U.S. DEPARTMENT OF ENERGY

Office of ENERGY EFFICIENCY & RENEWABLE ENERGY

A Cybersecurity Threat Profile for a Connected Lighting System

February 2022

(This page intentionally left blank)

A Cybersecurity Threat Profile for a Connected Lighting System

Paul Francik, Pacific Northwest National Laboratory: Methodology, Visualization, Writing-original draft, Investigation, Analysis

Michael Poplawski, Pacific Northwest National Laboratory: Conceptualization, Funding acquisition, Project Administration, Supervision, Writing-review & editing

Sri Nikhil Gupta Gourisetti, Pacific Northwest National Laboratory: Writing-review & editing, Analysis

Patrick O'Connell, Pacific Northwest National Laboratory: Validation, Software

Chance Younkin, Pacific Northwest National Laboratory: Conceptualization, Resources

Travis Ashley, Pacific Northwest National Laboratory: Methodology

Garrett Seppala, Pacific Northwest National Laboratory: Analysis

February 2022

Produced for the U.S. Department of Energy, Energy Efficiency and Renewable Energy, by the Pacific Northwest National Laboratory, Richland, Washington 99352

PNNL-31958

DISCLAIMER

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

Abstract

In anticipation of improved energy performance and cost savings, cities and building owners are increasingly considering "smart lighting initiatives" that aim to convert their collection of simple luminaires (i.e., lighting fixtures) into an intelligent connected lighting system (CLS) capable of remotely monitoring energy consumption and fault conditions, and possibly implementing adaptive lighting schemes. The U.S. Department of Energy (DOE) has set a national goal of tripling the energy efficiency and demand flexibility of the buildings sector by 2030, relative to 2020 levels.[1] DOE forecasts that connected lighting systems can contribute to that goal by delivering 125 TWh of annual energy savings by 2035,[2] equivalent to the annual output of 50 typical (500 MW) power plants. However, these energy savings and the DOE goal are put at significant risk if connected technologies are not adopted due to real or perceived cybersecurity concerns. Connected IoT devices have historically been rife with vulnerabilities, and security considerations are sometimes secondary to functionality and operability. What are the cybersecurity threats that will impact these systems, as formerly banal luminaires transition into intelligent connected devices that collect information about themselves, their surrounding environment, and possibly us?

In this paper we analyze a threat profile performed on a fault-detection use case for streetlights. A threat profile establishes security requirements, justifies security measures, yields actionable controls, and effectively communicates risk to stakeholders. This effort provides critical information for making threat-based decisions to increase security at a reasonable cost, and can effectively be used by development teams, software architects, and managers to make cybersecurity a part of their ongoing culture of awareness, training, and prevention. This leads to more secure systems and better-understood security. On-premise, cloud, and hybrid architectures with different authentication mechanisms were modeled and later categorized using the Microsoft STRIDE framework. An analysis of the recommended controls for each threat was performed to determine which controls could and should be put in place by manufacturers or third-party suppliers, and which controls need to be left up the end-user to implement. Fifty-seven threats were identified, as seen in the following table.

Distribution of threats mapped to the STRIDE framework.

Threat Type	High Priority	Medium Priority	Low Priority	Total
Spoofing	14	0	0	14
Tampering	4	10	0	14
Repudiation	2	1	0	3
Information Disclosure	11	2	0	13
Denial of Service	1	0	0	1
Elevation of Privilege	12	0	0	12
Total	44	13	0	57

Among our key findings:

- 65% (37/57) of the threats did not involve the luminaires, but rather the other components needed to communicate with and manage them
- 63% (36/57) of the threats could have been mitigated through manufacturer-implemented defensive techniques or "controls"
- 23% (13/57) of the threats were dependent on the network configuration.

Recommendations based on the results of this work are made to key stakeholder groups. Notably, lighting technology developers are advised to address all threats that can be reasonably controlled with baked-in technology solutions (e.g., encryption or authentication controls), and employ some form of secure supply chain management and tracking where other parts (e.g., sensors, microprocessors) of a luminaire must also be built and manufactured with the proper security controls in place. Developers should also review threats involving assets not developed in-house to understand how connectivity with other devices will affect their product during system operation and determine if a compensating control for a defense-in-depth strategy will be needed. Finally, those interested in deploying CLS should compare the differences between cloud and on-premise models to determine which is more suitable for their needs and the abilities of their security team.

Introduction

Buildings and city infrastructure are generating increasing types and amounts of data and using that data to enable valuable services.[3] Sensors are used in buildings to open doors and flush toilets, deployed in cities to monitor the weather and track air quality, and integrated into mobile phones to recognize faces and authorize access. Sensors that have traditionally been used to improve the energy performance of lighting systems include those that attempt to detect human presence or daylight.[4] Typically, these sensors were directly integrated with a single or small number of lights meant to be influenced by sensor outputs, and sensor data was not available to other lighting devices, or outside of the lighting system. Increasingly indoor and outdoor luminaires, which were once single-purpose devices that existed solely to provide light, are now being equipped with modern network interfaces that allow for broader sharing of data within and outside of the lighting system, thereby enabling the formation of intelligent connected lighting systems (CLS).[4] Further, sensors with other diverse capabilities are being incorporated into lighting systems in ways that are sometimes obvious and sometimes inconspicuous. The potential offered by the increased availability and use of data is significant. Data can be used to improve the service and energy performance of lighting and other connected building or infrastructure systems, and possibly even be shared with other IoT systems. The U.S. Department of Energy (DOE) has set a national goal of tripling the energy efficiency and demand flexibility of the buildings sector by 2030, relative to 2020 levels.[1] DOE forecasts that connected lighting systems can contribute to that goal by delivering 125 TWh of annual energy savings by 2035,[2] equivalent to the annual output of 50 typical (500 MW) power plants. However, as lighting and other systems are becoming more connected, they also become more vulnerable to potential cyber-attacks.[5] Lighting energy savings and the DOE goal are put at significant risk if connected technologies are not adopted due to real or perceived cybersecurity concerns. Data is expected to improve performance and create value—but is that data reliably secure? Are the systems and devices that collect and transmit the data reliably safe?

This work characterizes the attack surface of a CLS from the vantage point of an adversary wanting to manipulate and control the devices, and the requirements for securing these systems. Significant insight into the attack surfaces, or entry points into a system that an adversary could infiltrate if left unprotected, can be derived from threat models. Controls are the actions and implementations needed to mitigate a threat: for example, authenticating a user before granting access to a system by making sure the individual matches the identification credential provided. The method(s) of authentication is the control to the threat of someone spoofing or imitating someone else's identity and trying to gain system access. A threat profile will identify different threat types, patterns, and rates of occurrence in the attack surface, while suggesting controls that can be applied to mitigate the threats that are discovered.

This paper explores the use of threat profiles for lighting and other emerging IoT systems, and aims to answer the following research questions:

1. What cybersecurity threats are associated with a CLS?
2. How does CLS architecture, authentication mechanism, and installation size change the attack surface?
3. Who is responsible for implementing the controls suggested by the threat profile?

Background

Historically as devices have become internet enabled, or connected, cybersecurity has been an afterthought, taking a back seat to device functionality and ease of use. At times this was due to physical limitations of the device (e.g., low compute power that cannot support the encryption of its communication, so data is sent in plain text prioritizing availability and function over confidentiality and encryption). Among the key findings from a Unit 42 IoT threat report in 2020 was that, of the 1.2 million devices they tested, "98% of all IoT device traffic is unencrypted, exposing personal and confidential data on the network, allowing attackers the ability to listen to unencrypted network traffic, collect personal or confidential information, then exploit that data for profit on the dark web." They also reported that, "57% of IoT devices are vulnerable to medium or high-severity attacks, making IoT the low-hanging fruit for attackers." Additionally, the report found that, "While the security postures of IoT devices make them easy targets, in most cases, the devices are only used as stepping stones in lateral movement to attack other systems on a network."[6]

Cyber attacks can be expensive, complex, and sophisticated. According to the 2020 Cost of a Data Breach Report by IBM, an average breach in 2020 cost $3.9 million U.S. dollars, and on average takes an organization 280 days to identify and contain.[7] Cybersecurity Ventures estimates that cybercrime globally will cost $10.2 trillion U.S. dollars by 2025.[8] According to a new study in 2020 by Tenable, "94% of organizations have experienced at least one business-impacting cyber attack in the past 12 Months."[9] While loss of reputation may be hard to quantify in terms of dollars, it too is worth mentioning as it can destroy projected growth or consumer confidence for years into the future.

Connected lighting is still in the early introduction-to-market phase. Securing these types of systems is necessary to aid organizations and building owners wanting to actualize projected cost savings driving CLS adoption. When lights are connected over a network and can communicate with other devices, new attack vectors are opened to those looking to infiltrate insecure systems. This could allow for theft of sensitive information, pivots into other networks with even more valuable data, enslavement of devices into a botnet army, or disruption of vital services. In fact, in 2016 researchers at Dalhousie University demonstrated from over 200 feet away, a van equipped with "readily available" components costing only a few hundred dollars could compromise a popular name brand smart light bulb simply by driving around the block while malicious firmware updates were sent out to the luminaires.[10] These same researchers separately performed a war flying technique where a drone with an "autonomous attack kit" sent out malicious firmware updates as it flew by an office building equipped with proprietary smart light bulbs, setting off a chain reaction allowing the malicious code to jump from light to light until all units were compromised. In IEEE Extended Functionality Attacks on Iot Devices, other researchers demonstrated that while imperceptible to the human eye, "they were able to take control of LED lighting units and create strobes of light at frequency ranges that are known to induce seizures in people suffering from photosensitive epilepsy."[11]

Cybersecurity is a vast domain with processes that cover many aspects of an organization involving people, technologies, devices, networks, and frameworks. Security is also iterative, so users need to constantly define the most critical assets and the controls in place to balance security and functionality. Identifying a system's assets and evaluating the associated risks helps determine which are most relevant; for those that exceed the risk tolerance, testing can be implemented to identify whether sufficient controls are in place to mitigate that risk.

Defining an organization's risk tolerance, however, is not a perfect science and can be subject to bias depending on experience or background. For this reason, some organizations implement a holistic type of approach that balances a qualitative analysis, to hone in on the major threats, with some form of quantitative approach to model the likelihood of occurrence and financial impact those risks would pose to the system.

Technology developers and specification organizations' response to cybercrime has been the development of frameworks to guide cybersecurity best practices to mitigate existing known threats. Numerous frameworks

and guidelines exist for evaluating cybersecurity vulnerabilities, such as the NIST Cybersecurity Framework, NIST 800 series comprised of more than 150 resources, IEC 62443 series, UFC 4-010-06, UL 2900-1, and ISO 27001 and 27002. Like other ISO management system standards, certification to ISO/IEC 27001 is possible, but not obligatory. A variety of testing resources are also widely available, including the Open Web Application Security Project (OWASP) Testing Guide. While these frameworks, guidelines, and tests may apply to CLS in whole or in part, there is currently no mandatory requirement for cybersecurity testing or certification. The lighting industry, including technology developers and specification organizations, are currently evaluating the suitability of these frameworks and guidelines for CLS.

A new U.S. law, The Internet of Things (IoT) Cybersecurity Improvement Act of 2020, requires the National Institute of Standards and Technology (NIST) to issue standards and guidelines for the use of IoT devices owned or controlled by federal agencies. Furthermore, it directs NIST to work with cybersecurity researchers, industry experts, and the Department of Homeland Security (DHS) to publish guidelines on security vulnerabilities relating to information systems owned or controlled by a federal agency, including IoT devices, and the resolution of such security vulnerabilities. While this is a great starting point for federal agencies, it does not apply to cities, municipalities, college campuses, or individual building owners.

Organizations can select from many frameworks, methodologies, and tools to gain insight into their products and systems. The CIA triad, or orientation of cybersecurity, takes into account the confidentiality, integrity, and availability of the data and the impact of threats against it. The type of data collected, and type of industry, are considerations in determining CIA prioritization; for instance, if the data needs to remain secret, confidentiality is the top priority. If making sure data has not been tampered with and is correct, then integrity is the top priority. For reliable access and use of a system and its information, availability is the top priority, and where lighting has traditionally fit in. The availability of lights to function when needed and perform their specified lighting purpose is the priority for most CLS. The other aspects of information security are important—obviously, confidentiality and integrity are still considered and implemented wherever possible—but security analysts must prioritize threats which will most adversely affect the organization, as part of the risk analysis process. Sometimes the specific approach may be driven by the function of the industry, sector, and use case. NIST sp800-30 states that "organizations have great flexibility in choosing a particular analysis or approach. The specific approach taken is driven by different organizational considerations (e.g., the quality and quantity of information available with respect to threats, vulnerabilities, and impacts/assets; the specific CIA orientation carrying the highest priority for organizations; availability of analysis tools emphasizing certain orientations; or a combination of the above)."[12]

Vulnerability scanning is an automated process that checks known reported vulnerabilities known as CVEs (common vulnerability and exposures) against a system by scanning the software code for misconfigurations—which helps identify existing vulnerabilities and provides guidance on how they should be patched. This method, however, does not prioritize risk in terms of where an organization should focus testing and would not take the other connected devices (e.g., gateways, APIs, mobile devices) into consideration in relation to how a CLS is networked, and communications are sent and received. It should be noted that running vulnerability scans can be incorporated early on in the software development lifecycle. Designating a member of the security team to perform these types of scans and checks as early in the development process as possible mitigates coding and other software build errors that can be caught and fixed before deployment. If an organization does not have a security team that can run some of these scans, this type of work can be outsourced to a trusted professional organization.

Threat modeling provides an additional layer of security that can be baked in early on and addresses some of the ambiguity around new systems in development, with the intention of modeling a building, product, or system to identify potential issues before development begins or as early in the development process as possible. Defining a use case with a narrative about an end user allows a granular analysis to address all processes, components, and technologies involved from a human to computer, and device to device, interaction. The threat model's main objective is to identify assets within a product or system that need to be

secured, while visually mapping where they are in modeling software, with consideration of the CIA triad, system boundaries, events, and assets. Once the threat findings have been analyzed and assigned an impact rating, actionable mitigation strategies can be prioritized and implemented to enhance the overall security maturity and resilience from outside interference. This is of great value as it can prevent costly rebuilds, reveal threats that were not initially considered, and identify where to focus security budgets and controls. "Analysts and operational roles can benefit from threat modeling and analysis artifacts to uncover potential new attack vectors, especially in newer technologies and environments for which little intelligence will exist. Blending these complementary practices produces an agile and resilient cybersecurity practice and propels the organization to a more mature security posture," notes Michael Muckin and Scott C. Fitch of Lockheed Martin Corporation.[13] Pairing these discoveries from the model while also providing the necessary knowledge to control and mitigate or accept the risk based on the impacts it could have to that system is known as a threat profile.

In traditional IT networks, techniques such as penetration testing and chaos engineering are often used. In a penetration test (often referred to as pen testing), an experienced ethical hacker (or white hat hacker) examines the cyber resiliency of a system or network by attempting to discover vulnerabilities and exploit them. Pen testing involves seven phases: 1. information gathering, 2. reconnaissance, 3. discovery and scanning, 4. vulnerability assessment, 5. exploitation, 6. final analysis and review, and 7. utilization of testing results. Additionally, pen testing involves the use of active and passive scanning tools and is periodically performed in IT networks and organizations. Due to the nature of ICS/OT networks, which may involve legacy systems and real-time operations with little to no redundancy, pen testing in ICS/OT is often non-trivial. If not performed with immense care, pen testing these networks (similar to the connected lighting integration network discussed in this paper) can have severe consequences, including permanent damage to equipment. Similarly, the principle of chaos engineering involves performing robustness and resiliency tests/experiments on live systems and networks to test for the system's capability to withstand adverse conditions (expected, unexpected, malicious, and non-malicious in nature).

Industry (i.e., data that is being created, transported, shared, and stored) and infrastructure (i.e., devices and networks that are being used to create, transport, share and store that data) play a big part in what threats are faced, as well as the attack sophistication of the adversaries wanting access to the systems. Similarly, industry and organizational (i.e., a particular data/system owner and user) risk sensitivity play a big part in selecting cybersecurity management frameworks and methodologies. One way to categorize and communicate the risk each threat poses is to assign a security impact level to each threat, based on the likelihood and potential impact. These impact levels can be used to make framework and methodology decisions and prioritize related activities. One example of a risk matrix is demonstrated in Table 1.

Table 1: One example of an impact matrix chart used to determine and assign risk/impact levels.

		Likelihood				
		Rare	Unlikely	Possible	Likely	Almost Certain
Impact	Catastrophic	Moderate	Moderate	High	Critical	Critical
	Major	Low	Moderate	Moderate	High	Critical
	Moderate	Low	Moderate	Moderate	Moderate	High
	Minor	Very Low	Low	Moderate	Moderate	Moderate
	Insignificant	Very Low	Very Low	Low	Low	Moderate

Method

We conducted a threat profile for a relevant industry use case incorporating fault detection for street lighting. This included the ability of third-party enterprise software to utilize fault detection data for the management and optimization of lighting system maintenance. If a light goes out, the CLS would alert the system owner to the problem. This alert would include fault detection data (e.g., input voltage out-of-range, below lumen maintenance threshold) from a lighting system, regardless of lighting system vendor, enabling the system administrator to dispatch the appropriate work crew to resolve the issue. The CIA prioritization was 1. availability, 2. integrity, and 3. Confidentiality, as seen in Figure 1. The defining features for each CLS are listed in Table 2.

Figure 1: The CIA prioritization used to assign CLS threat impact levels.

Table 2: CLS implementations modeled in the threat profiling tool.

System	Defining Features / Description
CLS A	CLS comprised of single-vendor solution, using **on-premise** server and field gateway; **Active Directory** for authentication
CLS B	CLS comprised of single-vendor solution, using **cloud** server and gateway; **Active Directory** for authentication
CLS C	CLS comprised of single-vendor solution, using **on-premise** server and field gateway; **third-party application** for authentication
CLS D	CLS comprised of single-vendor solution, using **cloud** server and gateway; **third-party application** for authentication
CLS E	CLS comprised of two-vendor integration, using **both on-premise and cloud** servers and gateways; **Active Directory** for authentication
CLS F	CLS comprised of two-vendor integration, using **both on-premise and cloud** servers and gateways; **third-party application** for authentication

One of the most important aspects of performing threat-based analysis is understanding what trust boundaries are 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 2 locates and describes the trust boundaries as defined in the CLS diagrams.

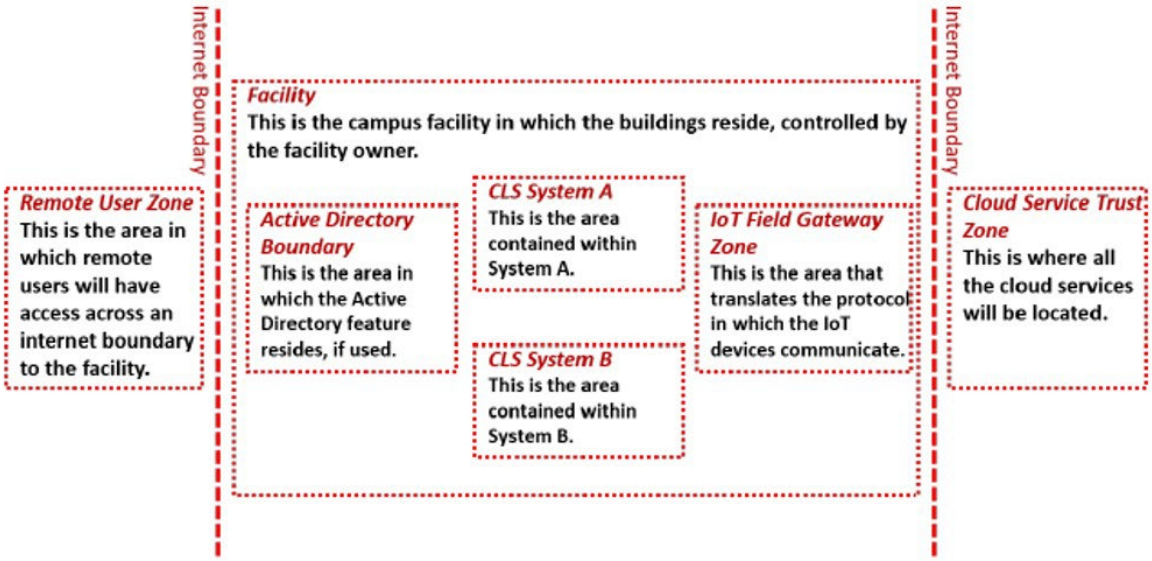

Figure 2: The trust boundaries defined for the CLS and modeled in the threat profiling tool.

CLS Threat Diagrams

The following conventions are used in the depicted threat diagrams to distinguish and categorize system components:

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

Squares – represent physical devices or data storage devices.

Arrows – represent interactions between components or between a person and a component. Arrows are labeled so they can be identified in the Threat Findings table (see appendix) and have mitigations that map directly to the interactions within the system.

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

Red dashed lines – represent internet boundaries between the CLS and external components as seen in Figures 3–8.

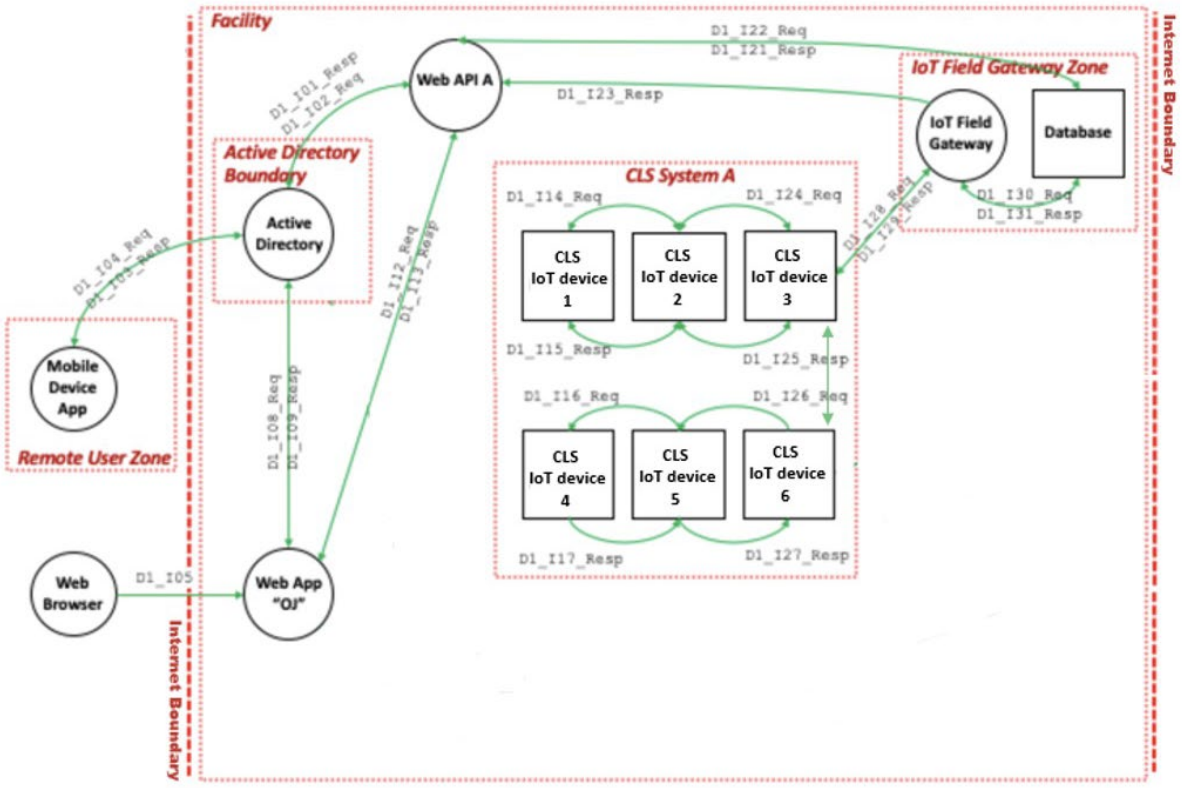

Figure 3: System diagram for CLS A.

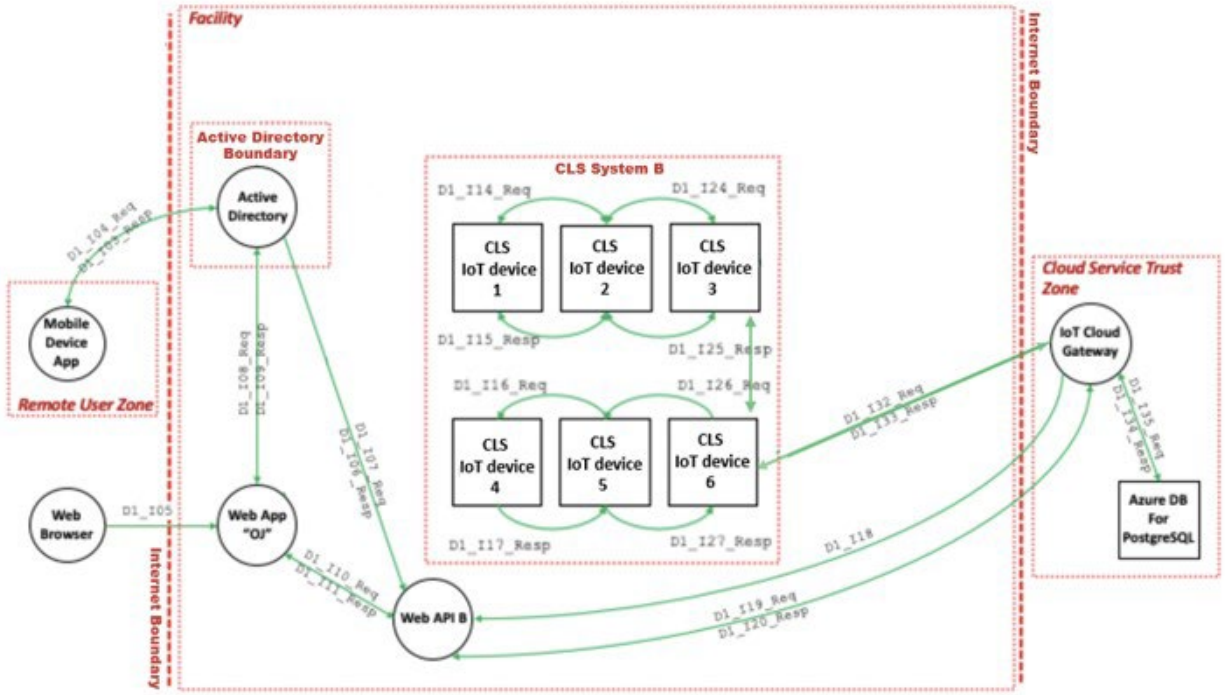

Figure 4: System diagram for CLS B.

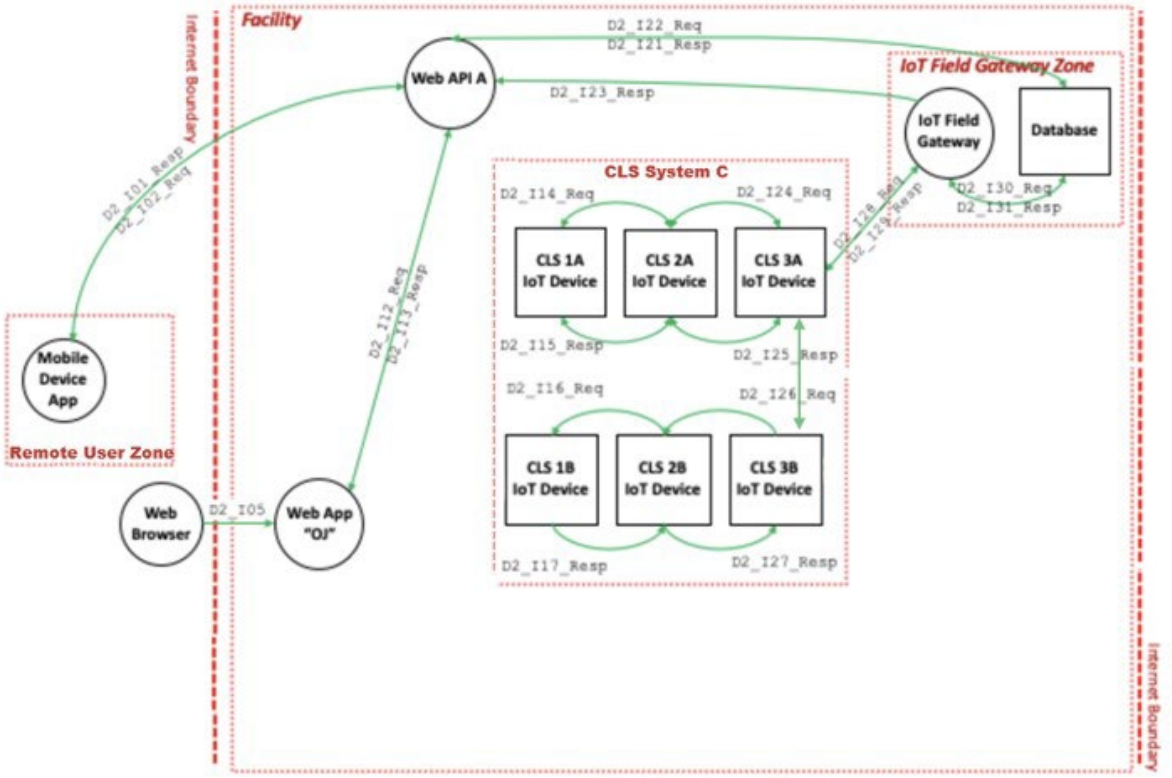

Figure 5: System diagram for CLS C.

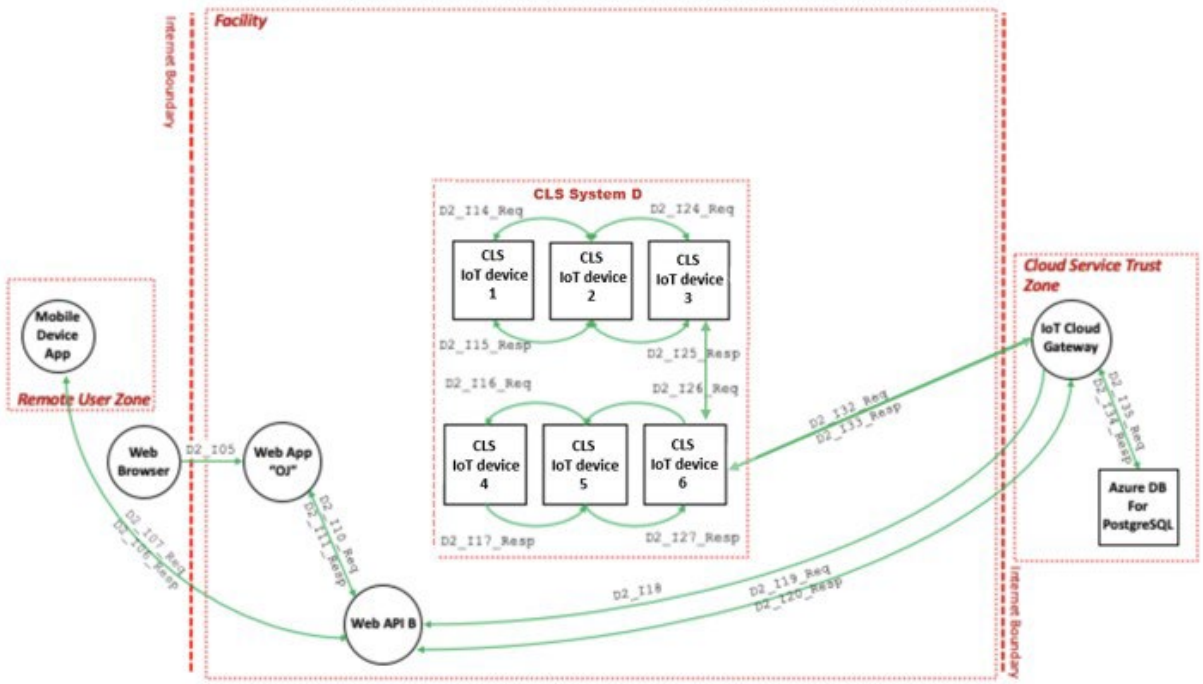

Figure 6: System diagram for CLS D.

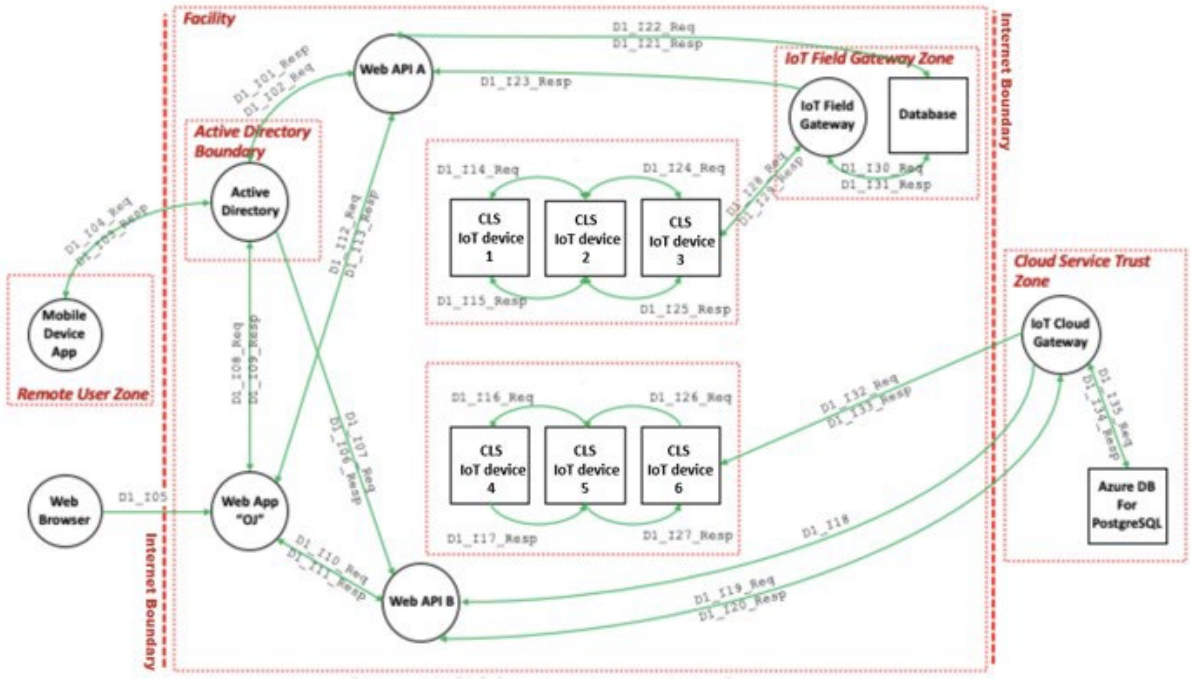

Figure 7: System diagram for CLS E.

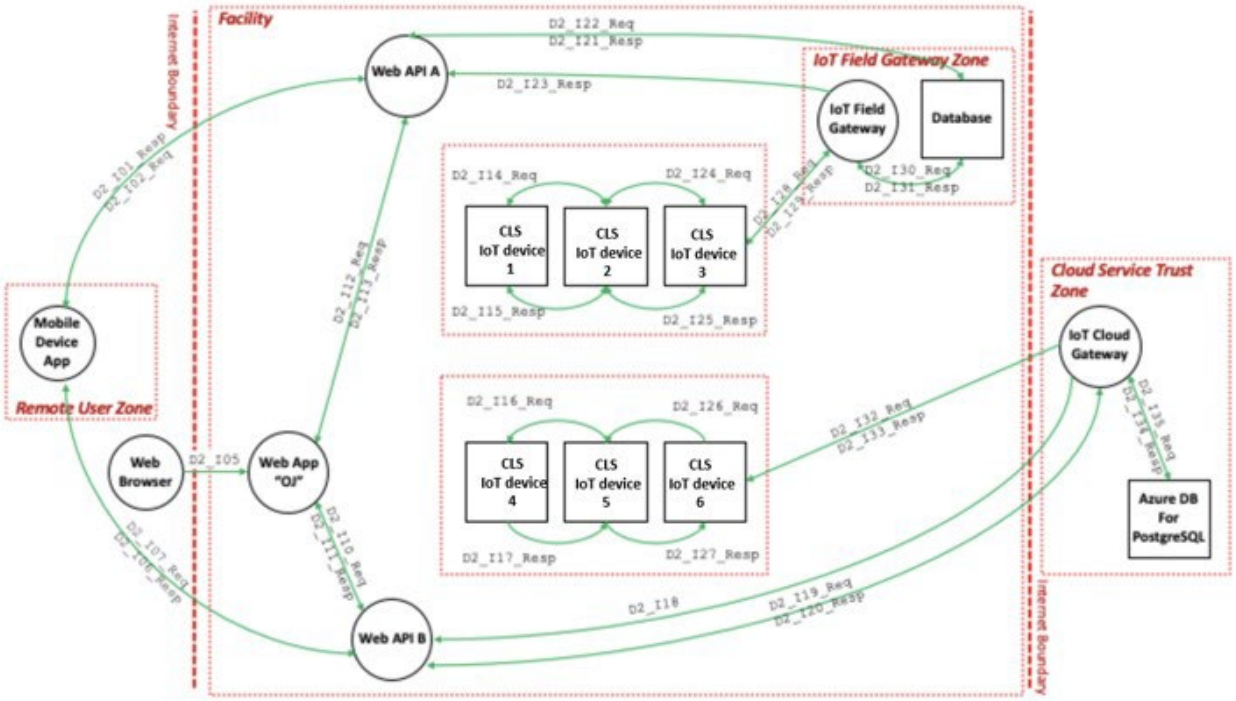

Figure 8: System diagram for CLS F.

There are a few different software options to choose from when it comes to threat modeling. OWASP Threat Dragon and Microsoft Threat Modeling Tool (MTMT) are the two leading open-source options. We chose to use the MTMT as it has been commonly used in security circles for years and our security team was already familiar with it. The MTMT contains a set of element and data-flow templates tailored towards IT networks and Microsoft cloud services. These templates are not necessarily well-suited for OT networks and IoT devices. For instance, it is not possible to specify a CLS that uses the Zigbee protocol as opposed to a more generalized HTTPS network. As a result, the element and data flow templates that most closely matched or represented those found in a CLS were selected. The MTMT does provide the option to customize templates and stencils within the tool, and thereby define application-specific assets, and address characteristics and threats associated with specific protocols and technologies. This capability facilitates the creation of an enhanced body of knowledge for CLS and other intelligent connected building systems, and possibly enables the modeling of an entire connected building. Furthermore, such an enhanced body of knowledge might save time during the analysis process by eliminating the generation of threats that are not applicable to the modeled system through the use of exclusion conditions.

The modeling tool generates threats based on the mapped assets and their configurations, and the security team reviews these results and uses their expertise to consolidate and prioritize the threats. "The Threat Modeling Tool is a core element of the Microsoft Security Development Lifecycle (SDL). It allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. As a result, it greatly reduces the total cost of development," according to Microsoft.[14]

Each threat is categorized as defined by the Microsoft STRIDE[15] framework in Table 3. STRIDE is a pneumonic for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privileges. Categorizing threats helps identify, organize, and prioritize them, and while there are many categorization models, STRIDE is already integrated into Microsoft's threat modeling tool and its processes. This helps with classifying the threat and defining the controls that will be assigned to mitigate the risk.

Table 3: The Microsoft STRIDE categories.

	Threat	Violation	Threat Definition	Example
S	Spoofing Identity	Authentication	Impersonating something or someone else	Pretending to be a website, service, or user to gain access
T	Tampering with Data	Integrity	Modifying data or code	Intercepting data in transit and modifying with malicious code or false text
R	Repudiation	Non-repudiation	Claiming to have not performed an action	"I didn't visit that website," "I never ordered that," "I didn't modify that file"
I	Information Disclosure	Confidentiality	Revealing information to someone not authorized to see it.	Publishing a list of customers to a website, allowing someone to read the source code of software
D	Denial of Service	Availability	Denying or degrading service to users	Crashing a website, SYN floods, rerouting packets
E	Elevation of Privileges	Authorization	Gaining access without proper authorization	A normal user gaining admin privilege, an external remote user accessing and running commands

The Threat Findings table in the appendix maps all threats into categories associated with the STRIDE framework and recommends how to control the threat so it does not become a vulnerability. The objective is to provide the knowledge 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. The threat profile provides critical information for making threat-based decisions to increase security. The tools used to perform the threat profile analysis presented in this work are shown in Table 4.

Table 4: Tools used for analysis.

Tools:	Type	Description
Microsoft PowerPoint	Software	Use case design and narrative
Microsoft Threat Modeling Tool	Software	Modeling software that generates threats based off interactions between devices and assets
Microsoft Excel	Software	Graphs / charts / analysis
Microsoft STRIDE Framework	Threat classification model	https://docs.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach

Results

The total attack surface for all threats when categorized using the STRIDE framework is shown in Figure 9. The CLS and associated color is on the y-axis; the associated Stride category is on the x-axis. All threats were tallied by type. Tampering produced the largest attack surface across all CLS. Attack surface size should not be confused with the impact a threat will have on an organization. Instead, it represents the number of opportunities for that type of threat to enter the system.

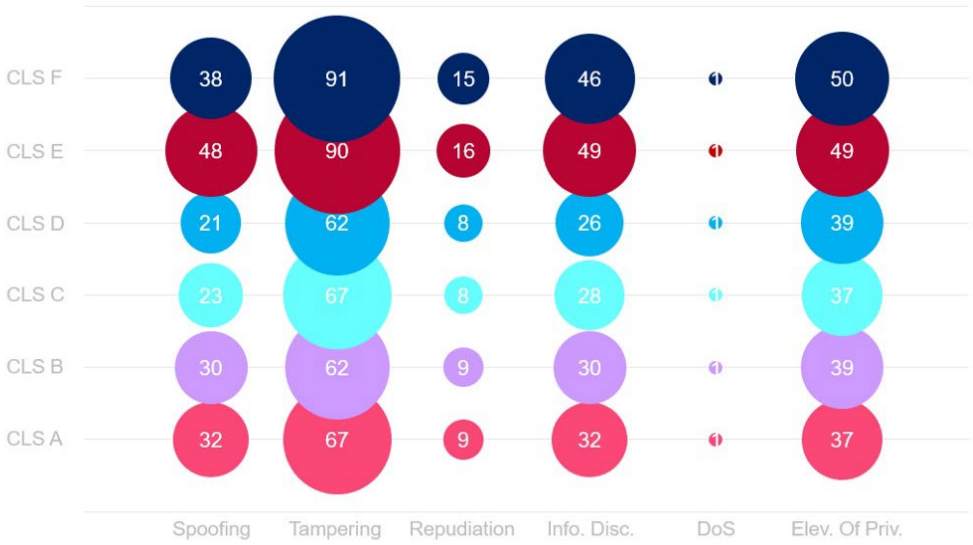

Figure 9: The attack surface for all six CLS mapped to the STRIDE framework.

Elevation of Privileges provides the largest attack surface for high-risk threats as seen in Figure 10. Systems A, B, and E with Active Directory have a larger Spoofing attack surface than systems C, D, and F that do not use Active Directory, as its use increased interactions regarding authentication checks. These added interactions increase the attack surface, which means if Active Directory is improperly set up, it actually makes a system more vulnerable, as opposed to the intended purpose of adding additional protections regarding access to systems. This is generally true for all technology. With each connected device that is added to a network, attack surface and footprint or traceable activities and communications grows, which in turn creates more opportunities for someone to enter the system.

As seen in Figure 11, Tampering provides the largest medium risk attack surface; single vendor systems A and C that incorporate on-premise servers have an additional five points of entry as opposed to single vendor systems B and D that incorporate cloud technologies. Dual systems increase the attack surface for systems E and F. All threats to the CLS were designated as high or medium risk; no low-risk threats were identified.

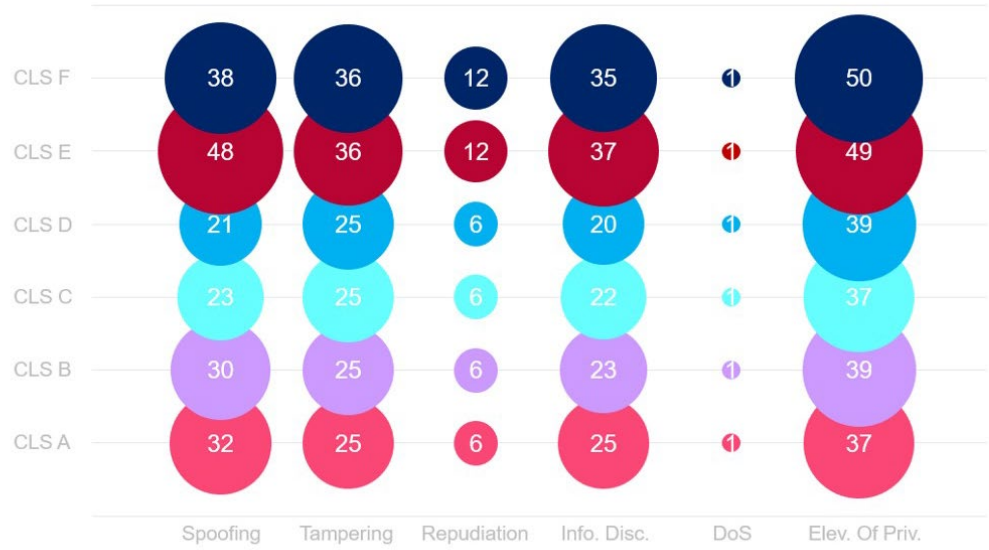

Figure 10: The attack surface for high-risk threats.

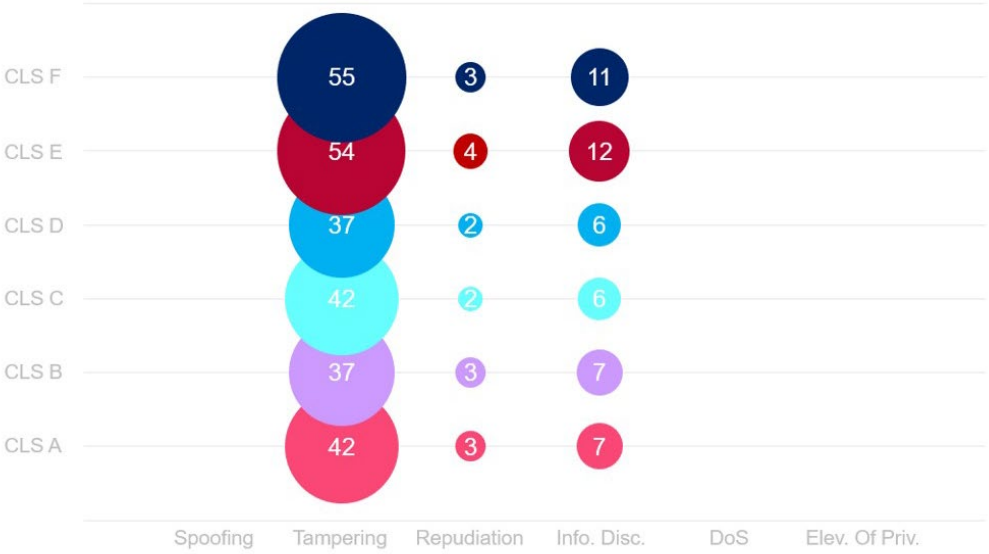

Figure 11: The attack surface for medium-risk threats.

Not all threats apply to all systems. For instance, in Figure 12, when looking vertically at Spoofing threat 1, it only shows up on systems C, D, and F—systems that do not use Active Directory. In contrast, when scanning horizontally we see bigger color bubbles indicating a larger attack surface by the time we get to Spoofing threat 11. In this instance, if left without the proper control that initiates a standard authentication mechanism, this threat will become a vulnerability in every asset authenticating to the web application. The control to mitigate the risk would need to be implemented in seven different places for CLS A and twelve different places for CLS E. This threat is not lighting specific and will be prevalent throughout the technology choices used for that system. Spoofing threat 1 applies to systems *not* using Active Directory and is present on systems C, D, and F whereas Spoofing threat 6 applies to systems *using* Active Directory and only shows up on systems A, B, and E. The use of Active Directory creates additional attack surfaces on systems A, B, and E for Spoofing threat 10 and Spoofing threat 13.

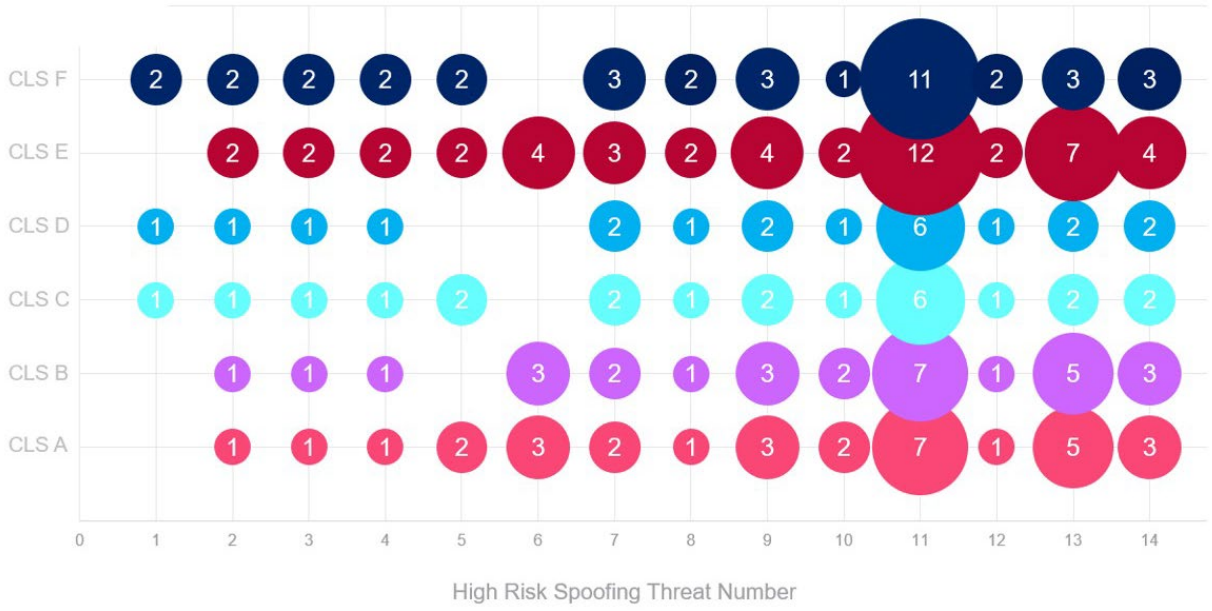

Figure 12: The attack surface for high-risk Spoofing threats.

Figure 13 shows high-risk Tampering, Repudiation, and Denial of Service (DoS) threats. High-risk Tampering threat 4 is identical for all systems because there are six IoT luminaires modeled on all systems and that threat will scale with each unit added. For CLS systems E and F, providing the necessary infrastructure for a second set of luminaires doubles the attack surface for high-risk Tampering threats 2 and 3 and high-risk Repudiation threats 1 and 2. DoS threats for individual systems look the same, regardless of configuration. For systems E and F, totals double on high-risk Tampering threat 2 and 3 and high-risk Repudiation threat 1 and 2, as these threats will scale with installation size related to API communication infrastructure.

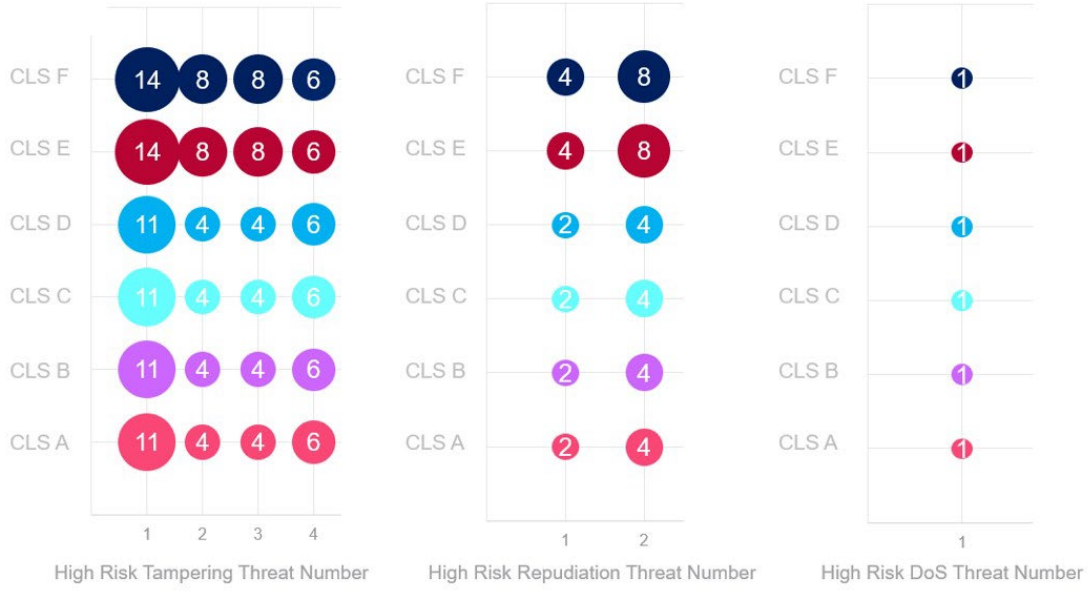

Figure 13: The attack surface for high-risk Tampering, Repudiation, and Denial of Service threats.

Many threats discovered are not lighting specific and are common among the assets needed to provide the connectivity among CLS. This is demonstrated when looking at the total high-risk attack surface for Information Disclosure threats related to the six different CLS as seen in Figure 14. Information Disclosure threat 2 applies to users who choose to configure their own database and at-rest encryption. Users cannot do this for cloud systems—it is up to the manufacturer or cloud owner to implement. Information Disclosure threat 4 appears in assets in the CLS needed to communicate with the lights but is not present in the lighting devices themselves. The attack surface is double for systems E and F, that need two APIs to communicate over HTTPS, as opposed to the other systems that only need one API.

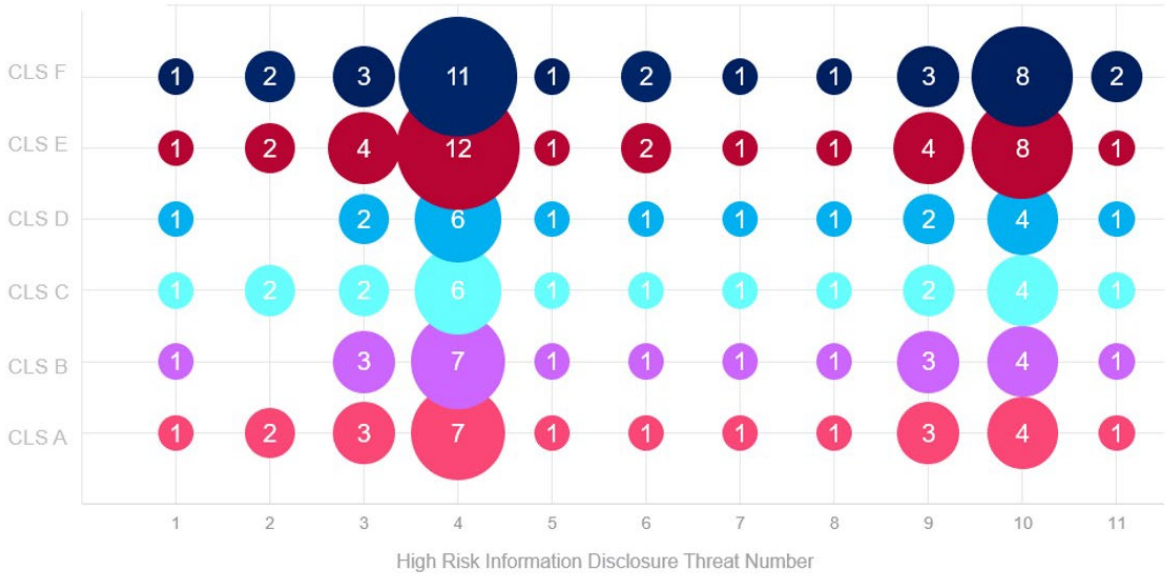

Figure 14: The attack surface for high-risk Information Disclosure threats.

High-risk Elevation of Privileges (EoP) poses a significant threat to all six CLS. In Figure 15, the total high-risk attack surface for EoP threats shows multiple points of entry throughout many assets for threats 3, 6, and 7. Threats 1, 10, and 12 are specific to cloud-based systems, and threats 9 and 11 apply to field-based systems. Threats that are present in the lighting devices scale with installation size when additional luminaires are added to the network. This will be true for EOP threats 3 and 7, and threat 8 as well, if every lighting device has the ability to execute sensitive commands remotely.

The largest majority of medium-risk threats that were discovered fell into the Tampering category. In Figure 16, a large attack surface for threats 4, 7, and 10 exists on assets in all six CLS, showing entry points throughout the system. Threat 1 and 2 are specific to the field database and API connection and only appear on systems with those assets; similarly, threat 9 only appears on systems using a field gateway. Additional entry points for threats 4, 7, and 10 can be seen as a result of dual systems housing two APIs and two databases, as opposed to other systems that utilize only one.

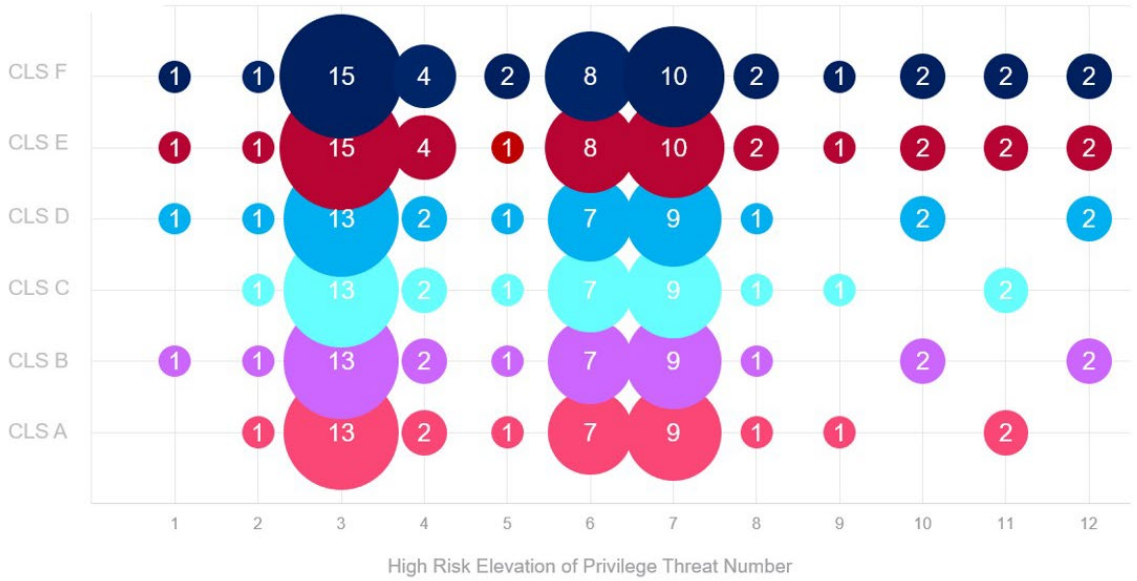

Figure 15: The attack surface for high-risk Elevation of Privilege threats.

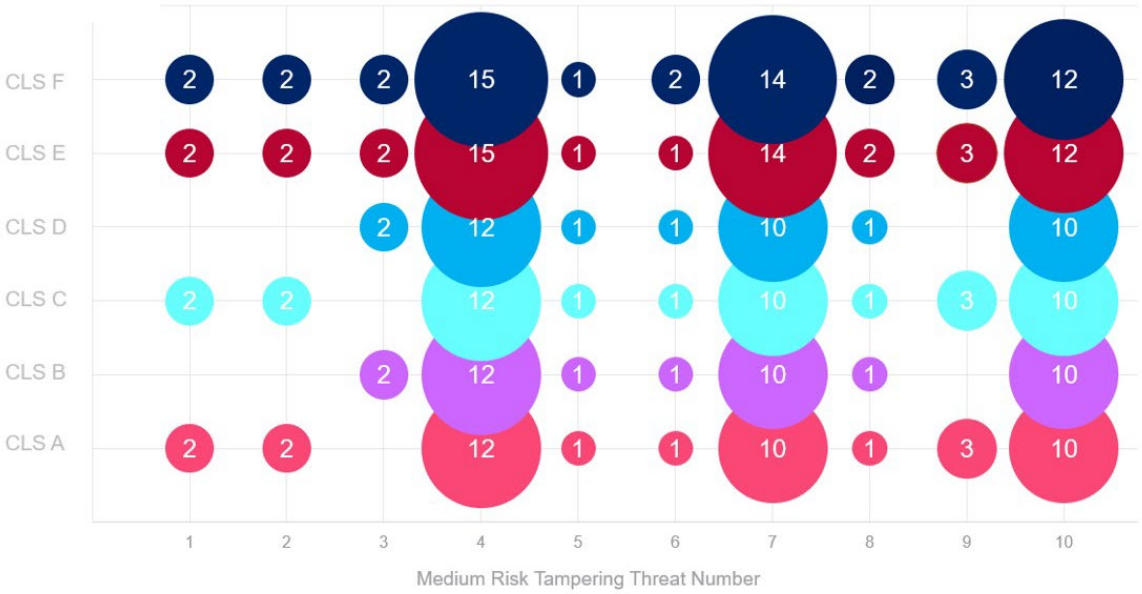

Figure 16: The attack surface for medium-risk Tampering threats.

In Figure 17, Active Directory increases the attack surface by one for systems that use it. It becomes another asset that needs logging and monitoring performed to prevent unauthorized individuals from traversing in a system undetected. The attack surface for dual systems E and F is larger due to the extra API, gateway, and database needed for communication with the on-premise and cloud systems.

The full output of the threat profile—in which identified threats are mapped to specific system components and controls are specified for each distinct threat-asset pairing—is provided in the appendix.

Figure 17: The attack surface for medium-risk Repudiation and Information Disclosure threats.

Analysis

A significant portion of the threats (seventy-seven percent, 44/57) were applicable to all systems. Considering all the different assets that go into making a CLS function (e.g., servers, routers, APIs, end-user devices) perhaps it is no surprise that common technologies and similar system architectures will share some common threats. Of the identified threats, sixty-five percent resided within system assets needed to communicate with and control the lighting devices: the very assets that enable connectivity, communication, and control of the data being sent between the lighting devices and the parts of the system that process and respond to that information.

An analysis of the recommended controls for each threat was performed to determine which controls could and should be put in place by manufacturers or third-party suppliers, and which controls need to be left up the end-user to implement. Sixty-three percent (36/57) of the primary controls were deemed to be the responsibility of the manufacturer to implement, representing a significant portion of the overall attack surface. If the manufacturer has done their due diligence in securing their products, the end user's attack surface appears much more manageable, and less expensive to operate and control. However, threats deemed to be the primary responsibility of the manufacturer—for which the end user may not be confident a sufficient control was put in place—might still require or justify mitigation by a complimentary control. The use of such compensating controls is generally recommended as part of a defense-in-depth strategy, where if one control fails, there is another in place to contain the incident.

The manufacturer responsibility varied with STRIDE category, as shown in Figure 18. Manufacturer controls can significantly reduce the attack surface and end-user responsibilities for Tampering, Information Disclosure, and Denial of Service threats. On the other hand, end users are largely responsible for addressing Repudiation and Elevation of Privilege threats. Hybrid systems consisting of devices from more than one vendor (i.e., CLS E and F) created larger attack surfaces for all STRIDE categories.

Figure 18: The attack surface for all threats (gray bubbles) and those left to the user to control and mitigate (colored bubbles).

Cloud, on-premise, and lighting technology choices, along with network configuration and installation size, will affect the attack surface. Twelve percent (7/57) of the threats found in CLS A, CLS C, CLS E, and CLS F were specific to interaction with the on-premise server, field gateway, and API, and seven percent (4/57) of the threats found in CLS B, CLS D, CLS E, and CLS F were specific to interaction with the cloud gateway, cloud database, and API. While the difference in threats that exists between cloud-based systems and on-premise systems may seem slim or marginal from a quantitative perspective, the decision to maintain on-premise servers versus outsourcing to a third party is not to be taken lightly. Having the proper team and upfront funding in place to operate and maintain, house, and secure the hardware; stay within compliance; and keep a reliable backup option are all considerations. Additional considerations for the end user would include whether utilizing cloud technologies with shared responsibility and access to systems is appropriate for their use case.

Threats will exist in all connected devices; this is true for any system. The more devices connected to a network, the larger the attack surface becomes. Additional insight can be gained when scaling the connected streetlights x1000 and the number of gateways x3 on each system, as seen in Figure 19 for high-risk Spoofing threats. The change in attack surface with additional streetlights helps to easily identify threats that pertain specifically to the luminaires as opposed to the other system assets. This type of visualization is beneficial to the lighting manufacturer and end user in understanding how the luminaires themselves affect the attack surface and if left uncontrolled, the sheer number of devices that would be left vulnerable.

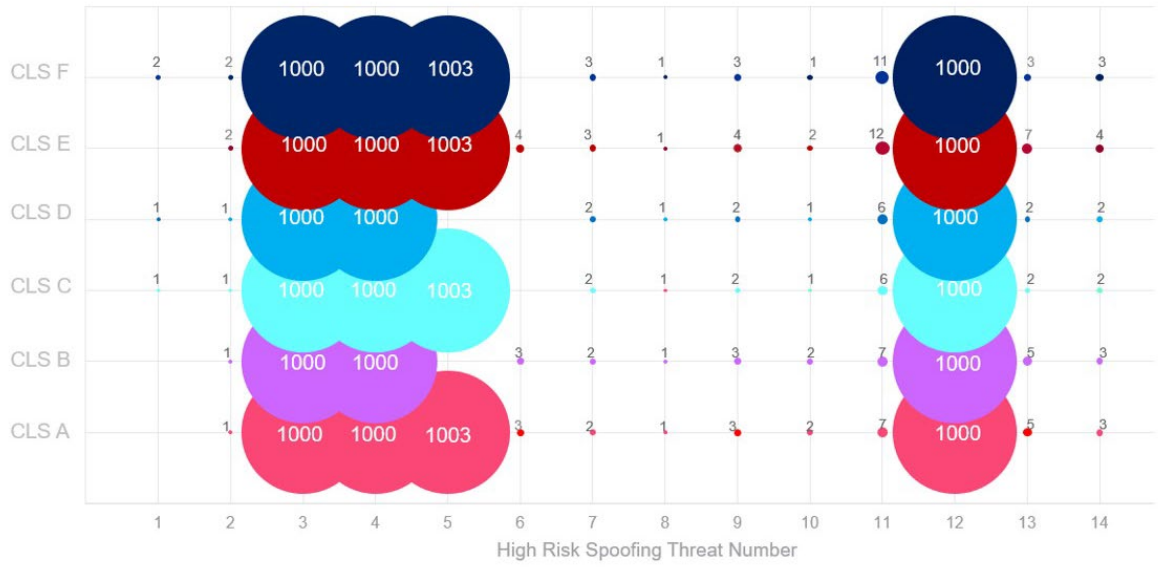

Figure 19: The attack surface for high-risk Spoofing threats when the number of lights are scaled x1000 and the number of gateways are scaled x3.

The end user depends on the device manufacturer to implement a control in some cases, like high-risk Tampering threats 1 and 4, as shown in Figure 20. High-risk Tampering threat 1 involves encrypting and signing the firmware image to ensure it is not tampered with. This should happen at the manufacturer level. When observing scaling results in a quantitative manner such as this, however, the total number of entry points or attack surface size should not be confused with the impact of an attack. One example is that a single uncontrolled DoS threat could leave the whole system inoperable. Looking at the bubble graph in a scaled manner such as this, however, does not convey that point. That is why it is important to approach security from both a quantitative and qualitative standpoint: to be able to prioritize what threats need which controls to obtain the most secure coverage possible.

Figure 20: The attack surface for high-risk Tampering, Repudiation, and DoS threats when the number of lights are scaled x1000 and the number of gateways are scaled x3.

In Figure 21, the EOP threats that scale when incorporating 1000 luminaires are easily identifiable. If improperly configured, the lighting devices could expose the following threats:

- Threat 3 – An adversary may get access to the admin interface or privileged services like Wi-Fi, SSH, file shares, etc. on a device.
- Threat 7 – An adversary may use unused features or services on CLS IoT devices such as USB ports, UI, etc.
- Threat 8 – An adversary may leverage insufficient authorization checks on the field gateway and execute unauthorized commands remotely.

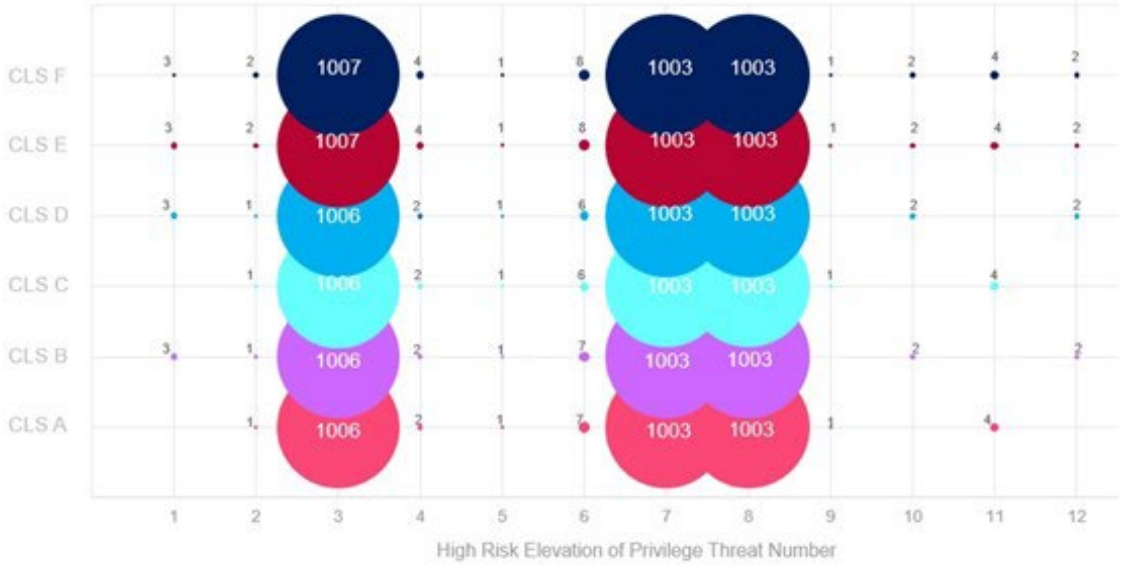

Figure 21: The attack surface for high-risk Elevation of Privilege threats when the number of lights are scaled x1000 and the number of gateways are scaled x3.

Summary and Recommendations

This CLS threat profile identifies threats that are mapped to specific system components. It also provides mitigations for each distinct threat–asset pairing. The outputs are actionable controls and facilitate an understanding of risk that informs the decision makers who are most concerned with optimizing impact or cost. Not all threats must be mitigated and perhaps not all threats can be addressed in a cost-effective way; however, without the proper controls in place, an organization will remain at risk. This CLS threat profile provides a foundation for a thorough understanding of possible threats for the CLS development team, testing team, management, and stakeholders. It will provide decision makers at all levels the opportunity to improve the security posture of the system. This will lead to more secure software and better-understood security.

The following research questions were answered through the findings of this study:

Q1) What are the cybersecurity threats associated with a CLS?

A1) This threat profile identified 57 threats spanning the six STRIDE categories, as summarized in Table 5.

Table 5: Distribution of threats mapped to the STRIDE framework.

Threat Type	High Priority	Medium Priority	Low Priority	Total
Spoofing	14	0	0	14
Tampering	4	10	0	14
Repudiation	2	1	0	3
Information Disclosure	11	2	0	13
Denial of Service	1	0	0	1
Elevation of Privilege	12	0	0	12
Total	44	13	0	57

Element and data flow templates that most closely matched or represented those found in a CLS were selected for the threat models. While the generated threats are still applicable and relevant, some may be of a more general nature as opposed to a specific known threat for a particular protocol or technology. The availability or development of a body of knowledge for the MTMT that specifically addresses building system protocols and technologies might produce more accurate or specific threats.

Q2) How does CLS architecture, authentication mechanism, and installation size change the attack surface?

A2) Utilizing hybrid systems with cloud and on-premise technologies created larger attack surfaces than other single systems. Twelve percent of the threats were attributed to on-premise architecture, and seven percent attributed to cloud architecture. Active Directory increased the interactions between systems and users regarding authentication checks. These added interactions increased the attack surface. Threats were identified in all assets and thus scaled accordingly with each added device. Threats associated with devices that will scale more significantly as a system grows will see a dramatically increased attack surface, which may or may not increase risk, and should not be confused with the impact of an attack.

Q3) Who is responsible for implementing the controls suggested by the threat profile?

A3) Sixty-three percent of the primary controls that would help secure the six modeled CLS should be built in by the manufacturer of the connected devices. The remaining thirty-seven percent fall on the end user or system owner to implement. Additionally, some form of defense-in-depth strategy to control potential weak points—or stop the kill chain, where an attacker attempts to methodically insert themselves into a system and gain persistence—is standard practice in many organizations.

Based on the results of this work, recommendations are made to key industry stakeholders, and a set of next steps are identified that might lead to the definition of future work.

Technology developers should:

- Mitigate all threats that can be reasonably controlled with baked-in technology solutions (e.g., encryption, authentication); see rows 1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41, 42, 48, 49, 50, 51, 52, 53, 56, 57 in the appendix table.
- Review threats involving sourced components (e.g., firmware, sensors, processors) used in the build of a CLS to understand how secure supply-chain management or a lack thereof will impact the final product.
- Consider adopting (and contributing to the development of, if/as necessary) standard interoperable protocols for CLS to aid in secure horizontal and vertical communications between IoT assets.

End users should:

- Conduct a threat profile of their existing or under-consideration CLS, together with any system that might be integrated with the CLS, and evaluate to determine if proper controls currently exist and are effective in mitigating threats.
- Compare the threat profile differences between cloud and on-premise systems or implementations of a particular vendor solution to determine which is more suitable for their needs and the abilities of their security team.
- Understand which controls they are responsible for implementing and maintaining, and which controls they could, and perhaps should, require from or collaborate on with their vendors or third-party suppliers; see rows 1, 3, 4, 5, 12, 13, 19, 22, 25, 27, 34, 35, 37, 39, 42, 43, 44, 45, 46, 47, 48, 52, 53, 54, 55 in the appendix table.

Standards and specification developers and regulators should:

- Define minimum recommended controls for CLS, as appropriate.
- Define minimum mandatory (regulated) controls for CLS, as appropriate.

Next Steps:

- Enhance and automate a portion of the threat analysis by creating templates and stencils in the MTMT that complement the current body of knowledge and define relevant elements, data flows, and processes that exist in CLS to provide more accurate and detailed models.
- Create additional threat profiles for other system architectures and integrations, using the refined template, to enhance the prioritization and identification of high-risk attack vectors for CLS.
- Characterize commercially available CLS to determine if controls that could and should be provided by the manufacturer are actually in place.
- Map all or a prioritized set of the identified threats to the MITRE ATT&CK® Matrix[16] to discern relevant techniques and tactics that might be used by malicious actors, and identify potential mitigations and best practices that might be adopted by CLS vendors and end users.

Citations

1. "DOE's National Roadmap for Grid-Interactive Efficient Buildings." *Office of Energy Efficiency & Renewable Energy DOE's National Roadmap for Grid-Interactive Efficient Buildings*, DOE, www.energy.gov/eere/articles/does-national-roadmap-grid-interactive-efficient-buildings.

2. Elliott, Clay. Energy Savings Forecast of Solid-State Lighting in General Illumination Applications. United States: N. p., 2019. Web. doi:10.2172/1607661.

3. M. N. I. Sarker, M. N. Khatun, G. M. Alam and M. S. Islam, "Big Data Driven Smart City: Way to Smart City Governance," 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1-8, doi.org/10.1109/ICCIT-144147971.2020.9213795.

4. Pandharipande, A. "Lighting Controls: Evolution and Revolution." *Lighting Research and Technology*, vol. 50, no. 1, 9 Jan. 2018, pp. 115–128., doi.org/10.1177%2F1477153517731909.

5. "Cybersecurity Building Control Systems." *Nationalacademies.org*, Federal Facilities Council, 24 Mar. 2015, www.nationalacademies.org/event/03-24-2015/federal-facilities-council-cybersecurity-building-control-systems.

6. 42, U., 2020. 2020 Unit 42 IoT Threat Report 2020 Unit 42 IoT Threat Report. *Unit42*. Available at: https://unit42.paloaltonetworks.com/iot-threat-report-2020/ [Accessed January 27, 2021].

7. IBM, 2020. Cost of a Data Breach Study. *IBM*. Available at: https://www.ibm.com/security/data-breach [Accessed January 27, 2021].

8. Morgan, S., 2020. Cybercrime To Cost The World $10.5 Trillion Annually By 2025. *Cybercrime Magazine*. Available at: https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/ [Accessed December 19, 2020].

9. Forrester, 2020. The Rise of the Business Aligned Security Executive. *Tenable.com*. Available at: https://static.tenable.com/marketing/whitepapers/Forrester-The_Rise_Of_The_Business-Aligned_Security_Executive.pdf [Accessed January 27, 2021].

10. Charlton, M., 2016. Hacking lightbulbs. *Dalhousie News*. Available at: https://www.dal.ca/news/2016/11/07/hacking-lightbulbs--phd-student-earns-international-attention-fo.html [Accessed February 4, 2021].

11. Ronen, E. and Shamir A., "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, 2016, pp. 3-12, doi: 10.1109/EuroSP.2016.13

12. NIST sp800-30 Initiative, J.T.F.T., 2012. Guide for Conducting Risk Assessments. *CSRC*. Available at: https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final [Accessed January 27, 2021].

13. Fitch, S. and Muckin, M., 2019 *A Threat-Driven Approach to Cyber Security* [white paper]. Available at: Lockheed Martin https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Threat-Driven-Approach.pdf [Accessed February 10, 2021].

14. Jegeib, Baldwin, M. & Kess, B., 2017. Mitigations - Microsoft Threat Modeling Tool - Mitigations. *Mitigations - Microsoft Threat Modeling Tool - Azure | Microsoft Docs*.

15. Shostack, Adam. “Stride Chart.” *Microsoft Security Blog*, Microsoft, 4 June 2020, www.microsoft.com/security/blog/2007/09/11/stride-chart/.

16. Anon, 2020. ATT&CK® for Industrial Control Systems. *attackics*. Available at: https://collaborate.mitre.org/attackics/index.php/Main_Page [Accessed March 8, 2021].

Glossary

Definitions from NIST.gov/glossary

Term	Definition
Attack Surface	The set of points on the boundary of a system, a system element, or an environment where an attacker can try to enter, cause an effect on, or extract data from, that system, system element, or environment.
Control	A safeguard or countermeasure prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined security requirements.
Mitigation	Prioritizing, evaluating, and implementing the appropriate risk-reducing controls/countermeasures recommended from the risk management process.
Threat	Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability.
Threat Model	A form of risk assessment that models aspects of the attack and defense sides of a particular logical entity, such as a piece of data, an application, a host, a system, or an environment [NIST SP 800 -154].
Threat Profile	A comprehensive report that illustrates a system's associated threats, security impact, likelihood, and suggested controls to mitigate the risk.
Vulnerability	Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

Acronyms

AAD	Azure Active Directory
ADAL	Active Directory Access List
AES	Advanced Encryption Standard
CA	Certificate Authority
CIA	Confidentiality, Integrity, Availability
CLS	Connected Lighting System
DTLS	Datagram Transport Layer Security
HBI	High Business Impact
IoT	Internet of Things
OSA	Open-Source Analysis
PII	Personally Identifiable Information
PNNL	Pacific Northwest National Laboratory
SaS	Shared access Signature
SAST	Static Application Security Testing
SSC	Secure Software Central
SSD	Secure Software Development
SSL	Secure Socket Layer
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
TBA	Threat-Based Analysis
TLS	Transport Layer Security
TMT	Threat Modeling Tool

Appendix

Threat Findings Table

Details for all the threats, the mapping of those threats to categories, example threats, and associated mitigations are documented in the following table. Mitigations are the main objective and describe what will be done to prevent, deter, or minimize the threat. The labels captured in parentheses in the "Interactions" column of the threat profile table below refer to the six CLS network diagrams shown in Figures 3–8. The label refers to an interaction (arrow) in the system diagrams, thus showing to which interaction and which components the threat corresponds. For example, a label such as D1_I09 refers to the three systems that use Active Directory and the arrow labeled D1_I09 will be the interaction corresponding to the threat. Systems without Active Directory will have a label that starts with D2 and are mapped accordingly. This strategy enables 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. The threat and mitigation columns provide the details to explain the situation for the purposes of due diligence, traceability, and risk management, while the responsibility column is meant as a general reference as to who bears the primary responsibility of implementing the control (manufacturer and/or end user). Not all interactions appear on all systems as a result of their varying architectures.

#	Threat Type	Threat	Mitigation	Interactions	Responsibility
1	Spoofing	On a public client (e.g., a mobile device), refresh tokens may be stolen and used by an attacker to obtain access to the API.	Depending on the client type, there are different ways that tokens may be revealed to an attacker and therefore different ways to protect them, some involving how the software using the tokens requests, stores, and refreshes them. For example, in an Azure environment, use Active Directory Access List (ADAL) libraries to manage token requests from OAuth2 clients to the Azure Active Directory (AAD) (or on-premises AD).	(D2_I02, D2_I07)	Manufacturer and User
2	Spoofing	An adversary may get access to Shared access Signature (SaS) tokens used to authenticate to the Internet of Things (IoT) Hub. If the lifetime of these tokens is not finite, the adversary may replay the stolen tokens indefinitely.	Determine and verify if vendor allows configuration of SaS token lifetime. If so, use finite lifetimes for generated SaS tokens.	(D1_I29,D1_I33,D2_I29,D2_I33)	Manufacturer
3	Spoofing	An adversary may replace the CLS IoT devices or part of the CLS IoT devices with some other CLS IoT device.	Assure that devices connecting to the field or cloud gateways are authenticated.	(D1_I29,D1_I33, D2_I29,D2_I33)	User
4	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.	Ensure that devices connecting to the field or cloud gateways are authenticated.	(D1_I29,D2_I29)	User
5	Spoofing	An adversary may gain access to the field gateway by leveraging default login credentials.	Ensure that the default login credentials of the field gateway are changed during installation. Integrate this check into the configuration guide for end-user configured systems.	(D1_I31, D1_I29)	Manufacturer and User
6	Spoofing	An adversary can get access to a user's session by replaying authentication tokens.	Ensure that <TokenReplayCache> is used to prevent the replay of ADAL authentication tokens.	(D1_I08, D1_I04, D1_I01,D1_I06)	Manufacturer
7	Spoofing	Attackers can exploit weaknesses in the system to steal user credentials.	Ensure web application implementation uses standard best practices for secure	(D1_I05, D1_I11, D1_I13, D2_I05, D2_I11, D2_I13)	Manufacturer

		Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if credentials are stored and sent in clear text, weak input validation is coupled with dynamic sql queries, or password retrieval mechanisms are poor.	transactions (i.e., does not store credentials in plain text on the client side, does implement input validation for forms, explicitly disables the autocomplete HTML attribute in sensitive forms and inputs).		
8	Spoofing	An adversary may predict and generate valid security tokens to authenticate to the IoT Hub by leveraging weak encryption keys.	Generate a random symmetric key of sufficient length according to standard best practices for the implemented encryption scheme for authentication to the IoT Hub. For example, the minimum key length for the Advanced Encryption Standard (AES) implemented scheme would be 256 bits (as of June 2020).	(D1_I29,D1_I33, D2_I29, D2_I33)	Manufacturer
9	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.	Implement finite lifetime for authentication credentials. Notify end user of last login time. Use a standard encryption scheme (i.e., Transport Layer Security (TLS)).	(D1_I09,D1_I05,D1_I11,D1_I13,D2_I05,D2_I11, D2_I13)	Manufacturer
10	Spoofing	The session cookies are the identifiers by which the server knows the identity of the current user for each incoming request. If the attacker can steal the user token, they would be able to access all user data and perform all actions on behalf of the user.	Set up session for inactivity lifetime.	(D1_I05, D1_I08,D2_I05)	Manufacturer
11	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.	Use a standard authentication mechanism to authenticate to the web application.	(D1_I09,D1_I05,D1_I11,D1_I13,D1_I20,D1_I18,D1_I23,D1_I21,D1_I10,D1_I12,D1_I02,D1_I07 ,D2_I05, D2_I11, D2_I13,D2_I20, D2_I18, D2_I02, D2_I23,D2_I21, D2_I10, D2_I12, D2_I07)	Manufacturer

12	Spoofing	An attacker may extract cryptographic key material from CLS IoT devices either at the software or hardware level and subsequently access the system with a different physical or virtual CLS IoT device under the identity of the CLS IoT device from which the key material has been taken. A good illustration is remote controls that can turn on any TV and that are popular prankster tools.	Use per-device authentication credentials. Do not use shared credentials among devices.	(D1_I29,D1_I33,D2_I29,D2_I33) * If each device can communicate independently with the IoT field gateway the threat would scale equal to the number of devices on the system.	Manufacturer and User
13	Spoofing	An adversary can bypass authentication due to non-standard Azure AD authentication schemes.	Use standard authentication scenarios supported by AAD.	(D1_I01,D1_I04, D1_I05,D1_I06,D1_I08,D1_I11,D 1_I13, D2_I05,D2_I11,D2_I13)	User
14	Spoofing	An adversary takes advantage of unsecured connections and transmissions on a network via HTTP instead of a more secure HTTPS due to misconfigured X.509/TLS certificate parameters.	Verify that the X.509 certificates used to authenticate the Secure Socket Layer (SSL), TLS, and the DatagramTransport Layer Security (DTLS) connections are using the correct Certificate Authority (CA).	(D1_I09,D1_I05,D1_I11,D1_I13,D 2_I05,D2_I11, D2_I13)	Manufacturer
15	Tampering	An adversary may launch malicious code into the CLS IoT devices and execute it.	Encrypt and/or sign firmware image (especially during transit) to ensure it has not been tampered with. When possible, configure device to only execute a specific set of known processes (process whitelisting).	(D1_I14,D1_I15,D1_I16,D1_I17,D 1_I24,D1_I25,D1_I26,D1_I27,D1_ I28,D1_I29,D1_I31,D1_I32,D1_I3 3,D1_I34) D2_I14,D2_I15,D2_I16,D2_I17,D 2_I24,D2_I25,D2_I26,D2_I27,D2_ I28,D2_I29,D2_I31,D2_I32,D2_I3 3,D2_I34)	Manufacturer
16	Tampering	An adversary may inject malicious inputs into an API and affect downstream processes.	Ensure that input validation is done on Web API methods. Limit the scope and functionality of each Web API method (Principal of Least Privilege).	(D1_I20,D1_I18, D1_I02, D1_I23, D1_I21,D1_I10, D1_I12, D1_I07, D2_I20,D2_I18, D2_I02, D2_I23, D2_I21,D2_I10, D2_I12, D2_I07)	Manufacturer
17	Tampering	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 less-direct attack injects	Ensure that type-safe parameters are used in web application for data access. Configure queries to be parameterized from web application. Limit the types of commands that can be executed on the database by the web application user (Principal of Least Privilege). Back up database transactions.	(D1_I20, D1_I18,D1_I02, D1_I23, D1_I21, D1_I10,D1_I12, D1_I07, D2_I20, D2_I18,D2_I02, D2_I23, D2_I21, D2_I10,D2_I12, D2_I07)	Manufacturer

		malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.			
18	Tampering	An adversary may launch malicious code into the CLS 1A IoT device and execute it.	When applicable, ensure that unknown code cannot execute on devices. Encrypt and/or sign firmware image (especially during transit) to ensure it has not been tampered with. When possible, configure device to only execute a specific set of known processes (process whitelisting).	(D1_I14,D1_I15, D1_I16,D1_I17,D1_I24,D1_I25,D1_I26,D1_I27) (D2_I14,D2_I15,D2_I16,D2_I17,D2_I24,D2_I25,D2_I26,D2_I27)	Manufacturer
19	Repudiation	An adversary may perform actions (e.g., spoofing attempts, unauthorized access) on the cloud gateway. It is important to monitor these attempts so adversaries cannot deny these actions.	Ensure that appropriate auditing and logging is enforced on the cloud gateway.	(D1_I34, D1_I33, D1_I31,D1_I29,D2_I34, D2_I33,D2_I31,D2_I29)	Manufacturer and User
20	Repudiation	Attacker can deny a malicious act on an API, leading to repudiation issues.	Ensure that auditing and logging is enforced on the Web API.	(D1_I20, D1_I10, D1_I18, D1_I02,D1_I23, D1_I21, D1_I12,D1_I07, D2_I20, D2_I10, D2_I18, D2_I02,D2_I23, D2_I21, D2_I12,D2_I07)	Manufacturer
21	Information Disclosure	An adversary may conduct a man-in-the-middle attack and downgrade the TLS connection to clear the text protocol or to 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 ensured.	Applications available over HTTPS must use secure cookies. Force all traffic to Web APIs over the HTTPS connection. Configure cookies with a finite lifetime.	(D1_I05,D2_I05)	Manufacturer
22	Information Disclosure	If an attacker gains access to the database, and if database security controls such as Transparent Data Encryption, Column Level Encryption, EKM, etc., are not being used, the attacker can more easily identify and extract high-value Personally Identifiable	Configure database to use at-rest encryption, if available. For databases without a configurable encryption feature, accept the risk of unencrypted data-at-rest (implementing encryption is not worth the risk of increasing the attack surface).	(D1_I30,D1_I22, D2_I30, D2_I22)	Manufacturer and User

		Information (PII) or High Business Impact (HBI) data.			
23	Information Disclosure	An adversary can reverse weakly encrypted or hashed content.	Do not expose security details in error messages. Implement default error handling page. Use a standard encryption library and best practice configurations (i.e., do not use known weak algorithms and key lengths). Verify that X.509 certificates are used to authenticate SSL, TLS, and DTLS connections.	(D1_I09,D1_I05, D1_I11,D1_I13,D2_I05, D2_I11, D2_I13)	Manufacturer
24	Information Disclosure	An adversary can gain access to sensitive data such as the following, through verbose error messages: 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.	Do not expose system, sensitive, or attributable details in error messages.	(D1_I09,D1_I05, D1_I11, D1_I13,D1_I20, D1_I10, D1_I18, D1_I02,D1_I23, D1_I21, D1_I12, D1_I07, D2_I05, D2_I11, D2_I13, D2_I20,D2_I10, D2_I18, D2_I02, D2_I23,D2_I21, D2_I12, D2_I07)	Manufacturer
25	Information Disclosure	If application saves sensitive PII or HBI data on phone Secure Digital (SD) card or local storage, then it may be stolen.	Do not store PII or sensitive data on a mobile device.	(D1_I02, D1_I07, D2_I02, D2_I07)	User
26	Information Disclosure	An adversary may eavesdrop and interfere with the communication between CLS IoT devices and gateways and possibly tamper with data that is transmitted.	Encrypt communication using SSL/TLS. (A Denial of Service attack on the communication requires additional mitigation and is addressed with that specific threat).	(D1_I33,D1_I29,D2_I33,D2_I29)	Manufacturer
27	Information Disclosure	An adversary may gain access to sensitive data from an uncleared browser cache.	Ensure that sensitive content is not cached on the browser.	(D1_I05,D2_I05)	Manufacturer and User
28	Information Disclosure	An adversary may gain access to unmasked sensitive data such as credit card numbers.	Ensure that sensitive data displayed on the user screen is masked.	(D1_I05,D2_I05)	Manufacturer
29	Information Disclosure	An adversary may gain access to sensitive data from log files.	Ensure that the application does not log sensitive user data.	(D1_I09,D1_I05, D1_I11, D1_I13, D2_I05, D2_I11, D2_I13)	Manufacturer
30	Information Disclosure	An adversary can gain access to sensitive data by sniffing traffic to the Web API.	Force all traffic to Web APIs over an HTTPS connection.	(D1_I20, D1_I10, D1_I18,D1_I02, D1_I23, D1_I21, D1_I12,D1_I07, D2_I20, D2_I10, D2_I18,D2_I02, D2_I23, D2_I21, D2_I12,D2_I07)	Manufacturer

31	Information Disclosure	An adversary can gain access to sensitive data by sniffing traffic from a mobile client.	Implement standard authentication mechanisms for mobile devices. Force all traffic to Web APIs over an HTTPS connection.	(D1_I04,D2_I02, D2_I07)	Manufacturer
32	Denial of Service	Failure to restrict requests originating from third-party domains may result in unauthorized actions or access to data.	Restrict web app requests to web app/client browser connection. Use standard authentication from the web browser.	(D1_I05,D2_I05)	Manufacturer
33	Elevation of Privileges	An adversary may gain elevated privileges on the functionality of the cloud gateway if SaS tokens with overprivileged permissions are used to connect.	Connect to the cloud gateway using least-privileged tokens (principle of least privilege). Configure SaS tokens with finite lifetimes.	(D1_I33,D2_I33)	Manufacturer
34	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.	Ensure that administrative interfaces are appropriately locked down. Enforce sequential step order when processing business logic flows. Ensure that proper authorization is in place and that the principle of least privileges is followed. Business logic and resource access authorization decisions should not be based on incoming request parameters. Ensure that content and resources are not enumerable or accessible via forceful browsing.	(D1_I05,D2_I05)	Manufacturer and User
35	Elevation of Privileges	An adversary may get access to the admin interface or privileged services like Wi-Fi, SSH, file shares, FTP etc., on a device.	Ensure that all admin interfaces are secured with strong credentials. Limit the number of services offered to those that are needed. Disable unneeded services.	D1_I14,D1_I15, D1_I16,D1_I17,D1_I24,D1_I25,D1_I26,D1_I27,D1_I28,D1_I29,D1_I30,D1_I32 D1_I33,D1_I35 D2_I14,D2_I15,D2_I16,D2_I17,D2_I24,D2_I25,D2_I26,D2_I27, D2_I28,D2_I29,D2_I30,D2_I32 D2_I33,D1_I35)	Manufacturer and User
36	Elevation of Privileges	Database user gains access to data access or configuration privileges that violate that user's need to know or access authority.	Ensure that least-privileged accounts are used to connect to the database server. Implement Row Level Security (RLS) to prevent tenants from accessing each other's data. Sysadmin role should only have valid necessary users.	(D1_I19,D1_I22,D1_I30,D1_I35 , D2_I19,D2_I22,D2_I30,D2_I35)	Manufacturer

37	Elevation of Privileges	An adversary may jail break into a mobile device and gain elevated privileges.	Implement implicit jailbreak or rooting detection. Limit the API calls the mobile device can make to only allowed calls. Enforce this at the system level (at API management) rather than only at the device application.	(D1_I04,D2_I02, D2_I07)	Manufacturer and User
38	Elevation of Privileges	An adversary may gain unauthorized access to the Web API due to poor access control checks.	Implement standard authentication for access to the Web API.	(D1_I20, D1_I10,D1_I18, D1_I02, D1_I23, D1_I21,D1_I12, D1_I07, D2_I20, D2_I10, D2_I18, D2_I02, D2_I23, D2_I21,D2_I12, D2_I07)	Manufacturer
39	Elevation of Privileges	An adversary may use unused features or services on CLS IoT devices such as UI, USB port, etc. Unused features increase the attack surface and serve as additional entry points for the adversary.	Limit the number of services offered to those that are needed. Disable unneeded services.	(D1_I14,D1_I15, D1_I16,D1_I17,D1_I24,D1_I25,D1_I26,D1_I27,D1_I28,D1_I32) D2_I14,D2_I15,D2_I16,D2_I17,D2_I24,D2_I25,D2_I26,D2_I27 D2_I28,D2_I32)	User
40	Elevation of Privileges	An adversary may leverage insufficient authorization checks on CLS IoT devices and execute unauthorized and sensitive commands remotely.	Perform authorization checks in the device if it supports various actions that require different permission levels. Limit the API calls the mobile device can make to only allowed calls. Enforce this at the system level (at API management) rather than only at the device application.	(D1_I32,D1_I28, D2_I32, D2_I28)	Manufacturer
41	Elevation of Privileges	An adversary may leverage insufficient authorization checks on the field gateway and execute unauthorized and sensitive commands remotely.	Perform authorization checks in the device if it supports various actions that require different permission levels. Limit the calls the IoT device can make to only allowed calls. Enforce this at the field gateway level rather than only at the IoT device.	(D1_I29,D2_I29)	Manufacturer
42	Elevation of Privileges	An adversary can gain unauthorized access to Azure Database for PostgreSQL instances due to weak network security configuration.	Restrict access to Azure Postgres DB instances by configuring server-level firewall rules to only permit connections from selected IP addresses where possible (e.g., whitelist access based on IP address).	(D1_I19, D1_I35, D2_I19, D2_I35)	Manufacturer and User
43	Elevation of Privileges	If there is no restriction at the network or host firewall levels, then anyone can attempt to connect to the database from an unauthorized location.	Restrict connection requests to known and expected IP sources (e.g., using IP whitelisting via a firewall).	(D1_I30,D1_I22, D2_I30, D2_I22)	User

44	Elevation of Privileges	An adversary can gain long-term, persistent access to an Azure Database for PostgreSQL instance through the compromise of local user account password(s).	Rotate user account passwords (e.g., those used in connection strings) regularly. Store secrets in a secret storage solution (e.g., Azure Key Vault). Log user account access to database and periodically audit for anomalous access.	(D1_I19, D1_I35, D2_I19,D2_I35)	User
45	Tampering	An adversary can tamper with critical database securables and deny the action.	Add digital signature to critical database securables. Back up database securables and log transactions.	(D1_I30, D1_I22)	User
46	Tampering	An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to database.	Enable threat detection on Azure SQL database or equivalent service as available.	(D1_I30, D1_I22, D2_I30, D2_I22)	User
47	Tampering	An adversary may read and/or tamper with the data transmitted to the Azure Database for PostgreSQL due to weak configuration.	Enforce communication between clients and Azure Postgres DB to be over SSL/TLS by enabling the Enforce SSL connection feature on the server. For MySQL, check that the connection strings used to connect to MySQL databases have the right configuration (e.g., ssl = true, or sslmode = require, or sslmode = true are set). Configure the MySQL server to use a verifiable SSL certificate (needed for SSL/TLS communication).	(D1_I19, D1_I35, D2_I19,D2_I35)	User
48	Tampering	An adversary may leverage known vulnerabilities and exploit a device if the firmware of the device is not updated.	Implement a patch management process to keep the connected device's firmware up to date.	(D1_I14,D1_I15, D1_I16,D1_I17,D1_I23,D1_I24,D1_I25,D1_I26,D1_I27,D1_I28,D1_I29, D1_I30,D1_I32 D1_I33, D1_I35) (D2_I14,D2_I15,D2_I16,D2_I17,D2_I23,D2_I24,D2_I25,D2_I26,D2_I27, D2_I28, D2_I29, D2_I30, D2_I32, D2_I33,D2_35)	Manufacturer and User
49	Tampering	An attacker steals messages off the network and replays them to steal a user's session.	Implement network encryption. Set user sessions to have a finite lifetime (note: this is a hard timeout, not an inactivity timeout).	(D1_I05,D2_I05)	Manufacturer

50	Tampering	An adversary can use various tools, reverse engineer binaries, and abuse them by tampering.	Limit the API calls the mobile device can make to only allowed calls. Enforce this at the system level (at API management) rather than only at the device application.	(D1_I04,D2_I02,D2_I07)	Manufacturer
51	Tampering	An adversary may partially or wholly replace the software running on CLS IoT devices, the IoT cloud gateway, database, or web APIs, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials are 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.	Store cryptographic keys securely. Only allow needed and known processes to execute. Disallow unknown or unneeded processes from executing.	(D1_I14,D1_I15, D1_I16,D1_I17,D1_20,D1_I23,D1_I24,D1_I25,D1_I26,D1_I27,D1_I29, D1_I30, D1_I33,D1_I34) (D2_I14,D2_I15, D2_I16,D2_I17,D2_20,D2_I23,D2_I24,D2_I25,D2_I26,D2_I27,D2_I29, D2_I30, D2_I33,D2_I34)	Manufacturer
52	Tampering	An adversary may perform a man-in-the-middle attack on the encrypted traffic sent to CLS IoT devices.	Verify that the X.509 certificates used to authenticate SSL, TLS, and DTLS connections are using the correct CA.	(D1_I28,D1_I32,D2_I28,D2_I32)	Manufacturer and User
53	Tampering	An adversary may gain unauthorized access to the IoT field gateway, tamper with its operating system, and access confidential information in the field gateway.	Implement standard authentication for access to web API.	(D1_I23, D1_I30,D1_I28, D2_I23, D2_I30, D2_I28)	Manufacturer and User
54	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.	When applicable and available, encrypt OS and additional partitions of the IoT field gateway (e.g., Bitlocker for windows OS). If storing data on the field gateway, log transactions and take steps to secure it as necessary (e.g., encryption).	(D1_I14,D1_I15, D1_I16,D1_I17,D1_20,D1_I23,D1_I24,D1_I25,D1_I26,D1_I27,D1_I29, D1_I33, D2_I14,D2_I15, D2_I16,D2_I17,D2_20,D2_I23,D2_I24,D2_I25,D2_I26,D2_I27,D2_I29,D2_I33)	User
55	Repudiation	If an attacker is present and proper logging of security events, log rotation/separation, auditing, and other secure logging practices are not in place, the attacker may be able to work without being detected.	Ensure that auditing and logging is enforced on the application. Ensure that log rotation and separation are in place. Ensure that audit and log files have restricted access. Ensure that user management events are logged.	(D1_I09,D1_I05, D1_I11, D1_I13,D2_I05, D2_I11, D2_I13)	User

56	Information Disclosure	An adversary can reverse weakly encrypted or hashed content.	Do not expose security details in error messages. Implement default error handling page. Set deployment method to retail in Internet Information Services (IIS). Use only approved symmetric block ciphers and key lengths. Use approved block cipher modes and initialization vectors for symmetric ciphers. Use approved asymmetric algorithms, key lengths, and padding. Use approved random number generators. Do not use symmetric stream ciphers. Use approved MAC/HMAC/keyed hash algorithms. Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections.	(D1_I09,D1_I05, D1_I11, D1_I13, D2_I05,D2_I11, D2_I13)	Manufacturer
57	Information Disclosure	An adversary can gain access to the config files and compromise those files if sensitive data is stored in them.	Encrypt sections of the Web API's configuration files that contain sensitive data. Limit access to configuration files to properly authenticated and authorized users.	D1_I09,D1_I20, D1_I10, D1_I05,D1_I18, D1_I11, D1_I02, D1_I23,D1_I21, D1_I12, D1_I13, D1_I07, D2_I20, D2_I10, D2_I05, D2_I18,D2_I11, D2_I02, D2_I23, D2_I21,D2_I12, D2_I13, D2_I7)	Manufacturer

(This page intentionally left blank)

U.S. DEPARTMENT OF
ENERGY

Office of
ENERGY EFFICIENCY &
RENEWABLE ENERGY

For more information, visit:
energy.gov/eere/ssl

PNNL-31958 • February 2022