Software Defined Internet of Things Security: Properties, State of the Art, and Future Research

Pritish Mishra, Ananya Biswal, Sahil Garg, Rongxing Lu, Mayank Tiwary, and Deepak Puthal

Abstract

The Internet of Things (IoT) is an emerging technology whose popularity and use cases grow with every passing day. Although the domain of IoT has proven to be transformative, as the deployment size of the network increases, it becomes a huge challenge to secure such a large number of heterogeneous devices connected by a complex network having a variety of access protocols. Software defined networking (SDN), with its unique capabilities of decoupling control plane from data plane and maintaining a centralized programmable controller, has been gaining a lot of prominence in solving the security challenges faced by the IoT domain. This work first analyzes the security threats faced by the IoT domain and the inherent architectural flaws for these vulnerabilities. It then goes on to understand how solutions presented by the modified architecture of SDN-IoT can help to tackle these challenges. Lastly, the work draws a realistic image of bringing in SDN-IoT by showcasing the critical issues faced by this domain and the current efforts attempting to solve these issues. Thus, it serves as a good primer for researchers looking to delve into the topic for tackling prevalent challenges.

Introduction

From smartphones to smart cities, the Internet of Things (IoT) infrastructure is slowly becoming an inseparable part of modern life, with its widespread applications only expected to multiply in coming years. However, the heterogeneous nature of IoT devices and the associated resource limitations become a cumbersome task for network and security administrators to manage. Mere encryption or anti-virus software cannot solve the intricate challenges of the domain. Over a period of time, many articles have come forward attempting to tackle the security challenges put forward by this domain. However, in recent times, software-defined networking IoT (SDN-IoT) infrastructure has gained a great reputation and much interest in the community for securing the traditional domain from the vulnerabilities in a clean and pervasive manner [1].

SDN-IoT provides orchestration of objects over the Internet for network management by decoupling the control plane and the data plane. It has been the result of extensive research efforts over the last decade aiming toward a more programmable, secure, reliable, and manageable infrastructure. SDN brings in its core concepts of segregation of the network control plane from the data plane and a logical centralized controller managing the operations of the entire network using standardized protocols. These features, armed with the developments in security frameworks, have made SDN-IoT a silver bullet for tackling the prevalent security challenges of the IoT network to make it efficient, secure, and reliable.

However, SDN-IoT architecture, in itself, is tedious to design to fit all the scenarios catered to by a traditional IoT infrastructure. The heterogeneity and interoperability of an IoT network make it very difficult for SDN-based technology to be practical, conform to all the previous features, and still provide the utmost security [2]. Still, comprehensive research in this field indicates that SDN-IoT is still our best hope to design and maintain the most secure IoT infrastructure.

There have been many works that have focused on documenting the challenges faced by the IoT network over the past years [3]. Similarly, there have also been a few works highlighting the solutions brought in by the SDN network [4]. This work aims to understand the challenges in IoT infrastructure by analyzing the gaps in the underlying architecture and how exactly does bringing in the SDN-IoT architecture help to overcome these challenges. The work further goes on to showcase the critical challenges faced by the SDN-IoT framework itself and what possible solutions have been attempted so far. Thus, the work showcases a comprehensive summary of the field, its current state of research, and the future scope.

Security Challenges in IoT Architecture

The domain of IoT is gaining a lot of popularity and adoption in the current generation. Hence, it makes it especially critical that the domain is secure against hostile attacks and threats. This section introduces a basic architecture of an IoT framework as showcased in Fig. 1. As a parallel, Fig. 2 classifies the prevalent threats in the IoT domain on the basis of the layer it affects. Furthermore, we try to understand, from the inherent properties of each layer, why a certain layer suffers from a particular type of threat.

Perception Layer

As is probably evident from the name, the perception layer is responsible for perceiving the physical nature of the things around the IoT deployment

Digital Object Identifier:
10.1109/MWC.001.1900318

Pritish Mishra and Mayank Tiwary are with SAP Labs; Ananya Biswal is with International Institute of Information Technology; Sahil Garg is with École de Technologie Supérieure; Rongxing Lu is with University of New Brunswick; Deepak Puthal is with Newcastle University.

1536-1284/20/$25.00 © 2020 IEEE

and gathers more information about it. This process involves devices including various types of sensors and actuators, RFIDs, smart wearables, and so on. Since this is the data collection plane of the deployment, this layer is always a source of attraction for attacks aiming to get access to sensitive information or disrupt the service of the network. This section delves deeper into the causes of these attacks and the mechanisms of their execution.

Sensor Attack: As per the traditional IoT architecture, each sensor has its own control logic and maintains its own flow rules. This is desirable, since the sensors are expected to operate even when there is no monitoring (due to hostile environments). However, attackers can utilize this feature to capture and manipulate the sensors escaping the notice of the network. Then the hostile sensors can leak critical information, provide erroneous data, or affect other devices by acting as trusted nodes.

Node Capture Attack: Leveraging the remote execution mechanism of an IoT node, which often requires no intervention or monitoring, the physical node can be tampered with by performing a forceful physical or electronic operation on the device. Then the physical access to the device leads to complete node capture providing full control to the attacker, which sabotages the device. If the attack goes unnoticed, it can lead to the rupture of the entire deployment.

Spoof-Node Attack: Since each node in an IoT network has its own flow rules, a malicious attacker could create a hostile node by replicating a certain identification feature of an actual node. The fake node can then be authorized by the network, thus leading to severe security vulnerabilities like allowing access privileges, extracting security keys, providing wrong feedback information, or revoking authorized nodes.

Hardware Manipulation Attack: Various types of malicious software like Trojans, worms, spywares, and viruses cause modification in the behavior of hardware for the benefit of the attacker. This could include giving access to sensitive sensor data to the attacker. Such an attack is possible primarily due to the tight coupling of the IoT hardware with the software components. The fact that each IoT device has its own software also makes it difficult to track the malicious code injections that could occur during fabrication or design to insert a trigger for activating malware.

Energy Manipulation Attack: Most wireless sensors and actuators rely on embedded batteries having a fixed energy capacity for the power source. This limitation can be exploited by a hostile attacker to bring down IoT hardware by draining off its power source. Nodes in an IoT network typically utilize enhanced duty-cycle procedures to extend the life cycle. In a *sleep deprivation* attack, an attacker can manipulate the normal sleep routines and force the affected node to be awake until the energy is exhausted. A huge number of trivial packets can also be bombarded to force resource consumption in devices while processing these packets.

Network Layer

The network layer accumulates and processes the received information from the perception layer and transmits the data to the application layer for analysis and consumption. Since the controllers are also responsible for communication between the application and perception layers, crippling attacks on this layer can render the breakdown of the entire system. This section attempts to understand the mechanisms of some of these threats and vulnerabilities.

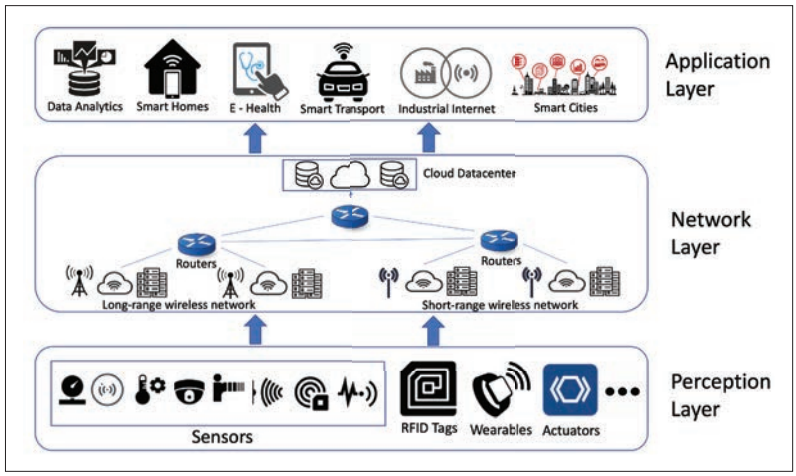

FIGURE 1. Classification of security threats in the IoT architecture.

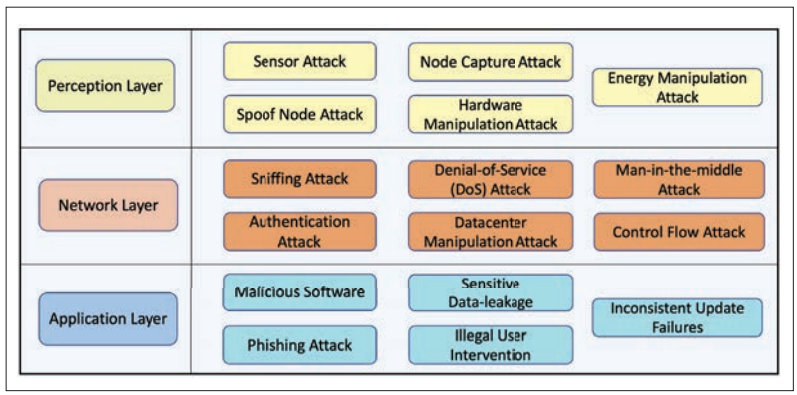

FIGURE 2. Architecture of the IoT framework.

Sniffing Attack: Data sniffing, also known as eavesdropping, is performed by sneaking pieces of information from private communication between the devices and the controller in an IoT network. This could occur if there is a lapse in implementing an encryption mechanism for any form of communication between two components. Since there are numerous points of communication, each needs to be encrypted to avoid this form of attack.

Authentication Attack: Most of the authentication protocols employed by current IoT frameworks employ mutual authentication, in which multiple IoT devices authenticate each other to maintain privacy and integrity. However, attackers can compromise one of these authenticating devices and can launch an authentication attack leading to hostile access to the network and its resources. Hostile authentication access leads to other security vulnerabilities like man-in-the-middle attack and impersonation attack.

Denial of Service Attack: One of the most prevalent security threats, denial of service (DoS) attacks are launched by compromising communication links and flooding the network with massive data, thus causing exhaustion of resources

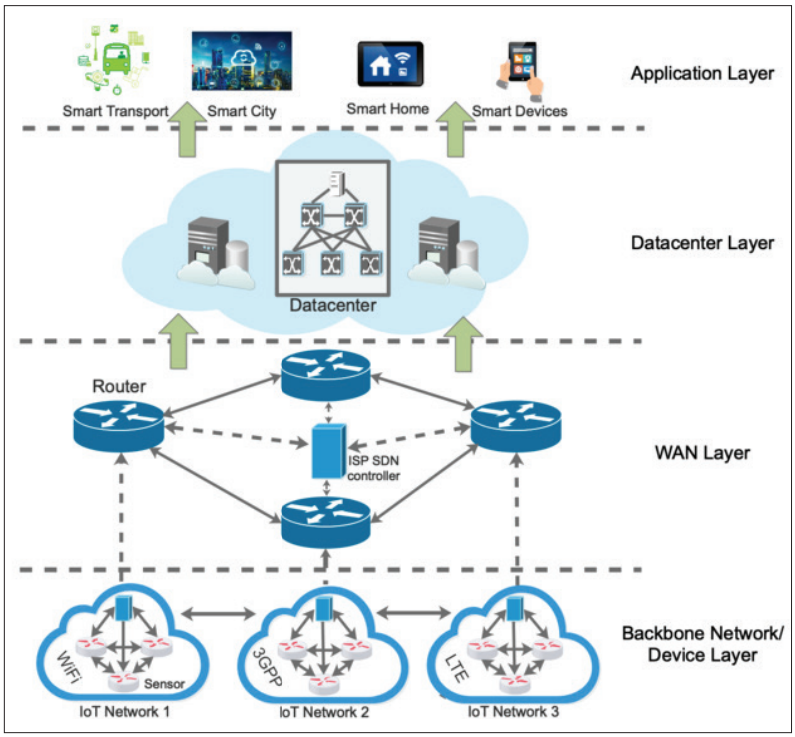

FIGURE 3. Architecture of the SDN-IoT framework.

and unavailability of the network. Since most of the devices use wireless communication links, jamming attacks can cause disconnectivity and serious performance degradation. These vulnerabilities can be further exploited to cause Distributed DoS attacks by creating large networks of bots.

Datacenter Manipulation Attack: Nowadays, IoT networks employ various cloud-provider solutions, in which a cloud data-center could be manipulated by an attacker to gain control of the deployed IoT instances. These manipulated instances could cause leakage in data, failures in data-conciliation, intentional lapses in data-collection, data integrity issues etc.

Man-in-the-Middle Attack: Exploiting the authentication lapses in a network, Man-in-the-middle attack is an enhanced spoofing attack in which both the devices in a communication are impersonated by the attacker to intercept the traffic and send fake messages to each target device. Both the parties are tricked to believe that the fake node is the trusted entity thus leaking sensitive information like security keys, private customer data, network control flow, etc. to the attacker.

Control Flow Attack: Since each device and corresponding controller maintains its own control flow logic in a traditional IoT architecture, it becomes easier for an attacker to take control of a controller and compromise the segment of the IoT network. This leads to intentional dropping of packets and bombardment of fake packets to the network, thus bringing down the performance of the network.

Application Layer

The Application Layer is the most popular subsystem, since the consumers of the IoT deployment interact with this layer. All popular applications like Smart homes, Smart transport, E-health, etc. are deployed in this layer. The analytics are performed on the received data and services are designed based on this. Although, the form of vulnerabilities for this layer are very different in nature from the rest of the domain, they are critical because of the nature and sensitivity of the data. The attacks might not bring down the system per-se, but the leaked data can have disastrous consequences.

Malicious Software: IoT applications, in the absence of updated security patches, are susceptible to malware, viruses, and worms. Malicious software can trigger data leakage and data integrity issues. Certain worms have propagation abilities that can further compromise the security of the subsequent components of the network.

Phishing Attack: Lucrative emails and false advertising websites contain malware that an attacker can use to launch a phishing attack. If an IoT application accesses such a hostile medium for its operation, the attacker can use phishing to gain control of sensitive information like users' credentials and access permissions to other IoT devices.

Sensitive Data Leakage: In the age of data privacy, where data is the most powerful currency, IoT applications need to be made especially secure from data leakage. The operational context of the applications can be exploited by attackers to not only attack the specific application but also to infer the data it is holding within. Even such a minor lapse can cause catastrophic consequences for the users of the application.

Illegal User Intervention: IoT applications are often meant to be used by several users, while the information used by the applications are often served by a single network deployment. If an attacker can pose as a user of the application by faking credentials, she can understand the context of the application and can attempt to reverse-engineer the process to gain access to the data of genuine users. The attacker can also fake interactions with the applications in order to launch a DoS attack on the network.

Incompatible Update Failures: Since each device of the IoT network maintains its own version of software, it becomes cumbersome to apply updates of software to each node. This becomes especially critical if an urgent security patch needs to applied to each software version to prevent a security vulnerability. In cases of such inconsistent updates, failures can occur due to incompatible versions of software or exposed security vulnerabilities that can be leveraged by an attacker.

SDN-IoT: Tackling Threats in IoT Framework

SDN-IoT architecture is widely being advocated as a silver bullet for solving the security challenges of the IoT framework in a graceful and pervasive manner. The question that arises from this claim is what features of the SDN-IoT architecture merit such graceful handling of the threats. The key differences in the architecture of the two technologies can easily be visualized from the illustrative implementation of an SDN-IoT framework as presented in Fig. 3. Furthermore, a taxonomy of solutions offered by the SDN-IoT architecture for combating the security threats faced by IoT, is presented as part of Fig. 4. These illustrations

serve as a primer for the detailed discussion that follows, focusing on the solutions presented by the various layers of SDN-IoT.

Device Layer

The device layer is the lowest layer of the SDN-IoT architecture and is similar in nature to the perception layer of the IoT architecture, with the key difference being that there are no individual flows or monitoring mechanism of the device. Rather, all the devices are monitored centrally and the flow rules are distributed across these devices from the centralized controllers. There are multiple device networks where each device network has multiple integration points like routers, gateways, and a controller centralized for that particular device network. These controllers further transmit information to the routers present in the WAN layer. This section further analyzes the security benefits of modifying the traditional IoT architecture with these changes.

Decoupling Hardware from Software: One of the key features of the SDN-IoT infrastructure is to remove the dependence of hardware devices on the specific software. The SDN deployment maintains one version of the software, which is applied to all virtual instances of the controller. These virtual instances are responsible for managing the traffic flow to the individual hardware devices (e.g., sensors). Thus, the hardware is protected from manipulation attacks, and regular updates to the software at the centralized location ensure minimal security threats.

Virtual Intrusion Detection System: The virtual intrusion detection system (vIDS) uses predefined signatures or attack patterns and event logs to determine abnormal traffic in a network, which is typically caused by energy-manipulation attack [5]. vIDS can be used to detect hostile activities including consumption of excessive bandwidth, flooding of trivial packets, DoS attacks, and so on. vIDS has a better vantage point, being implemented on the centralized network controller, to detect such attacks.

WAN Layer

The WAN layer is widely considered as the Internet layer of the architecture. Devices like routers and gateways form the data plane, while the centralized Internet service provider (ISP) SDN controller manages these devices and forms the control plane of the layer. While the routers and gateways are responsible for data forwarding and local data caching, the controller manages the data processing and governing of the network by determining the flow rules. The centralized position of the controller offers numerous advantages and helps prevent many security vulnerabilities.

Centralized Secure Device Monitoring: Due to its centralized positioning in SDN-IoT architecture, the controller performs the most effective supervision of the data plane. It also collects the status information of the entire network by performing periodic stats queries and health checks. Thus, the controller always maintains the updated status of the underlying device network and can modify the flow traffic as needed. This becomes especially helpful not only to detect capture of specific nodes or fake nodes, but also to facilitate the mitigation strategies in case of such attacks.

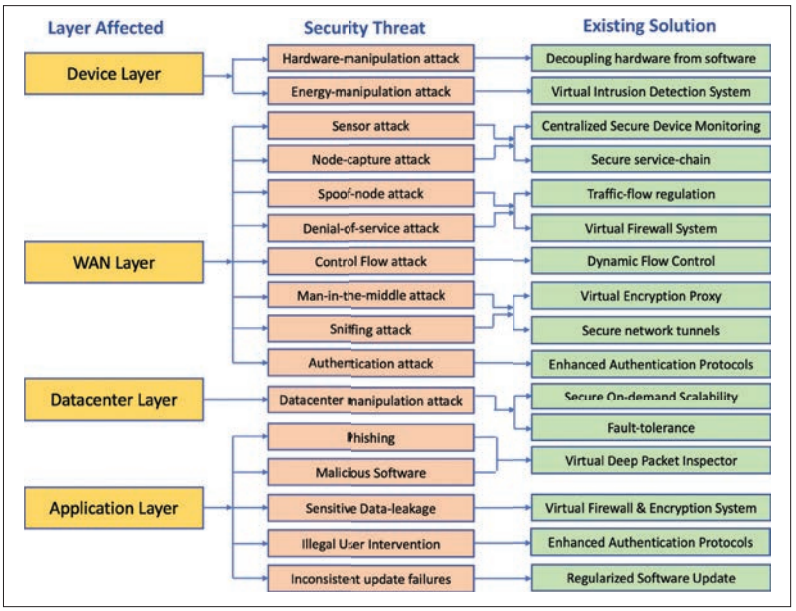

FIGURE 4. Taxonomy of solutions offered by SDN-IoT architecture for tackling security threats.

The modified flow rules, as the response to an attack, can easily be applied to the target nodes.

Secure Service Chain: Network service chaining, otherwise known as service function chaining, creates a secure chain of interconnected network services in a virtual chain. These network services, including firewalls, intrusion protection, NAT, and so on, provide essential security for the devices and enables security administrators to set up suites of security services into a single network connection for all the devices and services. This enables virtual network connections to handle the traffic flows for all connected services in an automated fashion.

Traffic Flow Regulation: As the denial-of-Service attacks aim to cripple the system by bombarding the controller with redundant packets, rate-limiting of control channel can allow the controller to resist system crashes due to exorbitant number of requests. Regulation of traffic can also include dropping packets based on priority, and QoS control of the packets received. Mobility-Aware and QoS-Driven solutions have also been proposed to handle traffic-flow challenges [6].

Virtual Firewall System: Virtual Firewall System in a SDN-IoT framework is a firewall service that provides network traffic filtering for the virtualised instances representing the devices of the network. It inspects the flow packets and uses pre-defined security policies to block out unwanted communication resulting from attacks like DoS attack. Virtual firewalls are especially desirable since they are very flexible and can be modified as per the changes in the policies of virtual networks.

Dynamic Flow Control: An integral feature of the SDN framework is the ability of the controller to dynamically modify the flow rules and propagate the updates seamlessly to the devices across the network. Leveraging this manageability, defense solutions can be designed to include features like network reflectors and dynamic quarantine to combat hostile packet flow in the network. This feature also helps to establish a network with specific privileges and uniform network policies.

The data center layer can be considered the persistence layer for application services. With the advent and popularity of cloud-based solutions, the SDN-IoT architecture maintains the storage of the data collected and processed by the device and WAN layers in the cloud-provisioned data centers for further analysis by the applications of the application layer.

Virtual Encryption Proxy: A virtual encryption proxy utilizes symmetric and asymmetric cryptosystems to protect the network from sniffing and eavesdropping attacks. The proxy acts as an intermediary between two communicating devices and helps to prevent attacks like man-in-the-middle, since the proxy becomes the sole entity responsible for determining the destination for both nodes.

Secure Network Tunnels: Secure network tunnels allow communication of data packets in an encrypted format to prevent the sniffing attacks. The network traffic can be rerouted through a vProxy using secure data tunnels to facilitate secure data transport and prevent man-in-the-middle attacks. Secure network tunnels can also be used to redirect the traffic via other security mechanisms (vIDS, vFirewall, etc.) to prevent other forms of attacks.

Enhanced Authentication Protocols: Enhanced authentication protocols, like two-way authentication, role-based authorization for each device, and multi-layered authentication, can be implemented in a centralized layer of the controller and be further distributed across the devices. The virtualized authentication, authorization, and accounting (vAAA) framework [7] can be utilized to design intelligent security protocols for the virtualized instances of IoT gateways.

Data Center Layer

The data center layer can be considered the persistence layer for application services. With the advent and popularity of cloud-based solutions, the SDN-IoT architecture maintains the storage of the data collected and processed by the device and WAN layers in the cloud-provisioned data centers for further analysis by the applications of the application layer [8].

Secure On-Demand Scalability: Leveraging on the concept of dynamic life cycle management of virtualized network resources in the SDN-IoT framework, the network administrators can design highly scalable secure solutions facilitating efficient resource optimization. Virtualized network functions (VNFs) can be implemented at the cloud data centers, which can be auto-scaled as per the workload of the incoming requests. The scalability feature of the security solutions reduces the probability of degraded performance during network analysis and ensures optimized threat detection, irrespective of the load on the network.

Fault Tolerance: Cloud deployments are susceptible to breakdowns due to either inherent failures or hostile attacks. Fault tolerance distributes the load across other nodes and ensures the availability of the service at all times, and thus enables the SDN-IoT network to be secured against unwanted failures and ingenious attacks. State-aware duplicates of the virtualized instances of controllers and multiple redundant nodes of the cloud servers ensure that, in case of a failure, a backup node is promoted as a master and the service operates as usual.

Application Layer

The most intuitive part of the architecture, the application layer deals with all the applications consumed by the customers of the network. This layer is not too different from the one proposed in the IoT architecture, but nevertheless, the security protocols need to be introduced and enhanced to handle hostile attacks on the information held by this layer.

Virtual Deep Packet Inspector: The deep packet inspector is one of the strongest players in the application security niche and has very strong potential to combat modern web and application attacks. It involves network packet filtering, which inspects the data part of a packet and detects any spam, viruses, and other intrusions. As per the system rule definitions, the packet can be either forwarded, blocked, or tagged for various characteristics. A lightweight high-performance deep packet inspector enabling anomaly detection features can be implemented for this seventh layer of Open Systems Interconnection (OSI) stack.

Firewall and Encryption System: An application firewall controls input, output, and access from or to an application or service. It not only prevents attacks from a hostile application to the parent, but also protects the parent application from accessing applications that might harm the integrity of the system. Additionally, the encryption system ensures that the data, if leaked while in storage or during communication, cannot be deciphered easily.

Enhanced Authentication Protocols: Application users must be authenticated using enhanced authentication protocols including biometric authentication, filtering medium access control (MAC) addresses, timer-based encrypted passwords, and so on. Role-based authorization ensures that not every entity can access the data, and no one single entity can have access to all sections of the data. These authentication and authorization schemes need to be updated regularly to anticipate and prevent modern threats.

Regular Software Update: A uniform version of software maintained at a centralized location ensures that the updates are consistently and regularly applied to a software and this version of software is distributed across the network. This also avoids the incompatibility of versions of softwares leading to update issues.

Open Challenges and Future Scope in the SDN-IoT Framework

SDN-IoT holds promising aspects for solving the security issues of the IoT framework and making it more robust. However, there are still many open challenges in the practical realization of SDN-IoT to replace the existing framework. There have been works attempting to tackle these issues, and yet many such issues still exist. Table 1 represents some of these critical issues faced by the SDN-IoT architecture and what solutions have been proposed until now in the literature. This section delves into these topics of interest that can be taken up by researchers to ease the realistic enablement of the technology.

Single Point of Failure: A centralized controller, which is the strongest feature in favor of an SDN architecture, sometimes becomes a major culprit susceptible to various forms of attacks. Since most of the control logic flows from this component, even though the attack vector

reduces, still the controller effectively becomes the source of a single point of failure. The use of standard prevention strategies, like an intrusion detection system, may not be sufficient since it becomes difficult to determine the exact chain of events triggering malicious behavior and detect it. Multiple controllers, like TinySDN, have been explored to tackle this challenge [9]. Several other techniques, such as replication strategies, diversity mechanisms of controllers and protocols, security policies restricting the applications having access to a controller, and periodic recovery of a controller to a clean slate, have been attempted to tackle this challenge. Since this is perhaps the most pertinent problem of the architecture, in spite of the prevalence of certain solutions, a thorough reliable solution is still lacking.

Data Confidentiality Issues: Lack of efficient encryption mechanisms between the switches and the SDN controller can give rise to a precarious violation of confidentiality of communication. Rigid authentication mechanisms and systematic trust models can be used to combat such identity attacks [10]. Sandboxing techniques allowing minimal operations and foolproof unhackable devices to store sensitive information are some of the approaches that need to be investigated in the design of practical prototypes. Autonomous trust management mechanisms must be designed and implemented to establish the authenticity of the communicating applications.

Fast Secure Mode Recovery: A fast and reliable recovery with minimal data loss requires a safe and reliable snapshot of the system and a point-in-time recovery mechanism to restore the network to its previous working state. Mechanisms guaranteeing such a procedure are currently missing from the domain. On the other hand, in order to investigate and establish information regarding an attack or a failure, reliable information needs to be retrieved from all components of the system. This data is useful only if its trustworthiness, integrity, and authenticity can be established. Immutable logging mechanisms, which guarantee that the log is indelible and cannot be tampered with, are challenging.

Orchestration Issues: The IoT architecture is an amalgamation of a variety of heterogeneous technologies, with each technology having its own set of complications and security vulnerabilities. Designing a secure SDN-IoT framework is incredibly challenging since it has to enforce a protection mechanism across various administrative and technological domains of the IoT network by accommodating the nuances of each subsystem. The heterogeneity of devices in IoT brings about a unique attribute of Interoperability, which cannot be compromised at the cost of designing secure architecture. An OpenFlow security framework, FRESCO, has been proposed to incorporate various security detection and remediation strategies [11]. Another prominent solution, OrchSec [12], aims at leveraging network monitoring applications and control functions of SDN to develop secure strategies. However, the field is still rife with challenges that need to be overcome.

Denial of Service Attacks: In spite of the existing security mechanisms, there have been investigations that confirm that DoS attacks still pose a major threat for the domain [2]. Rate limiting of the control channel, event filtering by a controller enabling selective handling of events, traffic prioritization, and timeout adjustment are approaches proposed to combat these attacks, some of which have even been standardized by the OpenFlow protocol. Proposed works focusing on localized central control, dynamic flow insertion/deletion, and automated traffic analysis to combat the attacks have gained confidence [13]. However, the issue still remains a very challenging one to tackle.

Critical issue	Layers affected	Existing solutions
Single point of failure	✓ Device layer ✓ WAN layer	• Multiple controllers • Replication strategies • Clean-slate recovery • Controller access restriction
Data confidentiality issues	✓ WAN layer ✓ Data center layer ✓ Application layer	• Rigid authentication mechanism • Systematic trust model • Autonomous trust management • Sandboxing techniques
Troubleshooting and speed recovery	✓ WAN layer ✓ Data center layer	• Reliable system snapshots • Immutable logs
Orchestration issues	✓ WAN layer ✓ Device layer	• FRESCO • OrchSec
Denial of service attacks	✓ WAN layer ✓ Device layer	• Rate-limiting of control channel • Event filtering • Traffic prioritization • Timeout adjustment • Localized central control
Man-in-the-middle attacks	✓ WAN layer ✓ Device layer	• Bloomfilter • Dynamic device association • Increase in data-plane programmability
Policy definition issues	✓ WAN layer	• HiPoLDS • HLP/MLP language protocols

TABLE 1. Open security challenges in SDN-IoT architecture.

Man-in-the-Middle Attacks: The security of communications is, at times, compromised by the certification/authentication mechanism used. Self-signed certificate, TLS/SSL-based communication, and public key infrastructure (PKI) have been determined to give rise to vulnerability to attacks like man-in-the-middle. Countermeasures like Bloomfilter, a lightweight protocol that monitors the system for such attacks, have been proposed [14]. Dynamic device association, a mechanism in which a switch can dynamically associate itself with several controllers based on requirements, has also been recommended to automatically handle faults arising due to these attacks. Increasing the data plane programmability using general-purpose CPUs or proxies for switches is also a possible remedy. The issue has been attracting a lot of attention lately.

Policy Definition Issues: Due to the criticality of unifying physical and data planes, definition of security policies has become an urgent issue. For adopting broad usability, contextual policy definitions need to be incorporated. The ideal goal is to have a unification of high-level security requirements, while simultaneously ensuring low-level configurations of applied security measures. For achieving this goal, solutions like Hier-

The open research areas have been showcased for guiding future works in this area from the reseachers of the community. Although the SDN-IoT domain has now been validated in terms of the concept and its advantages are no longer in question, there is need of practical and incisive research into the domain to establish its dominance.

archical Policy Language for Distributed Systems (HiPoLDS) [15] have been focused on decentralizing service-based execution frameworks. Another approach defines formulating security requirements using High-Level Policies (HLP) language, which enforces a subject-object-attribute structure. The process is two-step as the output of HLP is further processed by Medium Level Policies (MLP) language scripts for policy refining. However, policies handling the heterogeneity of the IoT network along with its flexibility are tedious to design.

Conclusions

This article has provided a comprehensive overview of the state of security in IoT and SDN-IoT frameworks. This work justifies the need of bringing in the enhanced model of SDN-IoT to tackle the challenges encountered by the traditional IoT infrastructure and showcases the solutions brought forth with the new model. It also takes a curated view at the much-acclaimed SDN-IoT infrastructure to analyze the weaknesses still existing in the domain. The open research areas have been showcased for guiding future works in this area from the reseachers of the community. Although the SDN-IoT domain has now been validated in terms of the concept, and its advantages are no longer in question, there is a need for practical and incisive research in the domain to establish its dominance.

References

[1] K. Kalkan and S. Zeadally, "Securing Internet of Things (IoT) with Software Defined Networking (SDN)," *IEEE Commun. Mag.*, vol. 56, no. 9, Sept. 2017, pp. 186–92.

[2] R. Kloti, V. Kotronis, and P. Smith, "OpenFlow: A Security Analysis," *Proc. 8th Wksp. Secure Network Protocols*, 2013.

[3] M. B. M. Noor and W. H. Hassan, "Current Research on Internet of Things (IoT): A Survey," *Computer Networks*, vol. 148, 2018, pp. 283–94.

[4] R. Kanagavelu and K. M. M. Aung, "A Survey on SDN Based Security in Internet of Things ," *Advances in Info. and Commun. Networks*, vol. 887, 2018, pp. 563–77.

[5] H. Upadhyay and T. Sherasiya, "Intrusion Detection System for Internet of Things," *IJARIIE Int'l. J.*, vol. 2, no. 3, 2016, pp. 2344–49.

[6] S. Garg *et al.*, "MobQoS: Mobility-Aware and QoS-Driven SDN Framework for Autonomous Vehicles," *IEEE Wireless Commun.*, vol. 26, no. 4, Aug. 2019, pp. 12–20.

[7] C. Rensing *et al.*, "AAA: A Survey and a Policy-Based Architecture and Framework," *IEEE Network*, vol. 16, no. 6, Nov./Dec. 2002, pp. 22–27.

[8] K. Kaur *et al.*, "A Big Data-Enabled Consolidated Framework for Energy Efficient Software Defined Data Centers in IoT Setups," *IEEE Trans. Industrial Informatics*, 2019, pp. 1–1.

[9] D. Oliveira *et al.*, "TinySDN: Enabling Multiple Controllers for Software-Defined Wireless Sensor Networks," *IEEE Latin America Trans.*, vol. 13, no. 11, 2015, pp. 3690–96.

[10] D. Kreutz *et al.*, "Towards Secure and Dependable Software-Defined Networks," *Proc. 2nd ACM SIGCOMM Wksp. Hot Topics in Software-Defined Networking*, 2013, pp. 55–60.

[11] S. Shin *et al.*, "FRESCO: Modular Composable Security Services for Software-Defined Networks," *ISOC Network and Distributed System Security Symp.*, 2013.

[12] A. Zaalouk *et al.*, "Orchsec: An Orchestrator-Based Architecture for Enhancing Network-Security Using Network Monitoring and SDN Control Functions," *Network Operations and Management Symp.*, 2014, pp. 1–9.

[13] M. E. Ahmed and H. Kim, "DDOS Attack Mitigation in Internet of Things Using Software Defined Networking," *IEEE 3rd Int'l. Conf. Big Data Computing Service and Applications*, 2017.

[14] P. Berde *et al.*, "ONOS: Towards an Open, Distributed SDN OS," *Proc. 3rd Wksp. Hot Topics in Software Defined Networking*, 2014.

[15] M. Dell'Amico *et al.*, "Hipolds: A Hierarchical Security Policy Language for Distributed Systems," *Info. Security Tech. Report*, vol. 17, no. 3, 2013, pp. 81–92.

Biographies

Pritish Mishra (pritish.mishra@sap.com) is working as a core developer in SAP Cloud Platform at SAP Labs Bangalore, India. He graduated from the International Institute of Information Technology, Bhubaneswar, India. He has been an active participant in many leading academic and industrial conferences. He has served as a peer reviewer for many key IEEE conferences and publications. He has numerous publications in the domain of SDN, IoT, and serverless, edge, and cloud computing.

Ananya Biswal (apriyadarshini3@gmail.com) is working as a tech analyst for Infosys Ltd., Bangalore, India. She graduated from the International Institute of Information Technology. Her primary research interests and contributions are in the areas of software-defined networks and the Internet of Things.

Sahil Garg [S'15, M'18] (sahil.garg@ieee.org) is a postdoctoral research fellow at École de technologie supérieure, Université du Québec, Montréal, Canada. He received his Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing. He has over 50 publications in high ranked journals and conferences, including 25+ IEEE transactions/journal papers. He received the IEEE ICC best paper award in 2018 in Kansas City, Missouri. He serves as the Managing Editor of Springer's *Human-Centric Computing and Information Sciences* journal. He is also an Associate Editor of *IEEE Network*, Elsevier's *Future Generation Computer Systems*, and Wiley's *International Journal of Communication Systems*. In addition, he also serves as a Workshops and Symposia Officer of the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He has guest edited a number of Special Issues in top-cited journals including *IEEE T-ITS*, *IEEE TII*, the *IEEE IoT Journal*, *IEEE Network*, and *Future Generation Computer Systems*. He serves/served as the Workshop Chair/Publicity Co-Chair for several IEEE/ACM conferences including IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, ACM MobiCom, and more. He is a member of ACM.

Rongxing Lu [S'09, M'11, SM'15] (RLU1@unb.ca) received his Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2006, and his Ph.D. degree (awarded the Canada Governor General's Gold Medal) in electrical and computer engineering from the University of Waterloo, Ontario, Canada, in 2012. He was an assistant professor at Nanyang Technological University, Singapore. Since 2016, he has been an assistant professor in the Faculty of Computer Science, University of New Brunswick, Canada. His research interests include computer, network, and communication security, and applied cryptography.

Mayank Tiwary (mayank.tiwary@sap.com) is working as a core developer in the SAP Cloud Platform at SAP Labs Bangalore. He graduated from Biju Patnaik University of Technology, Odisha, India. He has numerous publications in the domain of SDN, and distributed and cloud computing.

Deepak Puthal (deepak.puthal@newcastle.ac.uk) is a lecturer in the School of Computing, Newcastle University, United Kingdom. His research interests include cyber security, the Internet of Things, distributed computing, and edge/fog computing. He has received several recognitions and best paper awards from IEEE. He is an Associate Editor of *IEEE Transactions on Big Data*, *Computers & Electrical Engineering* (Elsevier), the *International Journal of Communication Systems* (Wiley), *IEEE Consumer Electronics Magazine*, and *Internet Technology Letters* (Wiley).