A Review Paper on Edge Computing: Architecture, Applications, Security Threats, Mechanisms, and Challenges

Farha Akhter Munmun, Adeeba Anis

Bangladesh University of Business and Technology, Bangladesh

Corresponding author: Farha Akhter Munmun, Email: farhapolin3@gmail.com

With the increasing popularity of Internet of Things (IoT) devices, the centralized and conventional architecture of cloud computing has become a bottleneck because of its limited bandwidth and resources. Therefore, edge computing has become an emerging technology nowadays that enables data processing and storing at the edge of networks. Instead of having some unique features (distributed architecture, parallel processing, mobility support, location awareness, proximity, and low latency) privacy and security have become an open concern for edge computing. This paper presents a comprehensive review in terms of the characteristics, architecture, and applications of edge computing. Also, some of the most recent security threats, mechanisms for avoiding these threats are highlighted with proper detail. Finally, we conclude our review analyzing some open issues and challenges in the field of edge computing.

Keywords: IoT devices, Cloud computing, Edge computing, Security threats, Security mechanisms.

1 Introduction

With the advent of internet technologies, people have become much more dependent on smart computing devices. Over the last few years, the popularity of IoT devices is increasing explosively due to lower cost and user convenience. According to the annual internet report of CISCO, there will be around 29.5 billion

2021. In Rahul Srivastava & Aditya Kumar Singh Pundir (eds.), *New Frontiers in Communication and Intelligent Systems*, 271–290. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-95502-00-4-29

devices connected to the internet by 2023 [1]. So, most individuals and organizations want such a high-performance and smart platform where they can have access to shared resources without acquiring them physically [2]. This was the initial idea behind the development of cloud computing. Instead of having many unique features such as resource pooling, scalability, and large network access, cloud computing increases the average response time and jitter because of its centralized architecture [3].

To solve the problem of the long delay, a new concept namely edge computing [4], [5], [6] has emerged. Edge computing is an extended architecture of cloud computing bringing the utilities and services of the cloud closer to the end-users. Nowadays, internet-based applications require fast data processing and lower response time [7], [8]. These applications run on the resource constraint small computing devices of end-users while the whole processing is performed in the cloud. Edge computing attains the requirements of these internet-based applications by shifting the functionalities of cloud computing into the edge of networks. The first technology of edge computing has been introduced in 2009, which is cloudlet [9]. Although this cloudlet technology has been developed to extend mobile cloud services, it was inefficient due to its restricted WiFi coverage. In recent times, some related technologies like mobile edge computing [10], [11], mobile cloud computing [12], [13], and fog computing [14], [15] also provide fast data processing and better user experience. Although edge computing has many advanced features such as parallel computing, distributed architecture, a large amount of data processing, mobility support, location awareness, proximity, and low latency, these features are not enough for preserving the security and privacy of data. Moreover, it is also a very challenging task to implement any strong encryption algorithm for maintaining confidentiality and security of data in mobile devices because most of these devices are highly resource constraint. The main contributions of our paper can be listed as:

- A detailed overview of the characteristics and architecture of edge computing is presented. Also, the real-life applications of edge computing are described.
- A discussion about the most recent security threats and mechanisms for avoiding these threats is presented.
- A detailed analysis of some open issues and challenges in the field of edge computing are highlighted.

The rest of the paper is organized as follows: Section 2 represents the background studies including the definition, features, architecture, and applications of edge computing. State-of-the-art security threats and mechanisms are described in section 3 and section 4 respectively. Section 5 presents some open issues and challenges. Finally, we have concluded our paper in section 6.

2 Background

This section presents the basic concepts of edge computing including it's three-layer architecture, features, and applications.

2.1 Edge Computing Architecture

Edge computing is basically a three-layer hierarchical architecture that occupies between mobile devices and the cloud. This three-layer architecture includes the cloud server layer, the edge server layer, and the edge device layer hierarchically. Fig. 1 shows the three-layer architecture of edge computing.

(i) **Cloud server layer**: This layer consists of several data servers and the central server. One of the primary responsibilities of the cloud server layer is to store and manage the massive amount of data generated by IoT and mobile devices. Also, this layer is responsible for providing the top level of authentication, authorization, and processing of data [4].

(ii) **Edge server layer**: This layer is a hierarchical structure of different sublayers with multiple edge servers. These sublayers are organized with increasing computational power from bottom to top. Edge servers located at the lowest sub-layer contain access points (APs) and wireless base stations. These access points and base stations receive data from the lower layer edge servers and forward the data to the edge servers located at the upper layer. Edge servers are mostly responsible for most of the core computing functions such as computation, task offloading, authentication, authorization, and data analytics.

(iii) **Edge device layer**: The lowest layer of edge computing is edge device layer. This layer contains those low-level IoT and mobile devices that perform the tasks of actuating, sensing, and controlling. Most of the IoT devices are highly resource-constraint, lightweight, and connected to the edge servers through wireless protocols such as WiFi, 4G/5G, and Bluetooth.

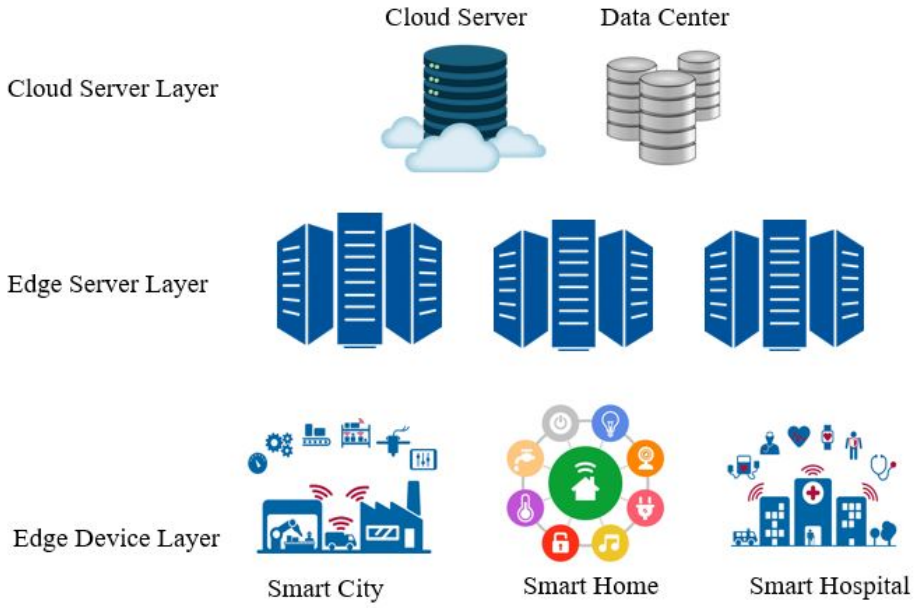

Figure 1: Three-layer architecture of edge computing.

2.2 Features of Edge Computing

As edge computing is an extended architecture of cloud computing it has few features similar to cloud computing. This section introduces the characteristics that make edge computing unique to cloud computing.

(i) **Distributed architecture**: Edge computing replaces the centralized architecture of cloud computing by bringing the cloud services into the edge of networks. In edge computing technology, data are processed at the edge nodes while in cloud computing data processing is performed in the central cloud server. Edge computing can run separately, without being attached to the rest of the network having access to local resources.

(ii) **Location awareness**: The property of determining the geographical location of a user device is defined as location awareness. When an application wants to provide this facility, the end-users of the application are needed to provide the details of the physical location to the cloud server. This may cause privacy leakage of users [16]. But, in edge computing, each node edge keeps track of the physical location of each device within its coverage area

and the users need not provide details of their location to the cloud server. Various edge computing applications such as edge-based disaster management and fog-based vehicular safety applications use this feature.

(iii) **Low Latency**: In edge computing architecture, the computation is done at the edge of the network and closer to the end-users. This reduces the response time and enables the end-users to access delay-sensitive and resource-constraint applications such as remote health monitoring, connected vehicles, and warehouse logistics.

(iv) **Mobility Support**: With the increasing number of IoT and mobile devices, edge computing provides mobility support that communicates with mobile devices directly. This is one of the most required properties of edge computing because mobile devices are most likely to move from one location to another covered by each edge node.

(v) **Proximity**: Edge computing avails the services and computational resources in the proximity of users for a better experience. The availability of services and computational resources in the local vicinity allows the users to leverage the network context information for making off loading decisions and service usage decisions [11].

(vi) **Heterogeneity**: Heterogeneity can be defined as the existence of various architectures, infrastructures, and communication technologies used by edge computing elements (edge servers, end devices, and networks).

(vii) **Bandwidth intensive use-cases**: With the exponential growth of data generated by IoT devices is bandwidth-intensive. Bringing computational resources as close as possible to high-bandwidth data sources implies that much less data need to be sent to the distant cloud data centers.

2.3 Applications of Edge Computing

Edge computing has several unique features compared to cloud computing and these features make this edge computing technology more suitable for different types of applications like autonomous cars, smart home monitoring, virtual reality, real-time traffic monitoring, smart cities, and smart industry as shown in Fig. 2.

(i) **Computation offloading**: With the increasing popularity of IoT devices (smartphones, smart TVs, smartwatches, laptops, and Internet TVs), the

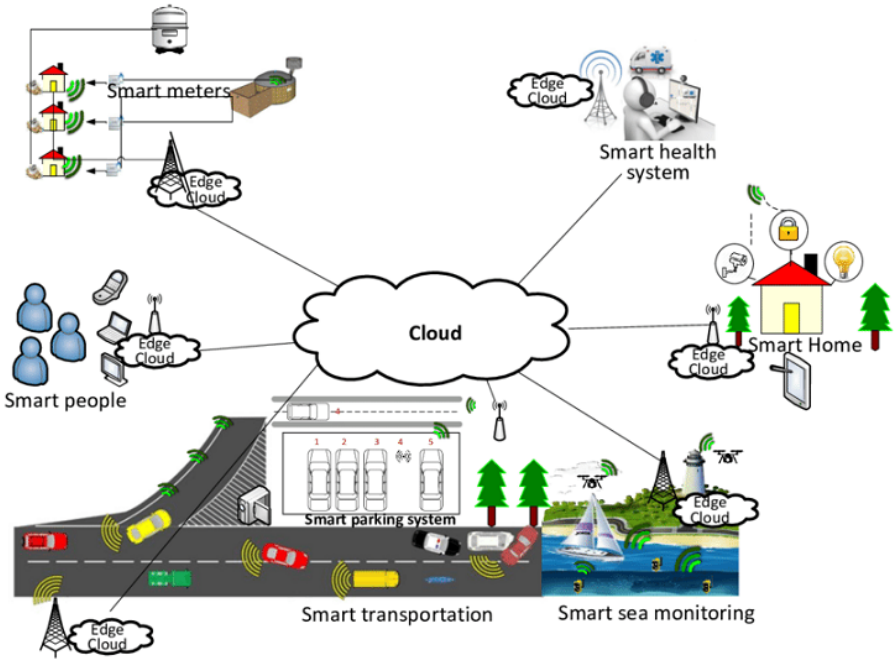

Figure 2: Edge computing applications.

requirement for real-time and low-latency processing is also increasing for these devices. Computational offloading is the process of transferring the computational process of resource-intensive devices to an external processor such as an accelerator or hardware. In traditional cloud computing, data processing is performed in the central cloud which increases the latency. In edge computing paradigm the computational overload can be handled by shifting the workloads at the edge of network.

(ii) **Internet vehicles**: Nowadays the number of smart vehicles on road is increasing rapidly. All the vehicles have a computation unit to acknowledge the intelligent traffic application. One of the greatest applications of edge computing is in internet vehicles. The vehicular network can achieve two-way communication such as Vehicle to vehicle (V2V), Vehicle with infrastructure (V2I) by using Road Side Units (RSUs). By applying communication and computation mechanisms cloud service is deployed in the edge server of the RSU.

(iii) **Video analytics**: It can be defined as a self-independent technology that analyzes any incident surveilled by several video cameras. Edge computing is playing a vital role in real-time video surveillance systems. Video surveillance system deployed in cloud computing is not so feasible for the fast increase of IoT devices. But edge computing has emerged with a new function processing the data at the edge of the network. As the data is processed close to the source of data it lowers the bandwidth consumption. Moreover, the data is processed in real-time and offers the fastest response.

(iv) **Smart grid**: A smart grid system can be described as an intelligent network system that establishes an efficient and reliable distribution of budget energy by amalgamating the actions of various users, producers, and consumers. Each smart grid system architecture contains separate functional entities such as operating system, communication gateway. Consumers' electricity usage is tracked by smart meters embedded in the smart grid which is used for grid analysis or pricing afterward. By deploying the system in the edge computing paradigm, all the computation can be done in an edge server which retains data privacy in a limited area and lowers the burden of cloud storage.

3 Edge Computing Security Threats

Security threat can be defined as a potential that can violate the security of any computer system and organization. In this section, some of the major security threats for edge computing are explored. The computational power of edge computing is generally low than cloud computing. For this reason, edge computing is less defensive to these threats. The security threats of edge computing are divided into four main categories which include: DDoS attacks, malware injection attacks, authentication and authorization attacks, and over priviledged attacks as shown in Fig. 3. Table 1 represents a comparison among the consequences of different security threats.

3.1 DDoS Attacks

In a DDoS attack, the attacker creates a group of devices and controls all the devices. Then each of the devices is ordered to offer a Denial of Service attack to the targeted device or edge server to stop the functionality of the targeted

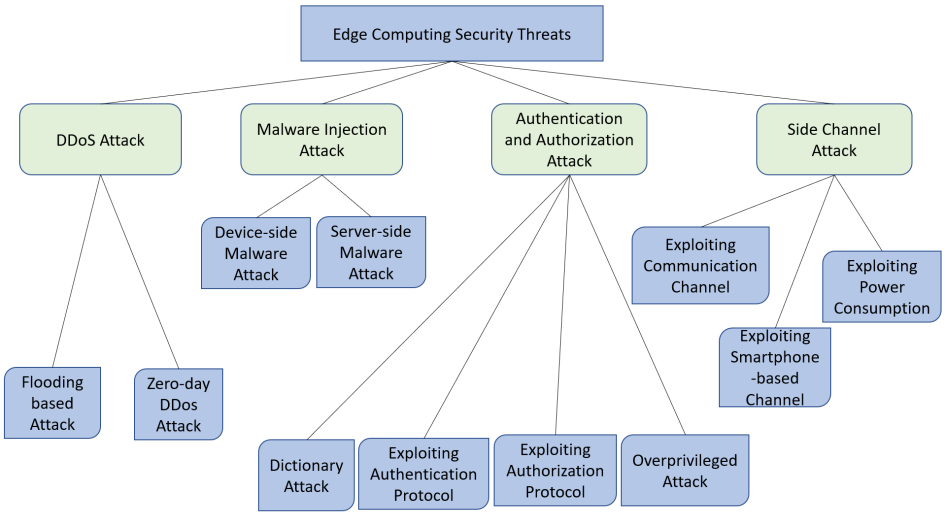

Figure 3: Edge computing security threats.

device [4]. Some recent examples of DDoS attack is Dyn, Mirai botnet [17], edge botnet [18], Zero-day DDoS attack, and Flooding based attack.

(i) **Zero-day DDoS attack**: A zero-day DDoS attack is one reasonably DDoS attack. In a zero-day attack, first, the attacker has to find an unspecified or secret vulnerability of the program code and then generate memory corruption to shut down the program. The attacker crashed the service while attacking the application layer of the edge architecture [19].

(ii) **Flooding-based attack**: Another type of DDoS attack is flooding-based attack. In this attack, attackers send a large amount of malware message packets simultaneously so that the system can not investigate it and the service is stopped. ICMP flooding, UDP flooding, SYN flooding, ping of death (PoD), and HTTP flooding are some kinds of flooding attacks. In UDP flooding, the attacker attacks a targeted edge server by sending a huge amount of disrupted IP packets containing User Datagram Protocol [20] to make the system incapable of controlling all the packets and terminating the normal services of the server. The attacker floods the victim edge server with a huge amount of ICMP (Internet Control Message Protocol) Echo Request packets so that the system is overwhelmed with a large number of messages and gets slow down. Another way of ICMP flooding by using some code and tool is scary and hping [21].

3.2 Malware Injection Attacks

In a malware injection attack, the attackers install malware or malicious program in the targeted edge server so that the server gives the output in a way that the attacker wants. Malware injection attacks can be categorized into two classes including server-side malware injection attacks and device-side malware injection attacks.

(i) **Server-side malware injection attacks**: In a Server-side Malware Injection attack, the malware is injected within the edge server or system. There are few types of server-side malware injection attacks: SQL injection, cross-site scripting (XSS), XML signature wrapping, Cross-Site Request Forgery (CSRF), Server Side Request Forgery (SSRF).

In an SQL injection attack, malicious codes are injected as SQL queries in the database to destroy the backend of the system. In XSS, the attack takes place on the client-side of the edge server where the developers visit to get the services of the server as a guest client. Malware is injected in HTML or Javascript codes by the attacker. XML messages are interrupted and modified in XML signature wrapping. Afterward, the modified data is re-transmitted to the predefined destination. CSRF is an attack where the attacker deceives the end-user to implement malicious codes through a web application. The internal data or services are altered by the edge server in SSRF.

(ii) **Device-side malware injection attacks**: Numerous types of IoT devices are used in edge computing, hence malware can be injected into the devices in various ways. The most frequent way of malware injection in the devices is Remote Code Execution (RCE) or command injection that can be done remotely without any physical existence of the attacker. For different devices, the malware behaves differently. A mighty IoT botnet named IoTroop or Reaper botnet was used to attack a minimum of one company in January 2018 [22].

3.3 Authentication and Authorization Attacks:

By authentication, any system validates the identity of a user. Authorization control the access rights for using any resources of the system for that particular user. In this attack, the attackers want to avoid the authentication and authorization process so that they don't get caught. Authentication and authorization attacks

can be categorized into four classes including dictionary attacks, exploiting weakness in authentication protocol, exploiting weakness in authorization protocol, and over privileged attacks.

(i) **Dictionary attacks**: This is one kind of authentication and authorization attack. A dictionary attack is the most facile attack among the authentication and authorization attacks. Here, the attacker creates a dictionary using the most workable passwords for the system and applies those by one.

(ii) **Exploiting weakness in authentication protocol**: In dictionary attacks, the uses of resources are more but the success amount is less. For this reason, the attacker tries to find the drawbacks of the edge computing authentication design. The authentication of identity is the first shield to protect the user data security [23].

(iii) **Exploiting weakness in authorization protocol**: The attacker looks for the design bugs in the authorization process to access the system as an authorized person and use all the delicate resources as well as perform all the works that are only permissible for the authorized users.

(iv) **Overprivileged attacks**: In an overprivileged attack, the attacker can access the system and insert malware to shut down the system or to get the authorized access to the system. By overprivileged attack, the attackers can change the door pin of a smart home, retrieving the voice history of a user [21], giving false alarm in smart home, etc.

3.4 Side-Channel Attacks

In a side-channel attack, the attackers first find some information of the system or user that is publicly accessible, then match up the data to find some private data to access the system. In this attack, the attacker breaks the cryptographic algorithm of the system. Three types of side-channel attacks include attacks exploiting communication channel, attacks exploiting power consumption, and attacks exploiting smartphone-based channel.

(i) **Attacks exploiting communication channel**: By exploiting a communication channel, the attacker has a huge possibility to get the sensitive information of a user. It is comparatively facile because the attacker does not need to access the edge server, the attacker can obtain user's private information by accessing any node of the communication channel.

(ii) **Attacks exploiting power consumption**: All the electrical devices need power consumption but different devices need different amounts of zeal as they are designed to consume different levels of power supply. So, by tracing the amount of power consumption the attacker might interpret the device. It can be subcategorized: the devices' power consumption which is collected by a meter (household activities like cooking, laundering) that is exploiting power consumption collected by meter, and the devices power consumption which is collected by oscilloscopes (voltage, current) that is exploiting power consumption collected by oscilloscopes.

(iii) **Attacks exploiting smartphone-based channel**: The devices with high-frequency modules behave like an antenna and radiate an electromagnetic emission. This confidential electromagnetic emission can be collected by any antenna [24]. These attacks can be subcategorized in two ways: one using the */proc* filesystem which is created by the kernel of Linux and another is using sensors of smartphones.

4 Security Mechanisms

Nowadays, providing security of data becomes one of the most crucial challenges in edge computing environment. Some defense mechanisms of the aforementioned security threats have been discussed in this section.

(i) **Defense against DDoS attacks**: The prime reason for flooding-based DDoS attacks is the design flaw in protocol-level within communication network protocol. Flooding-based DDoS attacks can be detected per-packet level or statistic level. In packet-based detection level per packet must be filtered and if any malicious packet is found that should be dropped before reaching the destination. In statistic level detection any group of DDoS traffic is detected hence every packet is not needed to be checked. Statistic level DDoS attack is solved using various machine learning tools [4]. K. Bhardwaj et al. [17] proposed an architecture named ShadowNet for defending DDoS attacks.

To defend against the zero-day attack, program code must be checked consciously so that there remain no vulnerabilities. It is possible to identify vulnerabilities in firmware using deep natural language processing (NLP)

Table 1: Comparison between edge computing security threats

Consequences	DDoS Attack	Malware Injection Attack	Authentication and Authorization Attack	Side Channel Attack
Generates memory corruption	Yes	No	No	No
Malware injection within edge server	No	Yes	No	No
Sends large amount of malware message simultaneously	Yes	No	No	No
Exploits any node of the system	No	No	No	Yes
Uses any resource of the system	No	No	Yes	No
Changes any password of the system	No	No	Yes	No
Breaks cryptographic algorithm	No	No	No	Yes

and deep learning models such as graph neural networks (GNNs), recurrent neural networks (RNNs) and the accuracy rate is high [25].

(ii) **Defense against malware injection attacks:** Design flaws in protocol level are the reason for the server-side malware injection attack. On the other hand, design flaws in code level are the reason for the device-side malware injection attack. To defend XML external entities, SQL injection S. K. Lala

et al. implemented a security mechanism using NodeJS [26]. S. Gupta et al. proposed an injection-based design by injecting context-sensitive sanitization to detect cross-site scripting (XSS) attacks [27].
For the device-side malware injection attack, S. Weiser et al. proposed a memory architecture named TIMBER-V, to isolate the sensitive code and data from other insensitive data [28]. B. Schmerl et al. implemented a prototype for identifying the malignant uses of Android APIs that are threatening using both static analysis element and run-time adaptation element [29].

(iii) **Defense against authentication and authorization attacks**: The main cause of dictionary attacks is weak and easy passwords. By giving a strong and difficult password the problem can be solved. But in edge computing, the computation power is limited so the complicating password can create problems. Moreover, edge computing has so many subscribers that's why difficult password overloads the storage of edge servers.
Enhancing the security of a communication protocol or securing the cryptographic factors can be a way to defend against the weak authentication protocol attack. S. Sivakorn et al. proposed a black-box testing framework based on automata learning algorithms to analyze SSL/TSL hostname verification implementations [30].
Implementing a secure authorization in a system is as important as authentication. R. Yang et al. proposed a static code analysis method based on three OAuth identity providers including Google, Facebook, and Sina for checking and fixing the OAuth implementation vulnerabilities [31]. H. Kim et al. proposed an approach called secure migration, by which an IoT device can migrate to a secured edge computer for authorization when its local authorization system is inoperative [32].
IoT and mobile devices mainly face the issues of overprivileged attacks. Z. B. Celik et al. implemented a static taint analysis tool (SAINT) to trace sensitive information flow and stop it from leakage [33].

(iv) **Defense against side-channel attacks**: The main reason for side-channel attacks is the hidden correlation of the sensitive data that is publicly accessible with private data. By data perturbation, side-channel attack might come to an end. Data perturbation is a process by which all the data of a database is altered with different data for the privacy and security of sensitive data. M. A. P. Chamikara et al. developed a reliable and efficient algorithm named P2RoCal for high stream data perturbation [34].

There is another way to stop the side-channel attack that is restricting the access of the side-channel information. By doing this, unauthorized users would not get access to sensitive information. T. Zhang et al. presented a system named CloudRadar, to detect and reduce cache-based side-channel attacks embedded in cloud systems [35].

5 Security Challenges

In edge computing, researchers have found so many edge computing threats and they have also developed defense mechanisms to defend the security threats. Despite all the security defense mechanisms, there remain some issues that are not fixed yet discussed in this section.

(i) **Programmability**: In cloud computing, the users program their code and offload it in the cloud. Then the programs are executed from cloud in a centralized way. The user does not know anything about this mechanism so it is protected. But in edge computing, there are different kinds of devices hence different power consumption and different speed. So, the programmers struggle to develop a program in this heterogeneous platform and thus the offloading mechanism is more open to the users than the cloud computing which is a matter of concern [36].

(ii) **Design of Security**: Edge computing is evolved to give a lightweight platform for more efficient computing in less time for a huge number of applications. Being more concerned about the performance of edge computing, the researchers gave less attention to the security issues of edge computing. The architecture of edge computing is exposed to the attackers in consequence the security becomes vulnerable [4].

(iii) **Isolated Defense Mechanism**: Previously, we have shown some edge computing security threats as well as their defense mechanisms. But the defense mechanisms work in an isolated manner hence one defense mechanism may defend one or few security threats still the other security threats are exposed to the attackers [4].

(iv) **Heterogeneity**: In edge computing heterogeneous devices are used and thus it needs heterogeneous security frameworks for various Software and OSes, diverse network protocols, etc. One security framework may be feasible for one or a few situations but it will not cover all the situations in edge computing.

6 Conclusion

With the exponential growth of IoT devices, it becomes very challenging to manage millions of devices and their required resources such as low latency, lower bandwidth utilization, low traffic, and less expense. Compared with the traditional and centralized cloud computing architecture, edge computing replaces data processing into the edge of networks resulting in lower latency and low bandwidth utilization for real-time IoT applications. Also, the transmission cost is reduced by bringing the computation into the edge nodes and near the end-users. Instead of having these advantages, the security and privacy of data produced by IoT and mobile devices have become an open concern. In this paper, we have presented a thorough review covering the features, architecture, and applications of edge computing. The review also includes the most recent edge security threats, mechanisms, and some open challenges. These findings can be a guideline for the network designers or engineers to work in the dynamic research area of edge computing.

References

[1] Cisco Annual Internet Report (2018–2023), https://www.cisco.com/c/en/us/ solutions/collateral/executive-perspectives/annual-internet-report/whitepaper-C11-741490.html .

[2] Choubey, R. and Dubey, R. (2011). A survey on cloud computing security, challenges and threats. *International Journal on Computer Science and Engineering*, 3(3): 1227-1231 .

[3] Satyanarayanan, M. (2015). A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets. *GetMobile: Mobile Computing and Communications*, 18(4): 19-23.

[4] Xiao, Y. et al. (2019). Edge computing security: State of the art and challenges. *Proceedings of the IEEE*, 107(8): 1608-1631.

[5] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1): 30-39.

[6] Shi, W. et al. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5): 637-646.

[7] Hassan, N. et al. (2018). The role of edge computing in internet of things. *IEEE communications magazine*, 56(11): 110-115.

[8] Liu, M. et al. (2018). Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. *IEEE Transactions on Wireless Communications*, 18(1): 695-708.

[9] Satyanarayanan, M. et al. (2009). The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4): 14-23.

[10] Hu, Y. C. et al.(2015). Mobile edge computing—A key technology towards 5G. *ETSI white paper*, 11(11): 1-16.

[11] Abbas, N. et al. (2017). Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1): 450-465.

2021. In Rahul Srivastava & Aditya Kumar Singh Pundir (eds.), *New Frontiers in Communication and Intelligent Systems*, 271–290. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-95502-00-4-29

[12] Raj, P. H., Kumar, P. R. and Jelciana, P. (2016). Mobile cloud computing: a survey on challenges and issues. *International Journal of Computer Science and Information Security*, 14(12): 165-170.

[13] Al_Janabi, S. and Hussein, N. Y. (2019). The reality and future of the secure mobile cloud computing (SMCC): survey. *In International Conference on big data and networks technologies*, (pp. 231-261) . Springer, Cham.

[14] Mouradian, C. et al. (2017). A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE communications surveys & tutorials*, 20(1): 416-464.

[15] Naha, R. K. et al. (2018). Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE access*, 6: 47980-48009.

[16] Guan, Y. et al. (2018). Data security and privacy in fog computing. *IEEE Network*, 32(5): 106-111.

[17] Bhardwaj, K., Miranda, J. C. and Gavrilovska, A. (2018). Towards IoT-DDoS prevention using edge computing. *In USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*.

[18] Liu, Z., Yin, X. and Hu, Y. (2020). CPSS LR-DDoS detection and defense in edge computing utilizing DCNN Q-learning. *IEEE Access*, 8: 42120-42130.

[19] Dao, N. N. et al. (2018). MAEC-X: DDoS prevention leveraging multi-access edge computing. *In 2018 International Conference on Information Networking (ICOIN)*, (pp. 245-248). IEEE.

[20] Sangodoyin, A. O. et al. (2021). Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning. *IEEE Access*, 9: 122495-122508.

[21] Zuin, N. K. and Selvarajah, V. (2021). A Case Study: SYN Flood Attack Launched Through Metasploit. *In 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)*, 520-525.

[22] Moriuchi, P. and Chohan, S. (2018). Mirai-variant iot botnet used to target financial sector in january 2018. *Recorded Future Cyber Threat Analysis Report.*

[23] Li, X. et al. (2020). Smart Applications in Edge Computing: Overview on Authentication and Data Security. *IEEE Internet of Things Journal*, 8(6): 4063-4080.

[24] Goller, G. and Sigl, G. (2015). Side channel attacks on smartphones and embedded devices using standard radio equipment. *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 255-270.

[25] Chua, Z. L. et al. (2017). Neural nets can learn function type signatures from binaries. *In 26th USENIX Security Symposium (USENIX Security 17)*, 99-116.

[26] Lala, S. K., Kumar, A. and Subbulakshmi, T. (2021). Secure Web development using OWASP Guidelines. *In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 323-332.

[27] Gupta, S. and Gupta, B. B. (2018). A robust server-side javascript feature injection-based design for JSP web applications against XSS vulnerabilities. *In Cyber Security*, 459-465.

[28] Weiser, S. et al. (2019). TIMBER-V: Tag-Isolated Memory Bringing Fine-grained Enclaves to RISC-V. *In NDSS*.

[29] Schmerl, B. et al. (2016). Raindroid–A System for Run-time Mitigation of Android Intent Vulnerabilities.

[30] Sivakorn, S. et al. (2017). HVLearn: Automated black-box analysis of hostname verification in SSL/TLS implementations. *In 2017 IEEE Symposium on Security and Privacy (SP)*, 521-538.

[31] Yang, R., Lau, W. C. and Shi, S. (2017). Breaking and fixing mobile app authentication with OAuth2. 0-based protocols. *In International Conference on Applied Cryptography and Network Security*, 313-335.

[32] Kim, H. et al. (2020). Resilient authentication and authorization for the Internet of Things (IoT) using edge computing. *ACM Transactions on Internet of Things*, 1(1): 1-27.

[33] Celik, Z. B. et al. (2018). Sensitive information tracking in commodity IoT. *In 27th USENIX Security Symposium (USENIX Security 18)*, 1687-1704.

[34] Chamikara, M. A. P. et al. (2018). Efficient data perturbation for privacy preserving and accurate data stream mining. *Pervasive and Mobile Computing*, 48: 1-19.

[35] Zhang, T., Zhang, Y. and Lee, R. B. (2016). Cloudradar: A real-time side-channel attack detection system in clouds. *In International Symposium on Research in Attacks, Intrusions, and Defenses*, 118-140.

[36] Shi, W. et al. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5): 637-646.