Cihan University-Erbil Scientific Journal (CUESJ)

Research Article

Distributed Denial of Service Attacks on Cloud Computing Environment: A Comprehensive Review

Israa T. Aziz[1], Ihsan H. Abdulqadder[2], Thakwan A. Jawad[3]

[1]Computer Center, University of Mosul, Mosul, Iraq, [2]Department of Informatics and Software Engineering, Cihan University-Erbil, Kurdistan Region, Iraq, [3]Department of System and Control Engineering, College of Electronic Engineering, Ninevah University, Mosul 41002, Iraq

ABSTRACT

The incorporation of numerous technologies into the traditional network makes it difficult to complete new demands such as security. The widespread conception of telecommunication technologies in the last decade has an increased in the number of more attractive security threats. However, the new technology also created many new security concerns, and the threat of distributed denial of service attacks (DDoS) attack is one of the major concerns. This paper presents a comprehensive review of state-of-the-art techniques to detect DDoS attacks. It aimed to describe various kinds of DDoS attacks such as peer-to-peer based, Web-based, and internet relay chat-based. In addition, the critical challenges against effective DDoS defense mechanisms are also explored. Finally, the limitations of several DDoS attack detection approaches are highlighted.

Keywords: Distributed denial of service attacks, cloud computing environment, peer-to-peer, internet relay chat

INTRODUCTION

Cloud computing has developed rapidly over recent years, and it is reshaping the information technology (IT) industry. The apparent transparency, large volumes of stored data, and comparable easiness of operability of cloud computing environment (CCE) make it vulnerable and easy targets for several kinds of predatory attacks, distributed denial of service (DDoS) being major ones like those against Cloudflare and Spanhaus which are alarmingly and increasingly used for exploiting sample network management protocol (SNMP). Some of the major DDoS attacks are flooding, spoofing, user to root, port scanning, oversized XML, coercive parsing, reflective attacks, and so forth. The DDoS attacks have been reported to have significantly high chances of occurrence, first, because the tools of launching DDoS are widely available, and second, because of the apparent lack of effective, timely mechanisms to defend against them. Due to the frequency of attacks such as DDoS, CCE's fullest gains and advantages stand highly compromised and negated. There is the need for in-depth, evidential, and research-validated studies on the topic and its many ramifications over time, DDoS has evolved, and remedial actions prompted to address them and offer permanent solutions. The statement of the problem is the exact degree and magnitude of destructive powers of DDoS cannot be quickly quantifiable, CCE could best address these issues and find solutions are also needed. The DDoS attacks on CCE hold immense potential to cause significant damages and detriment, especially if uncontrolled and unremedied.

In this paper, discussion of various studies carried out concerning CCEs is presented with more emphasis is given on the studies about DDoS is done. Authors in Kobusińska *et al.*[1] aimed to determine cloud computing's concerns and challenges, whereby they discussed grid computing and service-oriented computing, which are two related computing paradigms. They also sought to pinpoint the relationships of these two computing models with cloud computing and identified various challenges therein. On the other hand, Tahirkheli *et al.*[2] pointed out two primary reasons that complicate the assessment of the security impact of cloud computing. According to Tahirkheli *et al.*,[2] the current discourse on cloud computing security issues uses the terms "threat," "risk," and "vulnerability," interchangeably, while they have their respective definitions.

Moreover, the authors claim that not every issue that's raised is particular to cloud computing. In order to build a

Corresponding Author:
Israa T. Aziz, Computer Center, University of Mosul, Mosul, Iraq.
E-mail: israa_aziz@uomosul.edu.iq

Received: November 12, 2021
Accepted: January 27, 2022
Published: March 30, 2022

DOI: 10.24086/cuesj.v6n1y2022.pp47-52

broader perspective of the threats within the CCE, they delved into the meaning of CCE. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services). It can be rapidly provisioned and released with minimal management effort or service provider interaction, according to the US National Institute of Standards and Technology.[3] Within the terms used to define cloud computing and, by extension, CCE, it is clear that the environment is set on on-demand self-service, implying that consumers' resources automatically come into play, which, in itself result in fewer human interactions with the provider of the services.[4] The computing resources, in this case, include CPU time, software use, and network storage, which are engaged on a convenient, self-serve basis. Second, the term "Broad network access," as used in the definition, postulates a scenario whereby various clients' applications can use with diverse platforms access computing resources conveyed over networks.[5] Common user platforms include personal digital assistants, laptops, and mobile phones, which translates to increased exposure probability heightened vulnerability. Third, CCE's resource pooling aspect translates to the multi-tenancy or virtualization models.[6] These models make it possible for CCEs to serve multiple consumers, with the computing resources assignment varying now and then.

Rapid elasticity and measured service are the other terms in the definition of CCEs. Rapid elasticity pertains to the immediate availability of computing resources instead of persistent. The outstanding issues with such a setting are that consumption can increase rapidly during peak requirement times. On the other hand, measured service brings the CCEs metering individual consumers' usage even though the computing resources are combined and shared by numerous clients. In Mthunzi *et al.*,[7] the authors analyzed the cloud computing influences each risk factor in a pool of various risks in computing. They also assessed the vulnerability issue while singling out cloud computing from the general computing domain. Therefore, they asserted that cloud computing enhances some well-understood vulnerabilities and introduces new ones.

Akherfi *et al.*[8] argues that the distinctions between cloud computing, service-oriented computing, and grid computing should be investigated. As a result, they developed research topics to articulate future study problems and cloud computing objectives. More importantly, they sought to establish the benefits of the three paradigms within the context of their coexistence. The authors explain in the definition of CCE to point out the risks in cloud computing; they should have depicted the strengths associated with the terms used in the definition. Their dwelling on the vulnerability shows their biasness in their assessment. In Al-Dhuraibi *et al.*,[9] the authors investigate cloud computing's advantages, and they offer the future development issues on CCEs, among them, the opportunity to reduce the vulnerability of CCEs. However, the inherent nature of the technology used in the CCE implementation exposes vulnerabilities, including virtual machine escape, session riding and hijacking, and insecure or outdated cryptography. In addition to the intrinsic technologically based vulnerability, there are cloud-specific vulnerabilities.[10] Such vulnerabilities can be associated with the very nature of the cloud computing setting, thus implanted in the definition of CCE.

According to Chahal *et al.*,[11] the tools for launching DDoS are widely available. The author further posits a lack of an effective mechanism that defends DDoS attacks in a reasonable amount of time. After analyzing the various techniques available to reduce DDoS attacks, they concluded that none of the studied methods met their requirement. Their research sets a significant basis on the threat posed by DDoS to the CCEs. Their further assertion, based on their literature review, that DDoS attacks occur more frequently than reported, is a further ramification of DDoS.

Nevertheless, DDoS remain a daunting challenge in cloud computing environments, and it requires complex simulations to cope with it. Typically, DDoS flood traffic into servers, systems, or networks to overwhelm the victim resources, making it difficult for legitimate users to use them. Furthermore, the attacks are directed from multiple attack systems (distributed), thus complicating their detection and defending.

In Rossow *et al.*,[12] the authors closely analyzed twelve peer-to-peer (P2P) botnet variants still active. As shown in Table 1, each P2P botnet has its communication protocol, message propagation technique, communication direction, command-and-control (C&C) architecture, and purpose.

The major goal is written in capital letters. S = Spam, T = Credential Theft, C = Click Fraud, D = DDoS, M = Bitcoin Mining, N = Network Services=Pay-Per-Install Rossow *et al.*[12] adapted Table 1. Unlike P2P and internet relay chat (IRC) botnets, hypertext transfer protocol (HTTP) botnets employ a wide range of HTTP services, making it difficult to prevent them. Attacking with a Trojan allows the attacker to download a zombie agent, or the Trojan itself may contain one. The number of Trojan attacks has increased, with most of them aimed at web servers. Automated tools that exploit flaws in programs that listen for connections from remote hosts are also being used by attackers to access systems. Computer security experts and law enforcement agencies can easily take down centralized botnets. As a result, botnet operators have looked for novel ways to fortify their botnets' infrastructures. Some botnet operators have reconfigured their botnets to employ P2P infrastructures to achieve this goal. Because they have no single point of failure, many P2P botnets are significantly more resistant to takedown attempts than centralized botnets. P2P botnets, on the other hand, are vulnerable to specific

Table 1: Comparison and Overview of P2P botnet

Family	Protocol	Prop.	Dir.	C&C	Propose
Kelihos	Custom	Gossip	Pull	Hybrid	C,D,M,N,S
Miner	Custom	Gossip	Pull	Hybrid	D,M,P
Nugache	Custom	Gossip	Pull	P2P	D,T
Sality	Custom	Gossip	Pull	P2P	D,N,P
Storm	Overnet	Routing	Pull	Hybrid	D,S,T
Waledac	Custom	Gossip	Pull	Hybrid	D,S,T
ZeroAccess	Custom	Gossip	Pull	P2P	P
Zeus	Custom	Gossip	Both	Hybrid	D,P,T

types of assaults, such as node enumeration and poisoning.[13] The important outlines of this paper are providing a review for specific DDoS attacks on CCE, as well as the kind of detriment could cause. While the chief aims of this paper are to identify the various kinds of DDoS attacks, their destructive capabilities, best these issues could be counter-attacked and resolved for the benefit of all stakeholders along the cloud continuum, preferably as permanent solutions. This paper's structure is as follows: Section 2 offered the analysis of specific DDoS attacks. Section 3 explained the remedies for DDoS were offered previously. Section 4 concludes this paper.

ANALYSIS OF SPECIFIC DDOS

This section offers specific DDoS attacks on CCE based on published reports and other credible sources. Therefore, this paper will use reports on the nature and frequency of occurrence, the extent of detriment, and reports on efforts to deal with DDoS. The first step is to analyze the general architecture of DDoS, which is shown in Figure 1.

DDoS attacks are known to originate from multiple sources or multiple locations simultaneously, as evidenced by their general architecture. The participants in a DDoS attack are also visible.[14] Second, the importance of DDoS issues can be ascribed to (1) numerous attack machines can create more attack traffic than a single attack machine, (2) several attack machines are more difficult to turn off than a single attack machine, and (3) each attack machine's behavior can be stealthier, making it difficult to detect and shut down. Therefore, the key challenge from DDoS is the complexity of having a defense mechanism against them. Specific DDoS, based on their facilitation mechanism, plus the significance of the detriment they could cause, is as discussed here:

Botnet-Based DDoS

A botnet is a network of zombie computers that have been designed to accept commands without the owner's knowledge.[15] The infrastructure and tools currently evident in botnet launching areas are presented. Notably, the critical challenges against effective DDoS defense mechanisms are twofold: (1) To initiate DDoS flooding attacks, a large number of Zombies are used, and (2) Zombies IP addresses are usually faked under the attacker's control. Thus, the attacker's possible to add more attack machines dwindles the clients' ability to purchase more incoming bandwidth, eventually crashing a website completely over time.[16] An attacker (Master) controls a group of zombies, forming a botnet. Thus, botnets consist of masters, handlers, and agents (bots) whereby the master communicates to the bots through the handlers. Three categories of botnets are P2P-based, Web-based, and IRC-based.

P2P-Based

These are affected through a decentralized group of malware-compromised machines working in cooperation with an attacker without their owners' knowledge.[17] They, therefore, lack a C&C server. These botnets avoid detection and make it hard for security researchers to access communication by being decentralized. Moreover, investigators cannot take down an entire network when they detect a single bot since the P2P botnets lack a command-and-control server. While bots keep communicating, the botmaster retains commanding them, usually by digital signing. Mainly, digital signing is achieved through asymmetrical encryption, which requires two keys (public and private).[18] This ensures that while one key is used to encrypt a message, the message can only be decrypted with the other key. The private key is thus kept private, while the public key is embedded in the bot. The master encrypts commands with the private key, while the bots decrypt them with the public key. Bots that can accept incoming connections are called servers, while bots that cannot take incoming connections are called workers. Nodes or peers are the terms used to describe the servers. To receive commands, the workers must connect to one or more peers. Workers are distributed among several nodes to avoid being taken down, and they can relocate to another node if one is taken down. Since this infrastructure entails bootstrapping, which offers some sort of signing; these botnets are able to prevent themselves from being hijacked by security developers.

Still, the bootstraps are crucial nodes, and if all of the bootstrap servers were taken down simultaneously, it would have no effect on the bots currently in the botnet, but it would prevent new infections from happening to join. While such

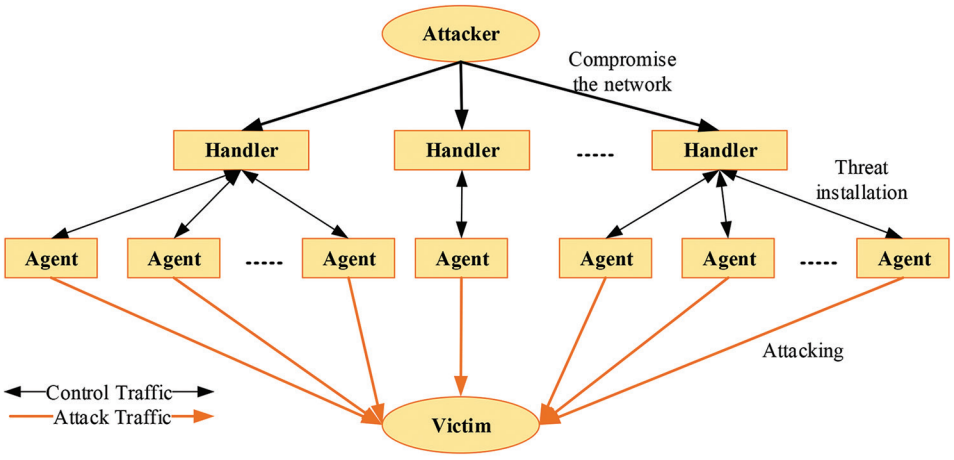

Figure 1: The general architecture of DDoS

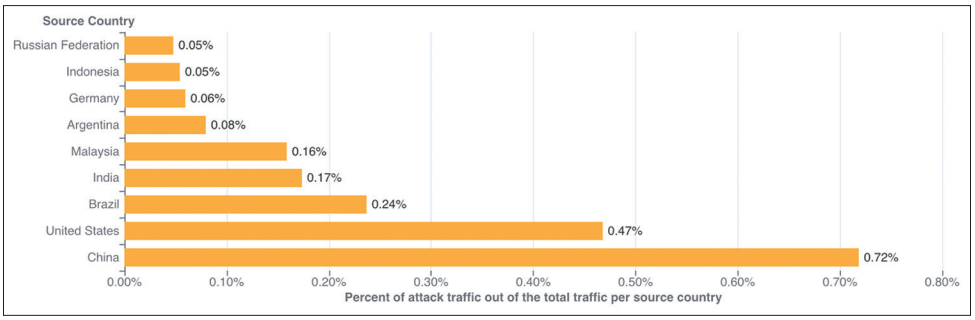

Figure 2: HTTP DDoS trends of prevalent based on country

a safeguard would be tough to implement, it would also be temporary, as the botmaster would temporarily stop infecting new computers while setting up new bootstrap servers. P2P bots include Sinit, Phabot, Spam thru, Nugache, and Peacomm (Storm).

Web-Based

These are more recent botnets. In this setup, the commands from the botnets to the bots are communicated through the HTTP communication protocol, making it challenging to track the control structure and the DDoS command. Such botnets conceal themselves within legitimate HTTP traffic; thus, they are stealthier.[19] Using the web requests, each web bots downloads instructions periodically as shown in Figure 2.

Although it might be expected that HTTP botnets would try to blend into regular HTTP traffic (since they use HTML), a significant number of them use specially registered domain names for their purpose.[20] A proof of this claim can be shown by Team-Cymru.com, which studied some identified HTTP C&Cs. 46% of HTTP botnets used a dedicated domain name, 26% used free hosting such as funpic.de and freehostia.com. Based on location, for the 131 unique addresses used, for the one day, 63 of the IP addresses were located in the United States, 11 were in Russia, and ten were in China.[21] An example of a web-based DDoS botnet is the Stacheldraht attack. These DDoS tool invaders employ a layered structure and link to handlers through a client program. The client program infiltrates the handler's systems and sends commands to the zombie agents, allowing the DDoS assault. Different types of internet packets, such as internet control message protocol (ICMP), user datagram protocol (UDP), and transmission control protocol (TCP), can be used by Stacheldraht. Each handler can control up to a thousand agents in some instances. The situation can even be made worse because the client can even consent to the DDoS attack to become part of their machine. For example, the group Anonymous used Stacheldraht called Operation Payback. Some examples of web-based bots include BlackEnergy, Kraken, Pushdo, Cyber bot, and BlackSun Rat.[22]

IRC-Based Botnets

This setup uses an online text-based instant messaging protocol to interact between servers and features a client/server structure. As a result, attackers use IRC channels as handlers to deliver commands to bots utilizing legal IRC ports. As a result, tracking the IRC DDoS's command-and-control system is challenging. Furthermore, because IRC is a stand-alone software or set of scripts that connect to IRC as a client, attackers do not need to keep bots on their site because they may access a list of all available bots by signing on to IRC servers.[23]

Although IRC botnets have historical operations such as Game Manager in games, bartenders, and Puppe, these bots have evolved over time to provide special services, including access to databases, maintaining access lists, and managing channels. Still, the usual large volume common to IRC servers makes it easy for the attacker to hide.[24] Defense against these DDoS can be affected by the defender capturing their centralized command and control servers. IRC botnets include SDbot, Agobot, GDbot, and Spybot.[25] Analyzing online social sites threats reveals that IRC botnets pose a significant threat. In fact, according to unpublished sources such as blogs on IRC sites, it is evident that IRC has continued to improve, which poses even more threat as they are building armies of bots for large-scale attacks. The worm Dorkbot, also known as Nrgbot, is one of the most common IRC botnets, capable of stealing passwords, blocking security updates, downloading more malware, and even conducting DDoS assaults using infected devices. "The bot joins a specific IRC channel on an IRC server and waits there for further command. This allows an attacker to remotely control this bot and use it for fun and profit. The communication between bots and their controllers is somewhat bloated; a more straightforward communication protocol would suffice. But IRC offers several advantages: IRC Servers are freely available and are easy to set up, and many attackers have years of IRC communication experience.[26]

REMEDIES AND DISCUSSION

In this section, we analyze the remedies for DDoS offered in the previous area by looking at the reported deliberations and analyzing their success. This paper seeks to set the way towards successful contemplation on DDoS. In general, defense tactics comprise a combination of attack detection, traffic classification, and response tools to block illicit traffic while allowing valid traffic.[27]

Application-Level Key Completion Indicators

Application-level DDoS attacks on CCEs can be based on an application layer analysis to identify whether incoming traffic packets are desirable or not, allowing elasticity decisions to be activated without the negative economic consequences of

a DDoS attack.[28] Upstream filtering with upstream filtering, as the name suggests, the traffic to the victim computer needs to be cleaned by passing it through digital cross-connects, proxies, direct circuits, or proxies. Such upstream cleaning ensures that DDoS is separated and only the "safe" traffic reaches the server.[29]

Applying Front End Hardware

Using front-end hardware, intelligent hardware can be deployed on the network before traffic reaches the servers. As data packets enter the system, this hardware examines them and classifies them as a priority, regular, or dangerous.

IPS-Based Prevention

Intrusion prevention systems (IPS) can be effective if the attack has signatures linked with it. Unfortunately, genuine content with malicious intent is a regular DDoS trend. Furthermore, intrusion-prevention systems designed for content recognition cannot detect DDoS attacks based on behavior. A rate-based intrusion prevention system (RBIPS) is designed to analyze traffic granularly and continuously scan the traffic array for anomalies.[30] The legitimate traffic must be let pass through while the suspicious one is filtered off.

Switches

Many switches include rate-limiting and access control list (ACL) capabilities. To identify and correct DoS assaults, some switches include deep packet inspection, delayed binding (TCP splicing), Bogon filtering, automatic traffic shaping, bogus internet protocol (IP) filtering, and system-wide rate restriction via wide area network (WAN) link failover and balancing and automatic rate filtering.[31] Switch treatments only if DDoS attacks can be avoided while they are used. The synchronize (SYN) flood attack, for example, can be mitigated by employing TCP splicing or delayed binding. Still, bogon filtering can be used for attacks going to dark addresses or coming from dark addresses, while deep packet inspection can be used for content-based DDoS. In addition, if both links have DoS/DDoS prevention mechanism, Wan-link failover will work. And more still, setting the rate thresholds correctly can ensure that the automatic filtering works.

Blackholing and Sinkhole

All communication to the attacked domain name system (DNS) or IP address is routed through a "black hole," a non-existent server, or a null interface. The internet service provider can manage such a hole to be more efficacious.[32] Moreover, a DNS sinkhole can route traffic to a valid IP address for analysis, rejecting the bad packets and allowing the good packets through.

Firewalls

With firewalls, a simple rule could be added, such that, based on the protocols, originating IP addresses, or ports, all incoming threat traffic is denied access.[33] However, this approach will not be able to block more complex attacks. For instance, web service; say through port 80, on experiencing an attack, it might not be possible to deny all incoming traffic through such a service port, as that would mean stopping the legitimate server's traffic.[34]

Routers

Routers and switches are similar in that they both include rate-limiting and ACL capabilities. They, like the switches, are also configured manually. However, it is essential to note that most routers are vulnerable to DDoS attacks.

P2P Remedies

Almost all P2P botnets have a vulnerability in the P2P sharing mechanism. As previously stated, the nodes must keep and share a list of other nodes with the workers in order to distribute the workers around the huge number of nodes.[35] Providing each node with a list of other nodes would take an excessive amount of time, if not impossible, for the botmaster to do manually.[36] As a result, the nodes do it on their own. If a new node is found to accept connections, the node to which it is linked adds it to its node list, which it then shares with other nodes. This could allow security researchers to inject hostile computers into the network through a loophole. They can introduce employees and other nodes with bogus IP addresses in this situation and share a list of other malicious nodes. If this is done and enough resources are put in place, the new rogue nodes can become a significant botnet component, thereby separating genuine nodes from workers. If the inserted malicious nodes populate the network, the workers will identify with the nodes and will have a much lower likelihood of joining the botnet.

CONCLUSION

DDoS remain a daunting challenge in cloud computing environments, and it requires complex simulations to cope with it. Typically, DDoS flood traffic into servers, systems, or networks to overwhelm the victim resources, making it difficult for legitimate users to use them. Furthermore, the attacks are directed from multiple attack systems (distributed), thus complicating their detection and defending against. IRC botnets pose a significant threat as they have continued to improve, which poses even more risk as they build armies of bots for large-scale attacks, the key challenges against effective DDoS defense mechanisms are two-fold: (1) A large number of Zombies is employed to launch DDoS flooding attacks and (2) Zombies IP address is typically spoofed under the control of the attacker. Thus, the attacker's possibility to add more attack machines dwindles the clients' ability to purchase more incoming bandwidth. Some remedies for DDoS include Application-level key completion indicators, upstream filtering, applying front-end hardware, IPS-based prevention, using switches and Firewalls, and Blackholing and sinkholing. Since DDoS keeps evolving and becoming more sophisticated, this field of their remedies requires more research and development to stay at par with their development, if not to be ahead of them.

In the future, this work is planned to be extended with the use of the software-defined network (SDN) and Networking function Virtualization (NFV) in a cloud environment to come up with new optimization algorithms to Solve DDoS attacks.

REFERENCES

1. A. Kobusińska, C. Leung, C. H. Hsu, S. Raghavendra and V. Chang. Emerging trends, issues and challenges in internet of things, big data and cloud computing. *Future Generation Computer Systems,* vol. 87, pp. 416-419, 2018.
2. A. I. Tahirkheli, M. Shiraz, B. Hayat, M. Idrees, A. Sajid, R. Ullah, N. Ayub and K. I. Kim. A survey on modern cloud computing security over smart city networks: Threats, vulnerabilities, consequences, countermeasures, and challenges. *Electronics,* vol. 10, p. 1811, 2021.
3. H. Tabrizchi and M. K. Rafsanjani. A survey on security challenges in cloud computing: Issues, threats, and solutions. *The Journal of suPERCOMPUTIng,* vol. 76, pp. 9493-9532, 2020.
4. Q. Han, W. Yu, Y. Zhang and Z. Zhao. Modeling and evaluating of typical advanced peer-to-peer botnet. *Performance Evaluation,* vol. 72, pp. 1-15, 2014.
5. L. Feng, H. Wang, Q. Han, Q. Zhao and L. Song. Modeling peer-to-peer botnet on scale-free network. In *Abstract and Applied Analysis*, vol. 2014, p. 212478, 2014.
6. P. Kumar and R. Kumar. Issues and challenges of load balancing techniques in cloud computing: A survey. *ACM Computing Surveys,* vol. 51, pp. 1-35, 2019.
7. S. N. Mthunzi, E. Benkhelifa, T. Bosakowski, C. G. Guegan and M. Barhamgi. Cloud computing security taxonomy: From an atomistic to a holistic view. *Future Generation Computer Systems,* vol. 107, pp. 620-644, 2020.
8. K. Akherfi, M. Gerndt and H. Harroud. Mobile cloud computing for computation offloading: Issues and challenges. *Applied Computing and Informatics,* vol. 14, pp. 1-16, 2018.
9. Y. Al-Dhuraibi, F. Paraiso, N. Djarallah and P. Merle. Elasticity in cloud computing: State of the art and research challenges. *IEEE Transactions on Services Computing,* vol. 11, pp. 430-447, 2017.
10. F. Shahzad. State-of-the-art survey on cloud computing security challenges, approaches and solutions. *Procedia Computer Science,* vol. 37, pp. 357-362, 2014.
11. J. Kaur Chahal, A. Bhandari and S. Behal. Distributed denial of service attacks: A threat or challenge. *New Review of Information Networking,* vol. 24, pp. 31-103, 2019.
12. C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C. J. Dietrich and Bos, H. Sok: P2pwned-modeling and evaluating the resilience of peer-to-peer botnets. In: *2013 IEEE Symposium on Security and Privacy*, pp. 97-111, 2013.
13. Y. S. Almashhadani and G. A. Qasmarrogy. Ad hoc on-demand distance vector inherent techniques comparison for detecting and eliminating the black hole attack nodes in mobile ad hoc network. *Cihan University-Erbil Scientific Journal,* vol. 4, pp. 77-81, 2020.
14. Y. Gao, Y. Ma and D. Li. Anomaly detection of malicious users' behaviors for web applications based on web logs. In: *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pp. 1352-1355, 2017.
15. P. Narang, S. Ray, C. Hota and V. Venkatakrishnan. Peershark: Detecting peer-to-peer botnets by tracking conversations. In: *2014 IEEE Security and Privacy Workshops*, pp. 108-115, 2014.
16. V. Hamon. Android botnets for multi-targeted attacks. *Journal of Computer Virology and Hacking Techniques,* vol. 11, pp. 193-202, 2015.
17. D. Zhuang and J. M. Chang. Enhanced peerhunter: Detecting peer-to-peer botnets through network-flow level community behavior analysis. *IEEE Transactions on Information Forensics and Security,* vol. 14, pp. 1485-1500, 2018.
18. Q. Yan, Y. Zheng, T. Jiang, W. Lou and Y. T. Hou. Peerclean: Unveiling peer-to-peer botnets through dynamic group behavior analysis. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 316-324, 2015.
19. Q. Shafi and A. Basit. DDoS botnet prevention using blockchain in software defined internet of things. In: *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 624-628, 2019.
20. I. T. Aziz, H. Jin, I. H. Abdulqadder, Z. A. Hussien, Z. A. Abduljabbar and F. M. Flaih. A lightweight scheme to authenticate and secure the communication in smart grids. *Applied Sciences,* vol. 8, p. 1508, 2018.
21. Team Cymru. *Team Cymru Services*. United States: Team Cymru, 2018.
22. A. Karim, S. A. A. Shah and R. Salleh. Mobile botnet attacks: A thematic taxonomy. In: *New Perspectives in Information Systems and Technologies.* Vol. 2. Berlin, Germany: Springer, pp. 153-164, 2014.
23. A. Houmansadr and N. Borisov. BotMosaic: Collaborative network watermark for the detection of IRC-based botnets. *Journal of Systems and Software,* vol. 86, pp. 707-715, 2013.
24. I. T. Aziz, H. Jin, I. H. Abdulqadder, S. M. Alturfi, W. H. Alobaidi and F. M. Flaih. T2S2G: A novel two-tier secure smart grid architecture to protect network measurements. *Energies,* vol. 12, p. 2555, 2019.
25. S. Hosseini, A. E. Nezhad and H. Seilani. Botnet detection using negative selection algorithm, convolution neural network and classification methods. *Evolving Systems,* vol. 13, pp. 1-15, 2021.
26. J. M. Cruz, J. P. Dias and J. P. Pinto. A hands-on approach on botnets for behavior exploration. In: *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security*, 2017.
27. M. Aresu, D. Ariu, M. Ahmadi, D. Maiorca and G. Giacinto. Clustering android malware families by http traffic. In: *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 128-135, 2015.
28. P. Wainwright and H. Kettani. An analysis of botnet models. In: *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, pp. 116-121, 2019.
29. A. Zarras, A. Papadogiannakis, R. Gawlik and T. Holz. Automated generation of models for fast and precise detection of HTTP-based malware. In: *2014 Twelfth Annual International Conference on Privacy, Security and Trust*, pp. 249-256, 2014.
30. F. Haddadi, J. Morgan, E. Gomes Filho and A. N. Zincir-Heywood. Botnet behaviour analysis using ip flows: With http filters using classifiers. In: *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 7-12, 2014.
31. A. Welzel, C. Rossow and H. Bos. On measuring the impact of DDoS botnets. In: *Proceedings of the Seventh European Workshop on System Security*, pp. 1-6, 2014.
32. P. Amini, R. Azmi and M. A. Araghizadeh. Analysis of network traffic flows for centralized botnet detection. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* vol. 11, pp. 7-17, 2019.
33. A. K. Sood, S. Zeadally and R. J. Enbody. An empirical study of HTTP-based financial botnets. *IEEE Transactions on Dependable and Secure Computing,* vol. 13, pp. 236-251, 2014.
34. K. Neupane, R. Haddad and L. Chen. Next generation firewall for network security: A survey. in *SoutheastCon 2018*, pp. 1-6, 2018.
35. L. Böck, S. Karuppayah, K. Fong, M. Mühlhäuser and E. Vasilomanolakis. Poster: Challenges of accurately measuring churn in P2P botnets. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2661-2663, 2019.
36. M. Eslahi, M. Rohmad, H. Nilsaz, M. V. Naseri, N. Tahir and H. Hashim. Periodicity classification of HTTP traffic to detect HTTP Botnets. In: *2015 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, pp. 119-123, 2015.