Pacific Northwest
NATIONAL LABORATORY

HPNNL-29991

# ACE IoT Solutions Threat Profile

## Provided by Secure Software Central

June 2020

Ryan Bays
Taylor Edwards
Emma McMahon
Patrick O'Connell
Garret Seppala
Torri Simmons
Sarah Sundgren
Chance Younkin

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Click here to enter a date.

**DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
*operated by*
BATTELLE
*for the*
UNITED STATES DEPARTMENT OF ENERGY
*under Contract DE-AC05-76RL01830*

**Printed in the United States of America**

**Available to DOE and DOE contractors from the**
**Office of Scientific and Technical Information,**
**P.O. Box 62, Oak Ridge, TN 37831-0062;**
**ph: (865) 576-8401**
**fax: (865) 576-5728**
**email: reports@adonis.osti.gov**

**Available to the public from the National Technical Information Service**
**5301 Shawnee Rd., Alexandria, VA 22312**
**ph: (800) 553-NTIS (6847)**
**email: orders@ntis.gov <https://www.ntis.gov/about>**
**Online ordering: http://www.ntis.gov**

Click here to enter text.

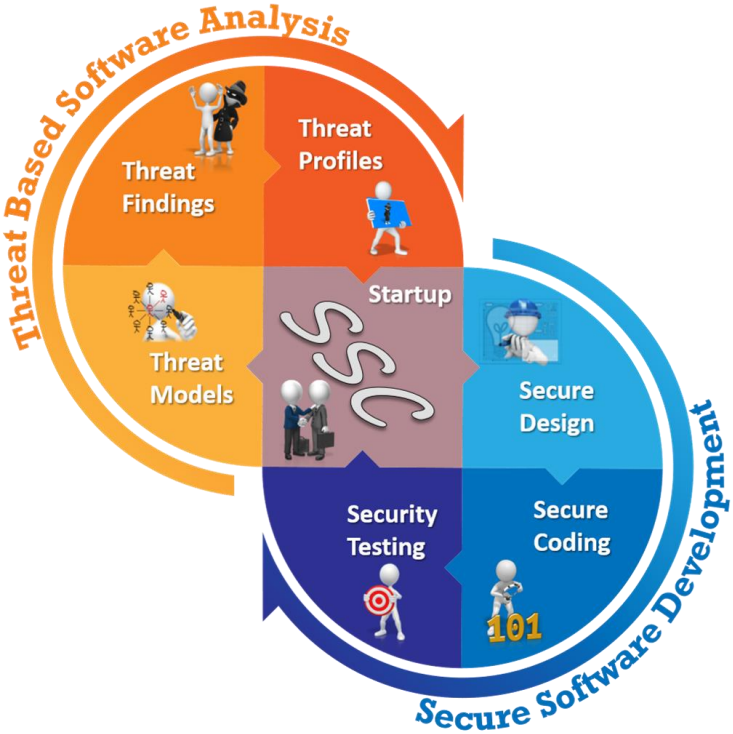
PNNL-29991

# ACE IoT Solutions Threat Profile

Provided by Secure Software Central

June 2020

Ryan Bays
Taylor Edwards
Emma McMahon
Patrick O'Connell
Garret Seppala
Torri Simmons
Sarah Sundgren
Chance Younkin




Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

# Contents

Contents ..... ii
Acronyms and Abbreviations ..... iii
Summary ..... iv
1.0 Introduction ..... 1
1.1 Purpose of the Threat Profile ..... 1
1.2 Categorizing and Prioritizing Threats ..... 2
1.3 Types of Mitigation ..... 2
2.0 Threat Model ..... 3
2.1 Production Environment Issues ..... 3
2.2 Threat Diagrams ..... 4
2.2.1 Understanding Trust Boundaries ..... 4
2.2.2 ACE IoT Threat Diagrams ..... 4
3.0 Threat Profile Table ..... 6
3.1 Interpreting the Labels ..... 6
3.2 The Detailed Threat Profile Table ..... 6
4.0 Conclusion ..... 13
Appendix A Brief on Threat Based Analysis ..... A.1
Appendix B Brief on Secure Software Development ..... B.1

# Diagrams

Diagram 1. VOLTTRON deployment for ACE IoT. ..... 5

# Figures

Figure 1. Secure Software Central services. ..... 1
Figure 2. Microsoft's STRIDE model described. ..... 2
Figure 3. ACE IoT priorities. ..... 2
Figure 4. Trust boundaries defined. ..... 4
Figure 5. The TBA half of SSC. ..... A.1
Figure 6. Lockheed Martin's methodology. ..... A.1
Figure 7. The CIA triad. ..... A.1
Figure 8. The SSD half of SSC. ..... B.1

# Tables

Table 1. Threat Profile table. ..... 6

# Acronyms and Abbreviations

| | |
|---|---|
| API | application programmer interface |
| AWS | Amazon Web Services |
| BAS | building automation system |
| BMS | building management system |
| CIA | Confidentiality, Integrity, and Availability |
| HMI | human–machine interface |
| HTTPS | Hypertext Transfer Protocol Secure |
| IDDIL-ATC | Identify Assets, Define the Attack Surface, Decompose the System, Identify Attack Vectors, List the Threat Actors, Analysis & Assessment, Triage, Controls |
| IoT | Internet of Things |
| OS | operating system |
| OSA | Open Source Analysis |
| PNNL | Pacific Northwest National Laboratory |
| PoLP | Principle of Least Privilege |
| RBAC | Role Based Access Control |
| SAST | Static Analysis Security Testing |
| SSC | Secure Software Central |
| SSD | Secure Software Development |
| TBA | Threat Based Analysis |
| SSH | secure shell |
| STRIDE | Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege |
| SQL | Structured Query Language |
| TMT | Threat Modeling Tool |
| TPM | Trusted Platform Module |
| UI | user interface |
| VM | virtual machine |

# Summary

The ACE IoT Solutions LLC (hereafter “ACE IoT”) and Pacific Northwest National Laboratory’s (PNNL) VOLTTRON™ team have engaged with PNNL’s Secure Software Central (SSC) Team to produce this Threat Profile. The Threat Profile gives ACE IoT, a software company based in Chattanooga, TN, the means to understand the potential threats against ACE IoT’s deployment of the Eclipse VOLTTRON platform. The objective is to provide the knowledge needed to mitigate or accept threats based on the impact those threats have on the system. Not all threats must be mitigated, and not all threats can be addressed in a cost-effective way. This Threat Profile provides critical information for making threat-based decisions to increase security at a reasonable cost and to reduce risk.

The ACE IoT Threat Profile establishes security requirements, justifies security measures, yields actionable controls, and effectively communicates risk to stakeholders. To that end, it can be effectively used by development teams, software architects, and managers.

Through the Threat Profile, threats to the system were categorized, prioritized, and directly mapped to affected system components. The Threat Profile shows mitigations that were already addressed at the time of engagement, as well as those that could be addressed or considered acceptable as-is.

Use cases and threat diagrams were created through engagement between SSC and ACE IoT. An SSC analysis followed, producing a Threat Findings document. Follow-on engagements with ACE IoT led to the full Threat Profile, which details threat type, threat category, and mitigations for every threat identified. Finally, this detailed table was summarized in a mitigations table that prioritizes and lists every mitigation, implemented or not, with references to the full Threat Profile. The mitigations map to the threats, which map to components in the diagrams, providing complete coverage of the system from a threat analysis perspective.

This Threat Profile provides the foundation for a thorough understanding of threats for the development team, the testing team, management, stakeholders, and customers of ACE IoT. It enables decision makers at all levels to improve the security posture of the system. This effort leads to more secure software and better-understood security; the ACE IoT team is to be commended for their rigorous approach to employing cybersecurity throughout the software development life cycle.

The results of this assessment are summarized in the table below.

| Threat Type | High Priority | Medium Priority | Low Priority | Total |
|---|---|---|---|---|
| Spoofing | 3 | 5 | 0 | 8 |
| Tampering | 12 | 2 | 1 | 15 |
| Repudiation | 0 | 0 | 2 | 2 |
| Information Disclosure | 1 | 2 | 5 | 8 |
| Denial of Service | 1 | 1 | 0 | 2 |
| Elevation of Privilege | 6 | 7 | 0 | 13 |

# 1.0 Introduction

Pacific Northwest National Laboratory (PNNL), with funding from the U.S. Department of Energy's Building Technologies Office, developed and maintains VOLTTRON as an open-source community project. VOLTTRON includes agent execution software; agents that perform critical services that enable and enhance VOLTTRON functionality; and numerous agents that utilize the platform to perform a specific function (fault detection, demand response, etc.). VOLTTRON supports energy, operational, and financial transactions between networked entities (equipment, organizations, buildings, grid, etc.) and enhances the control infrastructure of existing buildings through the use of open-source device communication, control protocols, and integrated analytics.

ACE IoT Solutions LLC (hereafter "ACE IoT") is a private commercial company that leverages open-source technologies, including VOLTTRON. They provide customers with low-cost approaches to acquire, access, and manage data from distributed control systems and sensors. ACE IoT offers Infrastructure as a Service, which enables customers to remotely access data from networks of connected Internet of Things (IoT) devices, including buildings with building automation systems.

The ACE IoT team is engaged with PNNL's Secure Software Central (SSC) Team to provide cybersecurity analyses of ACE IoT's deployment of the Eclipse VOLTTRON platform. SSC offers both threat-based analysis services and secure software development services, as defined in Figure 1. These services are ultimately used to understand and mitigate threats against software and to reduce vulnerabilities in software, thus improving overall cybersecurity and informing decision makers. SSC's threat-based analysis produced this document, a Threat Profile, which is composed of threat model diagrams, threat findings, and most importantly, controls that mitigate those threats.

***Threat-Based Software Analysis*** **–** determines and prioritizes threats against the software system and recommends mitigations. The result is a Threat Profile that contains a threat model, threat findings, and mitigations.

***Secure Software Development*** **–** applies security best practices to the software development life cycle. This includes secure design, secure code review, vulnerability scanning, and security testing.

Figure 1. Secure Software Central services.

## 1.1 Purpose of the Threat Profile

The Threat Profile establishes security requirements, justifies security measures, yields actionable controls, and effectively communicates risk. To that end, it can be used effectively by development teams, software architects, managers, and stakeholders. For stakeholders and managers, the Threat Profile shows what has been mitigated and what has not been mitigated, thus enabling decision makers to assess priorities in terms of the actual system and the threats against it. For development teams and software architects, the Threat Profile provides direct and actionable tasking that boosts the cybersecurity of the software product. In addition to providing information, the format of the Threat Profile maps mitigations to threats and threats to the diagram, making it clear where and how the controls are affecting and benefiting the system. This is advantageous for controls and vulnerability assessments that are not threat based and do not stem from system diagrams.

## 1.2 Categorizing and Prioritizing Threats

Categorizing threats helps identify, organize, and prioritize threats in any system—this holds true for the ACE IoT software that is being developed. To optimize the analysis process, streamline the engagements, and aid in mitigation implementations, SSC utilizes Microsoft's STRIDE model (see Figure 2). There are many categorization models, but STRIDE best lends itself to PNNL's processes, and tools are available to partially automate and expedite the initial analysis processes. SSC uses Microsoft's Threat Modeling Tool, which is based on the STRIDE model. The tool provides initial results, and SSC provides expertise to consolidate the threats.

***S***poofing **–** when a process, file, website, network address, etc. is not what it claims to be
***T***ampering – the act of altering the bits in a running process, data in storage, or data in transit
***R***epudiation – involves an adversary denying that something happened
***I***nformation Disclosure – when the information can be read by an unauthorized party
***D***enial of Service – when the process or data store is unable to service incoming requests
***E***levation of Privilege – when an adversary gains increased capability on a system or network

Figure 2. Microsoft's STRIDE model described.

Prioritizing threats is also critical to the process of developing a Threat Profile. With a list of mitigations, each with their own cost, level of effort, and consequences, it is necessary to understand which ones are most important and why. For a Threat Profile, priorities start with the standard CIA (Confidentiality, Integrity, and Availability) Triad, as used in Figure 3. The terms are defined simplistically as follows:

***Confidentiality –*** keep the data secret.
***Integrity –*** make sure the data is correct.
***Availability –*** make the data available.


Figure 3. ACE IoT priorities.

These terms are important considerations when prioritizing threats, but using the triad necessitates that one of the three must be ranked as the most important. Figure 3 shows the ACE IoT priorities for this Threat Profile.

## 1.3 Types of Mitigation

Mitigations identified in this Threat Profile fall into three categories:

***Physical*** – This is the traditional type of security in which valuable assets are guarded with guns, guards, and gates. However, physical security is becoming blended with cybersecurity in many ways because computers and networks are linked with gates, locks, and other access protection devices.
***Technical*** – This refers to cybersecurity technology that is applied to typically (but not always) digital assets. Multi-factor authentication is a good example of a technical mitigation for access control.
***Operational/Administrative*** – This is a method of following policy or procedural practices to implement security.

These three types are not identified directly in the Threat Profile, but most mitigations fall into the technical category, although both physical and operational mitigations can and do occur.

# 2.0 Threat Model

An SSC threat model is a set of use cases, a set of threat cases, and a set of system diagrams. Use cases are descriptions of how the system operates from a user's viewpoint. They are invaluable for deriving system diagrams, which are the framework for the Threat Findings and the Threat Profile documents. Threat cases are just like use cases but from the perspective of an adversary. Threat cases are used primarily to help derive and understand mitigations.

ACE IoT wishes to understand and explain how VOLTTRON is secure and remains available in a potentially hostile environment and/or understand the consequences and mitigations in a hostile environment. To that end, the production environment and specific use cases should be considered. Each are briefly outlined below.

## 2.1 Production Environment Issues

A Threat Profile depends on the environment in which the software is deployed. For VOLTTRON deployments in the ACE IoT use case, there are options to consider that may be important factors. This Threat Profile focuses on an open network environment. Other aspects of being on an open network are briefly described below.

***Open network –*** While much attention is paid to the security of the VOLTTRON software, it is important to consider that its production environment may be completely unprotected with compromised non-VOLTTRON systems.

***External hardware –*** ACE IoT customers will likely add hardware to the same network on which VOLTTRON is deployed. This exposes VOLTTRON to potential threats that should be considered.

***Modem interfaces –*** Cell modem, OT interfaces, or both could be present in an open network environment on which ACE IoT deploys the VOLTTRON platform. The presence of these interfaces exposes VOLTTRON to potential threats that should be considered.  Use Cases

***Facilities Manager & Building Engineer –*** These people log into a human–Machine interface (HMI) workstation within the building automation system (BAS), the building management system (BMS), or both.  They interact with the system either to gain situation awareness or to provide control over the BMS.

***System Administrator –*** This person accesses the Orchestrator, to establish configuration and to perform system administration for the Virtual Private Cloud Environment and its interaction with VOLTTRON Central.

***ACE IoT admin –*** This person logs into the virtual private network gateway, to perform system administration for VOLTTRON Central.

***WWW –*** This person logs into interact with the NGINX web server.

## 2.2 Threat Diagrams

The diagrams in this section represent ACE IoT's VOTTRON system and were derived from engagements between the SSC team, the ACE IoT team, and the PNNL VOLTTRON team. They contain some assumptions that are based on a mutual understanding about how the system will be designed and implemented.

### 2.2.1 Understanding Trust Boundaries

The most important aspect of doing threat-based analysis is knowing what trust boundaries exist and where they are located. Interactions that cross trust boundaries are the most likely place for an adversary to inflict damage on a system. Figure 4 shows the hierarchy of ACE IoT's VOLTTRON trust boundaries and explains what and where they are. The hierarchy of trust boundaries depicted in Figure 4 is maintained in all ACE IoT threat diagrams.

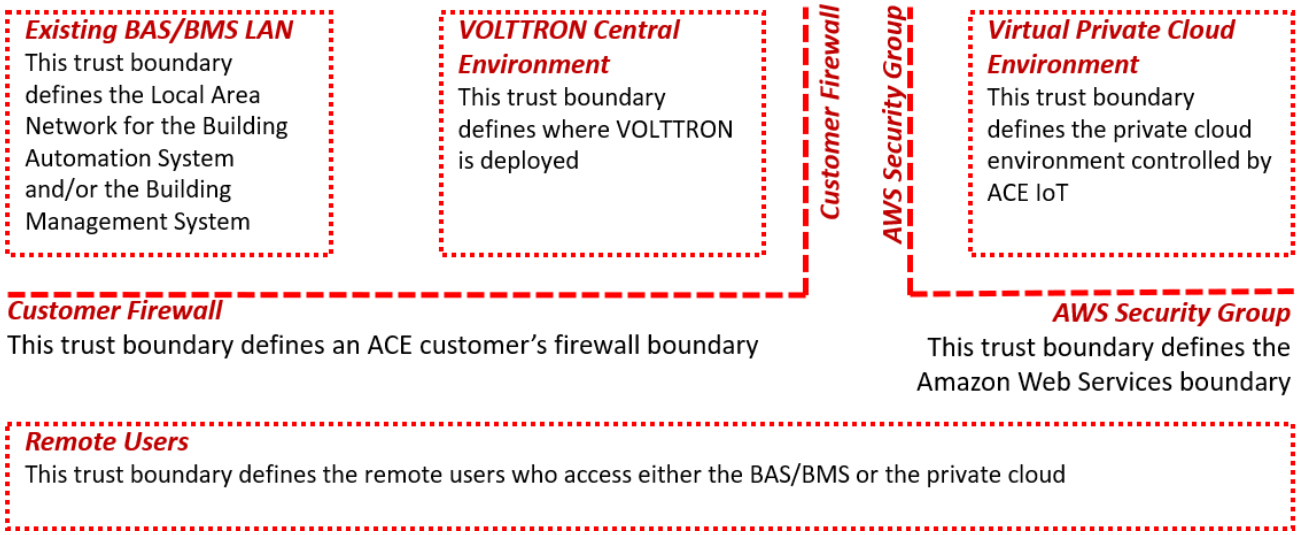



Figure 4. Trust boundaries defined.

### 2.2.2 ACE IoT Threat Diagrams

The conventions used in the diagram help distinguish and categorize the different components of the system as follows:

***Circles –*** these represent running processes or people interacting with system components.

***Squares –*** these represent physical devices or storage devices within the system.

***Arrows –*** these represent interactions between components within the system or between a person and a component. Interactions (arrows) are labeled so that they can be identified in the Threat Profile table (Table 1), which features mitigations that map directly to the interactions within the system.

***Red dotted boxes –*** these represent trust boundaries between components of the system.

***Red dashed lines –*** these represent internet boundaries between ACE IoT's VOLTTRON deployment and external components.

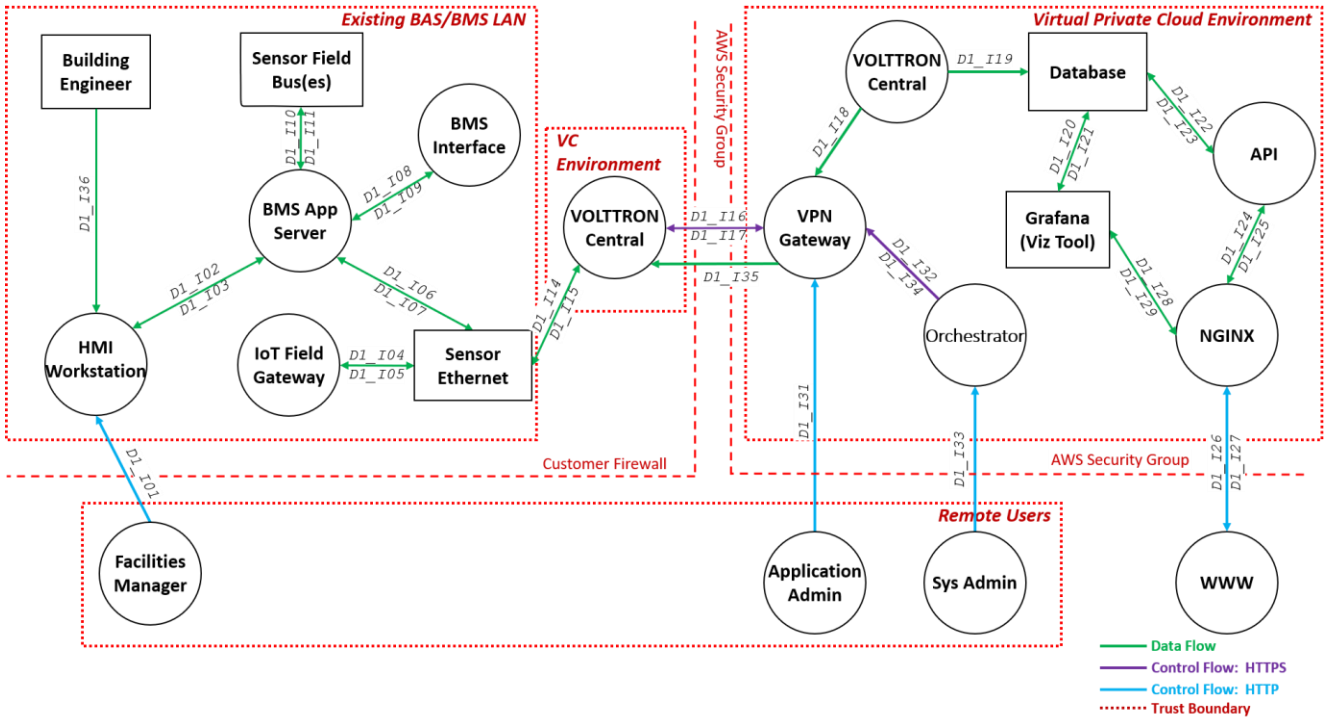



Diagram 1. VOLTTRON deployment for ACE IoT.

# 3.0 Threat Profile Table

The details for all the threats, the mapping of those threats to categories, example threats, and associated mitigations are documented here. Mitigations are the main objective and describe what will be done to prevent, deter, or minimize the threat.

## 3.1 Interpreting the Labels

The labels captured in parentheses in the Threat column of the Threat Profile table (Table 1) below refer to the diagrams above. The label refers to an interaction (arrow) in the diagram, thus showing which interaction and which components the threat corresponds to. For example, a label such as `D1_I15` refers to Diagram 1, Interaction 15. On Diagram 1 above, the arrow labeled `I15` will be the interaction corresponding to the threat. This strategy enables the tracking of a mitigation, the threat it addresses, and the area of the diagram where the threat could occur. Thus, the table provides complete traceability from mitigation to threat to interactions between components.

Note that bold items in the Mitigations column are mitigations that have yet to be implemented and are therefore potential issues that should be addressed. Non-bold items are either already in place, expected to be addressed outside of direct ACE IoT scope, or represent a risk that is accepted by the ACE IoT team. Whether bold or not, the description provides the detail to explain the situation for the purposes of due diligence, traceability, or risk management.

## 3.2 The Detailed Threat Profile Table

Table 1 below lists the threat type, threat, and mitigation. The table is arranged in order of priority.

Table 1. Threat Profile table.

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| **HIGH** | | | |
| **1** | Spoofing | An adversary can get access to a user's session by replaying authentication tokens. (D1_I32, D1_I18) | Use secure shell (SSH) Public Key Infrastructure for authentication. |
| **2** | Spoofing | An adversary may spoof a device and connect to the field gateway. This may be achieved even when the device is registered in the cloud gateway because the field gateway may not be in sync with the device identities in the cloud gateway. (D1_I05) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what a deployed sensor should provide.** |

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| **3** | Spoofing | An attacker may extract cryptographic key material from Sensor - Ethernet, either at the software or hardware level, and afterwards access the system with a different physical or virtual Sensor - Ethernet under the identity of the Sensor - Ethernet the key material has been taken from. A good illustration is remote controls that can turn on any TV and that are popular prankster tools. (D1_I05) | For devices with encryption, protect from malicious behavior through PoLP and role-based access control. Accept risk of complete spoofing without escalation of privilege. Take advantage of TPM for systems that support it. |
| **4** | Tampering | An adversary can tamper critical database securables and deny the action. (D1_I19) | Limit write access to CrateDB to VOLTTRON Central. The authenticator role has limited access to securables. |
| **5** | Tampering | An adversary may gain unauthorized access to the IoT field gateway, tamper its operating system (OS), and get access to confidential information in the field gateway. (D1_I04) | Protect from malicious behavior through PoLP and role-based access control because only confidential information here qualifies as credentials. |
| **6** | Tampering | An adversary may launch malicious code into the IoT field gateway and execute it. (D1_I02, D1_I05) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **7** | Tampering | An adversary may launch malicious code into Sensor - Ethernet and execute it. (D1_I04, D1_I07,D1_I15) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **8** | Tampering | An adversary may launch malicious code into the Sensor - Filed Bus(es) and execute it. (D1_I10) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| **9** | Tampering | An adversary may launch offline attacks made by disabling or circumventing the installed operating system or by physically separating the storage media from the device to attack the data separately. (D1_I11, D1_I05, D1_I14) | Deploy Edge VOLTTRON in a physically controlled location. **For external entities (such as a BMS application server), explore ways to detect malicious behavior.** |
| **10** | Tampering | An adversary may partially or wholly replace the software running on a BMS application server, which may allow the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example, an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material. (D1_I11) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **11** | Tampering | An adversary may partially or wholly replace the software running on an IoT field gateway, which may allow the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example, an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material. (D1_I05) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **12** | Tampering | An adversary may partially or wholly replace the software running on Sensor - Ethernet, which may allow the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material. (D1_I04) | Accept risk of legacy systems. **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **13** | Tampering | An adversary may partially or wholly replace the software running on Edge VOLTTRON, which may allow the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example, an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material. (D1_I14) | **Where possible, leverage TPM to verify that software has not been tampered with.** Reduce risk of compromise through unique credentials and PoLP. |
| **14** | Tampering | An adversary may perform a man-in-the-middle attack on the encrypted traffic sent to Sensor – Ethernet. (D1_I04) | Implement encryption Secure Sockets Layer (SSL) where available. |
| **15** | Tampering | An attacker steals messages off the network and replays them to steal a user's session. (D1_I27) | Implement HTTPS. |

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| **16** | Information Disclosure | Structured Query Language (SQL) injection is an attack in which malicious code is inserted into strings that are later passed to an instance of the SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A more indirect attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are later concatenated into a dynamic SQL command, the malicious code is executed. (D1_I19, D1_I08, D1_I27) | Only accept inputs from an authenticated source (VOLTTRON Central). Implement RBAC on database. |
| **17** | Denial of Service | An adversary may block access to the application or application programmer interface (API) hosted on the API through a denial of service attack. (D1_I23) | **Explore NGINX configurations to rate-limit requests and timeouts and other available Denial of Service mitigations.** |
| **18** | Elevation of Privileges | An adversary may get access to an admin interface or privileged services like Wi-Fi, SSH, file shares, FTP etc., on a device. (D1_I04, D1_I11, D1_I06, D1_I03, D1_I05, D1_I14) | For Edge VOLTTRON, only enable essential services and authentication to them. Disable access to privilege services altogether from untrusted sources. |
| **19** | Elevation of Privileges | An adversary may use unused features or services on a BMS application server such as a user interface (UI), USB port, etc. Unused features increase the attack surface and serve as additional entry points for the adversary. (D1_I11, D1_I06) | **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **20** | Elevation of Privileges | An adversary may use unused features or services on an HMI workstation such as a UI, USB port, etc. Unused features increase the attack surface and serve as additional entry points for the adversary. (D1_I03) | **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **21** | Elevation of Privileges | An adversary may use unused features or services on an IoT field gateway such as a UI, USB port, etc. Unused features increase the attack surface and serve as additional entry points for the adversary. (D1_I05) | **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **22** | Elevation of Privileges | An adversary may use unused features or services on Sensor - Ethernet such as UI, USB port etc. Unused features increase the attack surface and serve as additional entry points for the adversary(D1_I04) | **Explore ways to detect malicious behavior outside the norm of what deployed sensor should provide.** |
| **23** | Elevation of Privileges | An adversary may use unused features or services on Edge VOLTTRON such as a UI, USB port, etc. Unused features | Deploy Edge VOLTTRON in a |

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| | | increase the attack surface and serve as additional entry points for the adversary. (D1_I14) | physically controlled location.<br>Do not provide a UI.<br>Disable USB ports. |
| MEDIUM | | | |
| **24** | Spoofing | Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a web server, which is a trustworthy entity in electronic communication. (D1_I08, D1_I27) | Do not accept implementations with poor credential management as described in the threat.<br>Configure NGINX using credential management best practices. |
| **25** | Spoofing | An adversary can bypass authentication because of non-standard AWS IAM authentication schemes. (D1_I32, D1_I18) | Implement HTTPS.<br>Limit risk through role-based access control.<br>Limit risk by not sharing accounts. |
| **26** | Spoofing | An adversary can bypass authentication because of non-standard AWS IAM authentication schemes. (D1_I32, D1_I18) | Use standard AWS IAM authentication |
| **27** | Spoofing | If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the web application. (D1_I08, D1_I27) | Configure NGINX using credential management best practices. |
| **28** | Spoofing | The session cookie is the identifier by which the server knows the identity of the current user for each incoming request. If the attacker is able to steal the user token, they would be able to access all user data and perform all actions on behalf of the user. (D1_I27) | Implement HTTPS.<br>Implement session token lifetime to limit risk of spoof. |
| **29** | Tampering | An adversary may leverage known vulnerabilities and exploit a device if the firmware of the device is not updated. (D1_I04, D1_I11, D1_I03, D1_I05, D1_I14) | **When the firmware version is available, track current version and alert when versions are out of date.**<br>Keep firmware updated for devices within control. |
| **30** | Tampering | An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to the database. (D1_I19) | Only accept inputs from authenticated source (VOLTTRON Central).<br>Implement RBAC on database. |
| **31** | Information Disclosure | An adversary can reverse weakly encrypted or hashed content. (D1_I08, D1_I27) | Configure NGINX using credential management best practices. |

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| | | | Configure HTTPS connection using best practices. |
| **32** | Information Disclosure | If an adversary can gain access to AWS virtual machines (VMs), sensitive data in the VM can be disclosed if the OS in the VM is not encrypted. (D1_I31, D1_I26, D1_I33, D1_I27, D1_I17, D1_I35) | Configure AWS administrator access control to VMs and periodically review accounts. |
| **33** | Denial of Service | Failure to restrict requests originating from third-party domains may result in unauthorized actions or access of data. (D1_I27) | Configure NGINX using credential management best practices. |
| **34** | Elevation of Privileges | An adversary may gain long-term, persistent access to related resources through the compromise of an application identity. (D1_I23, D1_I25) | Implement AWS security groups to limit communication channels. |
| **35** | Elevation of Privileges | An adversary may gain unauthorized access to an API because of weak network configuration. (D1_I23, D1_I25) | Implement AWS security groups to limit communication channels. |
| **36** | Elevation of Privileges | An adversary may leverage insufficient authorization checks on the device and execute unauthorized and sensitive commands remotely. (D1_I04) | **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **37** | Elevation of Privileges | An adversary may leverage insufficient authorization checks on the field gateway and execute unauthorized and sensitive commands remotely. (D1_I05) | **Explore ways to detect malicious behavior outside the norm of what the deployed sensor should provide.** |
| **38** | Elevation of Privileges | An adversary may perform action(s) on behalf of another user due to lack of controls against cross-domain requests. (D1_I23, D1_I25) | Limit API requests to an authenticated and expected process within the system. Implement AWS security groups to limit communication channels. |
| **39** | Elevation of Privileges | Failure to restrict the privileges and access rights to the application to individuals who require the privileges or access rights may result in unauthorized use of data due to inappropriate rights settings and validation. (D1_I27) | Configure NGINX using credential management best practices. |
| **40** | Elevation of Privileges | If there is no restriction at the network or host firewall level to access the database, anyone can attempt to connect to the database from an unauthorized location. (D1_I22, D1_I19, D1_I20) | Implement AWS security groups and firewall. |
| LOW | | | |

| # | Threat Type | Threat | Mitigation |
|---|---|---|---|
| **41** | Tampering | An adversary can gain access to the configuration files, and if sensitive data is stored in there, it would be compromised. (D1_I08, D1_I27) | Configure NGINX using credential management best practices.<br>Do not store sensitive data in the configuration files. |
| **42** | Repudiation | Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system. (D1_I08, D1_I27, D1_I28, D1_I24) | Log critical transactions within the system.<br>**Explore reviewing logs for malicious and anomalous behavior.** |
| **43** | Repudiation | Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system. (D1_I19, D1_I20, D1_I22) | Log critical transactions within the system.<br>**Explore reviewing logs for malicious and anomalous behavior.** |
| **44** | Information Disclosure | An adversary may gain access to unmasked sensitive data such as credit card numbers. (D1_I27) | Do not implement unmasked, sensitive data fields. |
| **45** | Information Disclosure | An adversary may conduct a man-in-the-middle attack and downgrade the Transport Layer Security connection to clear text protocol or force browser communication to pass through a proxy server that the adversary controls. This may happen because the application may use mixed content or because the HTTP Strict Transport Security policy is not verified. (D1_I27) | Configure NGINX using credential management best practices. |
| **46** | Information Disclosure | Through verbose error messages, an adversary can gain access to sensitive data such as the following: server names; connection strings; usernames; passwords; SQL procedures; details of dynamic SQL failures; stack trace and lines of code; variables stored in memory; drive and folder locations; application install points; host configuration settings; and other internal application details. (D1_I08, D1_I27) | Limit information displayed in error messages. |
| **47** | Information Disclosure | An adversary may gain access to sensitive data from log files. (D1_I08, D1_I27) | Do not log sensitive information.<br>Store logs in a centralized and protected server in the AWS. |
| **48** | Information Disclosure | An adversary may gain access to sensitive data from an uncleared browser cache. (D1_I27) | Configure NGINX using credential management best practices. |

# 4.0 Conclusion

This ACE IoT Threat Profile identifies threats that are mapped to specific system components. It also provides mitigations for each unique threat-asset pairing. The outputs are actionable controls and an understanding of risk that informs decision makers who are most concerned with optimizing impact or cost. The contents of this Threat Profile inform threat-based decisions for increasing security at a reasonable cost and for reducing risk.

This threat-based software analysis illustrates the due diligence of ACE IoT. In seeking an external assessment, ACE IoT ensures that their deployment of VOLTTRON to their customers is as secure and reliable as possible.

The ACE IoT Threat Profile provides a foundation for a thorough understanding of possible threats for the development team, the testing team, management, stakeholders, and partner stakeholders of ACE IoT. It enables decision makers at all levels to improve the security posture of the ACE IoT deployments of VOLTTRON. This effort leads to more secure software and better-understood security. ACE IoT and VOLTTRON are to be commended for their rigorous approach to employing cybersecurity throughout the software development life cycle.

# Appendix A Brief on Threat Based Analysis

The Secure Software Central (SSC) team combines three stages of Threat Based Analysis (TBA), as shown in Figure 5. TBA utilizes portions of Lockheed Martin's IDDIL-ATC methodology (Figure 6) to perform threat analysis. SSC optimizes IDDIL-ATC for more cost-effective, time-efficient results that lead to immediately actionable controls. Using the Lockheed Martin nomenclature, SSC actually begins with ***Decompose the System***. To accomplish this, SSC requests that ***Use cases*** be written by members of the project team. The use cases provide the SSC team with valuable context in simple, non-jargon terms. With this context, the next step is to develop a set of use cases and data flow diagrams that represent the system. Generally, the assets and the attack surface can be identified using these diagrams, thus addressing the ***Identify Assets*** and


Figure 5. The TBA half of SSC.

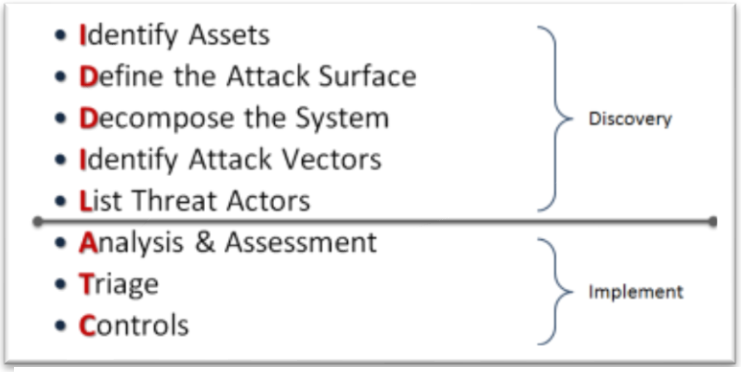



Figure 6. Lockheed Martin's methodology.

***Define the Attack Surface*** steps. From there, SSC attempts to ***List Threat Actors***, but this is not yet a rigorous exercise. The use cases, threat cases, and data flow diagrams represent the ***SSC Threat Model***, which is the foundation for developing the Threat Profile.

SSC asks the project team to set an initial expectation of threat priority based on Confidentiality, Integrity, and Availability (CIA). The CIA Triad of Figure 7 is a widely used cybersecurity model.


Figure 7. The CIA triad.

The SSC team uses the data flow diagrams as input to Microsoft's Threat Modeling Tool (TMT). The TMT is a free download that comes with standard threat templates used by SSC. The TMT reads the diagrams and uses the templates to provide initial ***Analysis and Assessment*** as well as ***Triage*** results. The TMT also uses Microsoft's STRIDE model (outlined above in (Figure 2) categorize threats. The initial results from the TMT are then analyzed by SSC subject matter experts to complete the ***SSC Threat Findings*** for review by the project team.

With the Threat Findings in hand, SSC goes back to the project team to collaboratively analyze and determine mitigations (***Controls***). When this exercise is complete, the SSC team organizes the information into the final product, the ***SSC Threat Profile***.

# Appendix B Brief on Secure Software Development

The Secure Software Central (SSC) Team is developing Secure Software development best practices in the areas depicted in Figure 8. While SSC will at some point offer ***Secure Design*** and ***Security Testing***, the current focus is on ***Secure Coding***. For SSC, secure coding combines Static Application Security Testing (SAST) and Open Source Analysis (OSA). The objective is to produce a Vulnerability Profile, which uses a SAST vulnerability scan of the code and an OSA scan to produce initial results. PNNL has adopted Checkmarx as the lab's vulnerability scanner, which does both SAST and OSA scans. SSC uses Checkmarx results to perform an analysis that eliminates false positives and condenses information into a simple report for use by the software development team. The full scan is also available in the Vulnerability Profile. The SSC process for creating a Vulnerability Profile is a straightforward set of steps:

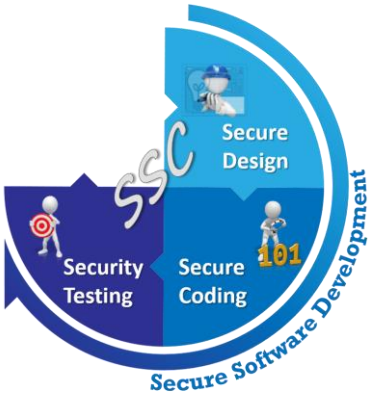



Figure 8. The SSD half of SSC.

1. Receive source code from development team in the form of a zip file

   The zip file will be unzipped and used as input to the Checkmarx scanner.

2. Run Checkmarx SAST scan

   Every file contained in the zip file will be scanned with results, forming the foundation for SSC analysis.

3. Run Checkmarx OSA scan

   Dependency libraries will be checked by Checkmarx, and vulnerable libraries along with out-of-date libraries will be documented, forming the foundation for SSC analysis.

4. Analyze SAST scan results

   Results of SSC analysis are in the SAST Profile section of a Vulnerability Profile.

5. Analyze OSA scan results

   Results of SSC analysis are in the OSA Profile section of a Vulnerability Profile.

6. Deliver a Vulnerability Profile, often accompanied by a Threat Profile.

# Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

***www.pnnl.gov***