Creating A Security Baseline and Cybersecurity Framework for the Internet of Things Via Security Controls

Tony Hoang and Yanzhen Qu*

ABSTRACT

The Internet of Things (IoT) has revolutionized how we interact with the world and has become an essential ingredient for various industries for service efficiency and effectiveness. IoT is a common building block for automation tasks to help businesses enhance productivity and performance. IoT is observed anywhere and everywhere and in almost every industry. The technology that was supposed to make our lives easier brought forth a cybersecurity storm for which the world is unprepared. To mitigate this issue, these devices need a proper security baseline and cybersecurity framework to support them. This design science study proposed a framework to create a security baseline based on the foundation of security controls. The proposed framework uses the NIST SP800-53 controls as requirements for management, operational, and technical implementations. The study investigated how and which controls are selected for requirements. This approach can be used as a guideline for organizations to develop their security baseline to support and secure the IoT systems.

Keywords: Cybersecurity framework, Internet of Things, risk management, security controls.

1. INTRODUCTION

The Internet of Things (IoT) started as a concept envisioned by Kevin Aston in 1999 to promote Radio Frequency Identification (RFID) to improve supply chain activities [1]. The vision and potential of IoT have quickly caught on, making it one of the fastest-growing and most popular technologies in recent times [2]. IoT technology has evolved quickly over the years, ranging from light bulbs, smartphones, watches, appliances, cars, and industrial and medical equipment. IoT has become a key ingredient in many industries, including finance, health, manufacturing, agriculture, and anywhere it can be envisioned [3]. IoT is a crucial building block for automation and controls where human involvement is problematic. IoT helped doctors diagnose and treat patients without the need for direct contact. IoT helped detect and predict weather patterns faster, earlier, and more accurately. The technology that was supposed to make our lives easier has brought along with it a cybersecurity storm that the world is unprepared for. The Jeep hack in 2015, the Mirai botnet in 2016, and the Verkada hack in 2021 were evidence of the hidden dangers [4].

IoT security has been frowned upon by cybersecurity experts due to their lack of standards for development, a diverse hardware architecture, and a lack of long-term support [3]. Many IoT devices are inherently weak due to their small form factor, weak processor, and limited power. Despite their weaknesses, these devices are still in use today. Many of these devices that were part of the original botnet attack still reside in the comfort of our own homes. Adversaries rely on these weaknesses for traversal attacks, or they may remain hidden for advanced persistent threat attacks or, more increasingly, for cryptocurrency mining [3]. In some cases, ignoring their benefits cannot be avoided. IoT is a powerful tool. IoT usage can be a business enabler or breaker, depending on how they are being used. Without adequately protecting them and the information they store and process, the risk is substantial and can outweigh the benefits IoT brings to the organization. Introducing a security baseline with security controls as requirements create accountability and enforceability to protect systems and environments [6], [7]. It is a type of countermeasure that has proven to be effective. When properly deployed, a baseline can be used to establish a standard for enforcing and measuring the security posture.
for IoT devices and to ensure an adequate infrastructure foundation that supports them.

2. Related Works

2.1. NIST SP 800-53

The National Institute of Standards and Technology Special Publication 800-53 “Security and Privacy Controls for Information Systems and Organizations” (NIST SP 800-53) is a set of standards and guidelines to help U.S. federal agencies and their contractors to meet regulatory requirements and mandates set by the Federal Information Security Management Act (FISMA) [2]. It consists of management, operational, and technical controls to maintain safeguards for any system that processes, stores, and transmits critical information. Because of its effectiveness, many non-government-affiliated organizations have begun adopting it to improve their own infrastructure [2]. Trust is an essential security component that should be controlled, accountable, and measurable. This study utilizes the NIST SP 800-53 to establish requirements, quantify the effectiveness or risk of the security implementations, and measure the system’s trustworthiness.

2.2. MITRE ATT&CK Framework

Cybercriminals are known to be notoriously intelligent, persistent, and adaptive to new technologies, vulnerabilities, successes, and failures of attacks. We can learn a lot from them to develop effective and threat-informed defense strategies [8]. The MITRE Adversary Tactics, Techniques, and Common Knowledge (ATT&CK) framework is a globally accessible and ever-evolving knowledge base of adversary behavior [9]. The ATT&CK outlines common tactics, techniques, and procedures by adversaries [8]. The model intentionally takes the attacker’s perspective to help provide insights into how the adversaries prepare, approach, and execute the attacks. Understanding the ATT&CK provides a common language and foundation defenders can understand [9]. The ATT&CK framework helps defenders reduce vulnerabilities and recognize threats before adversaries can exploit them.

3. Problem Statement, Hypothesis Statements, and Research Questions

3.1. Problem Statement

According to IoT Analytics, as of May 2023, there are approximately 16,700,000,000 connected IoT devices globally, ranging from simple functions to highly complicated systems [10]. The number is expected to be much higher considering those devices operating behind private networks or offline. The problem is the lack of a universal framework for creating an IoT security baseline to evaluate the IoT’s security posture and to ensure the security requirements are maintained. Organizations often rely on the manufacturer for service, maintenance, and security needs, resulting in a void in cybersecurity management, oversight, and monitoring that is difficult to track [11]. Furthermore, the lack of tight cybersecurity control requirements creates diverse IT infrastructure for organizations to handle IoT devices effectively [12]. IoT devices often operate on the same network as critical operation devices, leading to potential attacks and delivery of malicious software and ransomware via traversal techniques [13]. The lack of a formal framework often leads to accepting risk with total disregard and failure to meet regulatory requirements.

3.2. Hypothesis Statement

It is possible to use NIST SP 800-53 controls, risk management principles, adversary tactics, and security best practices to create a security baseline and cybersecurity framework to manage and assess the security posture of IoT devices.

3.3. Research Question

How can the application of NIST SP 800-53 controls, along with risk management principles, adversary tactics, and security best practices, be used to create a security baseline and cybersecurity framework for the management and assessment of the security posture of IoT devices?

4. Methodology

4.1. Method

The research method selected for this study is Design Science Research (DSR), which aims to create and evaluate innovative solutions to real-world problems and prescriptive knowledge of information science. It is particularly suitable for designing artifacts, such as processes, models, frameworks, and methods, to address specific challenges [14]. Developing a framework for creating a security baseline for IoT devices, DSR can be a foundation for creating a well-structured and effective process for ensuring adherence to regulatory requirements or standards [15]. The artifact in this study is formulating a security baseline for IoT devices by applying real-world applications, observation of phenomena, threat hunting using adversary tactics, risk management principles, and security best practices. The configuration verification and implementation steps were tied to a well-defined set of controls to ensure the security baseline is accountable, repeatable, and manageable.

Though not typically required, tying it to a set of controls helps to monitor and track requirements as the device or application evolves, migrates, or updates over time. Furthermore, the control can be tied to other systems as well. For example, if you enforce password complexity requirements for one system, the exact requirements can be enforced on other systems. The controls adopted for this study are the NIST 800-53 revision 5. The NIST SP 800-53 is a well-established control framework adopted by U.S. Government agencies, contractors, and organizations worldwide [16]. The selection of required controls changes as security requirements change. Which controls to apply depends on the type of organization, the data classification, and who has access to the data. In many cases, the controls should be established before enforcing any configurations. How and which controls were selected was part of the framework development process for the
baseline. The same way these experts selected their controls can be applied to non-government organizations.

4.2. Population and Sample

The target population for this study is high-end IoT devices that are organization-owned or adopted and can be managed. The organization solely manages some of these devices or has shared responsibilities between the organization and their manufacturers. The targeted controls are selected from the NIST SP 800-53 rev 5, which consists of 1190 different controls [17]. An opposing control or set of controls was selected for defence by evaluating the MITRE ATT&CK framework for threat intelligence and vulnerability identification [8]. A simulated IoT device with a system-on-chip and a Linux-based operation system was used as a validation and test tool to simulate a high-end IoT device. Most high-end IoT can handle more than simply sending and receiving sensor data but can perform advanced tasks such as user input, control interfaces, and processing [18]. These IoT devices are often used in major industries and labelled according to usage.

High-end IoT devices have a 4-layer architecture and are divided as follows: sensing, network, data processing, and application layer [19]. This research was not intended to exploit possible vulnerabilities but to understand how vulnerabilities work so that exploitation can be avoided when specific technical controls are met. The research focused on the most common technical controls for analysis and testing in creating the artifact. The targeted controls for baseline evaluation are selected from the 114 technical controls deemed IoT-relevant.

4.3. Artifact Creation

The system's foundation often relies on the underlying architecture as the foundation for system operations to interact with various software, hardware, and APIs that make them universally accepted. The architecture is observed in the layered design of systems ranging from simple two layers to seven layers found in complex systems. Reusing existing development frameworks makes development faster and often less prone to errors. Otherwise, developers must revert to the basic foundation when they begin a new application or hardware. Using a widely accepted Processor, chipset, kernel, operating system, libraries, and API is preferable and recommended to avoid unnecessary errors that arise from creating everything from the ground up [18], [20]. Despite focusing on small form factors, IoT developers often emphasize the importance of using a system of architecture they trust and are familiar with [21]. As these underlying systems become increasingly popular, the focus on security becomes more critical. They share security requirements because of their functionality, risk impact on humans or property, and the criticality of the information they handle, transmit, share, or store [22]. The following diagram represents the high-level assessment and development of the security baseline adapted from the NIST Risk Management Framework [23].

Fig. 1 presents a high-level security control selection and analysis strategy that can be used for the baseline. It was adapted from the NIST management framework [23]. Often presented as six and sometimes seven steps, the NIST RMF goes through a continuous process that grows with the organization. The diagram included the “Prepare” step but did not include the “Authorize” and “Monitoring” steps—the latter two focus on device operation rather than baseline development.

Step 1: “Prepare”. This step aims to leverage securities and activities already underway and what is coming into play with the new baseline. Before adopting a new baseline, an organization must establish a foundation to support it to ensure its security. Laws, regulations, stakeholders, data handling requirements, and organization requirements play a big part in identifying common and regulatory controls. One of the activities is establishing strong communication and commitment from senior executives and stakeholders. The involvement helps organization leaders understand the risks and benefits the organization would take on and its impact on the existing security posture [17].

Step 2: “Categorize”. This step aims to determine the new system’s impact on the information system [23]. Identifying and determining the impact value of the information system and data helps determine the appropriate security measure for the system. One way to identify the information's impact value is by using NIST 800-60 for information classification.

Step 3: “Select”. This step focuses on the control type selection. There are three types: management, operational, and technical [23]. Management and operational control deal with high-level organizational management and operations, whereas technical controls are those that can be implemented at the system level. Control responsibility can be shared between the organization and the providers. The providers may release patches and hotfixes, but it is up to the organization to be responsible for applying the patches when they are released [15].

Step 4: “Implement”. Once the controls are selected, it goes to the implementation step, which determines how they are carried out [23]. The organization also needs to document how the controls are implemented. These documents include configuration and implementation changes to make the system more secure as part of the control requirements [15].
Step 5: "Assess". This step aims to develop a way to assess and verify the controls. A baseline may be a form of a checklist that validates the configurations or a procedure to ensure that configurations were applied. Ideally, create one document to harden the system while the other to verify whether the system is hardened. The results of assessments can help verify and assure that the IoT devices are securely fit to be used in the environment [23].

4.4. Controls Selection Determination

One of the most daunting tasks is determining what control to select amidst the one-thousand-plus controls. One control may be tied to hundreds of configurations, operations, and technical changes. On the other hand, a single technical change may be tied to several controls. Also, how do we reject one control while accepting the other? The controls we selected may differ entirely from the next person attempting to recreate it. The variation is dependent on the circumstances that surround the selection. Consider the equation in Fig. 2 for assessment.

Each vulnerability has an impact severity (Impact\textsubscript{rev}) of high, moderate, low, or none that can be tied to security control. For quantitative purposes, we must assume that these values are not zero. Otherwise, what is the point of having these controls? There are several factors that will impact the selection of a control. Control is needed if the system does not handle sensitive data (Data\textsubscript{sen}) but is part of a network that contains sensitive data. Historically, we did not need control if there were no organization requirements (Req\textsubscript{org}) or management requirements (Req\textsubscript{mgt}). However, what if we want to start securing and tying our device to a control? Such as choosing to shut down my pacemaker’s wireless and Bluetooth interface. By tying the data sensitivity to the management requirements (Req\textsubscript{mgt} U Data\textsubscript{sen}), we establish a new control we chose to enforce. Doing so will establish accountability to management or personal requirements. In the case of personal IoT devices or organizations without an existing security framework, the decision to protect the system is typically the result of elevating the sensitivity of the data, collection of new sensitive data, or the information in the network that the system belongs to.

4.4.1. Applying Security Control to the Security Baseline

For the system and application to be versatile, they need to be able to adjust to the user’s needs [24]. One of the ways to do this is through changes in the configuration. A selection of screen size, a mapped drive, remote access, or multifactor authentication are some configuration changes that can be adjusted. The configuration can be implemented in many ways, through various user interface interactions, running a shell command, changing a group policy or chef recipe, or editing the configuration system registry. Of the possibilities listed, the change in configuration or registry is the most direct control implementation at the configuration level [23]. All other changes would ultimately transfer to these lower-level configurations. The alternative approach for making configuration changes is one of the reasons why many attacks often focus on misconfiguration [8]. Therefore, assessing and maintaining the security posture of the system based on these files and their dependent processes yields more reliable results [23].

4.4.2. Focusing on IoT Middleware Security Controls

It would take considerable time to address every aspect of security fully as it requires careful planning, documents, and collaboration from every level and personnel within the organization [25]. The work involves identifying administrative and physical control on top of technical control. Often, the administrative and physical controls are already in place if the organization is well established and has a wide range of systems and applications they protect [26]. This research intended to focus solely on a subset of security controls that can be managed by IT and security professionals working directly/indirectly with IoT devices. The artifact created for this study is to provide a window of opportunity for IT and cybersecurity professionals to be equipped with information that can be used to gain better visibility of the IoT device they own and a pathway to develop their process or to build on top of an existing process with information that can be obtained from the internet. Cybersecurity security controls can be implemented at all four layers (application, support, network, and perception) of the IoT architecture [19]. Despite their differences, the evaluation criteria remain the same.

1. Define objectives and scope: Determining the objectives and scope helps identify what you are trying to protect and why it is necessary. The objectives identified are information and configuration that can be controlled. This information helped derive the types of technical security control you intend to validate.

2. Identify validation criteria: This step aims to determine the criteria against which to evaluate the effectiveness of security controls. These criteria should be measurable, specific, and aligned with the intended security outcomes. Typical criteria include confidentiality, integrity, availability, and compliance. Validation criteria for IoT security controls being met typically involve assessing whether the implemented controls effectively address IoT devices and systems’ security and functionality requirements.

3. Establish metrics and indicators: This step aims to define quantitative metrics to measure the performance of each security control, which is the only way to determine if a security control has been successfully implemented and maintained. Metrics often include success rates, response times, incident counts, and compliance percentages. In this study, the metric focused on the compliance of established controls set forth by the organization and the security team.
Following the process in Fig. 3, readers can systematically evaluate the implementation of NIST SP 800-53 controls within their IoT environment. The evaluation helps identify security gaps, improve control effectiveness, and enhance the overall security of IoT systems and devices. The DoD STIGs and CIS Benchmarks were used as guidelines and references on how each control can be met and evaluated based on specific sets of commands. It is virtually impossible to create a single baseline for all IoT devices. However, an assessment can be made by correlating the controls and similar command types tied to a similar IoT device having a similar architecture, such as the same middleware or operating system.

5. Experiment and Results

5.1. Results

This study introduced a framework for establishing a security baseline for managing and accessing IoT devices. This study identified critical security controls to lay the foundation for a security baseline for IoT devices. The baseline can be used to identify vulnerabilities and verify compliance with organizational policies and requirements. The artifact presented in this study is intended to be flexible enough to be adapted to various types of organizations. The research considers that IoT devices are built differently; as such, the baseline can be adjusted to align with requirements, type of device, data sensitivity, functionality, environment, and other limitations. The research was broken into three separate stages for control identification and assessment:

- **Stage 1**: Cybersecurity foundation with management and operational controls,
- **Stage 2**: Identify security-relevant changes with operational and technical controls,
- **Stage 3**: Continuous oversight and long-term controls for IoT devices.

NIST SP 800-53 controls consist of management, operational, and technical categories [23]. A complicated IoT system resides on a highly classified network security and may include all the controls identified in all three stages. In contrast, more simplified ones may involve just technical controls identified in Stages 2 and 3 with the expectation that management and operational controls be enforced for all systems at a higher level. Introducing a new system may impact current systems or require changes to the existing environment [27]. Some changes, such as using cloud services for IoT data collection and processing, may require a review of the organization's overall infrastructure framework.

5.2. Evaluate the Effectiveness of the Framework

A technical control compliance evaluation (TCCE) checklist was created to ensure the validity and reliability of the framework for baseline creation. TCCE is a way to establish a baseline and validate that the security is met for a specific component of the IoT device. Keep in mind that a single system may have several baselines that it needs to meet (i.e., OS, application, communication). This framework aimed to identify security controls that come as close to the current computing system as possible since some IoT devices can perform and function in ways that rival existing enterprise computing systems. Cybersecurity capabilities are an essential consideration for applying a specific control or set of controls to it.

The TCCE for IoT middleware was created following the procedures highlighted in the research. The middleware was selected since it focused on one particular area of research that can be controlled and tested. The checklist, a spreadsheet, was used to evaluate the security posture of a default build Ubuntu Core OS for IoT installed on a Raspberry Pi 3b+. The IoT middleware is a central processor and authority for IoT operations and communication and the foundation for the system software and hardware interoperability. IoT middleware baseline can be assessed in these specific areas: Access Control (AC), Auditing and Accountability (AU), Identification and Authentication (IA), Configuration Management (CM), System and Information Integrity (SI), System and Communication Protection (SC), and Maintenance (MA) [17]. Table I presents the findings after evaluating these controls and tying them to the middleware.

The steps created were based on available and implementable controls that could be altered on the default configurations allowed by Ubuntu Core OS for IoT. In many cases, changing system settings could render the functionality of the device inoperable; in other cases, leaving the system at the default settings leaves the system with the potential to be hijacked, render the system useless, or even destroyed. The steps were selected because there were specific means to reveal the internal configuration of the system and tie the result to vulnerabilities found in the MITRE ATT&CK database. The DoD STIGs and CIS Benchmark were used as a foundation for best practices and a way to identify and mitigate some of the most common vulnerabilities. The commands used were based on the commonality with other Linux bash shells and system kernels to pull the required information from the system.

| TABLE I: IoT Middleware Technical Control Compliance Evaluation Checklist |
|-----------------------------|---------------------------------|
| **Categories**              | **Controls used**               |
| Control category used       | 8 (AC, AU, IA, CM, SI, SC, MA)  |
| Number of steps created     | 55 (OS specific)                |
| NIST SP 800-53 controls     | 48 including sub-controls       |
| Command used                | 66 different commands           |
TABLE II: EXAMPLE STEP FOR ASSESSMENT OF IoT MIDDLEWARE

<table>
<thead>
<tr>
<th>Step number</th>
<th>Control category</th>
<th>Description</th>
<th>Verifying action (shell command)</th>
<th>Expected results</th>
<th>Actual result</th>
<th>Control met</th>
<th>Control ID/Description</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>Identification and authentication</td>
<td>The system must enforce a 180-day maximum password lifetime restriction</td>
<td>grep -i pass_max_days/etc./login.defs</td>
<td>Verify that the password age is 180 days or less</td>
<td>PASS_MAX_DAYS 99999</td>
<td>No/Failed</td>
<td>IA-5(1)(d) The information system enforces maximum password lifetime restrictions</td>
<td>LOW</td>
</tr>
</tbody>
</table>

In the case of control AC-2(2), the expectation was that the temporary account be removed or disabled after a predefined period of creation, use, and creation [23]. This requirement is typically necessary when a temporary account is created for emergency usage for troubleshooting, maintenance, and fixing. To avoid using the account again later, it should be disabled. In this case, establishing a precise time guideline and what needs to be checked is necessary. Table II is representative of what a single step in the TCCE checklist looks like.

In the case of the example above, the result was pulled directly from the assessment of the Ubuntu Core OS for IoT. The result of the step failed because the system did not manage the expiration of accounts on the system as required by the intended threshold, which was 180 days [23]. This requirement is essential for systems without multifactor authentication. In many cases, requiring an even shorter number of days creates fewer opportunities for the password to be cracked. Advances in processing technology make cracking passwords even easier and faster than ever before [28]. One way to mitigate this risk is to use complex passwords and change them often. The 180 days defined here is just one example of a verification step in the TCCE.

Of the 1190 individual Control IDs specified in the NIST 800-53 r5, 114 controls were deemed IoT-related. Their relatability was formulated from the full review and assessment of NIST documents such as the NIST SP 800-231A and NIST SP 800-82. However, only 48 technical controls from eight categories were specific to IoT middleware. The same steps may also be used to implement other controls on the applications, software dependencies, APIs, libraries, hardware, management, and more. Some checks require visual verification, monitoring, tracking, and documentation to maintain the controls. Maintaining these controls can span across many disciplines and departments. This TCCE matrix tool focused only on the technical aspects that can be controlled by system administrators, IT professionals, cybersecurity engineers, and developers who work directly with these IoT devices. The results of the assessment summarized in Fig. 4 reveal some troubling facts.

Of the 55 steps that can be assessed, meaning that a check can be revealed with the command, 17 failed, and eight were non-applicable due to lack of implementation. These eight steps could translate to failed steps if the organization expects some of these settings and services to be configured. Despite only 17 of 55 steps failing, that is 31% failed, an overwhelmingly high number for a system that is intended to be ready for use. Some more noticeable failures include inactive host firewall, unrestricted commands given to the default user, no password expiration, unspecified ciphers, X11 forwarding, and session restrictions and timeouts. Not every failed step is weighted the same according to NIST impact score and priority level [23]. Most failed steps identified, including our IA-5 step, were considered low impact and a priority of 1 (P1). As it stands, the middleware needs to be assessed and further hardened prior to being put into production.

5.3. Summary

This study presents three distinct stages in developing a security baseline for IoT devices. The steps were broken up to ensure each stage was completed or met before moving on to the next one. Each IoT device is different, but the expectation for the control to be met remains the same. Otherwise, an acceptable alternative control must be in place to ensure the security and protection of the IoT devices and other systems in the organizations. Each device may have a comprehensive security baseline or multiple security baselines for its software and other components. The TCCE was created to evaluate the framework’s effectiveness by assessing a simulated IoT device with middleware, a simulated IoT with an Ubuntu Core OS for IoT. The TCCE is an example of how a baseline can be created for this middleware. The result reveals some default configurations that can be changed to improve the device's security posture.

The TCCE confirms the possibility of creating the baseline using the stated methodology. The validation of the
artifact resulted in creating a usable TCCE checklist with 55 separate steps and 66 unique shell commands to account for 48 different NIST 800-53 controls in eight different control categories. Of the 55 verified steps, 30 were configured securely, while 17 did not meet the requirements. These 17 steps offer possibilities for security improvements, mitigating vulnerabilities and potential threats. The result of the checklist is indicative that the artifact may be effectively used to create other baselines for other IoT devices or the software that resides within the IoT devices as part of its functionality. The artifact may be a foundation to establish a baseline for IoT devices using a similar approach. The results confirm the hypothesis that it is possible to use NIST SP 800-53 controls, risk management principles, adversary tactics, and cybersecurity kill chain to create a security baseline to manage and assess the security posture of IoT devices.

6. Conclusion

This Design Science Research aimed to create a security baseline and cybersecurity framework for IoT devices by using NIST SP 800-53 controls, risk management principles, adversary tactics, and security best practices. Establishing a security baseline through using controls establishes the foundation for safe security practices that rely on security best practices, known exploits, cybersecurity adversary behaviors, and requirements of laws and regulations [6], [7]. This study proposed a framework to select and match the control to the cyber capabilities and apply it to the security baseline.

The artifact was evaluated and validated through the creation of a TCCE checklist for the IoT middleware as a security baseline. The TCCE was created with the assessment of Common Vulnerabilities and Exposure (CVE) database, MITRE ATT&CK framework, NIST Risk Management Framework, DoD STIGs for various Operating Systems, and common security best practices. The vulnerabilities are then tied to the NIST 800-53 controls as a foundation for accountability and oversight. A simulated IoT device with an Ubuntu Core Operating System for IoT devices was used to evaluate the artifact. Commands designed to interact with Ubuntu Core OS operations and configurations were created. The artifact created for this study revealed the possibility of establishing control and manageability over these devices directly or indirectly through an in-depth analysis of the middleware of many IoT devices. Though it is only one component of the IoT device, it offers the most visibility into the underlying architecture, processes, functionality, and communication derived from services, applications, and sensors. However, securing the middleware is just one component that can be established with technical controls. Other controls (management and operational) need to be considered. In an environment where the managerial and operation framework has not been established, it must be incorporated into the security baseline for the IoT device.

This suggests a security baseline can be developed for an IoT device using proper means. The framework highlighted in this study is one of the ways to create a security baseline for an IoT device; the approach can be repeated and modified to meet the needs and requirements of other IoT devices and their components. Doing so would bring visibility and control back into the hands of the organization and owners and allow them to use and secure the device per their requirements and standards.

Conflict of Interest

Authors declare that they do not have any conflict of interest.

References


