Issues in Information Systems
Volume 27, Issue 4, pp. 238-249, 2026

DOI: https://doi.org/10.48009/4_iis_2026_120

Prompt and vulnerable: a security analysis of LLM-integrated IoT firmware

Jim Mahony, *Robert Morris University, mahony@rmu.edu*
Natalya Bromall, *Robert Morris University, bromall@rmu.edu*

Abstract

This study presents an analysis of firmware vulnerabilities of IoT devices that either include Large Language Model (LLM) communication tools or can be connected to a platform with such tools. We used device mapping with SBOM and static firmware analysis techniques with Netrise to analyze the firmware of 75 LLM-integrated IoT devices. The results showed that dangerous vulnerabilities were present in a large percentage of IoT firmware samples analyzed, exposing the LLM-integrated systems to unforeseen risk. In this paper, we review the vulnerabilities found in the firmware and discuss the possible mitigation steps needed to defend the networks from these vulnerabilities. This research contributes to the existing pool of studies in two ways: first, we analyzed a large sample of both common and non-mainstream devices. Second, we look at both the vulnerabilities of IoT device firmware and those of the LLM-integrated platforms to which these devices can be connected.

Keywords: IoT, Internet of Things, firmware security, AI, LLMs

Introduction

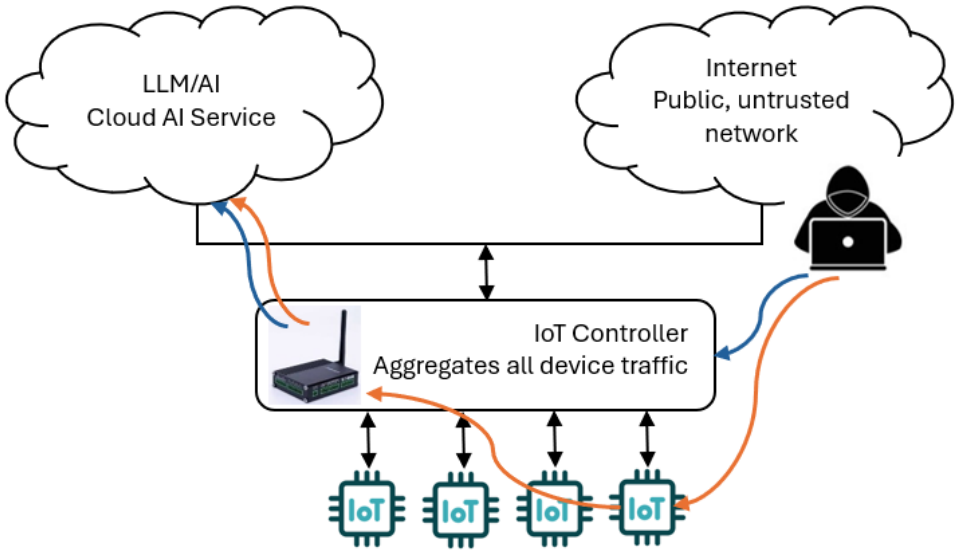
In October 2025, a high-impact DDoS attack paralyzed online gaming activities, causing router card failures and other damage (Paganini, 2025). The attack was caused by Aisuru, a new Turbo-Mirai class of Mirai botnet, consisting of over 300,000 compromised Internet of Things (IoT) devices such as home routers, DVRs, IP cameras, and smart TVs. In May 2021, Colonial Pipeline was attacked with ransomware; the impact was "so significant that President Biden declared a state of emergency" (Insurica, 2026). A hacker group known as DarkSide penetrated the Pipeline network using a compromised VPN password, obtained through vulnerable sensors, pipeline control systems, and other IoT devices on the network. According to the report from Zscaler (2023), there has been a 400% increase in IoT malware and ransomware attacks, and this number has been steadily growing during the past few years. At the same time, IoT devices remain vulnerable to attacks due to a lack of patching, weak factory settings such as default passwords, and many other issues. As an example, Kelly et al. (2020) used Mirai libraries to attack four random IoT devices, finding that three out of these four devices were vulnerable to Mirai malware.

Most people use IoT devices or IoT-integrated devices in some way, often unaware that, similar to computers and tablets, they can be vulnerable to network and other attacks through their firmware. It is more difficult to detect and evaluate firmware vulnerabilities, while their effects are faster and larger (Bhardwaj et al., 2023); additionally, firmware updates are more challenging than updating software on a conventional computer or a tablet (UL Solutions, 2024). The consequences of a hacked IoT device can be just as severe as those of any successful network attack: all network resources can be captured and compromised, opening it to ransomware attacks, malware injections, and many other catastrophic scenarios. Addressing IoT security requirements is often a challenge due to many factors, such as their limited memory, storage, and computational capability, as well as inaccessible firmware (Neshenko et al., 2019).

As a result, the internet is flooded with various vulnerable, unpatched IoT devices with limited or no auditing.

Integrating IoT devices with LLMs opened a new gateway to security vulnerabilities by adding the threats of prompt injection, data poisoning of IoT data streams, shared sensitive information through LLM interaction, and many others (Joshi and Baidya, 2026). In addition, non-LLM IoT devices that are connected to LLM-enabled controllers can introduce the same vulnerabilities to their network as LLM-enabled devices (Figure 1).

Figure 1: Propagation of a cyberattack through the network (attack path is shown in curvy arrows)



While there is a significant amount of research dedicated to IoT firmware security, this research is focused primarily on two directions: (1) general IoT and LLM-IoT security taxonomy and detection strategies, and (2) device-specific case studies related to IoT and LLM-IoT devices. Case studies mainly include mainstream smart home devices: IP cameras, smart speaker systems, doorbell cameras, and others. A very limited number of research studies focus not on mainstream but rather on specialized IoT devices (e.g., controllers for agricultural watering systems). Some of these devices are LLM-integrated on their own, and others include the possibility of integrating with LLM-supported controllers. Protecting all IoT devices is equally important; a security breach in any such device can lead to serious consequences, i.e., ransomware attacks, network penetration, a device being captured by a botnet similar to Mirai, and a data breach. As we can see in the Colonial Pipeline case discussed by Paganini (2025), the consequences of security breaches through IoT devices can be catastrophic. The goal of this research is to analyze the firmware security vulnerabilities of LLM-integrated IoT devices.

Research Question: What are some firmware security vulnerabilities of LLM-IoT devices and LLM-integrated controllers?

Literature Review

Firmware security in IoT devices is a well-known problem with an extremely broad scope, including any IoT devices such as cameras, doorbell cameras, smart speakers, and many others. While used in most organizations and a substantial number of households, IoT firmware is found to be one of the most neglected areas in device security (Bhardwaj et al., 2023), a problem that stems from the IoT diversity and the closed-

source nature of its firmware, making it challenging to audit. Researchers found multiple security gaps associated with such devices and addressed the security strategies needed to protect the networks that contain them. As such, Chen et. Al (2024) found that among all reviewed firmware, 67.3% of firmware included outdated libraries that their manufacturers failed to update. Kaushik et al. (2025) analyzed the firmware of common IoT smart home devices and identified ten critical network-based firmware vulnerabilities scoring 9.8 or 10 on the severity scale, along with multiple vulnerabilities scoring lower. In the following sections of this review, we begin by discussing the existing classifications of IoT vulnerabilities, followed by the three strategies for identifying those vulnerabilities. We then give examples of how these methods are used in device-specific studies and conclude by reviewing the studies that focus on the specifics of LLM-integrated IoT devices.

Classification and mapping of IoT vulnerabilities

Due to the abundance of IoT devices and security vulnerabilities, classifying devices and mapping them to vulnerabilities presents a challenge, which is why it is necessary in designing a comprehensive security and audit strategy. Researchers used multiple approaches to create such a taxonomy and mapping. Bakhshi, Ghita, and Kuzminykh (2024) proposed a taxonomy of vulnerabilities and mapped them into auditing tools according to the eight categories of the IoT ecosystems, from system properties and access control to user awareness, regulatory compliance, and adversarial factors. Neshenko et al. (2019) designed a similar classification of IoT vulnerabilities across the hardware, firmware, software, and network layers. With multiple classifications of vulnerabilities that stem from the IoT devices, the next task for the researchers is to recommend methods of detecting vulnerabilities.

Detecting firmware vulnerabilities

Researchers outline three methodological directions for detecting IoT firmware vulnerabilities: static analysis, dynamic analysis, and fuzzing, which is a special subset of dynamic analysis. In static analysis, firmware is analyzed without running it. In dynamic analysis, the firmware is run either on the device itself or in an emulated environment (Zhou et al., 2023). Both these types of analysis have merits and are utilized in research.

Redini et. al. (2020) used KARONTE, a static analysis approach that "tracks data flows across the binaries of a firmware sample to precisely uncover security vulnerabilities" (Redini et. al, 2020, p. 1545), on 53 firmware samples and demonstrated that it could successfully track multi-binary interactions. Four years later, Chen et al. (2024) introduced SaTC, which outperformed KARONTE in discovering bugs in embedded systems. At present, multiple tools and platforms are used in static analysis, often in integration with other steps. Tolay (2026) designed an effective lightweight pipeline for extracting file systems from IoT devices and integrated it with Software Bill of Materials (SBOM) for mapping and scoring vulnerabilities.

Due to a large variety of IoT devices, vendor-protected firmware, and other reasons, running a static analysis often presents a challenge. In 2016, Chen et al. presented FIRMADYNE, the first automated dynamic analysis system capable of dynamically extracting firmware from IoT devices, analyzing it, and adjusting to the changes in firmware images. A few years later, another notable system, FirmAE, was developed by Kim et al. (2020), who demonstrated that the new system was an improvement over FIRMADYNE, capable of running close to 80% of the given firmware images, while FIRMADYNE processed only 16.28% (Kim et al., 2020). Researchers often effectively combine static and dynamic methods. For example, in their research, Costin, Zarras, and Francillon (2016) used static pre-processing steps such as extracting and unpacking firmware images, followed by dynamic analysis of firmware.

Fuzzing is a special type of dynamic testing and involves "an automated software testing technique that provides invalid, unexpected, or random data as inputs to a computer program to test whether the program performs as expected" (Zhou, Shen, and Liu, 2025, p. 6). This method is widely used in research; one of

the first research groups to mention it was Chen et al. (2018), who demonstrated the possibility of probing the IoT devices without any knowledge of their firmware or protocol specifications. Fuzzing methods are extremely helpful in analyzing vulnerabilities of IoT devices where their firmware is unavailable. At the same time, researchers find certain flaws in firmware fuzzing, e.g., it can successfully identify memory corruption resulting from an attack on a regular computer, but not an IoT device (Muench et al., 2018).

Device-specific studies

A large number of firmware vulnerability case studies are dedicated to specific classes of mainstream IoT devices, such as IP cameras, smart speakers, and smart doorbells. The results of these studies are unsettling: Bhardwaj et al. (2023), in their study of IoT cameras, found that most such devices do not have a built-in firmware security updating mechanism and are focused on "business but not security". In addition to insecure firmware updates, IoT cameras and other smart home devices demonstrate a long list of critical vulnerabilities: hard-coded passwords, weak authentication, invalidated inputs, and many others (Kaushik et al., 2025). Li, Mandalari, and Straw (2023) analyzed a sample of popular IoT smart home devices, which includes smart cameras, finding a set of critical vulnerabilities to flooding, port scanning, and OS scanning. Another study of the popular IoT devices demonstrated that, in addition to firmware vulnerabilities, these devices have improperly encrypted device-to-app communication, which can result in compromised sensitive data and other consequences.

LLM-integrated IoT devices are a newer phenomenon, and researchers unanimously admit that more security vulnerabilities are added to the picture due to their integration. Sarhaddi et al. (2025) provide a comprehensive survey of LLM-integrated IoT devices, with over 300 devices included, and address security challenges specific to these devices. Four threat domains are identified specifically related to LLM-IoT systems: data and input attacks, communication and infrastructure attacks, model-level attacks, and privacy leakage (Joshi and Baidya, 2026).

In conclusion, device-specific research studies identified multiple firmware and communication vulnerabilities in IoT smart home and office devices, commonly used in our everyday life; however, there is little to no focus on specific IoT devices with narrow application domains. While a large amount of research is dedicated to the firmware security of IoT and LLM-IoT devices per se, fewer studies focus on vulnerabilities of the LLM-integrated platforms to which these devices connect.

Methodology

Operational technology environments typically include numerous technology devices and interfaces, which in many cases can be managed centrally, and can contain numerous IoT devices (NIST, 2023). As technology has advanced, these IoT devices have become more diverse, with many now including wireless interfaces and capabilities to communicate over different channels, some of which now integrate with controllers that share the collected data with LLM or AI systems.

We chose static firmware analysis as our research method. To run this study, we first needed to identify IoT control systems that integrated with and collected data from IoT devices and shared this data with LLM/AI technologies for decision-making or other data correlation reasons. The firmware of the devices considered for the study needed to be readily available for download without needing to purchase the product. Many software manufacturers offer trial or beta versions for potential customers to try the systems before they buy. Others are available for download because they are open-source projects, with many developers contributing code or updates to enhance capabilities or to patch vulnerabilities. Available controller packages were analyzed to ensure that those which were included in the study were only those that have been used in industry and could be found in other academic or industry publications to validate

that the systems are, in fact, being used. Then, the systems were researched to ensure that compatible IoT devices could be identified, with these devices also having firmware freely available for download.

Device Sampling Strategy

A combination of convenience and stratified sampling was used to select a set of firmware used in this study. First, we sampled the firmware of various IoT devices available through open platforms such as GitHub. Second, we rejected the firmware of devices with no LLM or possibility of LLM integration. Finally, we identified the device types (e.g., thermostat, gateway, sensor) as strata and selected similar numbers of IoT devices representing multiple strata. We separated the firmware of IoT devices and controllers into two groups. Three different control systems were found to be applicable to this research and were subsequently downloaded for security analysis. Additionally, 75 compatible IoT device firmware images were downloaded for analysis. This study sought to investigate the security posture of IoT devices that integrated into platforms that, in their turn, integrated with LLM systems. The devices that were in scope of the exercise included: thermostats, gas detection sensors, weather station systems, temperature sensors, humidity sensors, asset trackers, IoT displays, IoT gateways, network video recorders, and numerous types of IoT cameras.

Once systems were identified and downloaded, analysis began. This analysis included the use of a firmware analysis platform called Netrise. Netrise is a newer cybersecurity company that specializes in the analysis of firmware. The Netrise platform takes compiled software, deconstructs the binary into the SBOM, analyzing each individual component, leveraging known vulnerability databases. The SBOM is a list of all software components that make up the firmware image (SBOM, 2018). Identified vulnerabilities were then further analyzed to identify those that have known exploit code and those that have been leveraged in cyberattacks. Netrise showed all aspects of the security profile of the firmware, which included a list of vulnerabilities and their characteristics: risk rating; whether or not it had available exploit code or was weaponized; and if it was listed in the CISA Known Exploited Vulnerabilities Catalog (KEV). As an example, one of the firmware samples analyzed contained 6269 components, which were found to have 4,703 vulnerabilities, with sixteen being weaponized, and 5 KEVs.

Shodan

After finalizing the sample of IoT devices, we verified the actual presence of the selected devices and controllers in the networks. In other words, we did not want to run the analysis for non-existent, obsolete, or otherwise non-discoverable devices. As an additional step, we used Shodan to identify the controllers that were generally discoverable on the internet, making it a very easy job for an attacker to find them and exploit their vulnerabilities. Shodan is an IoT search engine that discovers and catalogs connected IoT devices discoverable on the internet. It is described as “a search engine for Internet-connected devices” (Shodan).

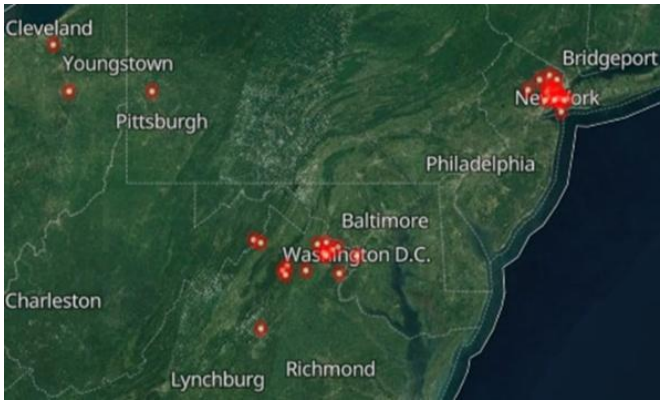


Figure 2: Shodan image showing the location of discovered instances of the IoT controller described

Figure 2 shows how a specific geographic area is represented in Shodan; we used Washington DC as an example, but this can also be an area outside of the US. Using Shodan, we were able to identify 1,516 instances of one of these IoT/OT controller interfaces globally (Figure 2), with 152 within the United States; this is often the first step of criminal reconnaissance that can lead to a cyber-attack. These devices are found by Shodan because they are configured with system defaults in place, such as standard TCP-IP ports or login webpages that are the default from the device manufacturer. The search parameters included words that very clearly identify the specific technology by name, and also included ports used for login. Therefore, this detail was excluded from the paper to protect the organization from further exposure.

Keeping these devices configured with system defaults is certainly the easiest way to bring them online, but it also introduces a level of risk that the organization may not anticipate or understand. Once these devices are discovered by a potential attacker, they then begin to enumerate the organization's technology footprint. Once the system is discovered, attempts to access the system can begin with a technique called Shodan Dorking, where default credentials are attempted for authentication (Harellevy, 2024). Should the system be inaccessible, then connected devices, if accessible, can be exploited for initial access. The devices that are connected are also often configured with internet access for communication with the controller system, and also to the manufacturer for monitoring or even for software updates.

Results

This research included an analysis of several OT/IoT controller systems and numerous different types of IoT technologies that can be connected to them. Throughout this research, an analysis of IoT device firmware and controller software was conducted using the Netrise platform. Figure 3 shows the summary graph of found vulnerabilities by their age (time that they were known) and their severity (Low through Critical). Using Netrise, we analyzed 75 pieces of firmware, which revealed 9,368 vulnerabilities, and determined which have already been exploited and which have exploits available for criminal use. The firmware images analyzed during this exercise revealed thousands of vulnerabilities, many of which are exploitable, and some that have been used in attacks. For security reasons, we chose not to utilize the names of any particular vendors or devices in this research.

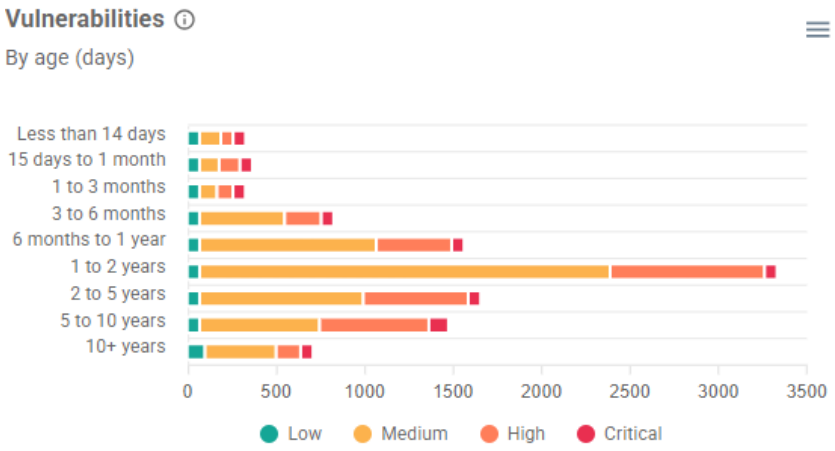
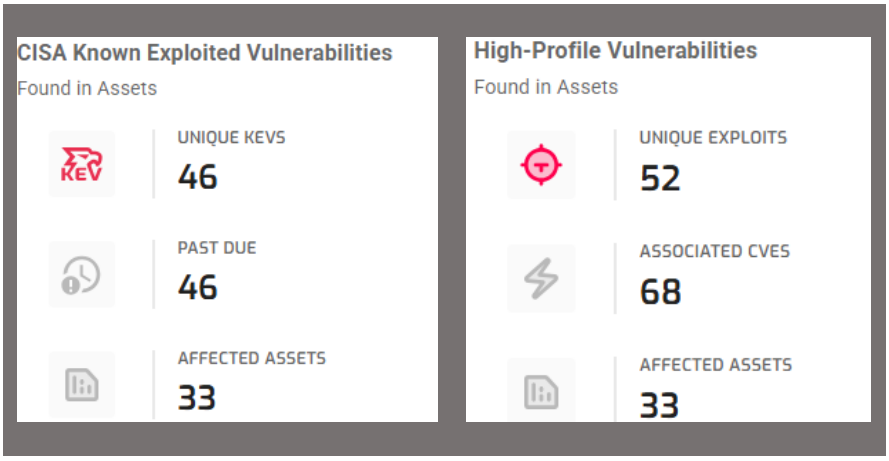


Figure 3: Netrise platform showing the array of vulnerability risk, instances, and age

Of the 75 assets analyzed, 33 contained CISA KEVs, with the total number of unique KEVs being 46 (Figure 4). These 75 assets also contained 9,368 Unique vulnerabilities, of which 461 have exploit code

available. Many of these vulnerabilities are categorized as weaponized, meaning that exploit code has been developed, and in some cases, though not all, has been used in cyber-attacks. The vulnerabilities discussed in the next section of this research paper represent only a small sample of all of the findings within the 75 samples examined. The details outlined in this section can also be found on the NIST National Vulnerability Database.

Figure 4: Netrise analysis of CISA KEVs and High-Profile Vulnerabilities



Out of all the unique vulnerabilities that were found, we excluded the ones that presented medium or lower risks and considered only the ones classified as high-risk or critical. Table 1 presents a summary of the remaining critical and high-risk vulnerabilities found as a result of our analysis. Any of these vulnerabilities, tied to an IoT device that integrates with an LLM, could lead to data poisoning, prompt injection, and data exfiltration. These attacks could be catastrophic, depending on the environment where the devices are installed. Alarms could be suppressed, data going into or coming out of the LLM could be stolen or manipulated to lead the decisioning based on the data to be incorrect. The outcomes of attacks on LLM-integrated IoT vulnerabilities are numerous and could be avoided if risk assessments are conducted and remediations or mitigations are put in place prior to deployment to production.

Table 1. Summary of critical and high-risk vulnerabilities by device

Vulnerability	Risk Impact	Impacted IoT Devices	Impacted LLM Controllers
CVE-2026-4800	Critical CVSS 9.8	0	2
CVE-2020-15999	Critical CVSS 9.6	21	0
CVE-2024-36971	High CVSS 7.8	27	0
CVE-2015-8104	Critical CVSS 10	5	0
CVE-2013-2094	High CVSS 8.4	12	0

Itemized analysis

The following section presents the itemized analysis of the Common Vulnerabilities and Exposures (CVEs). The CVEs identified and analyzed were an integral part of the research; their analysis demonstrates the breadth of the IoT vulnerability problem that existed before the introduction of LLM or AI technologies. Having these vulnerabilities still present in the firmware now introduces new risks to their usage.

Although there were many critical and high-risk findings discovered in this study, some notable vulnerabilities are described herein. After sorting these vulnerabilities by risk, we found many high-priority vulnerabilities, including CVE-2020-15999. This is a heap buffer overflow vulnerability with a critical risk CVSS score of 9.6, and it was found 21 times. This vulnerability has public exploit code available, involves low-complexity exploitation, and does not require physical access to the device, so an attack over a network is possible. This vulnerability has been used in cyber-attacks, making it a highest priority item and a critical finding (Narang, 2020). It is found in multiple IoT camera firmware images; if exploited, it could lead to potential interception or manipulation of images or video, possible credential theft, denial of service of the controller, and numerous other types of malicious actions.

Two analyzed devices were found to be vulnerable to CVE-2026-4800. This vulnerability is given a CVSS score of 9.8, a critical risk vulnerability. The affected devices were not edge IoT devices but controller units, to which the IoT devices would be sending their data. The data would then be sent to the LLM for processing and potentially decisioning. This is a code injection vulnerability in the Lodash JavaScript utility library, which, if successfully exploited, could lead to arbitrary code execution and potentially full system compromise. The best mitigation for this vulnerability is patching, which the manufacturers of the devices would ultimately be accountable for producing.

Another vulnerability of interest was CVE-2024-36971, which is a 7.8 CVSS score high-risk vulnerability, indicating that it is exposed to remote code execution (RCE). Remote code execution often leads to advanced compromised security incidents, and is often chained together with other vulnerabilities to further the efforts of an attacker. This vulnerability was found within 27 examined systems, including one controller system, an NVR video recording system, an IoT network gateway system, and numerous different cameras. This vulnerability can be found in the CISA Known Exploited Vulnerabilities (KEV) catalog. Systems prone to it included many different types of cameras and network video recording appliances. This vulnerability has five different exploit code samples available and was also found to have been associated with five separate cyber-attacks. If exploited, it could lead to image or video manipulation, lead to arbitrary code execution and root-level access to the system, and could lead to a full network compromise.

Also identified were 5 instances of CVE-2015-8104, which is a Linux kernel weakness within NVR systems analyzed. This vulnerability has a CVSS score of 10, making it a critical risk. If exploited, it could lead to the interception or manipulation of images or video, or denial of service to the devices. Attacks could be conducted to interfere with security cameras, which could be a risk that leads to other physical attacks, as a precursor to a more complex technical attack. Mitigation for this is patching and possibly isolating the system to prevent access to the devices.

Another dangerous vulnerability that was found was CVE-2013-2094, which is a Linux kernel privilege escalation vulnerability. This vulnerability was found in 12 samples. What is alarming about this and many other findings, this vulnerability was discovered and communicated over ten years ago in 2013. This item is also a CISA KEV, has 22 exploits available on the internet, and has been used in 2 documented attacks. The devices affected by this vulnerability include cameras and network infrastructure. Exploitation of this vulnerability could lead to arbitrary code execution and root-level access through privilege escalation and full network compromise.

Discussion and Limitations

As mentioned earlier, this study required a set of very specific IoT technologies. Their selection criteria dictated that the controllers be integrated with an LLM; in addition, we were limited to the systems that were freely available for download, thus restricting the scope of available technologies. The IoT firmware samples selected for the study were those compatible with the controller platforms. This left us with a small sample of IoT / controller systems that are in use currently, but at the same time are freely available for download. The Netrise platform's capability to analyze firmware is relatively new and not widely adopted by information security teams. Having access to this platform allowed for a unique study that exposed unknown risks within the IoT LLM landscape.

The vulnerabilities addressed in this section represent risk exposure and potential exploitation. These findings do not indicate that the devices in question possess these vulnerabilities in a condition that is easily exploitable, or even exposed to exploitation, simply that the firmware code includes vulnerable components. To fully understand whether or not these devices can be exploited through the found weaknesses, a dynamic firmware analysis needs to be launched in addition to the static analysis we conducted in this research. A lab environment would need to be developed, and hardware samples would need to be acquired and attacked in that lab setting. This academic analysis, therefore, represents the theoretical exposure of these systems. To mitigate or remediate these vulnerabilities, the owners of the systems would need to apply security patches, if they are available, or reach out to the vendor if they are not, and submit the findings to them for remediation. In hardware and software manufacturing, suppliers are typically motivated to fix systems for which vulnerabilities have been identified. Additional mitigations could be available through network segmentation and applied identity and access management; these mitigations can further isolate a vulnerable system by preventing access and limiting exposure of the vulnerabilities.

Conclusion

Vulnerabilities in devices whose data can interface with LLM or AI systems need to be carefully assessed, and any identified risks need to be mitigated or remediated. This study has shown that in many cases, IoT devices and systems can go into production environments vulnerable by default, without mitigations in place to protect the vulnerable systems, and attack vectors can be discovered and exploited by an adversary. The introduction of IoT into networks has created a new attack vector for adversaries. These IoT systems contain vulnerabilities, which makes it challenging for organizations to keep them secure. Integrating these networked devices with LLM/AI systems brings a new category of risks to companies leveraging the new technologies in conjunction with one another. Relying on an AI or LLM for decision-making, based on data that is collected by vulnerable IoT systems, is a mistake that companies are making across the globe. These systems must be assessed for security before deployment and regularly reassessed in order to maintain a known and acceptable level of risk. The assessments should include an understanding of the software and firmware that the systems use, leveraging a platform similar to Netrise, and also network vulnerability scanning on all ports.

Organizations choose to leverage IoT and AI systems for increased efficiencies and productivity. These technologies do allow for greater usage of resources; however, the risks associated with using them often fall beyond the bounds of the expertise of the teams that are charged with procuring, deploying, and maintaining their operation. For those who do choose to leverage these, it is imperative that they educate themselves on the risks that are involved and engage outside expertise during the process of identifying the providers and deployment of the infrastructure and software. Regularly risk-assessing these systems is often overlooked as an unnecessary expense; however, the alternative can be far more costly should an incident

occur. Vulnerabilities can be leveraged by a motivated attacker to infiltrate a network, allowing the operations to continue so that the attacker's presence is unknown, enabling the attacker to either manipulate the data that is being collected by the IoT devices or even to affect the processing of the systems and their data. These attacks can lead to full network compromise, ransomware, and other catastrophic consequences.

References

Bakhshi, T., Ghita, B., and Kuzminykh, I. (2024). A review of IoT firmware vulnerabilities and auditing techniques. Sensors, 24(2). DOI: https://doi.org/10.3390/s24020708

Bhardwaj, A., Kaushik, K., Bharany, S., and Kim, S. (2023). Forensic analysis and security assessment of IoT camera firmware for smart homes. Egyptian Informatics Journal, 24. DOI: https://doi.org/10.1016/j.eij.2023.100409

Chen, D. D., Egele, M., Woo, M., and Brumley, D. (2016). Towards automated dynamic analysis for Linux-based embedded firmware. Network and Distributed Systems Security (NDSS) Symposium, San Diego, CA. DOI: http://dx.doi.org/10.14722/ndss.2016.23415

Chen, J., Diao, W., Zhao, Q., Zuo, C., Lin, Z., Wang, Z., Lau, W.C., Sun, M., Yang, R., and Zhang, K. (2018). IOTFUZZER: discovering memory corruptions in IoT through app-based fuzzing. Network and Distributed Systems Security Symposium, San Diego, CA. DOI: http://dx.doi.org/10.14722/ndss.2018.23159

Chen, L., Wang, Y., Linghu, J., Hou, Q., Cai, Q., Guo, S., and Xue, Z. (2024). SaTC: shared-keyword aware taint checking for detecting bugs in embedded systems. IEEE Transactions on Dependable and Secure Computing, 21(4), pp. 2421-2433.

Chen, Y., Ma, F., Zhang, Y., He, Y., Wang, H., and Li, Q. (2024). AutoFirm: automatically identifying reused libraries inside IoT firmware at large-scale. Submission to ACM, 2024

Costin, A., Zarras, A., and Francillon, A. (2016). Automated dynamic firmware analysis at scale: a case study on embedded web interfaces. Asia Conference on Computer and Communication Security Proceedings, pp. 437-448. DOI: https://doi.org/10.1145/2897845.2897900

Hahn, Adam. (2016). Operational Technology and Information Technology in Industrial Control Systems. 10.1007/978-3-319-32125-7_4.

Harellevy. (2024, December 21). *Utilizing Shodan for large scopes*. Medium. https://medium.com/@harellevy159/utilizing-shodan-for-large-scopes-cb07ab5b49bb

Insurica (2026). Cyber case study: Colonial Pipeline ransomware attack. Retrieved on 4/14/26 from https://insurica.com/blog/colonial-pipeline-ransomware-attack/

Joshi, A. and Baidya, S. (2026). Securing the cognitive layer: a survey on security threats, defenses, and privacy-preserving architectures for LLM-IoT integration. Journal of Cybersecurity and Privacy, 6(2). DOI: https://doi.org/10.3390/jcp6020063

Kaushik, K., Bhardwaj, A., and Dahiya, S. (2025). Framework to Analyze and Exploit the Smart Home IoT Firmware. Measurement: Sensors, 37. DOI: 10.1016/j.array.2024.100382

Kelly, C., Pitropakis, N., McKeown, S., & Lambrinoudakis, C. (2020). Testing and Hardening IoT Devices Against the Mirai Botnet. IEEE International Conference on Cyber Security and Protection of Digital Services. DOI: 10.1109/CyberSecurity49315.2020.9138887

Kim, M., Kim, D., Kim, E., Kim, S., Jang, Y., and Kim, Y. (2020). FirmAE: towards large-scale emulation of IoT firmware for dynamic analysis. Annual Computer Security Applications Conference Proceedings, Austin, TX. DOI: https://doi.org/10.1145/3427228.3427294

Li, Y., Mandalari, A. M., & Straw, I. (2023). Who let the smart toaster hack the house? An investigation into the security vulnerabilities of consumer IoT devices. ArXiv, DOI: https://doi.org/10.48550/arXiv.2306.09017

Mauro, D., Melo, L., Lu. H., d'Amorim, M., Prakash, A. (2019). Beware of the app! On the vulnerability surface of smart devices through their companion apps. ArXiv, DOI: https://doi.org/10.48550/arXiv.1901.10062

Muench, M., Stijohann, J., Kargl, F., Francillon, A., and Balzarotti, D. (2018). What you corrupt is not what you crash: challenges in fuzzing embedded devices. Network and Distributed Systems Security Symposium, San Diego, CA. DOI: http://dx.doi.org/10.14722/ndss.2018.23166

Narang, S. (2020, November 2). *CVE-2020-15999, CVE-2020-17087: Google Chrome FreeType and Microsoft Windows Kernel Zero Days Exploited in the Wild*. Tenable®. https://es-la.tenable.com/blog/cve-2020-15999-cve-2020-17087-google-chrome-microsoft-windows-kernel-zero-day-vulnerabilities-exploited-in-wild-along-with-cve-2020-16009

National Institute of Standards and Technology (2023). *Guide to operational technology (OT) security* (NIST Special Publication 800-82 Rev. 3). U.S. Department of Commerce.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., and Ghani, N. (2019). Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. IEEE Communications Surveys & Tutorials, 21(3), pp. 2702–2733.

Paganini, P. (2025). Aisuru botnet is behind record 20 TB/sec DDoS attacks. Security Affairs. Retrieved on 4/15/26 from https://securityaffairs.com/183969/malware/aisuru-botnet-is-behind-record-20tb-sec-ddos-attacks.html

Redini, N., Machiry, A., Wang, R., Spensky, C., Continella, A., Shoshitaishvili, Y., Kruegel, C., and Vigna, G. (2020). KARONTE: Detecting Insecure Multi-binary Interactions in Embedded Firmware. IEEE Symposium on Security and Privacy, pp. 1544-1561.

Sarhaddi, F.; Nguyen, N.T.; Zuniga, A.; Hui, P.; Tarkoma, S.; Flores, H.; Nurmi, P. (2025). LLMs and IoT: a comprehensive survey on Large Language Models and the Internet of Things. IEEE Open Journal of Communication Society, 7, pp. 3585-3616. DOI: https://doi.org/10.1109/OJCOMS.2026.3680064

Shodan. (n.d.). *What is Shodan?* Retrieved on 5/12/26, from https://help.shodan.io/the-basics/what-is-shodan

Tolay, A., (2026). Automated SBOM-driven vulnerability triage for IoT firmware: a lightweight pipeline for risk prioritization. ArXiv, DOI: https://doi.org/10.48550/arXiv.2601.01308

UL Solutions (April 11, 2024). Unprotected firmware puts IoT devices at greater risk. UL Solutions industry report. Retrieved on 4/12/26 from https://www.ul.com/news/unprotected-firmware-puts-iot-devices-greater-risk

Zhou, W., Shen, S., Liu, P. (2025). IoT firmware emulation and its security application in fuzzing: a critical revisit. Future of Internet, 17(19). DOI: https://doi.org/10.3390/fi17010019

Zhou, X., Wang, P., Zhou, L., Xun, P., and Lu, K. (2023). A survey of the security analysis of embedded devices. Sensors, 23. DOI: https://doi.org/10.3390/s23229221

Zscaler (2023). Zscaler Threat Labz 2023 Enterprize IoT and OT threat report. Retrieved on 4/15/26 from https://info.zscaler.com/resources-industry-reports-threatlabz-2023-enterprise-ioT-ot-threat-report