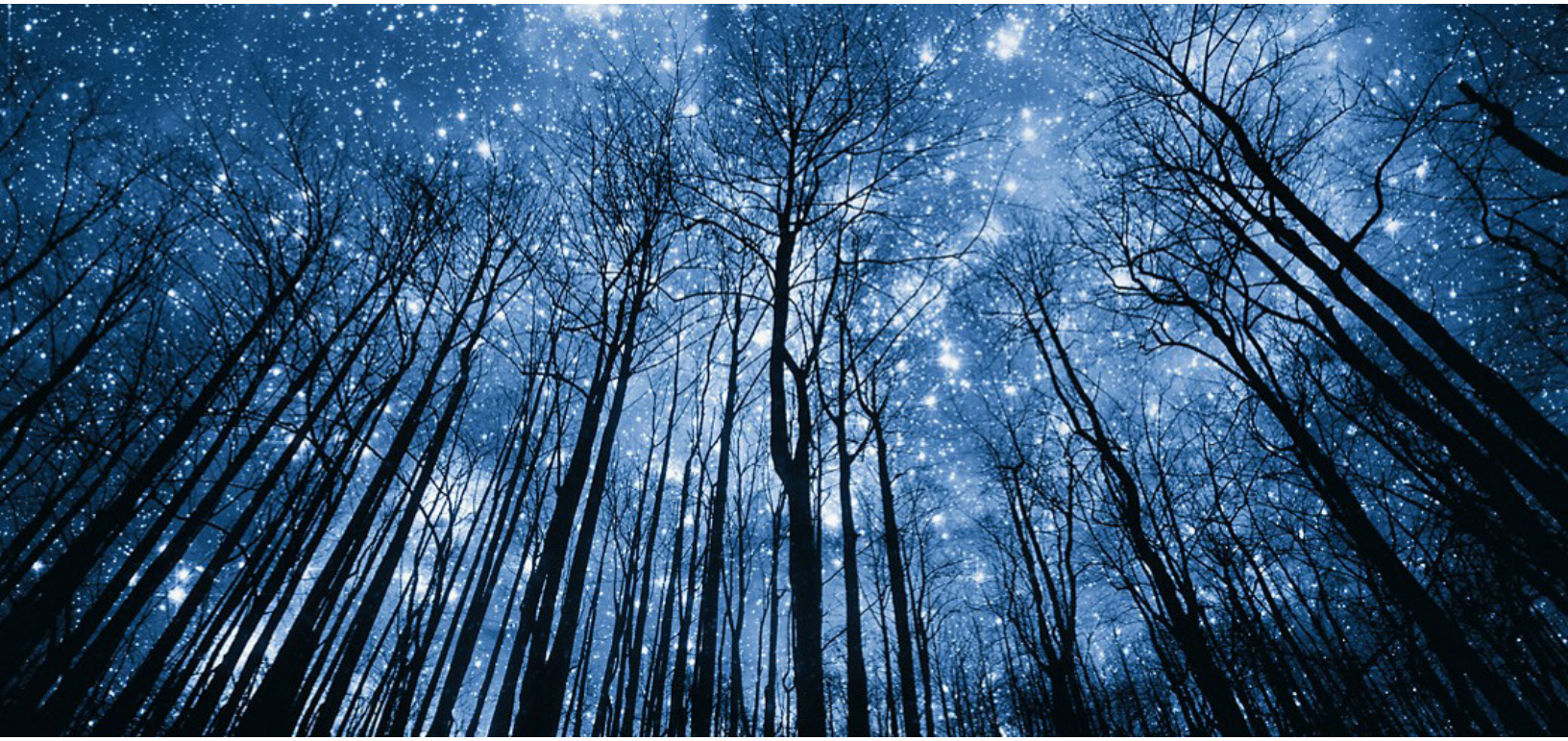
DELL Technologies

SDN FOR SECURING IOT SYSTEMS



Gaushil Patrick

Golla Venkata Harsha Vardhan

DELL Technologies
Proven Professional

Knowledge Sharing Article

The Dell Technologies Proven Professional Certification program validates a wide range of skills and competencies across multiple technologies and products.

From Associate, entry-level courses to Expert-level, experience-based exams, all professionals in or looking to begin a career in IT benefit from industry-leading training and certification paths from one of the world's most trusted technology partners.

Proven Professional certifications include:

- Cloud
- Converged/Hyperconverged Infrastructure
- Data Protection
- Data Science
- Networking
- Security
- Servers
- Storage
- Enterprise Architect

Courses are offered to meet different learning styles and schedules, including self-paced On Demand, remote-based Virtual Instructor-Led and in-person Classrooms.

Whether you are an experienced IT professional or just getting started, Dell Technologies Proven Professional certifications are designed to clearly signal proficiency to colleagues and employers.

Learn more at www.dell.com/certification

Table of Contents

Introduction

Evolution of technology has allowed various fields to converge and provide a well-rounded service by complementing each other. Internet of Things (IoT) is a system of correlated devices – mechanical or digital – able to transfer data across a network without human intervention. The IoT vision aims at seamlessly integrating the sensing and actuation features of common objects by leveraging their network capabilities to create pervasive information systems. The relevant data analysis systems can then derive appropriate control decision which can be enforced in the physical world. Cisco Systems forecasts that by 2020, over 50 billion connected "things" will be absorbed into the Internet, including cars, kitchen appliances, televisions, surveillance cameras, smartphones, utility meters, intrabody sensors, thermostats, and more. It's predicted that annual revenues could exceed 470 Billion dollars for IoT vendors selling the hardware, software and comprehensive solutions for the IoT.

While the benefits of IoT are undeniable, security of the systems against recent attacks is a major concern for IoT. If not appropriately considered, IoT security breaches can bring tremendous economical and reputation damages, thus undermining IoT's widespread adoption. IoT devices in homes and hospitals can carry sensitive information. Flaws in data integrity and confidentiality can cause critical information leakage. Misconfiguration of defense systems for a single node presents the weakest link of the chain, thus potentially compromising the interconnected devices and relevant service incomes. For example, researchers have found critical vulnerabilities in a wide range of IoT baby monitors, which could be leveraged by hackers to carry out nefarious activities, including authorizing other users to remotely view and control the monitor.

Traditional security mechanisms such as firewalls and intrusion detection systems aren't sufficient to protect the network from external attacks in IoT environments. This is due to the resource constraint nature and heterogeneity of IoT systems. Resources such as CPU, memory, and battery are limited in IoT devices as they tend to be wireless devices. Heterogeneity can range from smart microsensors to smart cars.

Due to its efficiency and flexibility, researchers strongly consider Software Defined Network (SDN) for securing the IoT system. SDN aims to increase network programmability by decoupling the data plane and control plane. It can account for dynamic routing of traffic as the controller takes all the routing decisions while reducing the network switches as forwarding elements. This can have a positive impact on performance, manageability and security as will be discussed in the following sections. SDN-based security mechanisms can cope with increasingly sophisticated security threats by accounting for the increasing blurring between physical and virtual IoT systems.

In this article, we discuss the architecture of the SDN and IoT systems, threats, the advantages introduced by SDN and challenges that must be addressed.

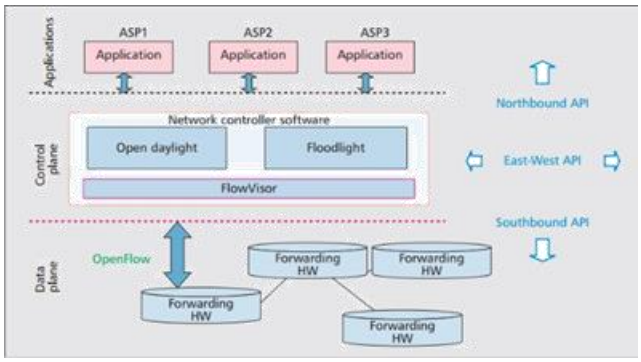
Background

Software Defined Network Architecture

Traditional networks are vertically integrated; that is the control plane and data plane has been bundled together. SDN breaks the vertical integration by decoupling the data plane and the control plane. It separates the network's control logic from the underlying routers and switches, promoting centralization of network control, and introduces the ability to program the network, improving policy enforcement and easing management. The SDN network architecture is shown in the Figure 1. Let's look at the figure from the bottom up to get a clear view of the architecture.

SDN consists of three layers – Control plane, Data plane and Management plane. The three layers are interconnected by the northbound APIs and the southbound APIs. Eastbound APIs and Westbound APIs are a special case of controllers required by the distributed controllers. The function of these interfaces includes exchange of data among the controllers and monitoring/notification capabilities.

Data plane: Data plane can be subdivided into two layers: 1) Infrastructure and 2) Southbound API. The main difference is that traditional routers are now physical devices in this plane. The forwarding elements or switches have no embedded control or software to take the forwarding/routing decisions. In essence, network intelligence is removed from the data plane



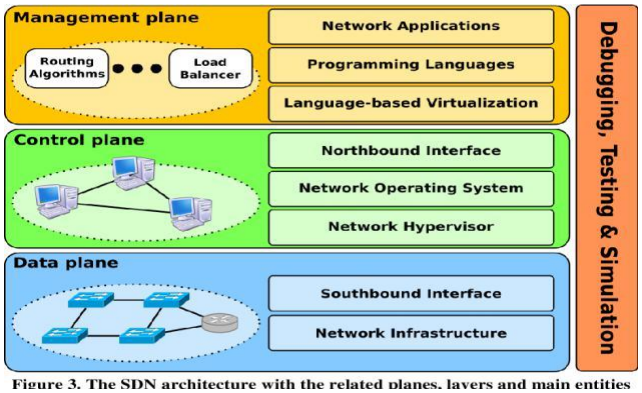


Figure 3. The SDN architecture with the related planes, layers and main entities

Infrastructure: This layer consists of switches which forward information/data packets according to the desires of the network controller. The path taken by the packet can be specified by the flow tables embedded in the switch which defines how a packet should be handled. When a new packet arrives, the lookup process starts at the first table and ends with a match in either the first table or subsequent tables in the pipeline. If no match occurs, it is called a miss and by default the packet is dropped. The default can be set such that in the event of a packet miss, the packet is sent to the control for further inspection. The controller can revert with an entry in the flow table to either forward it to an intermediate destination or to drop it.

Southbound Interface: This is the connecting bridge between the control and the forwarding elements thereby clearly separating control plane and data plane. As a central component of its design, it represents a major barrier to accepting new technology. Southbound interfaces consist of some standards which promote interoperability, allowing the deployment of vendor-agnostic network devices.

Control Plane: Control plane can be subdivided into three layers: Network Hypervisor, Network operating system and Northbound interface. This plane can be viewed as a single entity which can have a global view of the network. This layer can be hierarchical or fully distributed. In hierarchical, the controller operates on a partitioned network view while the distributed has a local view and they exchange synchronization messages to communicate with other distributed controllers.

> **Network Hypervisor:** Hypervisors enable distinct Virtual machines to share the same hardware resources. Hypervisors enable the virtual machines to easily migrate from one server to another server and can be created/destroyed on demand, enabling the provisioning the elastic services with flexible and easy management. This layer provides similar properties to that of any computing layer.
>
> **Network Operating System:** Provides abstractions to lower-level devices in the form of APIs. These functionalities help increase productivity. Network operating systems (NOS) ease the burden of management by means of a logical centralized controller offered by the NOS. A NOS is known to provide abstractions to developers so that they can define network policies where they need not care about the underlying details of data distribution among routing elements. A critical element in a SDN architecture, NOS is a key supporting piece for control logic to generate the network configuration determined by the network operator. Similar to the operating system, NOS abstracts the underlying details of connecting and interacting with the forward devices.
>
> **Northbound Interface**: The northbound interface is a bridge between the management plane and the control plane. The northbound interface is mostly software in contrast with the southbound interface. Open and standard northbound interfaces are crucial to promote application portability and interoperability among various platforms. Northbound interfaces abstract the inner details of the controller functions and the data plane behavior from the developers**.**

Management Plane: Management plane consists of traffic required for the network operator to manage the entire network. This plane consists of three logical layers; Language based virtualization, Programming languages and Network applications.

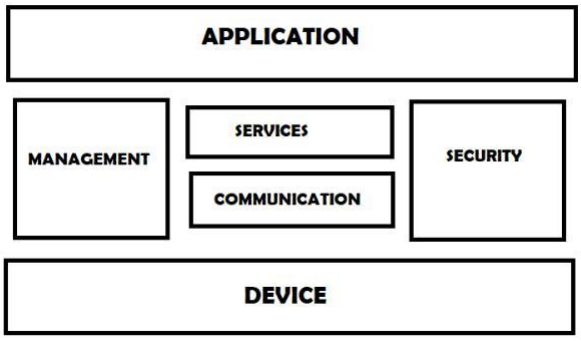
> **Language Based Virtualization**: This level is capable of expressing modularity and of allowing different levels of abstractions while guaranteeing desired properties such as protection. This level intrinsically simplifies the application developers' task. It incorporates such level of abstractions by introducing network objects. These objects consist of abstract network topology

and the sets of policies applied to it. This layer incorporates slicing which can be very helpful in isolation of various VMs in the cloud computing environment.

Programming Languages: In SDNs, higher level programming languages can be used to create higher level abstractions for simplifying the task of programming forwarding devices. They can enable more productive and problem focused environments for network software programmers, speeding up innovation and development. They can help in network virtualization and encourage software modularization and code reusability.

Network Applications: The applications can implement the control logic that will be translated into commands to be installed in the data plane, dictating the behavior of the forwarding devices. The main goal of these applications is to engineer the traffic with the aim of optimizing the network. SDN load balancing also simplifies augmentation of the network. Every time a new server is installed, the load balancing service can take appropriate actions to seamlessly divide the traffic among the available servers.

IoT Systems Architecture



IoT overview

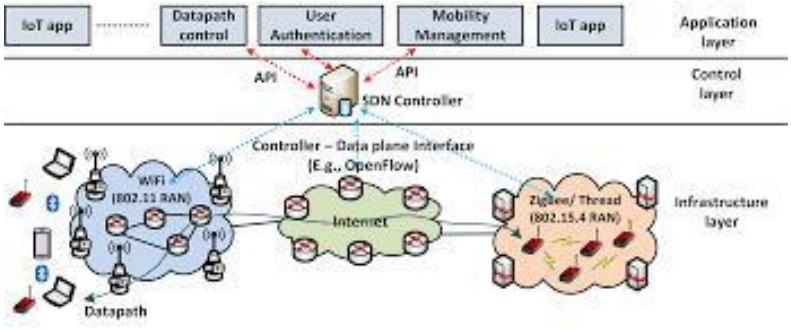
The IoT landscape has three layers: IoT device layer, IoT network platform and IoT application. This overview provides an end-to-end solution from devices to relevant IoT applications.

IoT Device layer: This layer includes all the devices that can interact with the physical environments by using identification and sensing capabilities. This layer is the prime source for data collection for analysis which could be carried out in the network platform. Technologies widely adopted in this layer are Radio Frequency Identification (RFID) and Wireless sensor networks (WSN). RFID tags are helpful in detecting and sensing whereas WSN have been used in various application scenarios. The most important property driving adoption of these technologies are the low cost involved in equipment setup and device management and the scalability of these technologies, from wireless in body sensors to unmanned aerial vehicles.

IoT Network layer: Considered the heart of the landscape, this layer provides various services such as network management, communications, and security, as shown in Figure 3. The platforms in this layer have been standardized. Cloud technologies have been the forefront for this layer because of the increased connectivity of the smart objects. A crucial aspect needed for this layer is low latency. To better cope with low latency requirement, edge computing has been viewed as an alternative to cloud solutions. Edge computing also helps promote distributed small scale cloud environments that can be deployed at the edge of the network to execute applications near the IoT devices. Several IoT systems have already embraced the edge computing paradigm as it is easier to split data processing between the edge and the cloud.

IoT Application layer: This layer consists of all the application modules required to provide the desired service to the end user. IoT platforms greatly differ to accommodate specific business logic requirements.

Integration of SDN in IoT environment



SDN architecture for IoT

Heterogeneity of the IoT devices makes it harder to propose a general solution using the existing network paradigm. Researchers consider SDN to be a hot topic for securing the IoT systems due to the inherent nature of the SDN paradigm which makes use of a centralized controller to perform routing/forwarding and isolation of parts of the network which can help in accommodating heterogeneous devices in a different part of the network. Figure 4 gives us a general solution to integrating SDN in IoT systems. In this architecture, IoT applications and services are implemented at the application layer. SDN controller-related functions are implemented in the control layer. IoT devices and gateways have been implemented in the infrastructure layer. The control software uses APIs to interact with the IoT application services at the application layer and IoT devices at infrastructure layer.

IoT Environment Attacks and Threats

IoT system threats were divided into sections depending on the domain where they are prevalent.

Attacks on IoT devices

Conventional security mechanisms are not guaranteed as the devices in this layer have constrained computation capabilities and limited energy. The devices can be wired or wireless and can have diverse range in terms of size. The attacks on IoT devices can be lethal as the compromised nodes may return altered measurements which can provide incorrect feedback and incorrect services.

Main attacks possible in this layer are:

Hardware Trojan Attacks: Trojan attacks are malicious modification on hardware which can allow unauthorized access to the attacker who can then exploit the compromised device to gain access to sensitive information or access to the software running on the device. For the attacker to insert a hardware trojan in the circuit, the attacker maliciously alters the design before/during fabrication. Trojans can be externally activated Trojans, i.e. actively triggered by an antenna or a sensor that interacts with the outside world or internally activated Trojans, i.e. activated after a certain condition is met inside the integrated circuit.

Replication Attacks: The attacker can create a new node which has the sensitive identification information of the victim. An adversary able to replicate one node can spread the influence through the entire network, thereby exposing the network to a variety of vulnerabilities. Using the replication of the nodes, the adversary can inject false information or suppress legitimate data.

Tampering Attacks: Refers to all scenarios where a malicious entity performs an unauthorized physical or electronic action on the device. This attack can be very dangerous as an adversary can exploit the physical access to the device to gain full control of the network. These attacks are known as node capture attacks causing internal malfunction or sabotage. These tampering attacks can be physical in nature. These attacks are quite simple to launch and can be extremely damaging.

Battery Draining Attacks: These attacks are very similar to denial-of-service (DoS) attacks as they render the IoT device useless by draining the battery. This attack is quite simple to launch; one way to drain the battery is to send a huge number of packets to the target device forcing their processing and the relevant resource consumption which can rapidly consume the available energy. Battery draining is an indirect attack which can lead to serious consequences. These attacks are most successful with wireless devices where the constant supply of power is missing.

Malicious Code Injection Attacks: IoT devices run a lightweight code to assist IoT applications. There is a possibility of code injection by the attacker to take control of the system. The injected code can disrupt normal behaviour and sabotage the network. Code injection can happen in the memory accessed by the lightweight code. The common way to inject this malicious code is through software by using buffer overflow attacks.

Attacks on IoT network layer

This layer is a crucial domain as it provides the connectivity between applications and devices. This layer also offers computation and storage capabilities of cloud environments. Attacks on this layer can be very damaging as they contain a large amount of information that can leaked to the attacker. Main attacks possible in this layer are:

Eavesdropping Attack: An attack where the attacker intentionally sniffs the data transmitted. As IoT involves wireless communication, it can be vulnerable to eavesdropping. IoT services typically carry sensitive information, hence it is very important to defend against sniffing attacks. The simple solution to this attack is to encrypt the data transmitted.

Denial-of-service attack: A DoS attack is one of the most common attacks in networked systems. In this attack, the attacker targets the communication links by flooding the IoT networks with massive data thereby compromising them. DoS attacks can exhaust the IoT device resources rendering them useless. Since a majority of IoT devices use some form of wireless communication, interference and jamming can be used to block radio transmissions. For example, using the DoS attack, the RF channels can be jammed such that the RFID tags cannot be read. The attacker can lock down all the doors in a building that can be accessed using RFID doors. Jamming devices can be continuous thus rendering the IoT device useless or jamming devices can be done in regular or irregular intervals to reduce performance of the device. Various approaches can be used to carry out the DDoS attacks, i.e. Ping of Death, Teardrop, UDP/SYN flood and SYN flood. Compromised IoT devices can be exploited to create large scale botnets and to launch massive Distributer DoS (DDoS) attacks.

Identity Spoofing: The objective of Identity spoofing is to generate and send malicious packets that seem legitimate to the network. Compromising the identity of an IoT device can be devastating to the network as they can produce illegitimate data thereby decreasing system performance. In such a scenario, an adversary can send malicious data using spoofed addresses. Identity spoofing can be used to launch DoS and DDoS attacks in the network.

Man-in-the-middle attacks: MitM attacks are the advanced version of the spoofing attack where the adversary impersonates both endpoints and makes independent connections with each target, to intercept the exchanged traffic. It then transfers and forwards the messages between them. Impersonation leads both sides to believe they are conversing directly but in actuality, the entire

conversation is controlled by the attacker. Authorization in the IoT architecture is attained by exchanging identified data between connected items. This procedure is vulnerable to eavesdropping, which can lead to a MitM attacks.

Routing attacks: Routing attacks affect where the network packet is being transmitted and ultimately, how a message is routed. The attacker can use such attacks to spoof, redirect, misdirect or drop the message at the communication level. In this way, the attacker can create routing loops and generate false error messages. In addition to routing attacks, many serious attacks have been proposed. For example, Black hole attack is an attack where the malicious node advertises the shortest path to its node. As a result, all the packets will be transferred to the malicious node. Hello Flood attack is based on a requirement that a node should broadcast" HELLO PACKETS" to show its presence to its neighbors. The receiving nodes assume that they are within the communication range of the sender. Sybil attacks are attacks where a Sybil node – nodes with fake identities. Sybil nodes can vote out honest nodes in the system. Worm hole attack is a severe attack that can be launched when authenticity and confidentiality are guaranteed in the system. The attacker first records packets in one location and then tunnels them to another location.

IoT Cloud Service Manipulation: A cloud/edge data center that is controlled by a malicious administrator can create a serious situation since the adversary can easily launch attacks against deployed Virtualized service instances, such as VMs. Thus, attackers may gain access to sensitive information gathered by the IoT devices.

Attacks on IoT application layer

The application layer can be used to implement the logic to make effective use of the IoT devices. Main attacks possible in this layer include:

Malicious Virus/worm: A computer worm is a type of malware that spreads copies of itself from computer to computer without any human intervention. Transmitted via software vulnerabilities, these worms or viruses allow data leakage and compromise the correct behavior of the system.

Application data leakage: The biggest concern in this domain is that of privacy leakage, whereby the sensitive information can be leaked by cyber criminals. The application data can be used to hack into the application itself as it contains application-related information. Application data can be used for further attacks and also installing a backdoor to the application.

Service logging failure: Logging activities can be extremely beneficial to the developers as they can used for data analysis. Authentication of the logs should be firm as they can contain information about device usage, etc. This information can be used to provide a customized service to the user. Application developers should be able to record authentication events and application errors in the relevant log. Compromised applications can produce a large number of incorrect logs which can impact the hypervisor logging analysis. Inefficient logging monitoring can limit the capabilities of implementing security controls in Cloud Computing environments.

Malicious Scripts: Malicious scripts can lead to data leakage which can severely impact software execution. The scripts are usually executed over IoT application portals. The scripts can be internal to the software that can be caused by untested bugs or scripts that may be external to the system, where the attacker may have intentionally placed it for the victim to execute when accessing a compromised application.

Phishing Attacks: Phishing attacks are malicious way to extract sensitive information by disguising as a legitimate application. Attackers can perform phishing attacks by leveraging infected emails, phishing websites, etc. The aim of these phishing attacks is to gain credentials of the victim.

SDN for IoT Security

The present network paradigm is proving to be insufficient for securing the IoT systems, resulting in heterogeneity and limited scalability. Researchers have considered SDN for increasing the IoT bandwidth and flexibility.

The main advantage of SDN is the decoupling of the control plane and the data plane. Decision making has been left to the SDN controller while the switches are just mere forwarding elements. This helps in network management and scalability. Another advantage of using SDN is its dynamic flow control. Switches can forward the packet to the controller if there is uncertainty to take future forwarding decisions. This way, the SDN applications can communicate their network requirements to the SDN controller for modification or addition of new flow rules. SDN makes network services agile and flexible as they can be automatically deployed and programmed. SDN with these added benefits has led to its effective adoption in IoT systems. The advanced countermeasures that have been provided by the adoption of SDN to IoT systems includes:

Traffic Isolation: SDN can be utilized to enable forwarding of different network traffic over the same physical infrastructure while ensuring desired isolation. This feature can help drastically reduce propagation and damages of security attacks via different network domains. This represents a fundamental feature in the IoT system, where sensitive operations can depend on data generated by other objects. We can also implement a SDN-based IoT access control application that is implemented on top of the SDN controller. In this way, each incoming or outgoing connections to the IoT system can be verified according to the security policies. Because of the dynamic programmability of SDN, malicious traffic can be blocked dynamically, resulting in a faster response security mechanism.

Dynamic Flow Control: SDN controller is capable of dynamically installing and updating forwarding rules, raising the potential for network applications to implement appropriate security mechanisms. This is possible due to decoupling the data plane and control plane. When a switch does not have a flow rule to decide how to process an incoming packet, a request can be forwarded to the controller and the controller responds back with an appropriate flow rule. This feature can enable a dynamic access control function, which helps ease management of the network and ensuring network privacy.

Network programmability: Data forwarding in the SDN network can be controlled by the network application program. This gives SDN enhanced flexibility to enable new security functions. Recent efforts in this area are worth mentioning. The FRESCO framework offers a scripting language to assist programmers in developing new SDN-based security mechanisms. Avant-guard has introduced two mechanisms; connection migration and actuation triggers. The former can reduce control traffic in the event of DoS attacks. The latter has improved network monitoring services. OFX framework allows installation of OFX software modules that carry out processing and monitoring tasks directly on the switches. Enterprise Centric Offloading System has been proposed to manage offloading the enhanced SDN programmability. SDN programmability ensures remote management and allows applications to be easily updated when new threats emerge.

Centralized network monitoring: The SDN controller has the visibility of the data planes under its supervision. Through the control plane, it can collect network status information by sending statistics query messages to the switches. The SDN controller can provide the necessary information such as status reports and flow request messages to the applications. This approach helps in creation and development of strategies for implementing anomaly network analysis and detection of nationwide networks.

Scalability and dynamic allocation of network and processing resources are the strongest aspects introduced by SDN. Most SDN controllers support distributed control planes and dynamic allocation of bandwidth and the switches ensure optimal routing and improving the reactivity and availability of the network, especially during attacks. Flexibility not only ensures infrastructure resilience against failures

but also enables multi-level security enforcement. SDN is also capable of providing platforms for a wide range of different vendors and applications by using multiple gateways and supporting multiple devices.

Security Threats from SDN Networks

There are many advantages to using SDN to enforce IoT security but security and dependability of SDN itself is a big concern. The increased flexibility and centralized controller can bring in additional attacks. Attackers can forge or fake traffic flows to launch DoS attacks by saturating the flow tables. Attacks on switches are not new as one switch can be used to drop or slow down packets in the network, leading to a bottleneck in the network. Attackers can also use control plane communications to launch a DoS attack or data theft. TLS/SSL communications are as strong as its weakest link such as a self-signed certificate. Moreover, SSL is known to be extremely vulnerable to MitM attacks. Even controllers are not safe from attackers, as a faulty or a malicious controller can compromise an entire network. Attackers could use exact combinations of attacks which could lead to a compromised controller that would be very difficult for IDS to catch. Attacks on administrative stations can also prove to be seriously debilitating as from one location, the entire network can be reprogrammed.

Open Research on SDN Networks

Researchers have been trying to inculcate extra hardware or software mechanisms to make the SDN more secure. Network function virtualization (NFV) seems like an ideal candidate as using NFV, we can make programmable switches which would be objects or instances making it easier to manage the network. Machine learning (ML) has crept its way in to SDN networks, providing better intrusion detection systems. ML is being used to detect ransomware which could seriously harm IoT systems. Frameworks are being developed to ease programmability of IoT systems. These frameworks tend to be lightweight as IoT systems are known to be resource constrained and are written in programmer-friendly languages for easy readability and increased collaboration. Federation schemes and security policies are being modified to provide fine-grained protection.

Conclusion

The IoT environment is ever-evolving, attracting an increasing number of cybercriminals who aim to exploit IoT system vulnerabilities to carry out malicious attacks on a potentially global scale. Conventional mechanisms are very inefficient in terms of dealing with security threats in this landscape. In this article, we have discussed various threats, the countermeasures and issues that may arise when adopting SDN in IoT systems. The future of SDN, despite all the threats, when combined with different fields like ML and data analysis can be solidified into a very secure system. I hope that this article carves a path to explore and tackle the challenges that may arise in this area.

Related and Referred Work

[1][2] give a very comprehensive idea about the SDN networks. The authors thoroughly explain the concept of SDN from scratch. A few recent papers have surveyed specific architectural aspects of SDN[3][4][5]. [6][7][8][9] gives an overview of the IoT architecture. [10][11][12][11] gives us an idea as to how we can integrate SDN with IoT. Attacks on IoT is compacted by [13]. Trojan attacks are given by [14]. [15], [16] examines replication attacks. Battery draining attacks are discussed in [17]. The authors of [18] detail malicious code injections. Eavesdropping attacks were discussed in[12]. DoS attacks and Distributed DoS attacks are discussed in[19][20] and [21]. Spoof attacks were given in mukaddam2014ip. The advantages of using SDN in IoT networks is given by [22]. The threats of using SDN in IoT environments is given by [22]. The adoption of machine learning in SDN networks for detection of security threats is given by [23]

References

D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, 2014.

J. Esch, “Prolog to,” software-defined networking: a comprehensive survey”,” Proceedings of the IEEE, vol. 103, no. 1, pp. 10–13, 2014.

B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and Turletti, “A survey of software-defined networking: Past, present, and future of programmable networks,” IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1617–1634, 2014.

A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation using openflow: A survey,” IEEE communications surveys & tutorials, vol. 16, no. 1, pp. 493–512, 2013.

Y. Jarraya, T. Madi, and M. Debbabi, “A survey and a layered taxonomy of software-defined networking,” IEEE communications surveys & tutorials, vol. 16, no. 4, pp. 1955–1980, 2014.

S. Chakrabarty and D. W. Engels, “A secure iot architecture for smart cities,” in 2016 13th IEEE annual consumer communications & networking conference (CCNC), pp. 812–813, IEEE, 2016.

A. A. Mutlag, M. K. A. Ghani, N. a. Arunkumar, M. A. Mohammed, and O. Mohd, “Enabling technologies for fog computing in healthcare iot systems,” Future Generation Computer Systems, vol. 90, pp. 62–78, 2019.

A. Krylovskiy, M. Jahn, and E. Patti, “Designing a smart city internet of things platform with microservice architecture,” in 2015 3rd International Conference on Future Internet of Things and Cloud, pp. 25–30, IEEE, 2015.

I. Azimi, A. Anzanpour, A. M. Rahmani, T. Pahikkala, M. Levorato, Liljeberg, and N. Dutt, “Hich: Hierarchical fog-assisted computing architecture for healthcare iot,” ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 5s, pp. 1–20, 2017.

O. Flauzac, C. Gonzalez,´ A. Hachani, and F. Nolot, “Sdn based architecture for iot and improvement of the security,” in 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, pp. 688–693, IEEE, 2015.

A. Molina Zarca, J. Bernal Bernabe, I. Farris, Y. Khettab, T. Taleb, and Skarmeta, “Enhancing iot security through network softwarization and virtual security appliances,” International Journal of Network Management, vol. 28, no. 5, p. e2038, 2018.

F. I. Khan and S. Hameed, “Software defined security service provisioning framework for internet of things,” arXiv preprint arXiv:1711.11133, 2017.

M. Abomhara et al., “Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks,” Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65–88, 2015.

A. Mosenia and N. K. Jha, “A comprehensive study of security of internet-of-things,” IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 4, pp. 586–602, 2016.

J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, “Wireless sensor network security: A survey,” Security in distributed, grid, mobile, and pervasive computing, vol. 1, p. 367, 2007.

B. Parno, A. Perrig, and V. Gligor, “Distributed detection of node replication attacks in sensor networks,” in 2005 IEEE Symposium on Security and Privacy (S&P’05), pp. 49–63, IEEE, 2005.

A. Becher, Z. Benenson, and M. Dornseif, “Tampering with motes: Real-world physical attacks on wireless sensor networks,” in International Conference on Security in Pervasive Computing, pp. 104–118, Springer, 2006.

M. Khouzani and S. Sarkar, “Maximum damage battery depletion attack in mobile sensor networks,” IEEE transactions on automatic control, vol. 56, no. 10, pp. 2358–2368, 2011.

G. Noubir and G. Lin, “Low-power dos attacks in data wireless lans and countermeasures,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, no. 3, pp. 29–30, 2003.

D. R. Raymond and S. F. Midkiff, “Denial-of-service in wireless sensor networks: Attacks and defenses,” IEEE Pervasive Computing, vol. 7, no. 1, pp. 74–81, 2008.

E. Bertino and N. Islam, “Botnets and internet of things security,” Computer, vol. 50, no. 2, pp. 76–79, 2017.

D. Kreutz, F. M. Ramos, and P. Verissimo, “Towards secure and dependable software-defined networks,” in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 55–60, 2013.
A. Wani and S. Revathi, “Ransomware protection in iot using software defined networking,” International Journal of Electrical and Computer Engineering, vol. 10, no. 3, p. 3166, 2020.