sustainability

MDPI

Article

IIoT: Traffic Data Flow Analysis and Modeling Experiment for Smart IoT Devices

Akashdeep Bhardwaj [1], Keshav Kaushik [1], Salil Bharany [2,*], Ateeq Ur Rehman [3], Yu-Chen Hu [4], Elsayed Tag Eldin [5,*] and Nivin A. Ghamry [6]

[1] School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India
[2] Department of Computer Engineering & Technology, Guru Nanak Dev University, Amritsar 143005, India
[3] Department of Electrical Engineering, Government College University, Lahore 54000, Pakistan
[4] Department of Computer Science and Information Management, Providence University, Taichung City 433, Taiwan
[5] Faculty of Engineering and Technology, Future University in Egypt, New Cairo 11835, Egypt
[6] Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt
* Correspondence: salil.bharany@gmail.com (S.B.); elsayed.tageldin@fue.edu.eg (E.T.E.)

Abstract: The Internet of Things (IoT) has redefined several aspects of our daily lives, including automation and control of the living environment, innovative healthcare services, and much more. Digital IoT devices and sensors, when integrated with home appliances, industrial systems, and online services in the physical world, have brought intense, disruptive changes in our lives. The industry and home users have widely embraced these 'things' on the Internet or IoT. However, the innate, intrinsic repercussions regarding security and data privacy are not evaluated. Security applies to Industrial IoT (IIoT) is in its infancy stage. Techniques from security and privacy research promise to address broad security goals, but attacks continue to emerge in industrial devices. This research explores the vulnerabilities of IIoT ecosystems not just as individual nodes but as the integrated infrastructure of digital and physical systems interacting with the domains. The authors propose a unique threat model framework to analyze the attacks on IIoT application environments. The authors identified sensitive data flows inside the IIoT devices to determine privacy risks at the application level and explored the device exchanges at the physical level. Both these risks lead to insecure ecosystems. The authors also performed a security analysis of physical domains to digital domains.

Keywords: IoT security; data privacy; sensitive data; IIoT apps; physical device; IIoT threat model

check for updates

Citation: Bhardwaj, A.; Kaushik, K.; Bharany, S.; Rehman, A.U.; Hu, Y.-C.; Eldin, E.T.; Ghamry, N.A. IIoT: Traffic Data Flow Analysis and Modeling Experiment for Smart IoT Devices. *Sustainability* **2022**, *14*, 14645. https://doi.org/10.3390/su142114645

Academic Editor: Andreas Kanavos

Received: 23 August 2022
Accepted: 2 November 2022
Published: 7 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

IoT has flourished its arches on every floor possible, be it industry, agriculture, energy projects, or transportation. Its application plays a major role in transforming the analog world into a digital one. Industrial IoT is topping the list while talking about the applications area. The industrial IoT application field encompasses various linked "things" initiatives within and outside factories and manufacturing facilities. Many IoT-based factory control and automation initiatives, for example, offer holistic innovative factory technologies with multiple features such as manufacturing department monitoring, wearables, and Augmented Reality on the production floor, remote Programmable Logic Control (PLC) control, or computerized quality management systems. An IoT system comprises sensors and devices communicating with the cloud over the Internet connection. Once the data reach the cloud, software analyzes them and may decide to take action, such as sending an alarm or automatically altering the sensors/devices without the need for the user's intervention. A user interface allows users to enter information or check in on the system if required. Any changes or actions taken by the user are then communicated back in the other manner via the system: from the user interface to the cloud and then back to the sensors/devices to effect change.

A smart factory is a computerized manufacturing facility that collects and shares data continually through linked devices, machines, and production systems. This information is then utilized to judge how to enhance procedures and deal with problems. Connectivity, data analysis, and diagnostics are important ideas underpinning the future factory, resulting in fewer shutdowns, enhanced processes, and optimized facilities. A smart factory makes use of cutting-edge technology and networking to optimize processes. Using IoT and artificial intelligence, for example, enables a more responsive yet also predicting, using available resources to produce cost-effective and efficient manufacturing. Assessing the manufacturing chain aids in selecting components, and assessing these key regions may reveal what should be improved next. This investigation should be led by a varied team of professionals with expertise in many business areas. IoT engineers collaborate with management and IT system professionals to identify areas for improvement, and a. Agy should be developed to optimize operations, boost sales, lower costs, and save time throughout the production process. Apart from Industry, IoT is revolutionizing with an unstoppable speed in every area possible such as transportation/mobility, healthcare, supply chains, and cities. The IoT was only a notion in the early 2000s; as we approach 2021, indications indicate that this innovation is here to stay. According to reports, 35.82 billion IoT devices will be deployed globally by 2021 and 75.44 billion by 2025 [1]. Devices in homes, healthcare, and electronics to industrial, mechanical, and manufacturing for monitoring, alerting, and automation, IoT devices running application services have transformed the human–digital interaction in the lives of home users and the industry. The contact between students and teachers and between students throughout the learning process can occur in synchronous and asynchronous forms, as well as face-to-face and electronic modes. Interaction with a smartphone app, visiting a website from a computer, and using IoT devices are all instances of human–computer interaction.

Although home users and the industry have embraced the systems supporting IoT, the security and privacy implication of these devices on our lives is still not fully understood. IoT installations have access to application functionality that, if exploited, might jeopardize user security. These IoT systems have complete access to sensitive private information, which, if released, might result in privacy concerns. As a result, it is necessary to identify potential hazards to any digital device before deploying it and to include suitable protections in the system as it is developed and architected. Understanding how an adversary might be able to identify common ground with a system helps guarantee that proper mitigation mechanisms are in place from the start. Thus, building the product with security in mind from the start is vital. In linked corporate IoT devices, their manner of conceivable contact surface areas and communication patterns must be studied to create a framework for safeguarding internet access to those gadgets. The term 'digital access' is used to distinguish actions carried out with direct device connection from those carried out with physical access control's access security. Place the gadget in a room with a locked door, for example.

Physical access cannot be prevented by software or hardware. However, efforts may be made to avoid physical access from communicating with the device. Evaluating the security of IIoT-based smart settings such as commercial and smart homes has become critical to effectively reducing security threats and dangers associated with deploying smart IIoT-based electronics devices. Since IIoT applications are exposed to a large amount of sensitive data from various sensors and devices connected to the central, one of the main criticisms of concurrent systems is that current commercial methodologies lack basic tools and services to analyze what they do with that data, pointing to application privacy. There are few tools available for assessing privacy threats in IIoT applications. The need is a set of analytic tools and methodologies aimed at platforms that may detect privacy problems in IIoT apps. This study investigates the methodologies and tools for defining the use of critical material and identifying vulnerable data flows in IoT deployments.

Conventional sensitive data tracking solutions built for mobile apps and other areas are insufficient. Existing tools may overlook sources such as sensor status (locked/unlocked)

and media such as IIoT network connections, making them easily evaded by rogue programs. Second, security-critical design defects in the permission architecture of IoT platforms, such as over-privileged device controls caused by present coarse-grained access restrictions, necessitate analysis sensitive to these privileges and their impacts. Finally, IoT-specific technologies such as system parameters and web application IIoT apps differ greatly from other platforms; hence, on-demand algorithms are necessary to ensure accuracy. Symmetric encryption techniques employ a single cryptographic key to encrypt and decode the data received. The technique is relatively simple because just one key is utilized for both actions. The main benefit of symmetric encryption is this. Because of its simplicity, this encryption technique is exceptionally quick, uses less power, and has no effect on Wi-Fi or internet performance. For data encryption and decryption, asymmetric encryption methods use multiple keys. Because asymmetric encryption is authenticated, the data can only be accessed by the person or organization that is supposed to receive it, which improves IoT security.

IoT ecosystems are implemented as devices sending data gathered on edge to the Internet and delivering smart 'tangible' services for society. Some examples are presented below.

- Smart greenhouse farming controls the environmental parameters—light, temperature, air pressure, and humidity of farms which has enabled huge increases in the yield.
- The use of actuators and sensors [2] allows farmers to monitor the farm conditions remotely.
- Glucose trackers continuously monitor Eversense and Dexcom [3] for diabetic patients and deliver insulin when a drop is noticed.
- A wearable defibrillator treats people [4] at risk of cardiac arrest as the implanted device monitors the heart rhythm. As soon as it detects any life-threatening rhythm, it produces a shock treatment and restores the heart rate to a normal rhythm.
- The list goes on as wearable around the neck to protect against concussion [5], EMG sensors for stroke patients [6], Apple watch for movement disorders [7], and even cancer treatment using Bluetooth-enabled blood pressure and cuff weight scale [8] linked to an app tracking the symptoms and monitored by doctors constantly.

The main IoT security measures to improve data protection include

- Introduction of IoT security during the design and each phase;
- Use of Public Key Infrastructure or PKI and digital certificates for networked node and server communications;
- Disabling port forwarding, closing all unused ports, blocking unauthorized IP address;
- Enhanced Application Programming Interface (API) security;
- Network access control to identify, track and authorize IoT devices connected to the network;
- Segmentation of IoT devices connecting to the Internet and restricting external access to the internal network;
- Use of security gateways as an intermediary between IoT nodes devices and the network;
- OS updates and patch management.

Threats cannot be eliminated and exist regardless of the security measures used to lower the likelihood of an attack. Controlling risks while recognizing the existence of threats is the goal of security measures in real-world deployments. A systematic strategy to effectively control and communicate dangers is threat modeling. Threat modeling determines how an adversary may access a system and ensures that appropriate protections are in place. Thanks to threat modeling, the design team may study countermeasures while the system is being designed instead of after it has been deployed. Because upgrading security features to a massive number of sensors in the field are impracticable, error-prone, and even harmful to users, this is crucial. Most real-world attacks are extremely targeted, aiming to exploit specific flaws in IoT items and connections. Many attacks are designed to take advantage of zero-day flaws. In the case of zero-day vulnerabilities, an exploit already exists and may quickly propagate throughout the Internet or corporate networks, resulting in a snowball effect. Because IIoT requires a significant investment of time and effort, the

bulk of attacks is carried out by nation-state threat actors seeking a massive impact. Some instances of common IIoT-related attacks are:

- Ransomware launched via malware;
- On wired and wireless networks, scanning and mapping attacks are common
- Attacks on network protocols;
- Infecting the intelligence of Industrial Control Systems and Supervisory Control and Data Acquisition (SCADA) systems;
- Attacks against cryptographic algorithms and key management;
- Data corruption attacks;
- Attacks on the integrity of the operating system and applications;
- Service denial and service jamming;
- Tampering or interface exposure are examples of physical security threats;
- Privilege escalation and other attacks on access control.

For SCADA, IoT threat modeling generates measurable data, including threats and related risks. This simplifies the approach to compliance while also saving money since it aids in detecting security problems, which can then be fixed safely and cost-effectively rather than relying on reactive measures. Because of its importance and complexity, a country's critical infrastructure necessitates systems that promote user interaction to deal with and communicate information about the monitored operations appropriately. SCADA architecture is used in the current systems used to monitor such facilities. The Internet's arrival and the necessity for quick information transmission at any time to enable correct decision-making processes have necessitated increased connectivity across SCADA and management systems. However, such interconnectedness might expose critical data handled by such systems to hackers, resulting in significant losses, financial losses, and even dangers to human life. As a result, unencrypted data may expose systems to cyber-criminals by compromising one or more cyber-security principles, such as confidentiality, integrity, or authenticity.

Given the explosive growth of IoT devices and the increasing importance of the domains they are used in, IIoT systems must improve on the largely ad hoc certifications present in current market practices. By recognizing sensitive data flows in IIoT apps and managing the flow from diplomatic sources, such as device state (door locked/unlocked) and user information (away and at home), to external sites, such as Internet connections and Short Message Service (SMS), the research proposes analytical tools and techniques for IoT data applications. The authors propose a system that prioritizes security and privacy characteristics, features with security-relevant failures, and features that breach a protective barrier.

The highlights of this research are

- Setup test-bed of IIoT devices (physical and virtual) to gather and simulate industrial IoT traffic;
- Present security analysis of IIoT device data flow in IoT applications;
- Provide secure data flow for physical and digitally integrated devices during implementations;
- Propose a new threat model to determine device mitigation controls and detect data flow compromise;
- Identified sensitive data flows from 256 IoT applications using the custom-designed tool and manual code analysis;
- This research demonstrated the proposed approach to effectively identify potential suspicious sources and network data points having sensitive data flows.

This research is structured as the introduction follows Section 2, which reviews and presents the process of shortlisting published literature and research by other authors. Section 3 introduces the research methodology and experiments performed for tracking the IoT data flow for inbound and outbound vectors and suspicious official and third-party applications. Section 4 presents experiments performed on the 256 IoT apps. The authors

categorize the applications based on their functions and suspicious data leak sources and present the red-flagged applications. The authors used Metricfire services to visualize and evaluate the IoT data for various metrics. Finally, Section 6 presents the conclusion, followed by references for the research.

2. Literature Survey

Several research efforts have focused on data security and privacy for IoT devices. For this research, 271 research studies published after 2018 were shortlisted from IEEE, Springer, and other referred journals. These papers focused on IoT security, threat modeling, IoT framework, data sensitivity, privacy, and device security. The authors followed a four-stage selection and rejection process by including only the relevant keywords, results, and methodology-based research work to finally shortlist 31 papers, as illustrated in Figure 1.

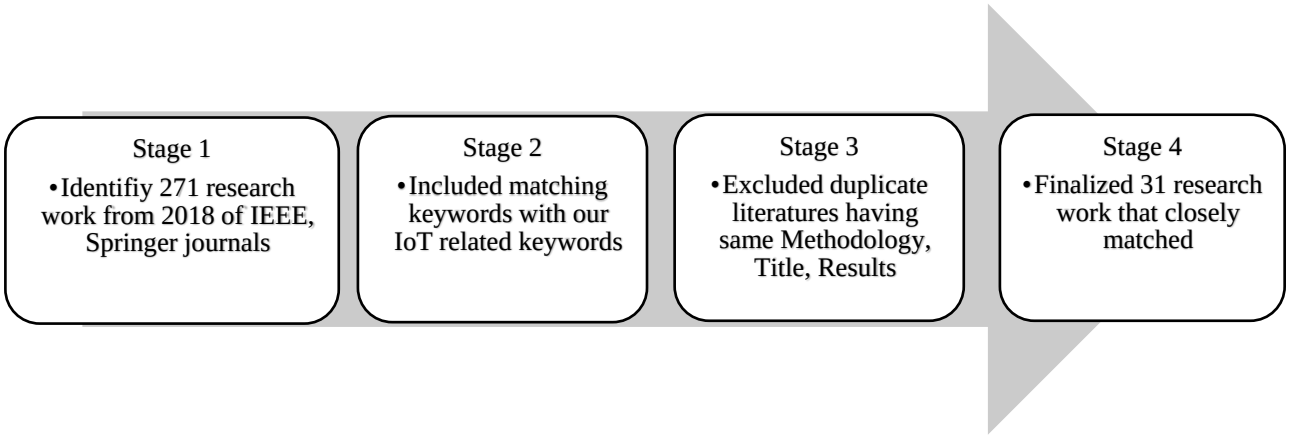


Figure 1. Four-Stage Selection Criteria.

Table 1 describes the overall spread of the research papers, the selected subcategories, and the latest reviews. Some of the closely matched and relevant references are presented below.

Table 1. IoT Security Literature Findings.

Classifying Papers	Stage 1	Stage 2	Stage 3	Stage 4	Final %
IoT Security	54	38	21	6	19.93%
IoT Threat Modeling	49	34	19	6	18.08%
Sensitive Data	51	36	20	6	18.82%
IoT Data Privacy	38	27	15	4	14.02%
Security Framework	47	33	18	5	17.34%
Device Security	32	22	12	4	11.81%
	271	190	104	31	

By collecting network data's temporal and geographical representations, Abdel-Basset et al. [9] introduced a unique federated deep learning model for hunting cyber threats against Industrial IoT. The model is then deployed as a threat-hunting micro-service on appropriate edge servers using a container-based industrial edge computing framework with good resource orchestration. An exploratory micro-service placement approach was devised to address the latency difficulties to allow for improved micro-service deployment based on the participants' computational capacity. In terms of accuracy (92.97%, 92.84%) and f1-scores, simulation data from two public benchmarks verify the usefulness of these techniques (91.61%, 90.49%).

The problem of software development while addressing these devices' security and safety concerns will continue to rise as the IoT and Intelligent Transportation Systems

(ITS) products grow in popularity. Tashtoush et al. [10] provided a thorough and in-depth examination of agile software development in the context of IoT, ITS, and their associated cybersecurity and risk issues. We also systematically compare the examined literature using a set of predetermined criteria. Finally, we offer a wide perspective and a framework for developing future secure, agile software development solutions for IoT and ITS systems.

Mills et al. [11] went above and beyond existing methods, combining anomaly detection and Cyber Threat Intelligence with parallel processing to profile and identify potential cyber threats. Citrus: a novel intrusion detection framework capable of tackling emerging threats through collecting and labeling live attack data by utilizing diverse Internet vantage points to detect and classify malicious behavior using graph-based metrics and various Machine Learning (ML) algorithms was demonstrated in the research. The findings confirmed that the proposed approach is a realistic and practical option for next-generation network defense and resilience techniques.

Creating security standards and evaluation frameworks that best match security expectations and completing testing and disclosing the security posture of IoT-based intelligent devices is generally recognized as a tough challenge. To identify those that could resolve some of the security needs of IoT-based smart environments, Karie et al. [12] presented a review of existing security assessment and evaluation frameworks, including many NIST special publications on security techniques highlighting their primary areas of focus. Throughout the review process, current and future security demands, as well as effectiveness, were examined. While the majority of existing security standards and assessment frameworks do not directly address the security needs of IoT-based intelligent devices, the findings imply that they might be updated. The difficulties and issues around security flaws in IoT-based smart environments were also discussed in this article.

Iqbal et al. [13] looked at IoT networks' hazards, security requirements, barriers, and attack vectors. Based on the gap analysis, a new paradigm was developed that combines network-based IoT design with software-defined networking. To offer a thorough overview of software-defined information security, the research presented an overview of Software-Defined networks (SDN) and a detailed discussion on IoT deployment patterns and produced SDN-based IoT security solutions. The authors addressed key difficulties, significant hurdles in bringing all IoT stakeholders together on a single platform, and a few key findings stressing the need for a network-based security solution for the IoT paradigm.

Rathore et al. [14] presented a deep learning and blockchain-enabled cybersecurity approach for intelligent 5G-enabled IoT, which relies on competence for efficient data analysis operations and blockchain for data security. Secure data storage and processing due to blockchain immutability and records of all data changes, data may be stored safely and securely. There are no single-point failures: due to the decentralized nature of blockchain systems, a single node failure does not affect the entire network. Thanks to blockchain, it is now simple to securely exchange, examine, and save digital data. Furthermore, every transaction is protected by cryptographic encryption. By doing so, banks may raise their present security and transparency standards to new heights. The framework's hierarchical structure represented the fog, edge, cloud, and user levels, showing the operations at each level. Common latency, accuracy, and security tests are performed to evaluate the architecture's usefulness in actual applications.

GDPR-related smart city structural and security solutions were presented by Badii et al. [15]. IoT systems, IoT edge on-premises, IoT programs on the Internet and on-premises, data analytics, and dashboard were all part of the system's comprehensive security capabilities that went beyond state of the art. The stress test also includes penetration testing to ensure the solution's resilience in the face of a wide variety of potential vulnerabilities.

Akil et al. [16] offered a complete literature study to answer the issue of what the literature presents types of privacy-preserving identifiers in IoT contexts for establishing pseudo-anonymity. It contains classifications and ratings of IoT scenarios for which privacy-preserving IDs have been suggested, as well as the types of pseudonyms and underpinning

identity management systems employed. Furthermore, it discusses trends and gaps in resolving privacy trade-offs.

Li et al. [17] developed the first distributed security outsourcing approach for arithmetic operations using a non-colluding edge node topology (fixed base and variable exponent). The authors devised a method for breaking an exponent into a predetermined number of pieces and a safe, distributed modular multiplication technique. The user can maintain privacy throughout the outsourcing relationship and detect erroneous findings from edge nodes with a high level of certainty. Finally, we demonstrate that the suggested techniques are cost-effective for both the user and the edge node.

Yu et al. [18] proposed a crowdsourced privacy protection approach based on multi-authority ciphertext regulation essential element encryption to improve privacy protection in a data-sharing environment. The authors developed an independent, crucial element distribution approach via different authorities in this paper, which might effectively reduce the measures that must be taken while utilizing the crowdsourcing platform. The authors advised partial network decryption to cut mobile users' processing costs and prevent corporations from snooping on their data. This study proposed an efficient attribute revocation method and a task search feature to provide dynamic on-demand services while ensuring task forward and backward security. The theoretical research establishes the correctness of decryption and phrase pairing and the security of each connected entity. According to simulation data, this technique beats other related systems in terms of time consumption.

At the convergence of security and privacy standards with the quest for new data uses, Sollines et al. [19] addressed difficulties in collecting, using, and managing big data. The authors devised a three-stage deconstruction of the design space to understand demands and limits better. Recognizing these many purposes for IoT big data management design, we believe that more effective design and control may be achieved at the intersection of these factors through an iterative review process and redesign.

Neshenko et al. [20] focused on the constantly evolving IoT vulnerabilities. The writers organized cutting-edge surveys into categories that address many facets of the IoT paradigm. The goal was to make IoT research possible by collating, comparing, and compare and contrasting disparate studies conducted and then propose a unique taxonomy that illuminates IoT security flaws, possible attacks, impacts on various security goals, attacks that exploit such weaknesses, corresponding remediation methods and techniques, and available online operations and maintenance cyber security functionalities to infer and supervise such weaknesses. This research aims to give the reader a multidimensional viewpoint on IoT vulnerabilities, encompassing technical specifics and repercussions, so they may be used to remediate them.

Data security vulnerabilities have also arisen, such as inadequate indication filtering and a lack of scientific verification of reputation assessment judgments. To address the concerns, Kong et al. [21] proposed a security reputation model that included convolutional neural networks and adaptive game theory. This framework protects the privacy of health data on the Internet of Things. The model was shown to be capable of handling the difficulties of poor dependability of health data filtering indexes and low accuracy of credit distinction in the cloud environment via experimental findings.

IoT has generated major security and privacy concerns in recent years while promising unprecedented ease, availability, and efficiency. While research towards reducing these risks is progressing, many obstacles still exist. Zhou et al. [22] investigated the concept of IoT features. They then mentioned the privacy and security repercussions of eight IoT features, such as the threats they cause, existing alternatives to threats, and research challenges yet to be solved, to truly comprehend the compelling reasons for new IoT threats and the challenges in current research. This study highlighted the rising trend of IoT security research and suggested how IoT characteristics affect recent security research by reviewing the bulk of existing research works connected to IoT security.

Xiong et al. [23] proposed a privacy and availability data clustering strategy based on the k-means algorithm and asymmetrical privacy that improves the identification of initial

center points and the closeness calculation methodology from other points to the center point. By detecting outliers during the clustering process, this research tried to reduce the influence of outliers. The security analysis demonstrates that our method achieves differential privacy and avoids the leakage of private information.

3. Research Methodology

IoT applications have access to user data that are sensitive and highly private. Tracking data flow dynamically or statically has been well-researched and applied with inbound and outbound configurations. The authors researched different IoT platforms, each having unique challenges and characteristics related to tracking data flow. Most operating systems run well-defined OS codes internally that represent the IoT source code. However, IoT platforms were diverse, with every vendor having its protocols, data formats, and programming languages. This research tested five IoT operating systems as

- Contiki-NG: an open-source OS focuses on network and memory-constrained, low-power devices.
- FreeRTOS: uses Amazon Web Services to run IoT apps with a 19 KB memory footprint. This is suitable for small microcontrollers.
- MicroPython: a compact, open-source python-based OS that runs high-level and low-level codes.
- Embedded Linux: is built for small, embedded devices, especially those with high-speed network features.
- ROIT: considered Linux of the IoT world, this open-source OS supports multi-threading, SSL/TLS, along with 8, 16, and 32-bit processor architectures.

The authors propose a secure zone-based design for threat modeling the IoT ecosystems to optimize sensitive data security modeling in IoT devices. The zones need to be subdivided as per the device components for threat modeling. The zones include user trust, IoT devices, cloud gateways, and application and service zones. Such compartments present a segmented design, with each zone having its authentication, authorization requirements, and data flow. These zones isolate any damage and restrict the impact across the infrastructure, with each zone segregated by the trust boundaries. IoT Threat model architecture separates the zones per the user, gateway, and device capabilities and services. This approach enables secure gateway routing and data flow. IoT devices with external network access should securely send data and communicate with the cloud gateways. This feature requires high encryption and processing overhead compared to devices that gather and display data locally.

The mind map in Figure 2 represents the data transition from one device source to the other, with the proposed zones elaborated as follows.

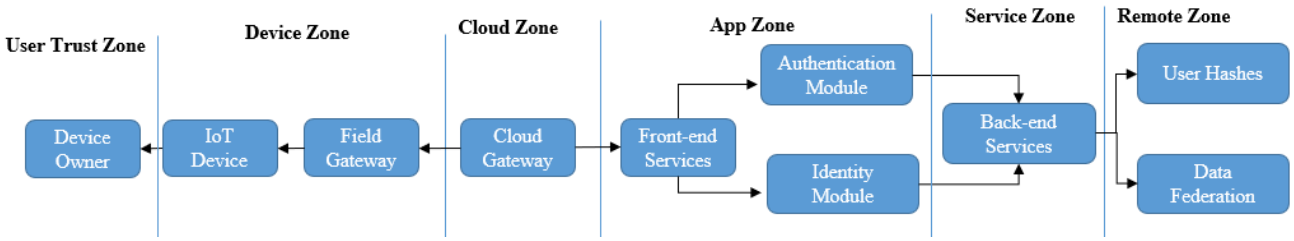


Figure 2. Zone-based Mind Map.

- User Trust Region: The device ecosystem is the obvious physical zone around the device that allows physical access and peer-to-peer digital access through the local network. Any short-range wireless radio technology that allows device-to-device communication and is independent of yet connected to the public Internet is referred to as a local network. Voice or data transmission between several communication devices can be accomplished using short-range wireless radio technology. When

paired successfully, a Bluetooth device may transmit a radio signal across a short distance to locate another Bluetooth device with which it can connect. As a result, it may be used to replace existing digital devices' cable connections. It excludes any network virtualization solution that creates the appearance of a local network and public operating company networks that force any two devices connected through peer-to-peer communication to interact across public network space.

- Internet of Things Device Region: A field gateway is a communication enabler, device management system, and data processing hub built into a device/appliance or general-purpose server software application. The field gateway zone is where the field gateway and any connected devices are kept. Field gateways, as their name implies, operate outside permitted data processing facilities, are often location-bound, vulnerable to physical intrusion, and have little operational redundancy. All of this suggests that a field gateway may typically be tampered with and broken without understanding what it is for.
- Cloud region: A cloud gateway is a system that allows remote communication from and to equipment or field gateways through a public network, usually to a cloud-based control and data analysis technique or a federation of such systems. Sometimes, a cloud gateway may offer instant access to special-purpose devices from endpoints such as tablets or phones. The term "cloud" refers to a customized information system not physically located near the associated devices or field gateways. Cloud Zone's operational constraints preclude targeted physical access and are not necessarily accessible in a public cloud architecture. Cloud gateway could be mapped into a virtualized network overlay to isolate it and all of its linked devices or field gateways from other network traffic. The cloud gateway is neither a device control system nor a data processing and storage facility; they are services provided by the cloud gateway. The cloud gateway zone contains the cloud gateway and any field gateways and devices that are either directly or indirectly connected to it. The zone', a separate surface region, is where all external parties interact. Examples of device control systems in our day-to-day life include an air conditioner, a refrigerator, an air conditioner, a bathroom toilet tank, an automatic iron, and many processes within a car—such as cruise control.
- Apps and Services Section: Any software component or module that connects with devices via a field or cloud gateway for data collection, analysis, and command and control is referred to as a service in this context. Service providers are acting as go-betweens. On behalf of gateways and other subsystems, they give information and control features to authorized end-users, store and analyze data, issue orders to devices autonomously based on data revelations or schedules, and operate on account of gateways and other subsystems.

Threat modeling helps design teams to think about mitigations early in the development process rather than after the system has been implemented. Because retrofitting security protocols to a multitude of devices in the field is cumbersome, error-prone, and even harmful to users, this is crucial. Many development teams excel at identifying the system's system requirements that benefit users. On the other hand, finding non-obvious ways for someone to abuse the system is more difficult. Threat modeling may assist developers in determining what an adversary would do and why. Threat modeling identifies and rates threats based on their likelihood of occurrence based on a comprehensive grasp of a system's architecture and implementation. This enables us to manage risks in a prioritized manner that is best-effective and efficient. Threat modeling is used during the design process and provides the most benefit. There is room for modification when designing to eliminate dangers. The goal is to eliminate dangers via design. It is a lot simpler than establishing mitigations, evaluating them, and making sure they're always up to date, and it is not always possible. Threat removal gets more difficult as a product matures, requiring substantially more work and far more nuanced decisions than threat assessment early in development. Threat modeling is a rigorous technique that generates a conversation about the system's security design decisions and modifications that impact security. A threat

model, albeit just a document, is an excellent tool to assure knowledge continuity, retention of lessons gained, and quick training of new teams. Finally, threat modeling enables you to think about other areas of security, such as the security assurances you want to provide your customers. These agreements serve as the framework for IoT solution testing and threat modeling. The threat modeling process is performed as illustrated in Figure 3 with the following steps:

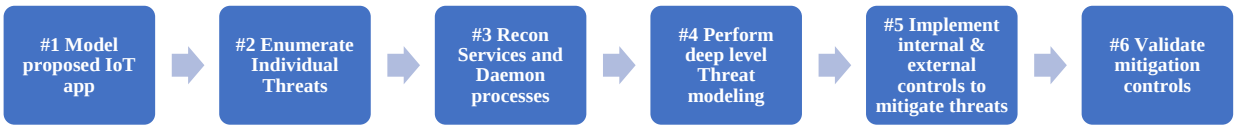


Figure 3. Threat Modelling Process.

The authors studied the device operating systems and apps to identify IoT-specific program codes, sinks, and sensor computation application structures. The authors translated the application source code to an intermediate representation. This provided the application lifecycle, which included the user inputs, data entry–exit points, log generated, and the sensor states. This helped identify IoT events and app actions, and using static code analysis, sensitive data flow was tracked from sources to the sinks. This helped identify vulnerable data flows. The significance of tracking this data flow is device depended and dependent on the IoT device, the setup configuration, and the environment where it was being deployed. Thus, different data flows had different impacts on the device's data security and privacy, which could present potential areas for exploitation and misuse.

The authors identified the data flow from potential risk sources to sinks due to default configuration setup, user carelessness, and malicious intent. Any app that leaks sensitive data, with or without device/user permissions, is considered malicious. Such apps act like trojans to violate user privacy and deviate from the original functional state. The IoT OS with root-level permissions leak information when granted default access to device-specific modules. External threats easily bypass any security controls of the IoT platform and exploit via the communication channels. The application source code is analyzed to identify any trace of sensitive data from potentially suspicious sources. Flag labels denote the data sensitivity level, type, and source. The static analysis tracks the data flow in the device and propagation in various stages by the application and the network interfaces. Any trace of the labeled data being transmitted out of the previously determined data flow in the app or at outbound sources to the messaging module or the Internet, an alert is raised by the proposed framework. However, the flagged data getting leaked is malicious or violates privacy is not flagged. This decision is left to the user to make an informed decision and rule about potential sensitive data leaks or privacy risks, as shown in Figure 4.

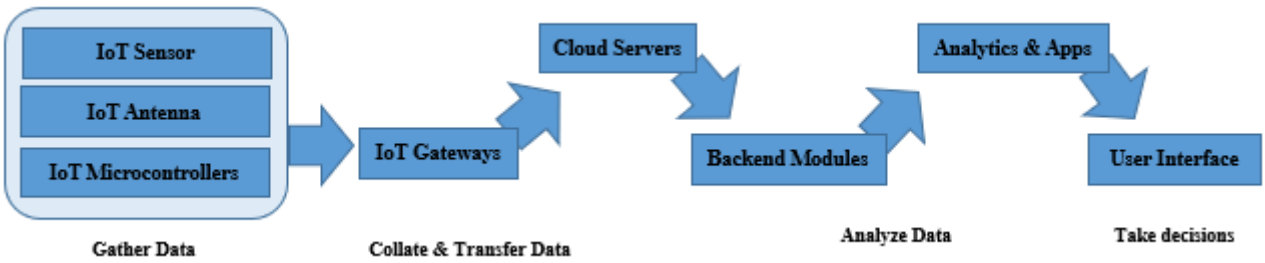


Figure 4. IoT Device Data Flow.

IoT applications perform various actions to generate data and control the flow during different events. The actions are primarily sent to event handlers that change the device state, such as calling an event handler module when a sensor alert is triggered due to motion detection to switch on the webcam and start recording or even change the state of the light bulb from off to on. However, other vendors and third-party functions are

sometimes invoked, such as sending messages, performing logic, and log device events, and saving the processed data to the device storage. During various actions around and in the IoT device, the event handlers execute their actions to make it necessary to track any sensitive data propagating inside the application, storage, and business logic modules. The information gathered at the preceding tiers is only useful if it leads to a problem-solving solution and achieving business objectives. New data must spur collaboration among stakeholders, implementing new methods to boost productivity. More than one individual working with many software solutions is generally involved in the decision-making process. As a result, the business logic layer is represented as a distinct level above the application layer. Any injection vulnerability pattern in the source code is detected via taint analysis. The analysis finds the untrustworthy input information flow that impacts the system's sensitive sink or portion. Taint may also be described as causing something to degrade or become polluted. A terrorist might poison the public water supply by putting a chemical in it. A taint is defined as something that contaminates or makes something unattractive. A criminal charge on a permanent record is an example of a taint. To determine the exact taint propagation, initially, the event handlers are initially ich propagate data via different entry-exit points and the storage in the device. Sensitive data flow from these sources, save update, or create states. If the flags are set, the data are deleted immediately, as shown in Figure 5.

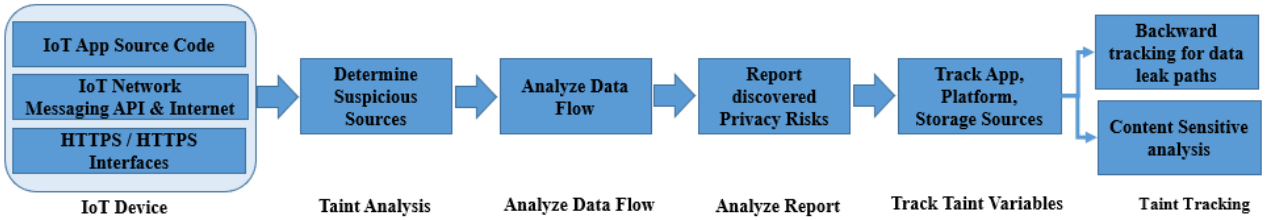


Figure 5. IoT Data Flow Analysis.

The common suspicious sources and network points posed a threat to privacy and potential leaking of sensitive data flowing in the device apps. These are presented in Table 2 as follows.

Table 2. IIoT Suspicious Sources and Investigations.

Suspicious Sources	Investigation Reveals
Device Info	Apps allow access to devise platforms and reveal access to interfaces to gather device information, Model, Make, ID, and Manufacturer. Based on this, vulnerabilities are found, which allow remote shell access to writing into device apps.
Device Geolocation	Device location refers to the site's geographical location, such as the time zone, PIN, latitudes, and longitudes, which can be used to control business logic-based apps.
Messaging Services	IoT apps send SMS messages to admin owners and user recipients or use APIs to deliver push notifications to apps, mobiles, and App users. Messaging service interfaces exfiltrate data and are the most vulnerable source.
Device State	This attribute obtains sensitive data for physical and privacy risks, which increase via the device state and interfaces.
State Variables	IoT apps often store 'persistent' execution events; platforms enable programs to save data to a proprietary external storage location and retrieve it later.
User Input	Apps require Admin control or user inputs via web forms, which are mostly HTTP or have expired certificates. This info is saved in the device that can have personally identifiable data. This can be exploited for profiling user behavior.

The research setup involved the use of real-world IioT devices as well as a virtualization application to implement and run virtual machines simulating the IoT devices. The authors also simulated IoT services with the hardware configuration involving an Intel i5 CPU, 8 GB RAM, 500 GB SATA drive running Windows 10 OS 64-bit with Java runtime

environment as the software with Python 3.1 and a few customs developed IoT security tools to evaluate the device data flow and applications. From a collection of open-source monitoring technologies, MetricFire [24] offers a comprehensive platform for infrastructure and application monitoring. Utilize Hosted Graphite to store your data and analyze analytics in real time on stunning dashboards. Without having to deal with the headaches of setting up your own server or worrying about scaling, backups, or maintenance, hosted Graphite enables you to monitor, analyze, and visualize massive volumes of data about your applications and back-end systems [25–30]. There are several supported tools, add-ons, and plugins that may help you obtain the precise data you want out, as well as many straightforward ways to obtain data [31–36]. Due to the confidentiality clause, the tool names or the code cannot be revealed. However, the results obtained during the research are presented in the results section.

4. Results

The authors researched 256 IoT apps, which included 105 official (industry-approved) and 151 third-party (unapproved). IoT app structure did portray some basic level of common structure, common entry-exit points for suspicion and networks, yet the results were alarming:

- This research red-flagged 84 out of 105 officially approved and 127 out of 151 third-party apps as insecure and capable of leaking sensitive data from their message services to the Internet.
- Over 60% of these apps were transmitting five types of sensitive data ranging from device information, geolocation, firmware version, device state, and user input.
- Over 75% of the apps are in three different sources and eternal URLs, Server Hostnames, and IP Addresses.

Table 3 presents the APIs used by the Internet and messaging apps in the IioT devices.

Table 3. IioT Messaging and Internet APIs.

Messaging Service	Internet API Calls
Send_SMS()	Http_Get()
Send_Notificaiton()	Http_Post()
Send_Notificaiton_event()	Http_Delete()
Send_Notificaiton_Contacts()	Http_Put()
Send_Push_Message()	Get (web service app)
Clear_Log()	Post (web service app)
Allow_Login()	Delete (web service app)
Reload_Daemon()	Put (web service app)

This research categorized each IioT device based on the application functionalities and source code for suspicious sources and data leaks. It was found that all IioT devices are leaking data and device information. The abnormally high percentage further emphasizes the research objective regarding IoT devices being insecure and potentially exploitable. The data are analyzed and presented in Table 4 and illustrated in Figure 6.

Table 4. IioT Application Categories.

IioT Application Type and Functions	Data Flow App Sources Leaking Info					
	Device Information	Device Geolocation	Messaging Services	Device State	Device Variables	User Input
Security-related	72%	45%	95%	88%	86%	89%
Plant Automation	88%	91%	67%	78%	88%	85%
Smart Sensor	93%	95%	83%	91%	99%	93%
Smart Camera	99%	99%	90%	94%	99%	99%
Pollution Monitor	98%	98%	95%	91%	98%	99%

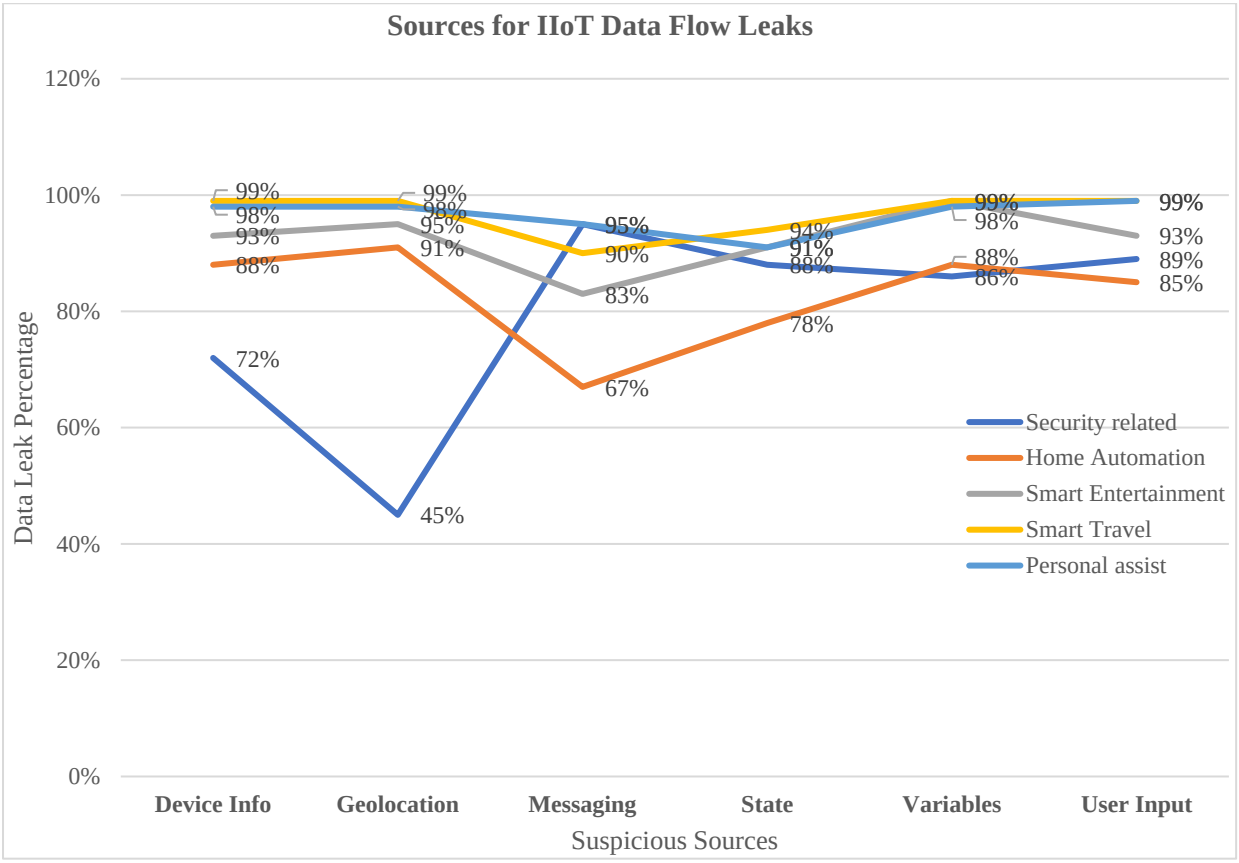


Figure 6. Sources for IioT Data Flow Leaks.

Each IoT device is analyzed to track the sensitive data flow for any outbound access to the Internet or when using API calls and messaging services by the device apps. Table 5 presents the IoT applications connecting to the Internet and messaging for transferring any sensitive information to outbound sources.

Table 5. Outbound Calls by IoT Apps.

Type of Apps	Apps	Red Flagged	Internet Access	Messaging	Both
Official/Vendor app	105	84	21	41	22
Unofficial/Third party	151	127	31	43	53

The authors performed an initial analysis of the IioT devices for the app permissions and data flow route of the 256 applications. The check was performed to detect outbound permissions to the Internet and messaging. From the initial analysis, around 80% of official applications of vendors and 84% of unofficial applications were red-flagged as suspicious and potentially leaking data to external sources. Of these red-flagged apps, official apps had 26% of both Internet and messaging access, while unofficial apps and 42%. Internet access was 24–25% for both types of applications. However, official apps displayed 49% of secure messaging calls, while unofficial apps had 34% of messaging counts. This revealed that the official applications followed good practices and were manually verified for concurrent connections to the devices, as presented in Figure 7. The authors gathered network traffic data from the IioT devices and performed network traffic analysis using a graphite monitoring service.

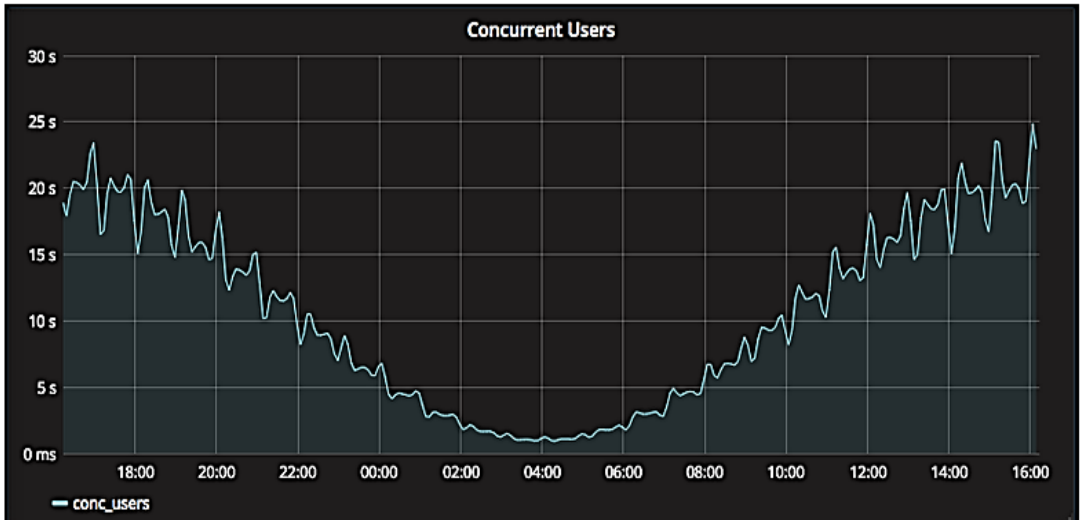
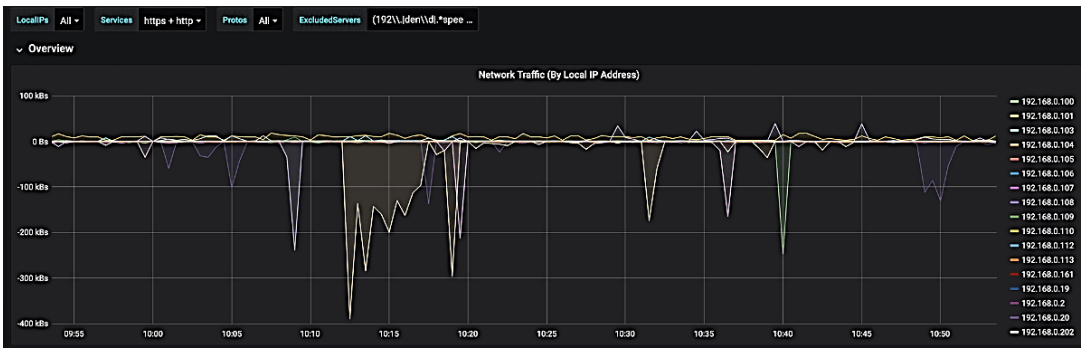
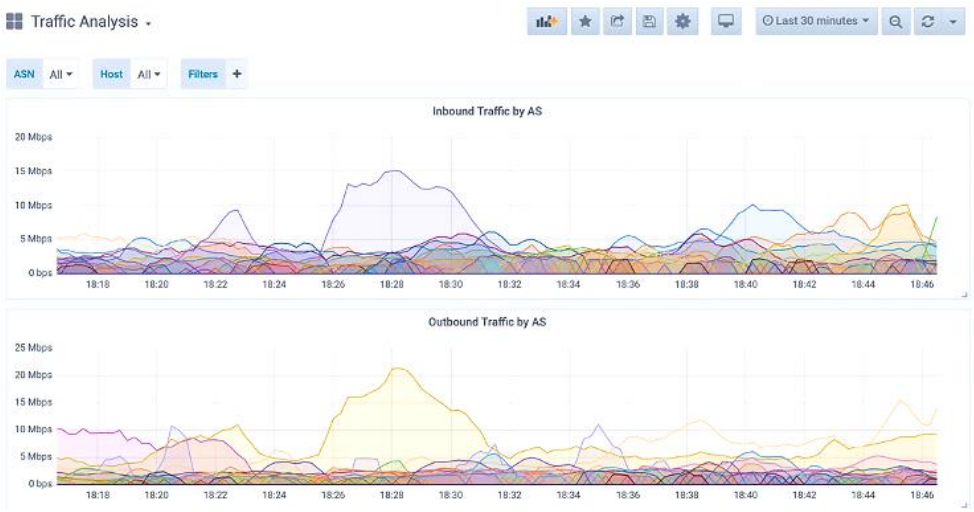


Figure 7. Concurrent Connections to IoT Device [24].

The advantage of using the MetricFire [24] services is the inherent ability to ingest time-series IoT logs and data, apart from storing and visualizing the traffic related to different trends, loads, and reliability from the IoT nodes as well as Cloud portals. Figure 8 illustrates the network traffic for the different IioT devices in the experimental setup.



(**a**)



(**b**)

Figure 8. Network Traffic Analysis. (**a**) Network Traffic for Different IioT Devices [24], (**b**) Detailed Network Traffic Flow [24].

The authors observed the applications interacting with external cloud portals via API and HTTP(S) requests. Most devices use web services to send and receive data to and from the cloud portals. These data included event logs, device states, and received commands for executing actions. This specific trend of having multiple inbound and outbound connections, as illustrated in Figure 9, is the primary cause of concern for IoT devices.

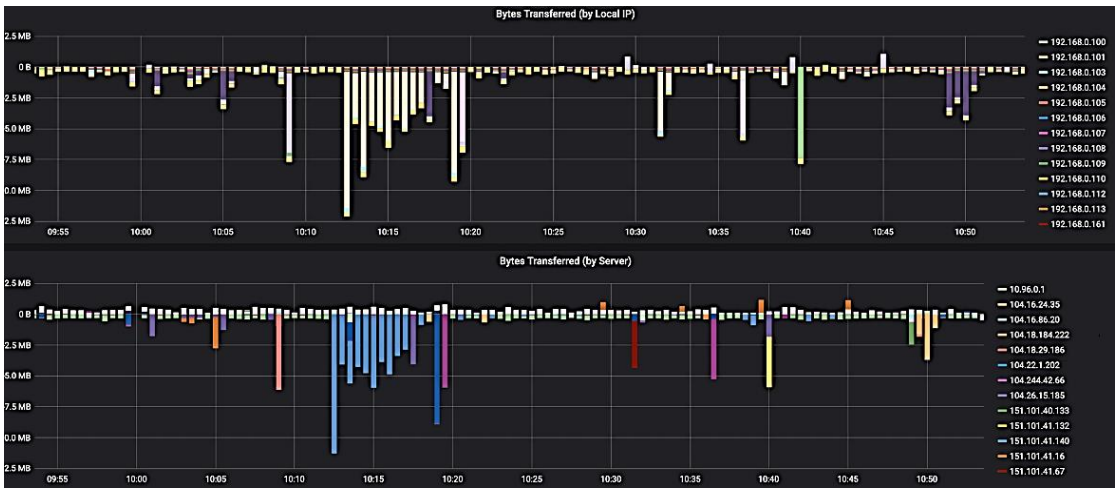


Figure 9. Inbound and Outbound Traffic.

The results also reveal that IoT applications tend to implement little or no security controls to protect data and logs. Data flows depend on application functionalities, so outbound flow to the Internet or messaging services cannot be avoided. The IioT devices are connect with external URLs, Autonomous System Numbers (ASNs), and IP addresses, as illustrated in Figure 10a,b.

(**a**)

Top Inbound ASNs

ASN	AS Name	Bandwidth
62744	QUINTEX - Quintex Alliance Consulting	3.84 Mbps
12876	Online SAS	3.52 Mbps
16276	OVH	3.35 Mbps
24940	HETZNER-AS	3.24 Mbps
200651	FLOKINET	2.87 Mbps
200052	FERAL Feral Hosting	2.59 Mbps
208323	APPLIEDPRIVACY-AS	1.86 Mbps
4224	CALYX-AS - The Calyx Institute	1.61 Mbps
53667	PONYNET - FranTech Solutions	1.37 Mbps
396507	EMERALD-ONION - Emerald Onion	1.28 Mbps

Top Outbound ASNs

ASN	AS Name	Bandwidth
16276	OVH	7.87 Mbps
24940	HETZNER-AS	5.63 Mbps
12876	Online SAS	2.94 Mbps
36492	GOOGLEWIFI - Google, LLC	2.23 Mbps
20473	AS-CHOOPA - Choopa, LLC	2.18 Mbps
63949	LINODE-AP Linode, LLC	1.69 Mbps
60781	LEASEWEB-NL-AMS-01 Netherlands	1.62 Mbps
11426	TWC-11426 - Charter Communications Inc	1.50 Mbps
49453	GLOBALLAYER	1.29 Mbps
53667	PONYNET - FranTech Solutions	1.08 Mbps

(**b**)

Figure 10. Analysis ASNs and IP addresses. (**a**) External URL and IP Address Connections by IioT Devices, (**b**) Top Inbound and Outbound Internet and Messaging Service Traffic.

The good news is that there is no direct correlation between the devices and the app managing the sensitive data flows. However, IioT applications have inbuilt data flow using SMS, API [37–40], or push notifications with external Internet requests when integrating with other devices or web portals. The cumulative distribution for external interface and outbound exfiltration is illustrated in Figure 11.

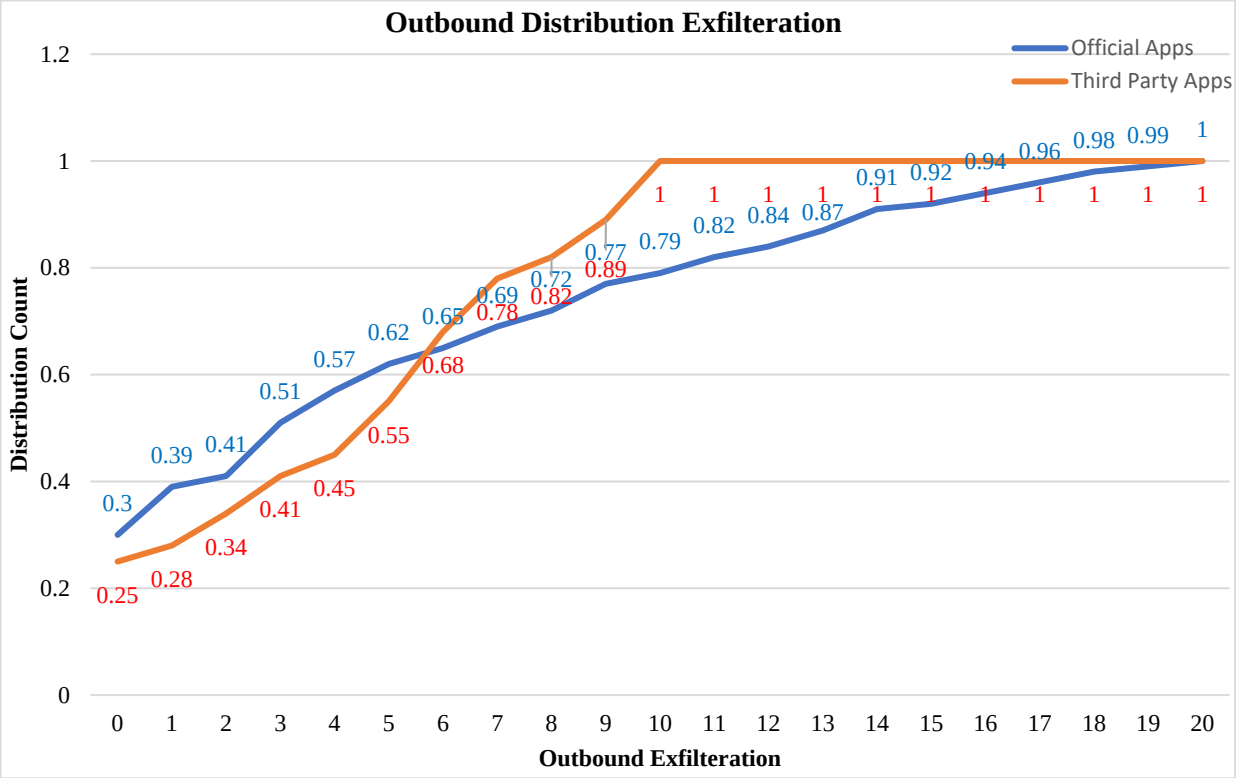


Figure 11. Distribution Function for Outbound Exfiltration.

The efficiency of this experimental investigation on official and third-party IoT apps in detecting sensitive data flows is shown. At least five distinct categories of sensitive data are sent to the Internet and instant messengers by IoT applications.

5. Conclusions

The authors performed research and threat modeling to study the features and response of data flow in IoT devices. This research set up a smart test-bed of IioT devices (physical and virtual) to gather and simulate industrial IoT traffic. The major challenge IoT devices face is the apparent lack of visibility and security control on the use and traffic access by IoT device applications. The authors also presented a threat modeling framework. The authors identified sensitive data flows within 256 IoT applications using a custom-designed tool and manual code analysis. This included reviewing potential suspicious sources ranging from device information, state, geolocation, messaging services, variables, and user input.

The IoT applications are categorized based on their functionalities for official and third-party apps. This research red-flagged 84 out of the 105 officially approved and 127 out of the 151 third-party apps as insecure and capable of leaking sensitive data due to their design of connecting to external sources with API, message services, and ASN outbound access to the Internet. In addition, 60% of these apps were transmitting five types of sensitive data ranging from device information, geolocation, firmware version, device state, and user input. Over 75% of the apps revealed different network interfaces and receivers, external URLs, server hostnames, and IP addresses. This research demonstrated

the proposed approach to effectively identify potential suspicious sources and network data points having sensitive data flows.

6. Future Work

The approaches and techniques performed in this paper can help future research provide a rigorous, solid framework for future evaluations of the use of sensitive information, as well as safety and security properties in IoT-based applications and environments, allowing IoT designers, developers, markets, and consumers to have a verified process for identifying potential threats to IoT data flow security and privacy.

Author Contributions: Conceptualization, K.K., A.B., S.B., A.U.R., Y.-C.H., E.T.E. and N.A.G.; methodology, K.K., A.B., S.B., Y.-C.H., A.U.R., E.T.E. and N.A.G.; software, K.K., A.B. and S.B.; validation, A.B., K.K. and S.B.; formal analysis, A.B., K.K. and Y.-C.H.; investigation. A.B., E.T.E. and N.A.G.; resources, E.T.E. and N.A.G.; data curation, S.B.; writing—original draft preparation K.K., A.B., S.B., Y.-C.H., A.U.R., E.T.E. and N.A.G.; writing—review and editing, K.K., A.B., S.B., Y.-C.H., A.U.R., E.T.E. and N.A.G.; visualization, S.B.; supervision, A.B. and S.B.; project administration, Y.-C.H. and E.T.E.; funding acquisition, E.T.E. and N.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Future University Researchers Supporting Project Number FUESP-2020/48 at Future University in Egypt, New Cairo 11845, Egypt.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IoT Devices Installed Base Worldwide 2015–2025 | Statista. Available online: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (accessed on 28 September 2022).
2. Top Sensors. 2021. Available online: https://www.arrow.com/en/research-and-events/articles/top-5-sensors-used-in-agriculture (accessed on 1 October 2021).
3. Livingston, M. Best Continuous Glucose Monitors for 2021. CNET. 2021. Available online: https://www.cnet.com/health/medical/best-continuous-glucose-monitors-for-2021 (accessed on 7 September 2021).
4. Mayoclinic.org. Implantable Cardioverter-Defibrillators (ICDs)–Mayo Clinic. 2021. Available online: https://www.mayoclinic.org/tests-procedures/implantable-cardioverter-defibrillators/about/pac-20384692 (accessed on 19 November 2021).
5. U.S. Food and Drug Administration. FDA Authorizes Marketing of Novel Device to Help Protect Athletes' Brains during Head Impacts. 2021. Available online: https://www.fda.gov/news-events/press-announcements/fda-authorizes-marketing-novel-device-help-protect-athletes-brains-during-head-impacts (accessed on 28 September 2021).
6. Rehman, A.U.; Naqvi, R.A.; Rehman, A.; Paul, A.; Sadiq, M.T.; Hussain, D. A Trustworthy SIoT Aware Mechanism as an Enabler for Citizen Services in Smart Cities. *Electronics* **2020**, *9*, 918. [CrossRef]
7. Gavidia, M. Apple Watch Effective in Monitoring Symptoms of Parkinson Disease. AJMC. 2021. Available online: https://www.ajmc.com/view/apple-watch-effective-in-monitoring-symptoms-of-parkinson-disease (accessed on 4 October 2021).
8. Gehealthcare.co.uk. Improving Oncology Outcomes with Connected Care Technology. 2021. Available online: https://www.gehealthcare.co.uk/article/improving-oncology-outcomes-with-connected-care-technology (accessed on 12 September 2021).
9. Abdel-Basset, M.; Hawash, H.; Sallam, K. Federated Threat-Hunting Approach for Microservice-Based Industrial Cyber-Physical System. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1905–1917. [CrossRef]
10. Tashtoush, Y.M.; Darweesh, D.A.; Husari, G.; Darwish, O.A.; Darwish, Y.; Issa, L.B.; Ashqar, H.I. Agile Approaches for Cybersecurity Systems, IoT and Intelligent Transportation. *IEEE Access* **2022**, *10*, 1360–1375. [CrossRef]
11. Mills, R.; Marnerides, A.K.; Broadbent, M.; Race, N. Practical Intrusion Detection of Emerging Threats. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 582–600. [CrossRef]
12. Karie, N.; Sahri, N.; Yang, W.; Valli, C.; Kebande, V. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. *IEEE Access* **2021**, *9*, 121975–121995. [CrossRef]
13. Iqbal, W.; Abbas, H.; Daneshmand, M.; Rauf, B.; Bangash, Y.A. An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security. *IEEE Internet Things J.* **2020**, *7*, 10250–10276. [CrossRef]
14. Rathore, S.; Park, J.H.; Chang, H. Deep Learning and Blockchain-Empowered Security Framework for Intelligent 5G-Enabled IoT. *IEEE Access* **2021**, *9*, 90075–90083. [CrossRef]

15. Badii, C.; Bellini, P.; Difino, A.; Nesi, P. Smart City IoT Platform Respecting GDPR Privacy and Security Aspects. *IEEE Access* **2020**, *8*, 23601–23623. [CrossRef]
16. Akil, M.; Islami, L.; Fischer-Hübner, S.; Martucci, L.A.; Zuccato, A. Privacy-Preserving Identifiers for IoT: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 168470–168485. [CrossRef]
17. Li, H.; Yu, J.; Zhang, H.; Yang, M.; Wang, H. Privacy-Preserving and Distributed Algorithms for Modular Exponentiation in IoT with Edge Computing Assistance. *IEEE Internet Things J.* **2020**, *7*, 8769–8779. [CrossRef]
18. Yu, Y.; Guo, L.; Liu, S.; Zheng, J.; Wang, H. Privacy Protection Scheme Based on CP-ABE in Crowdsourcing-IoT for Smart Ocean. *IEEE Internet Things J.* **2020**, *7*, 10061–10071. [CrossRef]
19. Sollins, K.R. IoT Big Data Security and Privacy versus Innovation. *IEEE Internet Things J.* **2019**, *6*, 1628–1635. [CrossRef]
20. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
21. Kong, F.; Zhou, Y.; Xia, B.; Pan, L.; Zhu, L. A Security Reputation Model for IoT Health Data Using S-AlexNet and Dynamic Game Theory in Cloud Computing Environment. *IEEE Access* **2019**, *7*, 161822–161830. [CrossRef]
22. Zhou, W.; Jia, Y.; Peng, A.; Zhang, Y.; Liu, P. The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved. *IEEE Internet Things J.* **2019**, *6*, 1606–1616. [CrossRef]
23. Xiong, J.; Ren, J.; Chen, L.; Yao, Z.; Lin, M.; Wu, D.; Niu, B. Enhancing Privacy and Availability for Data Clustering in Intelligent Electrical Service of IoT. *IEEE Internet Things J.* **2019**, *6*, 1530–1540. [CrossRef]
24. Metricfire.com. Hosted Graphite | MetricFire. 2021. Available online: https://www.metricfire.com/graphite-as-a-service/ (accessed on 8 November 2021).
25. Bharany, S.; Sharma, S.; Frnda, J.; Shuaib, M.; Khalid, M.I.; Hussain, S.; Iqbal, J.; Ullah, S.S. Wildfire Monitoring Based on Energy Efficient Clustering Approach for FANETS. *Drones* **2022**, *6*, 193. [CrossRef]
26. Zhang, C.; Zhang, S.; James, J.; Yu, S. FASTGNN: A topological information protected federated learning approach for traffic speed forecasting. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8464–8474. [CrossRef]
27. Hao, M.; Li, H.; Luo, X.; Xu, G.; Yang, H.; Liu, S. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Trans. Ind. Inform.* **2019**, *16*, 6532–6542. [CrossRef]
28. Bharany, S.; Badotra, S.; Sharma, S.; Rani, S.; Alazab, M.; Jhaveri, R.H.; Reddy Gadekallu, T. Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. In *Sustainable Energy Technologies and Assessments*; Elsevier BV: Amsterdam, The Netherlands, 2022; Volume 53, p. 102613. [CrossRef]
29. McMahan, H.B.; Ramage, D.; Talwar, K.; Zhang, L. Learning differentially private recurrent language models. *arXiv* **2017**, arXiv:1710.06963.
30. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; pp. 1–11.
31. Hu, R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized federated learning with differential privacy. *IEEE Internet Things J.* **2020**, *7*, 9530–9539. [CrossRef]
32. Bharany, S.; Sharma, S.; Bhatia, S.; Rahmani, M.K.I.; Shuaib, M.; Lashari, S.A. Energy Efficient Clustering Protocol for FANETS Using Moth Flame Optimization. *Sustainability* **2022**, *14*, 6159. [CrossRef]
33. Triastcyn, A.; Faltings, B. Federated learning with bayesian differential privacy. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2587–2596.
34. Bharany, S.; Kaur, K.; Badotra, S.; Rani, S.; Kavita; Wozniak, M.; Shafi, J.; Ijaz, M.F. Efficient Middleware for the Portability of PaaS Services Consuming Applications among Heterogeneous Clouds. *Sensors* **2022**, *22*, 5013. [CrossRef] [PubMed]
35. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
36. Bharany, S.; Sharma, S.; Khalaf, O.I.; Abdulsahib, G.M.; Al Humaimeedy, A.S.; Aldhyani, T.H.H.; Maashi, M.; Alkahtani, H. A Systematic Survey on Energy-Efficient Techniques in Sustainable Cloud Computing. *Sustainability* **2022**, *14*, 6256. [CrossRef]
37. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things J.* **2020**, *7*, 7751–7763. [CrossRef]
38. Zeng, T.; Guo, J.; Kim, K.J.; Parsons, K.; Orlik, P.; Di Cairano, S.; Saad, W. Multi-task federated learning for traffic prediction and its application to route planning. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–15 July 2021; pp. 451–457.
39. Chen, J.; Pan, X.; Monga, R.; Bengio, S.; Jozefowicz, R. Revisiting distributed synchronous SGD. *arXiv* **2016**, arXiv:1604.00981.
40. Dewri, R. Local differential perturbations: Location privacy under approximate knowledge attackers. *IEEE Trans. Mob. Comput.* **2012**, *12*, 2360–2372. [CrossRef]