IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

Improving Network Security through Software Defined Networking (SDN)

Rakshitha S M, Shravana, Riya Biswas, Sushmita Shetty
Students, Department of Computer Science and Engineering
Alva's Institute of Engineering and Technology, Mangalore, India

Abstract: *The Internet of Things (IoT) connects many of the world's home appliances, from intelligent thermostats to intelligent cars. We had 9 billion connected things by the end of 2015. According to Gartner, the total number of networked devices will approach 50 billion by 2020. The majority of linked devices in the existing network use outdated security technology and encryption, and many do not allow for remote device updates. Given the vast number of IoT devices available, ranging from off-the-shelf motion monitoring to machine tools, it is impossible to determine which functions are associated with any given service or product. In a nutshell, this is the issue: you can't trust the device's security and integrity. K-times anonymous authentication (k-TAA) is an important access control approach used in ecoupon and e-bill. It allows a user to authenticate himself anonymously to a distant server a set number of times. However, most known k-TAA techniques need a lot of processing, which makes them difficult to use on devices with low resources. In 2018, Tian and colleagues published Tian et al presented a remote user authentication system that protects users' privacy. Considering untraceability of k-times In contrast to the typical k-TAA,Tian et almethod .'s is better suited to mobile devices.as a result of avoiding costly pairing operations They're also claim that their system ensures user trust and authenticity. Untraceability k times Unfortunately, we are unable to do so in this paper Analyzing their scheme reveals that it is insecure. Their Neither scheme can stop a rogue user from sending information neither the authentication.*

Keywords: Internet of Things

I. INTRODUCTION

The Internet of Things (IoT) is a collection of embedded devices that are linked together by a network and gather, distribute, and process aggregated data in order to perform a task. Sensors are the most popular IoT devices, which sense data and send it to a processing device or the cloud. Everything in today's world has to be smart. Because of the advancement of IoT, smart homes, smart infrastructure, and smart transportation are now possible. Professor Ashton of MIT's American Auto-ID Center proposed the concept of IOT as early as 1999. The International Telecommunication Union (ITU) gave it the appropriate definition in 2005. Because individuals understood that it has immeasurable potential, the Internet of Things has become a hot topic for scientific research technology personnel, Intelligent power networks, intelligent traffic, intelligent logistics, intelligent buildings, GPS navigation, industrial monitoring, modern agriculture, public security, environmental management, remote medical treatment, and digital urban management, for example. People anticipate that the Internet of Things will bring a great deal of convenience to their vision. The development of IoT applications and the controllability of information have become mutually exclusive. These issues will make the user panic if there is no in-depth study

Network supervisors can monitor networks using Software-Defined Networking (SDN) [1, a developing network architecture]. Manageable, dynamic, cost- effective, and adaptable are the core qualities of SDN. The following are some of the most important elements of SDN**:**

- **Programmable Networks**: The separation of the control and forwarding planes allows for network programming without regard for forwarding services.
- **Flexible Development**: This decoupling also enables for quick and adaptive changes to flows and network services.
- **Centralized Management**: The SDN architecture is built on a centralised controller that maintains a global view of the network and interacts with it by changing flows.
- **Allows to Automate the Device Programming**: SDN enables administrators to set up self-written programmes

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

(since SDN is not a piece of software) to speed up the configuration, optimization, and management of network resources.

- **Based on Open Standards**: allows for network management to be simplified because it is not dependent on any specific providers, adaptation devices.

The goal of SDN architecture is to address issues with traditional network architecture that have become static.The control and data planes are separated from the network plane to achieve this goal. Packets are transported to the same destination via the same path on a legacy network using pre- loaded primitives. In an SDN system, on the other hand, the centralised controller manages the forwarding rules.An SDN Controller, as well as southbound and northbound APIs, are included in any SDN model.

The controller is a programme that runs on a server that is located someplace on the network. Controllers and network devices must communicate through some interface; the OpenFlow [2] protocol is one of the most common interfaces used by SDN controllers. OpenFlow provides network managers with a collection of entities that allow them to describe network flows and the direction they will take without interfering with current traffic. It also includes ways for defining rules that aid in achieving particular features, such as larger band width, lower latency, fewer hops, and lower the amount of energy necessary for traffic to reach its destination. The OpenFlow protocol lets administrators migrate network control away from proprietary network switches in an indirect way. Northbound APIs allow applications to communicate with the controller. Similarly, all commands from the controller to the network devices are sent through the southbound APIs. Furthermore, southbound APIs provide a layer of abstraction, allowing the controller and applications to ignore network device types.

- **Southbound APIs:** They allow the controller and network parts to communicate with each other. They enable network pieces to be dynamically changed in order to react to changing requirements in real time. Furthermore, they provide for independence between the controller's underlying pieces. This is due to the fact that the controller just has to be aware of transmitting instructions, which the adaption APIs will interpret and route to the correct network element.
- **Northbound APIs**: They are used to communicate the controller with applications that are running over the network. These Northbound APIs boosts research and enable effective automation and orchestration of the network. Northbound APIs are the vital APIs in the SDN technology, since they play key role in providing interaction between network and the business applications. They allow the controller and network parts to communicate with each other. They enable network pieces to be dynamically changed in order to react to changing requirements in real time. Furthermore, they provide for independence between the controller's underlying pieces.

II. Related Work

Enhancing performance and achieving greater connection are two of today's commercial and technical network requirements. Companies are required to comply with an increasing number of industry-specific security rules, and there is an increasing need for mobility.

Networking protocols have progressed greatly during the previous few decades in terms of these requirements. However, because of the way traditional networks are built up, adopting a single protocol to meet these demands across the board is rather difficult. There are two fundamental characteristics of the classic networking approach:

- Network functionality is mainly implemented in a dedicated appliance. In this case, dedicated appliance‟ refers to one or multiple switches, routers and/or application delivery controllers.
- Most functionality within this appliance is implemented in dedicated hardware.

The business organizations are defied with the limitations that get together with this hardware centric approach, such as:

- Traditional configuration is time-consuming and inefficient.
- Multi-vendor environments require a high level of expertise.
- Traditional architectures complicate network segmentation Networking Software Defined

Networking (SDN) is rapidly becoming the new buzzword in the networking business. Expectations are that this emerging technology will play an important role in overcoming the limitations associated with traditional networking. This concerns the two network device planes [3],

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

- The plane that determines where to send traffic (control plane).
- The plane that implements these decisions and forwards traffic (data plane)

Software Defined Networking is projected to have several business paybacks, viz., More configuration accuracy, consistency, flexibility and Data flow optimization, Integration, Visibility and reporting (Device visibility, Events and alerts and Traffic reporting), Security (Network Zoning, Live Threat Protection and Port Security), Reliable connectivity (Offline measures, Uplink Balancing, Traffic Prioritization).

The current scenario states that growing number of physical objects are being connected to the Internet at an alarming rate realizing the idea of the Internet of Things (IoT) [4]. IoT applications require communication compatibility between heterogeneous things. In addition, the traditional Internet architecture needs to be revised to match the IoT challenges. The IoT should be matured enough to interconnect large number of heterogeneous objects through the Internet, so there is anacute need for a flexible layered architecture.

Middleware is indispensable to ease the development of the diverse applications and services in IoT [2]. Although the existing middleware solutions address many requirements associated with middleware in IoTs, some requirements and related research issues remain relatively unexplored, such as scalable and dynamic resource discovery and composition, system-wide scalability, reliability, security and privacy, interoperability, integration of intelligence and contextawareness.

UbiFlow [5] is a software-defined IoT system for efficient flow control and mobility management in urban Multinetworks. Besides flow scheduling, it shifts handover optimization, mobility management and access point selection functions from the relatively resource constrained IoT devices to more capable distributed controllers. The distributed controllers are organized in a scalable and fault tolerant manner. The system was evaluated through simulation and testbed.

The paper [3] presents an original SDN controller design in IoT Multinetworks whose central, novel feature is the layered architecture that enable flexible, effective, and efficient management on task, flow, network, and resources. A novel vision on tasks and resources in IoT environments, and how to bridge the gap between abstract high level tasks and specific low level network/device resources, is illustrated. The Network Calculus model is modified to accurately evaluate the end-to-end flow performance in IoT Multinetworks, which is further serving as essentials of a novel multi-constraints flow scheduling algorithm under heterogeneous traffic pattern and network links. The semantic modeling approach performs resource matching and the GA-based algorithm schedules flows. Those techniques can be viewed as plug- ins and can be adjusted or replaced in different IoT scenarios.

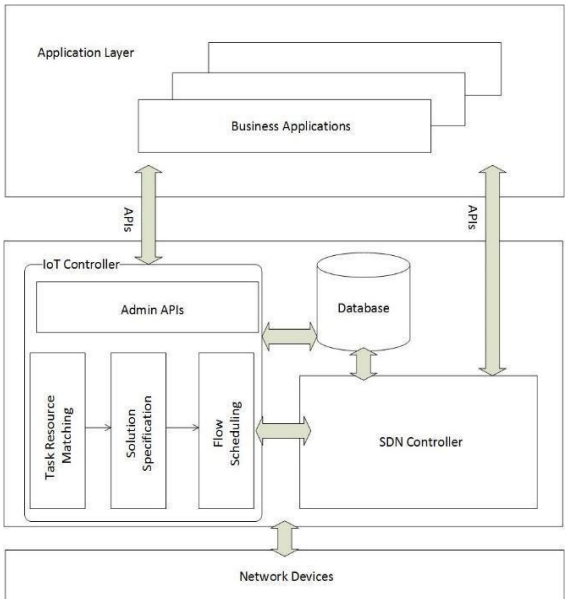

Figure 1: System Architecture

In Fig. 1 the task-resource matching component of the controller maps the task request onto the existing resources in the network. Once resource set solution is selected, the service solution specification component of the controller maps the

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

characteristics of the devices and services involved in that solution to specific requirements for devices, networks, and application constraints. The Flow Scheduling component takes these requirements and schedules flows that satisfy them. Finally the controller triggers the necessary communications in the IoT network. There are many SDN controllers designed by network vendors, which provides SDN infrastructure for development. OpenDayLight (ODL) [6] is one of the SDN controller which is much developed than its competitor.

ODL builds infrastructure for SDN deployments. It provides a model-driven service abstraction (service abstraction layer) that allows users to develop applications that easily work across a wide variety of hardware and southbound protocols. It relies on the following technologies:

- **Maven**: Project management tool that simplifies and automates dependencies within a project or between different projects. This tooling will help developers to manage all the required plugins and dependencies, as well as to provide a project start-up using its defined archetypes.
- **Java**: It is the programming language that is used to develop applications and features in the OpenDaylight‟s controller. Developing in Java provides a valuable compile-time safety, as well as an easy way to implement defined services.
- **Open Service Gateway Interface (OSGi)**: It is the back-end of OpenDayLight controller as it allows to dynamically load bundles and JAR packages, and bind modules for exchanging information.
- **Karaf**: It is an application container built on top of OSGi, which simplifies aspects of packaging and installing applications.
- **Yet Another Next Generation (YANG):** it is the key- point of the model-driven behavior in the controller. Developers will use YANG to model an application functionality, and to generate APIs from the defined models, which will be later used to provide its implementations. YANG supports modelling operational and configuration data, as well as RPC and notifications.

Genetic Algorithm (GA) is stimulated by the theory of natural evolution and its principles. It is one of the directed random search techniques, employed to find a near- optimal solution for many larger problems in complex multi-dimensional search spaces. Paper [1], proposes a multipurpose optimization method for QoS routing based on GA. The proposed method is a source based routing method and has a flexible and adaptive behavior. The proposed method can support QoS routing for multiple metrics. The adaptive routing mechanism has a load balancing system among alternative paths. The individual genes are used to express the connected nodes ofa route.

III. Conclusion

The SDN-based infrastructure and technologies will encounter the IoT at the crossroads of VPN enervation, uptime challenges and limited network resources. The projected result is that Software-Defined Networking will assist in boosting the expansion of IoT-enabled devices, enable more effective network resource sharing and enhance IoT service-level agreements (SLAs). In return, many vendors expect IoT will support SDN decisions and nourish greedy policy engines.

References

[1] “ONF SDN Evolution,” 2016.
[2] K. Sood, S. Yu, and Y. Xiang, “Software defined wireless networking opportunities andchallenges for internet of things: A review,” 2015.
[3] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, “A software defined networking architecture for the internet-of-things,” in 2014 IEEE network operations and management symposium (NOMS), pp. 1–9, IEEE, 2014.
[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet ofthings: A survey on enabling technologies, protocols, and applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.
[5] D.Wu, D. I. Arkhipov, E. Asmare, Z. Qin, and J. A. McCann, “Ubiflow: Mobility management in urban-scale software defined iot,” in 2015 IEEE Conference on Computer Communications(INFOCOM), pp. 208–216, IEEE,.
[6] “Opendaylight Project”https://wiki.opendaylight.org/