DDoS Attacks against Cloud-Computing Environment

Subjects: Computer Science, Information Systems

Contributor: Ziyad R. Alashhab, Mohammed Anbar, Manmeet Mahinderjit Singh, Iznan H. Hasbullah, Prateek Jain, Taief Alaa Al-Amiedy

Cloud computing (CC) plays a significant role in revolutionizing the information and communication technology (ICT) industry, allowing flexible delivery of new services and computing resources at a fraction of the costs for end-users than traditional computing. Unfortunately, many potential cyber threats impact CC-deployed services due to the exploitation of CC's characteristics, such as resource sharing, elasticity, and multi-tenancy. Researchers provides a comprehensive discussion on security issues and challenges facing CC for cloud service providers and their users. Furthermore, a new taxonomy for classifying CC attacks is proposed, distributed denial of service (DDoS) attacks, and DDoS attack detection approaches on CC. Researchers also provides a qualitative comparison with the existing surveys. Finally, they aims to serve as a guide and reference for other researchers working on new DDoS attack detection approaches within the CC environment.

Keywords: cyber-threats ; cloud computing (CC) ; cloud security ; distributed denial of service (DDoS) attacks ; cybersecurity

1. Introduction

Cloud computing (CC) enables the flexible delivery of services and computing resources through the Internet, including data storage, computation, networking, and software resources on demand. The elastic feature of the CC allows the resources to be dynamically allocated whenever needed without a large investment in infrastructure and software licensing for the users [1].

However, the same feature that makes CC flexible is also responsible for exposing it to security threats. One of the most serious threats is distributed denial of service (DDoS) attacks. Unfortunately, the impact of DDoS attacks on CC has not been investigated enough although there is some research that provides an in-depth examination of the above-said issue, shedding light on HTTP flooding DDoS attacks in the CC environment and various DDoS attacks [2].

CC has significant advantages over traditional computing, including lower capital and operational expenditures (CapEx and OpEx) and the ability to deliver dynamic and extensible virtualized computing resources [3][4]. Many recent research and surveys anticipated a dramatic rise in CC adoption based on past trends. For example, in the past three years, 75 percent of business applications allowed direct access to the application programming interface (API) of their most critical applications [5]. In upcoming years, 90 percent of large commercial IT companies will focus on CC. However, researchers believe that the coronavirus (COVID-19) pandemic could increase that percentage further [2][6].

Internet usage has increased by 50% in several European countries and 30% in the United Kingdom, reflecting the growth in the ICT sector spurred by surges in demands for online-based utilities and services. As a result, more issues and threats with CC will surface in due time, such as security and privacy breaches, data storage issues in the CC, and application-layer attacks [7][8]. Nowadays, major attacks on CC are growing, and the effects on its security are becoming more apparent. For example, attacks include malicious attempts by adversaries to deny legitimate users from obtaining services offered by the targeted servers [9].

Cloud service providers (CSP) charge their clients based on the services rendered or resources utilized on a pay-per-use basis. A new form of DDoS attack, called Economic denial of sustainability (EDoS), manipulates the CC charging model to hurt the financial bottom line of the users and the CSPs [10] by targeting the resource utilization at the application layer. EDoS attack is an economic security threat on the CC [10], which differs from a typical DDoS attack. DDoS attacks aim for service availability by debilitating the server's resources until the server crash or are unresponsive; however, EDoS exploits the resource flexibility provided by the CC to force over-consumption of resources [11][12].

2. Taxonomy of the Attacks on CC

The CC is defenseless in the face of several attacks that provide serious security dangers. The threat of an attack is determined by the target of the adversary's attack. The CC service availability can be disrupted partially or entirely for a short or long period.

2.1. Suffocative Attacks

Suffocative attacks are those that are bandwidth-based or involve volumetric attacks. This attack overloads the targeted system with garbage data to consume the bandwidth of the targeted network system, flooding the network and affecting system resources. Its magnitude is measured in bits per second (Bps). Adversaries can launch this attack using UDP, ICMP flooding, or flooding the target with various spoofed packets [13].

2.2. Protocol Attacks

Protocol attacks consume real server resources and equipment used for networking communication, like load balancer (LB) devices or protection devices. The protocol attack is measured in packets per second (Pps). It exploits the weaknesses of network protocols to increase the burden on the victim's resources. Some examples of protocol attacks include a smurf attack, fragmented packet attack, SYN floods, and ping of death [14].

2.3. Economic Denial of Sustainability (EDoS)

A type of DDoS attack on CC that impacts the financial bottom line of victims is called an EDoS attack [15]. It is a malicious attack specific to the CC that focuses on impacting the CSP's OpEx more than the physical resources. It exploits the CC's elastic or auto-scaling characteristic by attacking targeted services, especially on the application layer, forcing maximum consumption of CC resources until the services become inaccessible. Avoiding this situation requires the CSP to keep providing additional resources to fulfill the SLA for the CCC accessibility, which increases the cost for the CSP [16][17], resulting in EDoS [2].

Adversaries could trigger DDoS attacks on CC networks or resources using legitimate service requests to generate EDoS attacks [17]. In other words, the basis of this financial damage came from DDoS attacks (the result of EDoS is DDoS attacks usually, and the opposite is also true) that exploited available resources to increase the bill. The bloated cost has to be borne by the CSP or passed along to their clients.

2.4. Permanent Denial-of-Service Attacks

Permanent denial-of-service (PDoS) or Plashing is a fast-moving attack designed to disable the victim's and prohibit it from providing services. This type of cyber attack, a strain of DDoS attacks with more emphasis on the victim's hardware, started to increase in frequency in 2017 as more occurrences involving this hardware-damaging attack were found [18][19]. Furthermore, PDoS aims to cause perpetual harm to network equipment via programming, especially configurable network hardware, such as routers. Although PDoS attacks are rare, successful attacks are highly damaging to CC resources to the point of requiring the replacement or re-installation of equipment. Unlike DDoS attacks that disable a service temporarily, PDoS causes permanent hardware damage. It exploits the CC's security flaws or misconfigurations in the remote administration function on the hardware management interface to alter the device's firmware with a faulty version, damaging the device to the extent that requires fixing or even destroying essential system functions. All CC resources, such as LB, firewalls, VM, physical servers, storage units, and processors, are vulnerable to PDoS attacks [20]. Moreover, since a PDoS attack focuses on the hardware, it requires far fewer resources than a DDoS attack. PDoS is more destructive and has been gaining popularity among adversaries. CSP encountering a PDoS attack will incur business loss since services will be affected, and it could take a very long-time fixing fault. For instance, in 2009, the federal bureau of investigation (FBI) raided DCs in Texas because of fraud against several organizations that worked from out of the DCs [21]. In another case, a significant information loss occurred to a CSP providing storage services in Magnolia after experiencing Omni drive failure, leading to its shutting down without notice in 2008.

A Help Net Security site ran a report [22] on a universal serial bus (USB) device that disables a machine just by putting it into the USB port. According to the report, the most recent PDoS USB attack works by injecting some electrical power through the machine with the help of a voltage transformer to release a flood of negative electricity into the USB port. An example of a PDoS USB device is PhlashDance, built by Rich Smith in his security lab in 2008 to exhibit the inner working of a PDoS attack [23].

3. Taxonomy of DDoS Attacks on CC

DDoS attacks on CC have rapidly risen to the top of most cyber security threats lists. Attacks on CC could affect not only the CSP but also CC resources, including VMs and the networks [24]. Nevertheless, whatever the motive of an adversary to carry out DDoS attacks, any deterioration in services offered to CCC decreases its value.

CERT experts (a variety of researchers) say most DDoS attacks against the CERT target the application layer. The vast majority of DDoS attacks use a tremendous amount of requests of a standard communication protocol, making them hard to detect. Furthermore, they typically use well-known patterns to imitate legitimate traffic to throw off detection attempts. Therefore, standard network security strategies are not well suited to detect or prevent such attacks.

3.1. SYN Floods

An adversary executes SYN flood attacks on CC by sending SYN requests using a spoofed source IP address, forcing the VM to respond and allocate the necessary resources to handle the requests. The VM waits for an acknowledgment from the 'sender' that never arrives. Continuous attacker's requests finally exhaust the VM of all its resources, such as memory and CPU [25]. Consequently, the VM is forced to reject all subsequent user requests, including the legitimate ones.

3.2. UDP Floods

In this attack, the network bandwidth of CC is fully exhausted, although no user exists. Adversary injects a huge number of UDP packets into the network [12].

3.3. Ping of Death

A ping of death (PoD) attack involves an adversary using unusually large packets to cripple the CC's VM or resources. The adversary changed the ping instruction by modifying the Fragment Offset field in the IP header to create a packet larger than the maximum permissible value for that field, which is 65,536 bytes. A ping packet with a size larger than the limit set by TCP/IP could overflow the buffer of the destination OS [26], affecting the victim's computer connected to the CC networks and influencing the CC services linked to those networks. However, nowadays, all modern network equipment and OS ignore 65,535-byte IP packets that may cause a crash or slowdown of the OS, making today's network and machines less susceptible to this attack.

3.4. ICMP Flood Attack

ICMP messages are used to locate hosts on a network, map network structure, and determine the OS in use. It can also be used as a vehicle for various DDoS attacks on CC. For example, an adversary could crash the targeted host with ICMP Echo Request (ping) packets by broadcasting them quickly without waiting for replies, similar to the UDP flood attack principle. The targeted VM's resources would deplete rapidly, affecting the VM's availability. All Internet protocols permit specific data packets. In this attack, the destination CC resources or VM receives more data packets than the protocol allows, forcing the TCP/IP stack to fragment all data packets on the sender side and assemble them on the receiving side. When large amounts of fragmented data must be reassembled, the destination system's performance will suffer. In other words, adversaries flooded the victim machine by sending a huge number of ICMP echo requests. When the infected machine tries to respond, the maximum bandwidth used will be near the maximum amount. As a result, legitimate users could not connect to the CC network. When the CCC tries to send the reply, adversaries send an ICMP echo request packet. The bandwidth utilization will reach the maximum, and new users cannot connect to the network during this time [27]. Furthermore, the adversary could leverage a compromised CC device as an intermediary to send ICMP echo requests to flood the local network, resulting in an insider attack.

3.5. HTTP Flood

The application layer is vital for CC since the CSPs deliver many essential services to their users using application layer protocols, such as HTTP. HTTP is the primary application layer protocol used by web servers. Since CC usually hosts many web application servers, a massive number of HTTP requests can easily overwhelm web services. An example of an application layer attack is an HTTP flood. In an HTTP flood, the attackers may send enormous volumes of malicious HTTP requests to the victims to exhaust the resources and services running in the cloud and cause an EDoS attack, which is explained in Section 5.1.

A client, via a web browser or terminal, "talks" to a VM or web application server by sending a POST or GET request. The client uses POST queries to access dynamic resources, while GET requests retrieve static information like images. The two main categories of HTTP flood attacks are HTTP-POST and HTTP-GET. Attackers could overwhelm a targeted site or VM with HTTP-GET requests using valid packets without reflection or spoofing. Because many requests are sent to the web application server, and the VM generates many more responses than the zombie army receives, this attack is achievable by small botnets [28].

In this situation, an attacker sends an HTTP-GET request to the target application to test its availability. If the attacker receives an acknowledgment from the target application, the attacker transmits new HTTP-GET requests successively without waiting for acknowledgments. Since the web application server does not filter HTTP-GET requests to check if they are legitimate or not [29], it will continue accepting and processing the requests.

4. Taxonomy of DDoS Attack Detection Approaches on CC

4.1. Signature-Based Detection

Signature-based, misused-based, or rule-based approaches detect a DDoS attack if the incoming packets or traffic patterns match the predefined signatures or rules in its attack signatures database [30]. The drawback of these approaches is they cannot detect zero-day attacks.

The authors in [31] outlined the design of an offline signature-based network IDS that uses distributed processing and a Naive Bayesian classifier to detect DoS and DDoS attacks against HTTP servers. They or other researchers should do more work to build an inline IDS to identify attacks in real time. Because the current technique can only detect known attacks, more research is needed to detect new ones. The performance of the Naïve Bayesian classifier, having different classification methods, was evaluated on a testbed, achieving 97.82% classification accuracy for slow read attacks and 96.46% detection accuracy for normal behavior.

Anitha and Malliga [32] attempted to solve the problem of HTTP and XML Denial of Service (HX-DoS) attacks using CLASSIE, a rule-based detection system, and the modulo marking approach, which prevents spoofing attacks. For decision and packets dropping on the victim side, the Reconstruct and Drop method is employed. It helps improve the detection and filtering of DDoS attacks while lowering the false-positive rate. These attacks can be quickly detected on the adversary side by utilizing a packet-based marking mechanism. It can be filtered using the discovered packets on the victim side by dropping the marked packets. As a result, the overhead of packet marking and the false-positive rate of DoS attacks are considerably decreased.

Wang et al. [12] presented a dataset shift attack detection system based on a graphic model. The simulation findings suggest that their architecture can deal with the security difficulties posed by the new network paradigm effectively and efficiently. Additionally, the simulation result indicates that their attack detection system can effectively report numerous threats using real-world network traffic.

They proposed a new IPS service that uses signature-based devices, known as service-based intrusion prevention systems in CC (SIPSCC), to prevent SQL injections on CC websites (CCW). They used three VMs to test a model. Their implementation proposes, investigates, and evaluates SIPSCC from three perspectives: vulnerability detection, average time, and false positives. The suggested technique identifies and prevents key vulnerabilities in CCW [33].

Khatri and Khilari [34] proposed an architecture that includes the implementation of Suricata IDS for securing virtualized servers on CC and the validation of the IDS in detecting DDoS attacks against virtualized environments, effectively protecting the CC from vulnerabilities.

Sangeetha et al. [35] proposed combining a multi-threaded network IDS (NIDS) and host IDS (HIDS) to provide an efficient, quick, and secure HIDS. Cloud-IDS now captures packets from the network, analyses them, and sends reports to the CC Administrator based on the analysis. The K-Nearest neighbor and neural network (KNN-NN) hybrid classifier analyze packets. Further, the NSL-KDD dataset is used for training and testing purposes. After receiving the notification from Cloud-IDS, the CSP will alert the user and keep a log list of the malicious IP addresses. This approach effectively manages huge data packets, analyses them, and generates reports while detecting anomalies and misuse.

The E-CARGO model [36] is used to present a collaborative intrusion detection architecture. The components of an IDS are described by the common intrusion detection frame (CIDF). They also create and clearly outline the agent's behaviors and their relationships. The experiments show that their proposed technique can detect slow-scanning and DDoS attacks, which validates their model. The authors planned to study combining cooperative computing with IDS to deal with real-world problems in future work.

4.2. Anomaly-Based Detection

Anomaly-based detection is based on a profiling program that will be created for the normal behavior of the network, which the anomaly-based detection system will use as a baseline. Deviation from this baseline will be treated as an anomaly or a possible intrusion [37]. Anomaly-based detection approaches can trigger multiple false alarms due to the changing nature of network behavior or zombies and suspicious requests on the application layer if the detecting algorithm parameters are not properly tuned. Without any tuning to optimization, the classifier will not increase the detecting accuracy. If not, collecting the correct logs in a good way to choose the features well will not contribute to the best detection. The input to detection could be in the form of a vector, object, point, or observation named as single data instances [38] or a combination of data instances. Several anomaly-based approaches are using DL and ML to detect HTTP flooding DDoS attacks.

Alqahtani and Gamble [39] came up with a DDoS attack detection technique for the CC service and developed a four-layer algorithm to resolve the originating service for the attack. The levels are so structured that each level is suitable for detecting the attacks' symptoms using local data. Their detection techniques achieved O(n2) time in the worst-case scenario. They also reported a link between DDoS attacks and unauthorized messages passed across services.

Abusitta et al. [40] proposed a correlation mechanism by employing hypervisors to determine the predicted resource load of current compromised VMs based on specified metrics. The calculated resource load is then compared to the total resource load. The proposed approach collects system metrics to train the SVM classifier to distinguish between normal and malicious (i.e., DoS attack) VM activities. The results show that when utilizing the model to make resource adjustments, the detection accuracy hits 97.60%. Their findings also demonstrate that the accuracy of the revoking and

granting adjustments was reduced by just 1.79 percent and 1.43 percent, respectively, under the effect of resource adjustments, which have minimal impact and may be ignored.

Choi et al. [41] proposed a way to detect HTTP-GET flood DDoS attacks using MapReduce. This method ensures the target system's availability for precise and reliable detection of HTTP-GET flood attacks. The experimental results show that the proposed approach outperforms Snort detection because its processing time decreases as congestion increases.

Chen et al. [42] proposed a CC-based network monitoring and threat detection mechanism comprising monitoring agents, CC infrastructure, and operation center components. The proposed mechanism used Hadoop, Spark, and MapReduce to speed up data processing using separation and concurrent processing of data streams. Furthermore, they conducted a real-world experiment to evaluate the effectiveness of the developed network monitoring and threat detection and system performance to limit the risk of DoS attacks. The evaluation results reveal that the mechanism successfully detects and mitigates these attacks. Furthermore, the defensive system detects all published vulnerabilities and can identify unknown attacks [43].

Xiao et al. [44] proposed a protocol-free detection (PFD) algorithm to detect ransom denial of service (RDoS) attacks against the CC regardless of the protocol utilized in the attack. PFD calculates the flow correlation coefficient (FCC) between flow pairs and issues a warning once suspicious flows have been identified. The simulation result indicates it is effective in detecting RDoS attacks and can help detect and isolate adversary flows.

Dhanapal and Nithyanadam [29] used the OpenStack CC platform to implement their solution that detects, mitigates, and prevents low-rate HTTP DDoS attacks in the CC environment. The experiments yielded accurate findings in identifying attacks in the early phases.

The authors in [45] studied the existing DDoS attack detection frameworks and their flaws, then proposed a CC testbed framework on top of an OpenStack platform [46] for testing HTTP flood DDoS attack solution. They also looked into numerous attack paths to the web server on the CC, internally and externally.

The authors present a novel approach to protecting mobile-based systems from DDoS attacks. The model is built on anomaly detection to defend the public/private CC against zero-day attacks. By preventing CC DDoS attacks, the availability of CC applications significantly improved, and users will receive high-quality services [47][48]. Evaluations of the proposed model's efficiency and performance were promising in safeguarding mobile-based CC systems against DDoS attacks. The focus is on detecting and protecting mobile-based systems from DDoS attacks [49]. [50] reported the approach's complexity analysis, efficacy, and performance assessments, and the improved version is documented in [49].

Hazavehi and Rahmani [16] proposed and developed a mechanism called TPANGND for detecting DDoS attacks based on anomalies. Their mechanism uses flow-based classifier (FBC) to group similar input patterns into several clusters to determine an attack. Unique scenarios exist where FBC cannot distinguish between benign and malicious traffic. The suspect traffic is recognized in this situation by looking at the correlation between the VM instance issued by the CSP at a specific timestamp and the suspicious source list. The experimental results show that the suggested technique has a higher detection rate than existing K-means, fuzzy c-means clustering, bat clustering, and Bartd methods. It can detect unknown threats with fewer false alarms [51][52].

Abbasi et al. [53] proposed a new framework to detect various EDoS attacks by creating a profile that learns from and categorizes normal and abnormal activities. The more demanding resources are only allocated to VMs with a normal state in this framework, preventing the propagation of attacks and resource misuse in the CC.

Singh et al. [54] proposed collaborative IDS (CIDS), a system that combines cascading decision trees (DTs) and SVM to increase detection accuracy. DT speeds up the learning process and divides the dataset into smaller subsets; SVM on each sub-dataset (e.g., KDD99, NSL KDD, and ITOC) reduces SVM learning time, overcomes over-fitting, and reduces the size of the DT, allowing faster detection.

Raja Sree and Mary Saira Bhanu [52] proposed a method that scans log files to extract essential information related to HTTP flooding threats by grouping similar input patterns using fuzzy bat clustering and determining unusual behavior using deviating anomaly scores. They compared the findings with existing methodologies such as k-means clustering, fuzzy c-means clustering, bat clustering, and the Bartd method, showing the proposed method accurately diagnoses anomalies with low false alarms.

4.3. Hybrid Detection

A hybrid detection approach combines multiple detection approaches, including signature- and anomaly-based approaches. However, it has some drawbacks, such as a conflict between the two approaches, resulting in increased detection time. Therefore, the hybrid approach requires balancing options and complimentary features for each approach to improve discovery and detection rates.

Several researchers have adopted this approach and have presented architecture and methods for performing intrusion detection utilizing hypervisor performance metrics using virtualization technology based on CC. Furthermore, it is demonstrated that suspicious activities can be profiled without detailed knowledge of the OS running within the VMs using VM performance metrics gathered from hypervisors, such as packets transmitted/received, block device read/write requests, and CPU utilization [55].

Patil et al. [56] have designed an efficient security framework called Protocol Specific Multi-threaded Network IDS to detect DDoS attacks in a CC. It works by separating the incoming packets based on the protocol. These packets are sent in a queue for processing therein. The framework thread is responsible for handling each queue which also extracts the relevant features and applies protocol-specific classifiers for each packet in the queue. They used the KDD'99 dataset.

SaiSindhuTheja and Shyam [57] proposed an efficient DoS attack detection system based on the oppositional crow search algorithm (OCSA), which combines the crow search algorithm (CSA) and the opposition-based learning (OBL) technique. The proposed method has two stages: feature selection with OCSA and classification with an RNN classifier. The OCSA method identifies the key features, then feeds into the RNN classifier. The RNN classifier is used to classify incoming data during the testing process. It ensures that standard data (saved in the CC) is isolated from compromised data. The results show that this strategy outperforms other conventional methods by 98.18%, 95.13%, 93.56%, and 94.12% in terms of Precision, Recall, F-Measure, and Accuracy, respectively, using the benchmark data set. In addition, the suggested approach surpasses existing efforts by 3% on average across all metrics.

Many existing ML algorithms, such as neural classifiers, can detect DDoS attacks. The researchers in [58] discussed the findings of a survey on DDoS attacks in the CC environment. DDoS attacks are frequently categorized as bandwidth and resource consumption attacks. SYN Flood and Flash Crowd are prevalent DDoS attacks in a CC context. Nagaraja et al. [58] also tested many ML algorithms to detect DDoS attacks; some are more accurate than others. The use of ML techniques resulted in a higher false-positive detection rate. According to their study, after examining several studies on network attack detection in the CC environment, the most extensively utilized technique to detect DDoS attacks in the CC is ANN, SVM, KNN, J48, feature rank, and feature selection.

4.4. Entropy-Based Detection

Entropy is the ratio of arbitrariness in the data. Entropy-based detection approach analyzes random data, the entropy, or the Shannon-Wiener index to evaluate uncertainty associated with the data. Maximum randomness in the data implies a maximum entropy value [59][60]. For example, if the data only has one class, its entropy value will be lower. On the contrary, the data with numerous classes will have a higher entropy value. This way, the tested headers are broken down for port and IP, and their entropy is computed.

Entropy is usually used to calculate the randomness of IP source addresses or port numbers. A high entropy value indicates the traffic originates from various sources, which is the clue to detecting DDoS attacks [61]. A threshold can be put in place to distinguish DDoS attack traffic from normal traffic. The administrator should be alerted of DDoS attacks if the entropy value exceeds the threshold. If the detection of DDoS attacks involves multiple levels, the procedure can be partitioned into three stages:

- First stage: The client is permitted to go through the switch, and the detection calculation confirms that it is genuine.
- Second stage: The entropy is calculated based on the data packet size and the client's authentication.
- Third stage: The entropy value is compared with the threshold to determine if it is a DDoS attack or not.

Once the location of any abnormality is discovered, an information message is sent to CSP owners to take necessary action. The authors in [62] proposed an approach to detect HTTP flooding DDoS attacks in a CC using information-theoretic entropy (ITE) and ML to improve the false-positive rates. They are planning a real-world deployment of their approach for evaluation using several HTTP DDoS attack tools in the future.

The authors in [63] developed an entropy-based detection technique for DDoS attacks, achieving a 90 percent accuracy without extra packet overhead, resulting in excellent QoS. In addition, they have used CCs to implement the same algorithm. Meanwhile, the authors in [64] used a Gossip-based DDoS attacks detection apparatus for attack detection in a computer network by exchanging a stream of traffic-over-line.

The authors in [65] used an improved entropy to detect the cause of overload and locate the source of the problem, but [66] is similar in its approach to these authors. It appears that a reduction in traffic and improved response time could be feasible with the data simulated.

Girma et al. [67] examined and compared various DDoS attack detection techniques against multiple parameters. After discussing their benefits and drawbacks, they proposed a hybrid statistical model that could significantly mitigate DDoS attacks, providing a better solution to current detection issues. The authors of [68] looked at the standard EPA-HTTP

(environmental protection agency-hypertext transfer protocol) dataset. They chose the input parameters for the classifier model to distinguish an attack from a regular profile.

4.5. Filtering Tree-Based Detection

A technique proposed in [69] identifies flood attacks by analyzing network logs and keeping track of the connection states, such as the active IPs of incoming requests. It alters the window size (number of time slots) and measures the sliding window of dynamic entropy, which is dependent on traffic load. In a CC setting, traditional DDoS attacks on servers and network resources could morph into a new breed of attack called EDoS attacks, which target the CCC's economic resources. The researchers have presented a unique mitigation strategy against EDoS threats, utilizing source checking, counting, and Turing Test. The simulation results suggest that their technology can mitigate CC EDoS attacks.

Researchers in [70] proposed a CC defender system named cloud service queuing defender (CSQD) to detect and remediate XML vulnerabilities in online services. CSQD, a self-learner, employs a traceback solution to determine the source of the attack. Suppose an attack successfully shuts down the server; the CSQD system will detect the malicious requests and store them in its database to prevent similar attacks in the future. The authors presented a game-theoretic model and study that predicted widespread strategy adoption, reducing the risk of DNS amplification attacks. They have demonstrated the ability to implement their concept as a CC-based service to cut costs further and provide additional defenses for DNS servers.

A new solution dubbed an enhanced DDoS-mitigation system (Enhanced DDoS-MS) has been developed to combat EDoS attacks by leveraging firewall capabilities to control a verification process to protect the targeted system. Researchers used a simulated environment to assess their proposed system, showing the firewall successfully mitigates DDoS attacks by increasing users' services in response time and server load under attack [71][72].

Fontaine et al. [73] proposed a simplified CC security utilizing ML approaches to address the challenge of complex and platform-specific CC security architectures. It leads to a more general design that employs decision trees and neural networks as classifiers, trained using data gathered by CC apps. Iyengar et al. proposed a multilevel thrust filtration (MTF) mechanism as a solution against DDoS attacks in a CC environment. The mechanism authenticates incoming requests and detects various types of DDoS attacks at various levels at the early stage to prevent unnecessary traffic from reaching the DC [74].

References

1. Bahashwan, A.A.; Anbar, M.; Abdullah, N. New architecture design of cloud computing using software defined networking and network function virtualization technology. In Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2020; Volume 1073, pp. 705–713.
2. Alashhab, Z.R.; Anbar, M.; Singh, M.M.; Leau, Y.B.; Al-Sai, Z.A.; Abu Alhayja'a, S. Impact of coronavirus pandemic crisis on technologies and cloud computing applications. J. Electron. Sci. Technol. 2021, 19, 100059.
3. Song, S.m.; Yoon, Y.i. NIST Cloud Computing Program Overview. Available online: https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp (accessed on 30 November 2022).
4. Ficco, M.; Palmieri, F. Introducing fraudulent energy consumption in cloud infrastructures: A new generation of denial-of-service attacks. IEEE Syst. J. 2017, 11, 460–470.
5. Newmark, E.; Brien, A.O.; Arend, C.; Morris, H.D.; Nebuloni, G.; Versace, M.; Futurescape, F.D.E.I.D.C. IDC FutureScape IDC FutureScape: "Worldwide Cloud 2018 Predictions". Available online: https://www.sapvirtualagency.com/FileExplorer/Partners/SAPCloudPlatform/esCO/ManageYourBusiness/US42014717_esCO_Final_delive (accessed on 30 November 2022).
6. Kupreev, O.; Badovskaya, E.; Gutnikov, A. DDoS Attacks in Q1 2020. Available online: https://securelist.com/ddos-attacks-in-q1-2020/96837/ (accessed on 30 November 2022).
7. Khandelwal, S. 602 Gbps! This May Have Been the Largest DDoS Attack in History. Available online: http://thehackernews.com/2016/01/biggest-ddos-attack.html (accessed on 30 November 2022).
8. Yevsieieva, O.; Helalat, S.M. Analysis of the impact of the slow HTTP DOS and DDOS attacks on the cloud environment. In Proceedings of the 2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, (PIC S&T), Kharkov, Ukraine, 10–13 October 2017; pp. 519–523.
9. Al Ashhab, Z.R.; Anbar, M.; Singh, M.M.; Alieyan, K.; Ghazaleh, W.I. Detection of http flooding ddos attack using hadoop with mapreduce: A survey. Int. J. Adv. Trends Comput. Sci. Eng. 2019, 8, 71–77.
10. Singh, P.; Manickam, S.; Ul Rehman, S. A survey of mitigation techniques against Economic Denial of Sustainability (EDoS) attack on cloud computing architecture. In Proceedings of the 3rd International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, Noida, India, 8–10 October 2014; pp. 1–4.
11. Swami, R.; Dave, M.; Ranga, V. Software-defined Networking-based DDoS Defense Mechanisms. ACM Comput. Surv. 2019, 52, 1–36.

12. Wang, B.; Zheng, Y.; Lou, W.; Hou, Y.T. DDoS attack protection in the era of cloud computing and Software-Defined Networking. Comput. Netw. 2015, 81, 308–319.
13. Osanaiye, O.; Choo, K.K.R.; Dlodlo, M. Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. J. Netw. Comput. Appl. 2016, 67, 147–165.
14. Sanjalawe, Y.; Anbar, M.; Al-E'mari, S.; Abdullah, R.; Hasbullah1, I.; Aladaileh, M. Cloud Data Center Selection Using a Modified Differential Evolution. Comput. Mater. Contin. 2021, 69, 3179–3204.
15. Wang, H.; Xi, Z.; Li, F.; Chen, S. Abusing public third-party services for EDoS attacks. In Proceedings of the 10th USENIX Workshop on Offensive Technologies, WOOT 2016, Austin, TX, USA, 8–9 August 2016.
16. Mahdavi-Hezavehi, S.; Alimardani, Y.; Rahmani, R. An Efficient Framework for a Third Party Auditor in Cloud Computing Environments. Itnow 2020, 62, 66.
17. Baig, Z.A.; Sait, S.M.; Binbeshr, F. Controlled access to cloud resources for mitigating Economic Denial of Sustainability (EDoS) attacks. Comput. Netw. 2016, 97, 31–47.
18. Radware. BrickerBot: Back with a Vengeance. Available online: https://www.radware.com/security/ddos-threats-attacks/brickerbot-pdos-back-with-vengeance/ (accessed on 30 November 2022).
19. Rao Varre, D.N.M.; Bayana, J. A Secured Botnet Prevention Mechanism for HTTP Flooding Based DDoS Attack. In Proceedings of the 2022 3rd International Conference for Emerging Technology, INCET 2022, Belgaum, India, 27–29 May 2022; pp. 1–5.
20. Kumar, S.N.; Vajpayee, A. A survey on secure cloud: Security and privacy in cloud computing. Am. J. Syst. Softw. 2016, 4, 14–26.
21. Wired; Zetterl, K. FBI Defends Disruptive Raids on Texas Data Centers|WIRED. Available online: https://www.wired.com/2009/04/data-centers-ra/ (accessed on 30 November 2022).
22. Helpnetsecurity. USB Killer 2.0: A Harmless-Looking USB Stick that Destroys Computers—Help Net Security. Available online: https://www.helpnetsecurity.com/2015/10/15/usb-killer-20-a-harmless-looking-usb-stick-that-destroys-computers/ (accessed on 30 November 2022).
23. Sue, P. Types of DDoS Attacks. Available online: https://www.globaldots.com/blog/types-ddos-attacks (accessed on 22 January 2022).
24. Meng, B.; Andi, W.; Jian, X.; Fucai, Z. DDOS Attack Detection System Based on Analysis of Users' Behaviors for Application Layer. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, Guangzhou, China, 21–24 July 2017; Volume 1, pp. 596–599.
25. Maghrabi, L.A. The threats of data security over the Cloud as perceived by experts and university students. In Proceedings of the 2014 World Symposium on Computer Applications and Research (WSCAR), Sousse, Tunisia, 18–20 January 2014; Volume 18–20, pp. 1–6.
26. Neupane, R.L.; Neely, T.; Calyam, P.; Chettri, N.; Vassell, M.; Durairajan, R. Intelligent defense using pretense against targeted attacks in cloud platforms. Future Gener. Comput. Syst. 2019, 93, 609–626.
27. Paraszczuk, M. "Software Reviews, Opinions, and Tips—DNSstuff." Software Reviews, Opinions, and Tips—DNSstuff. Available online: https://www.dnsstuff.com/network-throughput-bandwidth (accessed on 30 November 2022).
28. Alieyan, K.; Kadhum, M.M.; Anbar, M.; Rehman, S.U.; Alajmi, N.K. An overview of DDoS attacks based on DNS. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 19–21 October 2016; pp. 276–280.
29. Dhanapal, A.; Nithyanandam, P. The slow http ddos attacks: Detection, mitigation and prevention in the cloud environment. Scalable Comput. 2019, 20, 669–685.
30. Al-mashhadi, S.; Anbar, M.; Hasbullah, I.; Alamiedy, T.A. Hybrid rule-based botnet detection approach using machine learning for analysing DNS traffic. PeerJ Comput. Sci. 2021, 7, e640.
31. Katkar, V.; Zinjade, A.; Dalvi, S.; Bafna, T.; Mahajan, R. Detection of DoS/DDoS attack against HTTP servers using naive Bayesian. In Proceedings of the 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015, Pune, India, 26–27 February 2015; pp. 280–285.
32. Anitha, E.; Malliga, S. A packet marking approach to protect cloud environment against DDoS attacks. In Proceedings of the 2013 International Conference on Information Communication and Embedded Systems, ICICES 2013, Chennai, India, 21–22 February 2013; pp. 367–370.
33. Alqahtani, S.M.; Al Balushi, M.; John, R. An intelligent intrusion prevention system for cloud computing (SIPSCC). In Proceedings of the 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014, Las Vegas, NV, USA, 10–13 March 2014; Volume 2, pp. 152–158.
34. Khatri, J.K.; Khilari, G. Advancement in virtualization based intrusion detection system in cloud environment. Int. J. Sci. Eng. Technol. Res. (IJSETR) 2015, 4, 1510–1514.
35. Sangeetha, S.; Gayathri Devi, B.; Ramya, R.; Dharani, M.K.; Sathya, P. Signature based semantic intrusion detection system on cloud. In Advances in Intelligent Systems and Computing; Springer: New Delhi, India, 2015; Volume 339, pp. 657–666.

36. Teng, S.; Zheng, C.; Zhu, H.; Liu, D.; Zhang, W. A cooperative intrusion detection model for cloud computing networks. Int. J. Secur. Its Appl. 2014, 8, 107–118.

37. Xiang, Y.; Li, K.; Zhou, W. Low-rate DDoS attacks detection and traceback by using new information metrics. IEEE Trans. Inf. Forensics Secur. 2011, 6, 426–437.

38. Alzubi, Q.M.; Anbar, M.; Sanjalawe, Y.; Al-Betar, M.A.; Abdullah, R. Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization. Expert Syst. Appl. 2022, 204, 117597.

39. Alqahtani, S.; Gamble, R.F. DDoS attacks in service clouds. In Proceedings of the Annual Hawaii International Conference on System Sciences, Kauai, HI, USA, 5–8 January 2015; pp. 5331–5340.

40. Abusitta, A.; Bellaiche, M.; Dagenais, M. An SVM-based framework for detecting DoS attacks in virtualized clouds under changing environment. J. Cloud Comput. 2018, 7, 9.

41. Choi, J.; Choi, C.; Ko, B.; Kim, P. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft Comput. 2014, 18, 1697–1703.

42. Chen, Z.; Xu, G.; Mahalingam, V.; Ge, L.; Nguyen, J.; Yu, W.; Lu, C. A Cloud Computing Based Network Monitoring and Threat Detection System for Critical Infrastructures. Big Data Res. 2016, 3, 10–23.

43. Vissers, T.; Somasundaram, T.S.; Pieters, L.; Govindarajan, K.; Hellinckx, P. DDoS defense system for web services in a cloud environment. Future Gener. Comput. Syst. 2014, 37, 37–45.

44. Xiao, L.; Wei, W.; Yang, W.; Shen, Y.; Wu, X. A protocol-free detection against cloud oriented reflection DoS attacks. Soft Comput. 2017, 21, 3713–3721.

45. Dhanapal, A.; Nithyanandam, P. An OpenStack based cloud testbed framework for evaluating HTTP flooding attacks. Wirel. Netw. 2021, 27, 5491–5501.

46. Albaroodi, H.; Manickam, S.; Anbar, M. A proposed framework for outsourcing and secure encrypted data on OpenStack object storage (Swift). J. Comput. Sci. 2015, 11, 590.

47. Osanaiye, O.; Choo, K.K.R.; Dlodlo, M. Change-point cloud DDoS detection using packet inter-arrival time. In Proceedings of the 2016 8th Computer Science and Electronic Engineering Conference, CEEC 2016, Colchester, UK, 28–30 September 2016; pp. 204–209.

48. Kiruthika Devi, B.S.; Subbulakshmi, T. A comparative analysis of security methods for ddos attacks in the cloud computing environment. Indian J. Sci. Technol. 2016, 9, 1–7.

49. El-Sofany, H.F. Proposed a Novel Mechanism to Detect and Prevent XML and HTTP-Based Denial-of-Service Attacks for Cloud Computing. In Proceedings of the 2018 International Conference on Network Technology (ICNT 2018), and 7th International Conference on Software and Information Engineering (ICSIE 2018), Cairo, Egypt, 2–4 May 2018; pp. 4–6.

50. El-Sofany, H.F.; Abou El-Seoud, S. Performance Analysis of an Effective Approach to Protect Cloud Systems against Application Layer Based Attacks. Int. J. Online Biomed. Eng. (iJOE) 2019, 15, 82.

51. Muthukrishnan, R.K.; Hoy, J.R.; Iyer, S.R.; Kapadia, K.K.; Nagaratnam, N. User state tracking and anomaly detection in software-as-a-service environments. US Patent 10,200,387, 2019.

52. Raja Sree, T.; Mary Saira Bhanu, S. Detection of HTTP flooding attacks in cloud using fuzzy bat clustering. Neural Comput. Appl. 2020, 32, 9603–9619.

53. Abbasi, H.; Ezzati-Jivan, N.; Bellaiche, M.; Talhi, C.; Dagenais, M.R. Machine Learning-Based EDoS Attack Detection Technique Using Execution Trace Analysis. J. Hardw. Syst. Secur. 2019, 3, 164–176.

54. Singh, D.; Patel, D.; Borisaniya, B.; Modi, C. Collaborative IDS framework for cloud. Int. J. Netw. Secur. 2016, 18, 699–709.

55. Nikolai, J.; Wang, Y. Hypervisor-based cloud intrusion detection system. In Proceedings of the 2014 International Conference on Computing, Networking and Communications, ICNC 2014, Honolulu, HI, USA, 3–6 February 2014; pp. 989–993.

56. Patil, R.; Dudeja, H.; Gawade, S.; Modi, C. Protocol Specific Multi-Threaded Network Intrusion Detection System (PM-NIDS) for DoS/DDoS Attack Detection in Cloud. In Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018, Bengaluru, India, 10–12 July 2018; pp. 1–7.

57. SaiSindhuTheja, R.; Shyam, G.K. An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. Appl. Soft Comput. 2021, 100, 106997.

58. Nagaraja, A.; Boregowda, U.; Vangipuram, R. Study of Detection of DDoS attacks in cloud environment Using Regression Analysis. In Proceedings of the International Conference on Data Science, E-Learning and Information Systems 2021, Ma'an, Jordan, 5–7 April 2021; ACM: New York, NY, USA, 2021; pp. 166–172.

59. Aladaileh, M.A.; Anbar, M.; Hintaw, A.J.; Hasbullah, I.H.; Bahashwan, A.A.; Al-Sarawi, S. Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks against SDN Controller with Various Traffic Rates. Appl. Sci. 2022, 12, 6127.

60. Aladaileh, M.A.; Anbar, M.; Hasbullah, I.H.; Chong, Y.W.; Sanjalawe, Y.K. Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller—A Review. IEEE Access 2020, 8, 143985–143995.

61. Shah, S.B.I.; Anbar, M.; Al-Ani, A.; Al-Ani, A.K. Hybridizing entropy based mechanism with adaptive threshold algorithm to detect RA flooding attack in IPv6 networks. In Lecture Notes in Electrical Engineering; Alfred, R., Lim, Y., Ibrahim, A.A.A., Anthony, P., Eds.; Springer: Singapore, 2019; Volume 481, pp. 315–323.
62. Idhammad, M.; Afdel, K.; Belouch, M. Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest. Secur. Commun. Netw. 2018, 2018.
63. Zakarya, M. DDoS verification and attack packet dropping algorithm in cloud computing. World Appl. Sci. J. 2013, 23, 1418–1424.
64. Zakarya, M.; Jan, Z.; Ullah, I.; Dilawar, N. DDoS Confirmation & Attack Packet Dropping Algorithm in On-Demand Grid Computing Platform. Bahria Univ. J. Inf. Commun. Technol. 2012, 5, 64–68.
65. Jeyanthi, N.; Iyengar, N.C.S.; Kumar, P.C.; Kannammal, A. An enhanced entropy approach to detect and prevent DDOS in cloud environment. Int. J. Commun. Networks Inf. Secur. 2013, 5, 110–119.
66. Agrawal, N.; Tapaswi, S. A Lightweight Approach to Detect the Low/High Rate IP Spoofed Cloud DDoS Attacks. In Proceedings of the 2017 IEEE 7th International Symposium on Cloud and Service Computing, SC2 2017, Kanazawa, Japan, 22–25 November 2017; pp. 118–123.
67. Girma, A.; Garuba, M.; Li, J.; Liu, C. Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment. In Proceedings of the 12th International Conference on Information Technology: New Generations, ITNG 2015, Las Vegas, NV, USA, 13–15 April 2015; pp. 212–217.
68. Singh, K.J.; Thongam, K.; De, T. Entropy-based application layer DDoS attack detection using artificial neural networks. Entropy 2016, 18, 350.
69. Shameli-Sendi, A.; Pourzandi, M.; Fekih-Ahmed, M.; Cheriet, M. Taxonomy of Distributed Denial of Service mitigation approaches for cloud computing. J. Netw. Comput. Appl. 2015, 58, 165–179.
70. ManouchehriSarhadi, R.; Ghafori, V. New Approach to Mitigate XML-DOS and HTTP-DOS Attacks for Cloud Computing. Int. J. Comput. Appl. 2013, 72, 27–31.
71. Bakshi, A.; Yogesh, B. Securing cloud from DDOS attacks using intrusion detection system in virtual machine. In Proceedings of the 2nd International Conference on Communication Software and Networks, ICCSN 2010, Singapore, 26–28 February 2010; pp. 260–264.
72. Alosaimi, W.; Alshamrani, M.; Al-Begain, K. Simulation-Based Study of Distributed Denial of Service Attacks Prevention in the Cloud. In Proceedings of the NGMAST 2015: The 9th International Conference on Next Generation Mobile Applications, Services and Technologies, Cambridge, UK, 9–11 September 2015; pp. 60–65.
73. Fontaine, J.; Kappler, C.; Shahid, A.; Poorter, E.D. Log-Based Intrusion Detection for Cloud Web Applications Using Machine Learning. In Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2020; Volume 96, pp. 197–210.
74. Iyengar, N.C.S.N.; Ganapathy, G.; Kumar, P.C.; Abraham, A. A multilevel thrust filtration defending mechanism against DDoS attacks in cloud computing environment. Int. J. Grid Util. Comput. 2014, 5, 236–248.