat

[11] Known Exploited Vulnerabilities Catalog. CISA. URL: https://www.cisa.gov/known-exploited-vulnerabilities-catalog

actor's computing resources, level of access, information, or operational capabilities. Strong supporting reference material, such as peer-reviewed research reports documenting the threat activity, is necessary to address factors such as those listed above and to contextualize it in relation to other vulnerability and weakness categories. Examples include:

- Threats with ambiguous/hypothetical computing resources. An example could be assumptions that the threat actor has a functional quantum computer and therefore can break asymmetric encryption keys.[12]
- Threats with unclear assumptions about the required adversarial access, information, or algorithmic complexity necessary for execution. For example, there is a wide number of academic efforts discussing False Data Injection Attacks on theoretic controls or sensor models[13] that lack demonstration on real networks/software platforms.
- Threats lacking demonstrated/reliable exploitation mechanisms. These threats could include exploits that inconsistently work but have been proved to be feasible under specifically scoped environments/testing set-ups.

A.3 Key Challenges

Developing a threat model that is broadly useable for a wide-ranging set of organizations—including device vendors, security researchers/testing organizations, and integrators/asset owners—presents multiple challenges, including:

- **Identifying Relevant Threats:** Information about threats to devices can be found across a diverse set of sources. Existing knowledge bases such as MITRE ATT&CK[14] include information about known adversary behavior, while platforms such as CVE, CISA ICS Advisories,[15] and CWE define weaknesses and vulnerabilities within hardware and software platforms. Additionally, there exists a wide range of theoretic and proof-of-concept vulnerability research efforts that have been demonstrated by security researchers and are documented through various security conferences, papers, blogs, and other sources.
- **Understanding the Evolving Threat Landscape:** Threats continually evolve, both in the set of techniques they utilize and in the environments they target. Security researchers are continually identifying new methods by which systems can be manipulated. These often begin as theoretical ideas, which often lead to proof-of-concept demonstrations, and then may eventually be adopted and used by threat actors. Further, threat techniques that are identified in certain sectors or devices, such as traditional IT environments, may be adopted and tailored to critical infrastructure sectors. Therefore, the model must include

[12] Ford, Pete. "The quantum cybersecurity threat may arrive sooner than you think." Computer 56.2 (2023): 134-136.

[13] Mo, Yilin, and Bruno Sinopoli. "False data injection attacks in control systems." Preprints of the 1st workshop on Secure Control Systems. Vol. 1. 2010.

[14] MITRE ATT&CK: URL: https://attack.mitre.org

[15] Cybersecurity Alerts & Advisories. URL: https://www.cisa.gov/news-events/cybersecurity-advisories

threats of varying maturity levels and across the diverse sectors where the threats have been observed.

- **Scoping/Prioritizing Threats:** There is a range of potential threats a device could be vulnerable to, ranging from those in active use by threat actors to theoretic threats that are currently not technically possible. Further, threats are typically aligned to specific technical properties or architectures, which may have varying prevalence across embedded devices. Therefore, the threat model must bound the included threats based on the (i) credibility of the threat, (ii) technologies/architectures for which threats are identified, and (iii) level of detail regarding the threat such that it is sufficiently applicable to a broad set of embedded devices.
- **Identifying Commonalities with Embedded Devices and Properties:** Devices used across the critical infrastructure embedded technology domain are very diverse. The threat model must scope and abstract the large set of technologies and device features down to a manageable number of unique properties. In many cases, this will also require abstracting threats—for example, identifying and combining general threats to network protocols in general, rather than defining individual threats to the large set of specific protocols or implementations.
- **Identifying Mitigations and Countermeasures:** Where practical, the threat model identifies mitigations and countermeasures against enumerated threats. The mitigations and countermeasures proposed in the model are not intended to be used as a definitive security checklist or one-size-fits-all approach, as devices with unique architectures, implementations, and constraints may necessitate customized mitigations and countermeasures. Additionally, not all mitigations are equally effective and may leave residual risks that still constitute a threat to the device. For example, encrypting/authenticating network traffic will mitigate certain threats, but introduce additional threats associated with known weaknesses in cryptographic mechanisms.

A.4 Scoping

Threat scoping is provided below and is intended to inform users on where EMB3D™ should be used, what is included in the threat model, and what is specifically excluded. Key elements of scope are enumerated below:

- **No Network Architecture or Sector Operational Properties:** The threats within EMB3D™ are mapped to specific properties and features of a specific device. They do not include any context or reference to the device's position within a specific network architecture or its specific operational functions within an organization. This is because devices are often used across a diverse set of architectures or operational roles, which inhibits the development of a complete and accurate enumeration of these uses for a device.
- **Focused on Atomic Threat Techniques Rather than Exploit Chaining:** The threats within the model are scoped to events that are used to bypass the confidentiality, integrity, or availability of some property of a device. EMB3D™ does not explicitly model threats that consist of a sequence of events that must be chained together to achieve a specific effect on a device. For example, a threat actor may be able to reverse

engineer a device to gain a valid authentication key and then use that key to remotely access a device. Within EMB3D™, this scenario would be categorized as two discrete threats.

- **Access to Devices:** The threat actor is considered to have physical access to instances of the device that can be used to conduct vulnerability research and exploit development prior to execution of an attack, including performing reconnaissance and reverse engineering.
- **Relevance to Embedded Devices:** The threat model is focused on embedded devices; therefore, the properties and features have been scoped to these devices. Embedded devices include a broad set of unique technologies used within embedded applications, along with dependencies on more diverse technologies used across general computing platforms/environments. Therefore, the scope will include both categories, as defined in the following table.

	Properties of Defined Technologies and Platforms
In Scope	Unique to critical infrastructure embedded equipment and environments, such as the execution of custom logic/programs (e.g., IEC 61131)
	Frequently observed in embedded environments due to foundational use across computing/networking, including protocols (e.g., HTTP, TCP/IP, HTTP, SQL), hardware (e.g., USB, JTAG, FPGA), and software (e.g., operating systems functions, firmware, memory management)
Out of Scope	Minimal/no documented use in embedded platforms such as cloud or mobile platforms
	Specific to vendors or product lines, including specific design technologies by Microsoft or specific embedded device vendors

A.5 Related Work

While there has been substantial previous work developing methodologies and knowledge bases to help organizations model threats against software and devices, EMB3D™ provides a unique approach that builds on and leverages several other related efforts. This section reviews related efforts and how they inform the EMB3D™ approach.

- MITRE ATT&CK® is a knowledge base of observed adversary behavior that categorizes the associated tactic, technique, and procedure examples. EMB3D builds off these observed adversary TTPs and includes theoretical and proof of concept threats to embedded devices.
- STRIDE is a threat modeling methodology that that can be broadly adapted to different environments or systems. It leverages data flow diagrams of a system/device to enumerate potential threats from Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, or Escalation of Privileges. EMB3D complements this theoretical approach by incorporating specific technical artifacts to ensure relevant threats can be identified to specific device properties, especially prioritizing threats that have more credible exploitations.

- CWE™ is an enumeration of common weaknesses, which are defined as “a condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities”. The weaknesses defined within CWE are general categories of weaknesses observed across broad computing environments. CWE’s are categorized at 4 different Abstractions:

  Pillar – “the most abstract type of weakness and represents a theme for all class/base/variant weaknesses related to it”,

  Class – “a weakness also described in a very abstract fashion, typically independent of any specific language or technology”,

  Base – “a weakness that is still mostly independent of a resource or technology, but with sufficient details to provide specific methods for detection and prevention,”

  Variant – “a weakness that is linked to a certain type of product, typically involving a specific language or technology.”

- EMB3D connects these weaknesses to threat actions and maps them to specific device properties. CAPEC™ is a dictionary of observed and theoretical patterns of attack employed to exploit known software weaknesses. EMB3D leverages similar concepts of observed and theoretical attack patterns and focuses on mapping to embedded devices.