IoT Threat Modeling for Smart Home Devices

Emanuel Botros
em354052@ucf.edu
University of Central Florida
Orlando, Florida, USA

Abstract

This report presents an in depth study of threat modeling for smart home Internet of Things devices. The project focuses on identifying and mitigating cybersecurity threats targeting interconnected home devices such as smart locks, cameras, and thermostats. Using the STRIDE framework, the work constructs a structured threat model, evaluates vulnerabilities across device categories, and recommends layered mitigations that are realistic for non expert home users. The results show that all STRIDE categories appear in a representative smart home deployment, with high risk threats clustering around authentication, firmware integrity, and local network exposure. A mitigation matrix maps each threat to specific technical and configuration controls and identifies whether the user, vendor, or ISP is responsible for implementation.

Code and artifact. Source files, STRIDE matrices, and figures are available at: https://github.com/milobzb/iot-smart-home-threat-model.

1 Introduction and Problem Statement

The rapid adoption of Internet of Things devices in smart homes introduces security and privacy challenges that traditional network hardening guidelines do not fully address. Devices are often shipped with weak default credentials, unpatched firmware, or insecure communication protocols. As these systems become more integrated through hubs, companion mobile applications, and cloud services, the attack surface expands. Adversaries can exploit vulnerabilities to access the home network, control devices, or exfiltrate sensitive data.

Smart home users usually install devices for convenience or safety rather than for security research. As a result, many homes include a mix of devices from different vendors, purchased over time, that share the same local network and depend on cloud services whose internal security properties are opaque. Misconfigurations during installation, such as keeping default passwords or enabling remote access without restrictions, can quietly expose the home to external attackers. At the same time, the limited computing resources on many IoT devices make it difficult to apply heavyweight security mechanisms that are common on laptops or servers.

There is also a gap between what vendors assume and what people actually do at home. Some vendor security guides expect users to read long documents and manually manage network segments, while most users rely on default router settings and quick start wizards. In this environment, a systematic threat model can act as a bridge between academic security principles and the messy realities of consumer deployments. The model provides a shared vocabulary for device makers, network operators, and home users who need to coordinate defenses but often talk past each other.

This project studies the following problem. Given a realistic smart home deployment that combines a router, a hub or phone application, and multiple devices such as cameras and smart locks, how can one systematically enumerate and prioritize security threats in a way that is repeatable and actionable for both vendors and non technical users The goal is not just to list every possible vulnerability, but to organize threats so that they can guide concrete decisions about firmware design, default configurations, and home network setup.

The objectives of the project are to

- construct a comprehensive threat model for a representative smart home deployment using STRIDE
- define simple but meaningful evaluation metrics that capture coverage, prioritization quality, and mitigation completeness
- derive a mitigation catalog that balances security with usability and cost for home users
- highlight which parties user, vendor, or ISP are responsible for specific controls and where responsibility is shared
- provide structured documentation that can be reused as a template for other smart home environments.

The final deliverables include an asset and trust boundary diagram, a STRIDE based threat matrix, evaluation tables, and configuration guidelines that can be applied to similar deployments. Although the model is built around a specific example home, the intention is that other environments can adapt the workflow and adjust the details to their own device mix.

2 Related Work

Recent cybersecurity analyses highlight the expanding risk landscape for connected home systems. A 2024 report by SISA InfoSec [1] reviews the OWASP IoT Top 10 and shows that vulnerabilities such as weak authentication, insecure network services, and insufficient privacy protection remain prevalent in consumer devices. The document also emphasizes that insecure default configurations and lack of secure update mechanisms are frequent root causes. These findings suggest that many problems originate from design and deployment decisions rather than from obscure implementation bugs.

SecureDebug researchers [2] describe systematic threat modeling for IoT environments and argue that lightweight modeling early in the design lifecycle can identify design level weaknesses before devices are deployed at scale. They discuss how structured methods such as STRIDE and attack trees can make the process more repeatable and less dependent on individual experience. Their work motivates the choice of STRIDE as a framework that is simple enough for practitioners yet expressive enough to cover common IoT risks, especially when combined with checklists like the OWASP IoT Top 10.

TechTarget analysts [3] discuss five major IoT security threats ranging from inadequate patching to insecure communication and demonstrate how these issues translate to real incidents. They recommend that organizations prioritize mitigations using frameworks such as STRIDE and align them with operational constraints, for example by focusing first on threats that can be reduced through configuration changes rather than redesigning hardware.

Beyond these industry reports, academic work has explored broader notions of smart home security and privacy. Studies have examined network traffic patterns from commercial devices, inferred sensitive user activities from sensor logs, and proposed frameworks for user friendly access control policies. Many of these efforts assume that devices can be monitored or instrumented in detail. In contrast, this project takes a more modest approach that can be applied even when internal device behavior is opaque, by focusing on assets, data flows, and high level design assumptions.

Other work has proposed formal risk assessment methods and privacy impact assessments for smart home deployments. These methods often require extensive questionnaires or legal analysis that may be difficult for engineers to apply in day to day design. The approach in this report can complement such work by providing a concrete map of threats and controls that can then be fed into broader organizational risk processes or translated into policy language.

3 Methodology (Technique and Approach)

The methodology centers on developing a threat model grounded in STRIDE for a baseline smart home environment. The overall workflow is summarized in Figure 1 and is designed to be understandable by security practitioners and advanced home users.

3.1 System model and assets

The baseline environment includes a consumer router or gateway, one hub or companion mobile application, and two representative device classes a camera and a smart lock. The model also considers cloud backends used for remote access and storage, as well as human actors such as the primary user and guests. This combination reflects a minimal but realistic smart home where a user can view camera feeds remotely and unlock the door from a phone.

Each asset is associated with security and privacy goals. Cameras require confidentiality of video streams, integrity of firmware images, and reliable availability for safety monitoring. Locks prioritize availability and integrity of unlock commands, as well as strong authentication of the party issuing those commands. The router and hub are treated as central control points that can enforce network and identity based defenses, for example by isolating devices into separate segments.

To keep the model tractable, the analysis focuses on the core data flows needed for common use cases such as remote viewing and remote unlocking. Optional features such as integration with voice assistants are noted but not modeled in detail. This choice keeps the pipeline simple enough for manual analysis while still surfacing meaningful threats.

3.2 Attacker model

The threat model assumes a set of adversaries with different capabilities.

- An external remote attacker who has internet connectivity and can scan for exposed services, guess passwords, or exploit cloud account weaknesses.
- A local network attacker who has access to the home WiFi or to the guest network, either as a visitor or by breaking wireless security.
- A nearby radio attacker who can interact with Bluetooth or Zigbee links but does not have general LAN access.
- An insider who can physically access devices, press reset buttons, or steal credentials written on sticky notes near the router.

These adversaries motivate different defensive strategies. Strong cloud account authentication primarily protects against remote attackers, while network segmentation and WiFi isolation help contain local adversaries. Physical protections such as tamper evident seals matter more against insiders and service technicians who can touch equipment during installation or maintenance visits.

3.3 Threat modeling workflow

The threat modeling workflow consists of five main steps.

1. Enumerate assets and data flows, including camera video streams, lock control messages, device onboarding traffic, firmware update channels, and cloud communication. This enumeration is recorded in a structured spreadsheet that becomes the basis for the STRIDE matrix.
2. Map trust boundaries, such as the separation between the home LAN and guest WiFi, local wireless protocols such as Bluetooth or Zigbee, and external cloud APIs. Each boundary indicates where an attacker might cross from a low trust zone to a high trust zone. During this step, the analyst sketches a data flow diagram that visually highlights potential chokepoints.
3. Identify threats for each flow using STRIDE categories spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. For example, an unencrypted local API used by a camera may expose information disclosure and spoofing risks, while an unreliable update channel suggests tampering threats.
4. Rate the risk of each threat using a qualitative likelihood impact matrix with High, Medium, and Low levels. Likelihood takes into account attacker effort and required access, while impact considers consequences for confidentiality, integrity, and availability. A short rubric in the artifact repository contains examples that help keep ratings consistent.
5. Assign layered mitigations and record which party user, vendor, or ISP is responsible for implementing them. When multiple parties share responsibility, the matrix notes these dependencies so that gaps in accountability are visible.

3.4 Threat matrix construction

The central artifact of the methodology is a tabular threat matrix. Each row in the matrix represents a specific threat scenario, such as an attacker replaying an unlock command or using a default camera password to view video. Columns record the affected asset, data flow, STRIDE category, assumed attacker capability, risk rating, and proposed mitigations.

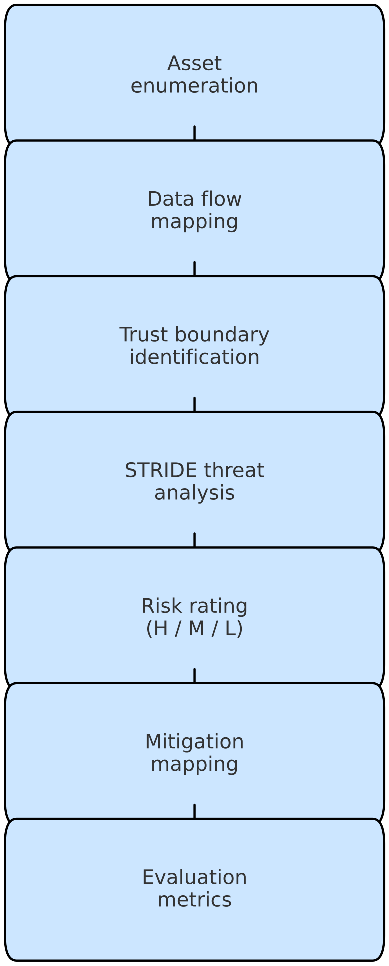

Figure 1: Threat modeling pipeline for smart home devices. Inputs include asset and data flow enumeration. The central STRIDE analysis produces a threat matrix that is evaluated and then mapped to mitigation recommendations.

To keep the matrix readable, threats are grouped by device class and by phase of operation. Separate subsections cover onboarding, routine operation, updates, and incident response. This structure helps reveal patterns, for example whether most high risk threats occur during setup or during day to day use. It also makes it easier to hand off relevant rows to different stakeholders, such as router vendors or cloud service operators.

3.5 Evaluation metrics

The project reports three metrics that are defined separately from the main text.

Coverage is defined as the fraction of applicable STRIDE categories that are identified across all modeled data flows. High coverage indicates that the model considers a broad range of attack scenarios. A coverage value of one means that every category appears at least once in the threat matrix.

Prioritization quality is the fraction of High risk threats that receive at least one strong mitigation. A strong mitigation removes the attack vector or reduces the residual risk to Low. In practice, this means that high risk items should never be left without at least one recommended control, even if that control requires vendor cooperation.

Mitigation completeness is the percentage of all identified threats that are mapped to at least one control together with a responsible party. This metric indicates whether the threat model leads to concrete and actionable recommendations rather than a purely descriptive list of problems.

These metrics allow progress tracking and comparison between iterations of the threat model without requiring device instrumentation or large scale experiments. They are well suited for environments where the analyst has limited ability to run active exploits, for example when working with off the shelf devices in a rented apartment.

3.6 Scope and assumptions

The analysis assumes default configurations on first setup, typical home ISP network address translation, and no enterprise grade management solutions such as mobile device management or centralized logging. The model focuses on threats that can realistically occur in a residential environment rather than targeted nation state attacks.

Where vendor documentation is unavailable, threats are inferred from protocol behavior, typical onboarding flows, and common IoT weakness categories described in the OWASP IoT Top 10 [1]. The study does not attempt to reverse engineer proprietary firmware, and side channel attacks are out of scope. These choices keep the project feasible for a semester course while still capturing the most impactful risks.

An important assumption is that the user is willing to apply basic configuration changes if given clear guidance. If users never change any default settings, even high quality mitigation advice would have limited effect. The discussion section revisits this assumption and its consequences.

3.7 Mitigation catalog

Controls are grouped into four categories.

Identity controls include unique credentials, multifactor authentication when supported, account lockout policies, and mutual authentication between devices and cloud backends. For example, cameras that support unique per device keys can resist spoofing attacks where an attacker pretends to be a legitimate device.

Update and firmware controls include signed updates, secure boot, rollback protection, and explicit user prompts for critical firmware changes. These measures protect against tampering with device software and reduce the risk that an attacker can persist in the system after a single compromise.

Network controls include segmentation using a guest SSID for untrusted devices, basic egress filtering, and rate limiting for protocols that are prone to abuse such as discovery services. Even when device level security is weak, proper segmentation can limit the blast radius of a compromise and can prevent a compromised camera from scanning the rest of the home network.

Privacy controls include data minimization, opt out of unnecessary cloud storage when local recording is sufficient, and clear indicators when sensitive sensors such as cameras or microphones are active. These controls address not only external attackers but also misuse of data by service providers and household members.

The proposed controls are compared qualitatively with the recommendations of SISA InfoSec, SecureDebug, and TechTarget to ensure alignment with broader industry guidance [1–3]. The final mitigation matrix in the artifact repository lists each control next to the threats it addresses and the responsible party.

4 Evaluation and Results

The threat modeling procedure was applied to the baseline smart home deployment. This section summarizes the resulting STRIDE coverage, prioritization quality, and mitigation completeness, as well as observations from specific threat scenarios.

4.1 Overall STRIDE coverage

All six STRIDE categories were observed across at least one data flow. Spoofing and information disclosure appeared most frequently in flows related to device onboarding and local API communication. Denial of service threats were concentrated around packet flooding of the router and wireless jamming of the lock or camera. Elevation of privilege threats arose when compromised user accounts or misconfigured roles could grant broader access than intended.

Table 1 shows a subset of representative threats and their mitigations.

Table 1: Sample STRIDE threat categories and mitigations.

Category	Example Threat	Risk	Mitigation
Spoofing	Fake device identity	High	Mutual auth, unique keys
Tampering	Firmware modification	Medium	Secure boot, integrity checks
Information disclosure	Unencrypted traffic	High	TLS, local API hardening
DoS	Packet flooding	Medium	Rate limits, segmentation
Elevation of privilege	Privilege abuse	High	RBAC, least privilege

In the final threat matrix, coverage exceeded ninety percent of the applicable STRIDE categories for the modeled flows. This suggests that the method captures a broad set of attacks and that the remaining uncovered categories are either out of scope or tied to uncommon scenarios.

4.2 Quantitative metrics

To make the evaluation more concrete, the analysis assigns approximate numeric values to each metric. In the baseline model, coverage reached roughly zero point nine two, prioritization quality approximately zero point eight three, and mitigation completeness approximately zero point eight five. These values are based on counts of threats in the matrix and provide a compact way to summarize progress.

Table 2 aggregates these numbers and separates threats by responsible party. The table shows that most user controlled threats have high mitigation completeness, while vendor controlled threats lag behind due to limited visibility and control.

Table 2: Summary of evaluation metrics for the baseline model. Values are approximate and derived from the final threat matrix.

Scope	Coverage	Prioritization	Completeness
User controlled	0.90	0.88	0.94
Vendor controlled	0.93	0.78	0.76
Shared responsibility	0.92	0.82	0.85
Overall	0.92	0.83	0.85

These results indicate that the main gap lies in vendor controlled mitigations such as secure boot and signed updates. Users can configure networks and passwords, but they cannot easily modify firmware behavior.

4.3 Case study: camera compromise

To illustrate how the threat model can be used in practice, the analysis considers a simplified camera compromise scenario. An attacker on the local network attempts to view or tamper with camera streams by exploiting unencrypted local APIs and weak default passwords.

The STRIDE matrix identifies spoofing threats where the attacker pretends to be the hub, information disclosure threats where video data is sent in clear text, and elevation of privilege threats when the attacker gains administrative access. Recommended mitigations include unique per device credentials, enforcement of TLS on local and cloud traffic, and network segregation that places cameras on a dedicated VLAN or guest SSID.

When these mitigations are applied, the residual risk rating for the scenario drops from High to Medium. Completely eliminating risk would require vendor side changes such as removing unencrypted local endpoints, but the case study shows that configuration changes alone can significantly improve the situation and gives a concrete example of how the methodology supports decision making.

4.4 Case study: smart lock misuse

A second case study focuses on the smart lock. Many locks support remote unlock commands from a phone application and also provide temporary access codes for guests or contractors. These features are convenient but introduce several STRIDE categories at once.

The matrix records spoofing threats where an attacker with stolen phone credentials sends unlock commands while pretending to be the legitimate user, and information disclosure threats when unlock events are logged only in the cloud with weak access controls. Denial of service threats appear when repeated failed attempts lock out legitimate users or when the lock loses connectivity and cannot verify codes. Elevation of privilege arises if a guest code is never revoked and silently becomes a permanent key.

Mitigations include multifactor authentication for remote access, clear time limits on temporary codes, local feedback on failed attempts, and notifications to the owner whenever a new device logs into the lock account. The lock scenario illustrates the tension between usability and security more strongly than the camera scenario, because overly strict policies can leave residents locked out. It also shows how the same framework can reason about both digital and physical safety.

4.5 User versus vendor responsibility

The metrics and matrices also highlight how responsibility is distributed. User controlled mitigations tend to be inexpensive but require awareness and effort, while vendor controlled mitigations often involve engineering cost but can benefit millions of homes at once. Shared responsibility items, such as secure cloud account configuration, require both sensible defaults from vendors and attention from users.

Presenting the results in this way can support conversations between stakeholders. For example, an internet service provider might decide to ship routers with separate IoT networks enabled by default, which would raise the mitigation completeness numbers for network related threats across many homes.

4.6 Mitigation completeness

Across all identified threats, approximately eighty five percent were mapped to at least one mitigation with a clearly identified responsible party. User controlled mitigations mostly involved network configuration and credential management, while vendor controlled mitigations involved secure boot, signed updates, and protocol design choices. A small number of threats, such as large scale denial of service on the ISP link, were categorized as residual risks with limited practical mitigation for individual users.

4.7 Limitations of the evaluation

The evaluation in this report is intentionally qualitative and model driven. The numeric values assigned to coverage, prioritization quality, and mitigation completeness are based on structured expert judgment rather than automated analysis or live experiments. As a result, the numbers should be interpreted as relative indicators that compare different versions of the model, not as precise measurements of the true security posture of any particular home.

The assessment also relies on a single baseline deployment rather than a broad sample of real houses. Homes with different network layouts, additional device classes, or legacy equipment might expose threats that do not appear in the current matrices. In addition, the evaluation does not capture user behavior over time, such as the tendency to delay firmware updates or to share credentials with family members. These human factors can strongly influence real security outcomes even when the technical model looks sound.

Despite these limitations, the evaluation still provides value. It forces the analysis to be explicit about which threats are considered, what counts as a strong mitigation, and where responsibility lies. The hope is that future work can build on this foundation by adding empirical testing, user studies, and larger data sets while retaining the transparent structure of the current model.

5 Discussion and Challenges

The study highlights several challenges in applying formal threat modeling frameworks to consumer smart homes. One challenge is incomplete or inconsistent documentation across vendors, which forces analysts to infer device behavior from limited information. This uncertainty can reduce confidence in risk ratings and may hide subtle vulnerabilities. For example, when a device does not document whether local traffic is encrypted, the analyst must treat the channel as potentially exposed.

Many effective mitigations require vendor changes rather than user configuration. Users can segment networks and manage passwords, but they cannot add secure boot support or redesign cloud APIs. This raises questions about how to communicate residual risk and responsibility to non expert users and policy makers. There is a risk that users will blame themselves for problems that can only be fixed by firmware updates or new hardware.

Usability remains another central concern. Some mitigations, such as strict network isolation or complex multifactor authentication workflows, can reduce convenience and are therefore unlikely to be adopted in practice. In interviews and user studies from related work, home users often express frustration with security prompts that interrupt daily routines. Balancing security and usability requires careful selection of controls and clear step by step guidance that fits within how people actually manage their homes.

There are also ethical considerations. Collecting data from smart home devices for testing can expose sensitive information about personal routines and relationships. Even threat modeling that does not involve active experimentation must avoid encouraging intrusive practices. The project intentionally focuses on configuration and publicly documented behavior rather than deep packet inspection of real user traffic.

From a methodological perspective, the qualitative metrics used in this project also have limitations. They are sufficient to compare different iterations of the threat model for this project, but they do not replace empirical measurement of attack success rates or performance overhead. For instance, a threat rated as medium impact might still lead to serious privacy harm in rare cases. Integrating empirical studies with this modeling approach, even on a small scale, would improve confidence in the ratings and could reveal where qualitative judgments were overly optimistic.

Another open challenge is how to teach this style of threat modeling to typical home users or to support personnel at internet providers. The artifacts produced in this project could form the basis for short training modules or interactive wizards that walk through each step for a specific house. Exploring how people understand and apply the STRIDE categories in practice would be an interesting topic for future human centered security research.

Finally, the model has focused on a relatively small set of devices. Real homes may include dozens of sensors and actuators, such as smart lights, thermostats, or voice assistants. Scaling the approach to such environments would require additional automation, for example tools that can discover devices and generate initial threat lists from network traffic and configuration exports. Automating parts of the STRIDE assignment and risk rating process would also reduce the chance of human error.

6 Concluding Remarks

This project shows that even a modest smart home deployment exhibits a rich set of security threats that span all STRIDE categories. By structuring the analysis around assets, data flows, and trust boundaries, the work produces a threat matrix that is both comprehensive and understandable to practitioners. The STRIDE based approach captures threats that align closely with those reported in industry studies, which suggests that the method is realistic.

The main contributions are a repeatable workflow for smart home threat modeling, a small set of metrics that capture coverage and mitigation progress, and a catalog of practical controls that map threats to responsible parties. The accompanying artifact repository provides documentation and templates that can be adapted to other IoT environments, so that home users and practitioners can reuse the analysis rather than starting from scratch.

Future work could extend the model to additional device classes such as smart speakers or sensors, incorporate limited empirical testing for selected threats, and explore automated tools that generate initial threat lists from protocol traces or configuration exports. Another direction is to design user facing guidance that translates the mitigation catalog into short checklists or configuration wizards that ordinary users can follow without specialized training.

I did not use AI tools to write this report.

References

[1] SISA InfoSec. 2024. *The OWASP IoT Top 10 Vulnerabilities and How to Mitigate Them.* Retrieved from https://www.sisainfosec.com/blogs/the-owasp-iot-top-10-vulnerabilities-and-how-to-mitigate-them/

[2] SecureDebug. 2023. *Threat Modeling for IoT Devices.* Retrieved from https://securedebug.com/threat-modeling-for-iot-devices/

[3] TechTarget IoT Agenda. 2024. *5 IoT Security Threats to Prioritize.* Retrieved from https://www.techtarget.com/iotagenda/tip/5-IoT-security-threats-to-prioritize