Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 11 Issue 7, July - 2024

Enhancing DDoS Attack Prevention And Detection Using IDS And XDR

Christian Bassey
Department of Computer Science
Innopolis University
Innopolis, Russia
C.bassey@innopolis.university

Ajayi Ayojesu Samuel
Department of Computing,
Teesside University, Middlesbrough,
United Kingdom.
ajayiayojesu@gmail.com

Success Imakuh
Department of Computing
Teesside University
Middlesbrough, United Kingdom.
success.imakuh@owasp.org

Opeyemi Oloruntola
Department of Computer Science
School of Computing and
Mathematical Sciences,
University of Greenwich, London,
United Kingdom.
yemi.sasonel@gmail.com

Ifeanyi Onyia-Odike
Department of Computer Science
Innopolis University
Innopolis, Russia
i.onyia-odike@innopolis.university

***Abstract*—Distributed Denial of Service (DDoS) attacks threaten web applications, servers, and networks. They can greatly impact the availability of critical services and systems. This research investigates various DDoS attacks and the current detection mechanisms, then proceeds to enhance DDoS attack prevention and detection using intrusion detection systems and extended detection and response systems with the ability to add custom detection rules. This work was implemented in a virtual environment running a web server, a DDoS attack controller and an XDR server. Open-source tools and platforms were leveraged in this study, with Suricata being used as the IDS and Wazuh as the XDR platform. The results of the study showed that DDoS attacks executed on a web server or application without a response mechanism impacted system resources, while once a response mechanism such as an IP address blocking was in place, the combined solution of the IDS, XDR, and custom rules triggering the blocking mechanism terminated malicious connections from attackers and drastically reduced the impact of the DDoS attacks on system resources. Additionally, this research provides valuable insights for security administrators concerning implementing mechanisms for DDoS attack detection and prevention, as well as emphasizing a multi-faceted approach leveraging multiple techniques and tools for responding to detected DDoS attacks.**

Keywords—Web application; Cyber attack; Intrusion detection systems; Wazuh; Extended detection and response.

I. Introduction

Cyber-attacks have recently affected numerous businesses and organizations with increasing regularity and sophistication. Denial of service attacks is one such attack that continues to be executed against websites and businesses. Distributed denial of service attacks, in particular, can render businesses inoperational, cripple websites, and impact service delivery.

In DDoS attacks, “the attacker can greatly degrade the quality or fully break the victim's network connectivity. The attacker first compromises many agents or hosts and then uses these agents to launch the attack by depleting the target network”[1]. Depleting a target network or system’s resources can render a service that relies on those resources unusable. Given the impact of DDoS attacks on service delivery, putting in place tools and technologies for detecting and blocking DDoS attacks is crucial. Several cybersecurity solutions can aid in the detection and blocking of DDoS attacks. They include security information and event management (SIEM) tools for detection, intrusion detection systems (IDS), and extended detection and response (XDR) tools.

An Intrusion detection system (IDS) is a “software application or device that monitors the system or activities of a network for policy violations or malicious activities and generates reports to the management system”[2]. IDS are monitoring systems; as such, they focus on detecting possible intrusions but not preventing them. To prevent detected incidents or respond to ongoing incidents, it is necessary to have a response system. This is where an IDS combined with

an XDR may provide a possible solution for responding to detected DDoS attacks.

An extended detection and response (XDR) tool is an evolution of security tools that unify endpoints, networks, intelligence, and other security data from various sources on one platform to detect, respond to, and stop threats. “XDR offers detection and response of security-related incidents through multiple layers of the Information Technology environment”[3]. It gathers data from endpoints, email, cloud infrastructure, servers, networks, and security solutions. The data collected is used to discover evasive threats and enable security specialists to review and respond quickly.

Since an XDR collects data from various security solutions and endpoints, including an IDS and the web server being attacked, it is a useful tool for responding to DDoS alerts detected by the IDS system. In this research work, we aim to utilize the detection capabilities of an IDS and the response features of an XDR to detect and respond to DDoS attacks on web servers efficiently.

II. Problem statement:

While various cybersecurity solutions exist for detecting and mitigating DDoS attacks, there is a need for a comprehensive and integrated approach that combines the detection capabilities of an Intrusion Detection System (IDS) with the response features of an Extended Detection and Response (XDR) tool. The current problem is the need for a unified architecture that efficiently detects and responds to DDoS attacks on web servers, leaving businesses vulnerable to service disruptions and potential financial losses.

III. Motivation

The research aims to combat the growing threat of DDoS attacks against web servers, disrupting website availability and impairing businesses' capacity to provide services. It develops a reusable architecture for deploying an IDS and XDR to web servers and a custom detection integration for an open-source XDR solution for responding to DDoS attacks identified by the IDS.

IV. Literature Review

In this section, DDoS attacks are reviewed. Then, the various ways of detecting and responding to DDoS attacks are identified, and corresponding existing literature from various authors is reviewed with a focus on practical implementations by authors that can be used in production environments and integrated with intrusion detection systems (IDS) and extended detection and response (XDR) tools.

A. Distributed Denial of Service

To understand what a DDoS is, it is first necessary to understand what a denial of service (DoS) is. Denial of service attacks are cyber attacks that “focus on running out the client’s resources so that he will not be able to service a request which is coming on from a legal or a legitimate user”[4]. The aim of a DoS attack is for the attackers to deny legitimate users of the service access to the requested resource. DoS attacks can be targeted against specific infrastructure, industrial control systems, and websites hosted by web servers, among other targets. It is worth noting that a DoS attack will come from one source. As such, given that DoS attacks have a single originating source, distributed denial of service attacks or distributed DoS attacks or DDoS attacks come from multiple attack sources, which may be botnets or zombies. Below is the architecture of a DDoS attack on a target service or infrastructure. It involves a control master and multiple zombies.

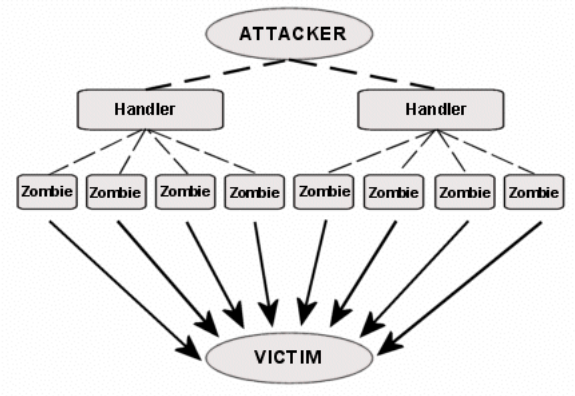


Fig. 1: Architecture of a DDOS attack [14].

DDoS attacks can be broadly classified into three (3) categories [10].

1. **Volumetric DDoS attack:** These attacks flood the target server with traffic intended to overwhelm it, consuming its bandwidth and resources and making it inaccessible to legitimate users. The basic aim of a volumetric attack is “to make a system unavailable by saturating the communication links used to access the victim” [9].

2. **Resource exhaustion DDoS attack**: These attacks abuse vulnerabilities and the operating processes of network protocols to execute DDoS on a target system. “Attacks in this category aim to deplete hardware resources such as memory, CPU, and storage, and thus make servers unavailable by exploiting vulnerabilities in protocols that are usually implemented at the network layer” [9].

3. **Application DDoS attack**: Application layer DDoS attacks focus on exploiting vulnerabilities in the application stack to overwhelm the application with requests and make it unable to respond to legitimate requests. “These attacks are often mistaken for implementation errors, as low rates of malicious traffic are needed to reproduce the behaviour of legitimate customers” [25].

While the above are the categories of DDoS attacks and examples below, some may fall into one or more categories.

B. Intrusion Detection System (IDS)

An intrusion detection system (IDS) is a solution that monitors network traffic or system activity for signals of malicious or unauthorized activity. It analyzes packets, log files, and/or system events and matches them to patterns or signatures of known attacks or anomalous behaviors that may indicate a security incident. “The IDS system uses various devices and software applications that help to monitor the network and predict malicious activities” [28]. It can detect attacks like brute force, DoS, network probing, etc. There are signature-based, rule anomaly-based, and machine learning-based IDS.

C. Extended Detection and Response (XDR)

XDR is a security platform that collects security data, logs, and telemetry from endpoints, networks, and other security solutions in an infrastructure for automated threat detection, investigation, and response. It has become an increasingly important platform because the “increasing complexity of several products from various manufacturers, together with the number of alerts generated, could easily overload businesses, particularly considering a systemic shortage of cybersecurity skills” [3]. XDR greatly simplifies the analysis, investigation, correlation, and response to security incidents using data sources from different platforms. They can collect the detection alerts from the IDS and execute responses based on those detections. XDRs can be broadly classed into two types based on their mode of collection of security data - “open and native” [8]. Where open XDRs rely on third-party integrations to collect specific types of telemetry and respond to those activities, native XDR combines security tools from a single vendor to collect various data types and perform response actions.

D. Related Research

In the paper “Statistical Approaches to DDoS Attack Detection and Response” by Feinstein, Schnackenberg, Balupari, and Kindred, the authors analyze the flow of a DDoS attack and its effects on service availability. They then outline statistical approaches for detecting and responding to these DDoS attacks [12].

Roopak, Tian, and Chambers proposed an IDS specifically designed to mitigate DDoS in IoT networks. The IDS would use machine learning capabilities to analyze network traffic and detect DDoS attacks. The paper outlines the stages of implementing the proposed solution, such as data preprocessing, feature selection, and machine learning model training. They highlight the use of features in their machine learning models, such as packet size, packet rate, and protocol type, as well as how they might be utilized to detect DDoS attacks [21].

Çakmakçı et al. proposed a framework combining SIEM for collecting, analyzing, and correlating security events with ontology-based techniques for representing and reasoning knowledge in a structured format. The authors describe the framework's design, including data gathering modules, data analysis, ontology-based reasoning, and response creation. Then, they show how the SIEM system collects and processes security events from various sources and how the ontology-based reasoning module analyzes and correlates the events to detect DDoS attacks using a domain-specific ontology [7].

George, A. S. et al. discuss the evolution of endpoint security solutions in the article “XDR: The Evolution of Endpoint Security Solutions - Superior Scalability and Analytics to Meet Future Organizational Needs,” with a focus on the concept of Extended Detection and Response (XDR). The authors highlight the growing sophistication of cyber threats targeting endpoints such as computers, laptops, and mobile devices and the need for advanced security solutions to detect and respond effectively. The paper introduces the concept of XDR, a next-generation approach to endpoint security that goes beyond traditional endpoint detection and response (EDR) solutions and discusses how XDR integrates with various security tools and technologies, such as threat intelligence, security analytics, and automation, to provide a more comprehensive and proactive approach to threat detection and response [3].

In the dissertation “Real-time Detection and Response of Distributed Denial of Service (DDoS) Attacks for Web Services” by Shiaeles, the author provides an overview of DDoS attacks, their identifiers, and their impact on web services. After that, the dissertation identifies the current approaches to detecting DDoS attacks, including traffic analysis, anomaly detection, and traffic filtering techniques. The work proposed a new detection technique using a fuzzy estimator based on the mean packet inter-arrival times and a combination of multiple methods in various modules, such as a detection module for analyzing network traffic patterns, an anomaly detection module for identifying abnormal traffic behavior, and a response module for filtering and blocking malicious traffic [22].

In “Intrusion Detection System Performance Against Distributed Denial of Service Attacks” the authors Sousa Araújo, Matos, and Moreira evaluate the performance of different intrusion detection systems (IDS) for detecting distributed denial of service (DDoS) attacks. Existing IDS techniques and tools for detecting DDoS attacks are reviewed, and their strengths and limitations in detection accuracy, efficiency, and timeliness are evaluated. The authors then ran tests to determine the performance of various IDS tools (such as Suricata, Snort, and Zeek) in detecting multiple DDoS attacks in a controlled environment. The paper also discusses the factors that affect the performance of IDS in detecting DDoS, such as the type and intensity of the attack, traffic characteristics, and configuration of the IDS. The authors emphasize the importance of tuning and

optimizing IDS parameters for better performance in detecting DDoS attacks. This paper provides characteristics we can look for in our IDS when choosing a tool to detect DDOS attacks on the web server [24].

V. Methodology

This study employs qualitative and remedial methods. The qualitative technique involved researching various DDoS detection and response strategies and deployment architectures, then crafting a detection and response architecture that leverages an IDS and XDR and includes our custom integration for response to detected DDoS attacks. After that, the designed architecture will be deployed, and then experiments in the form of DDoS attacks will be executed on a web application in the deployed architecture. The responses to these attacks and other predetermined research metrics will be observed and recorded. The results of these trials will be used in the remedial study to determine how the proposed architecture of an IDS with an XDR using our custom response integration may be improved and optimized for better responses.

A. Research Architecture

The architectural diagram for this research work contained the following elements:

1. **Web Application:** OWASP Mutillidae running on PHP with an SQL-based database. Various attacks were executed against it and the web server running it.

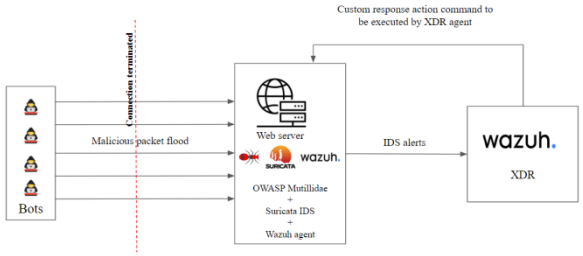
2. **Web Server**: This system hosted our web application. This research used an Ubuntu 22.04 OS for the web server, Apache for the web server, and MySQL for the database.

3. **Intrusion Detection Systems (IDS)**: This security mechanism monitors network traffic to the web server. Suricata was used as the IDS.

4. **Extended Detection and Response (XDR)**: The XDR used an agent to collect logs from the web server and execute custom responses to DDOS attacks. The XDR of choice was Wazuh.

5. **Attack machine**: the endpoints used in executing the DDoS attacks. A Kali 21 OS was used.

Fig. 2: Architecture for detecting and blocking



DDOS attacks on a web server with IDS and XDR.

The IDS is deployed inline on the web server. The inline configuration has been chosen over the passive configuration to ensure that the detection of DDoS attacks happens in real-time. The XDR agent is also installed on the webserver to collect the relevant detection logs and ship them to the XDR server for further analysis. Upon the completion of analysis by the XDR, if the activity from the detection logs proves to be malicious, then the custom response action is executed.

B. DDoS Attacks Executed

Table 1 below shows the tools used to execute the associated attacks.

Table 1: Attack tools and their respective attacks.

S/N	Tool	Attack(s)
1.	hping3	UDP Flood, ICMP flood, SYN flood, Network Bandwidth Exhaustion
2.	SSLDoS	SSL/TLS Exhaustion
3.	Scapy	Malformed Packets
4.	ApacheBench (ab)	HTTP/S Flood
5.	Slowloris	Slowloris
6.	Siege	HTTP GET/POST attack

C. Research Measurement Metrics (RMM)

To determine the impact of our research and how it affords websites better protection, the following metrics are measured and used for further statistical analysis in section 4 to determine the performance of our setup. The metrics noted were:

1. **DDoS attack executed:** This is the record of the specific DDoS attack executed against the web server.

2. **Duration of DDoS attack:** This records how long the attack was executed against the service. This is useful in determining the attack's impact on the victim service over a specific duration extrapolated for a longer period. For this research, each attack will be executed for 120 seconds where possible.

3. **CPU usage:** This is the record of the CPU consumption for 120 seconds before and during the attack.

4. **RAM usage (MU):** This is the measurement of the RAM consumption for 120 seconds before and during the DDoS Attack. It would allow us to determine the attack's impact on the system. Measured in Mebibyte MiB.

5. **Network Bandwidth:** This allows us to determine the impact of the DDoS attack on the web server's bandwidth. Measured in Kilobytes per second

VI. Results

Before executing the DDoS attacks, the system metrics are taken to know the system's benchmark under normal performance with the IDS and XDR installed. Using a resource monitoring script [29], nine readings were gotten 60 seconds apart each. The metrics were as follows for the total duration of 120 seconds:

Table 2: Application metrics in normal operational mode after IDS and XDR deployment.

Count	Memory Usage (MiB)	CPU Usage (%)	Network Bandwidth (Kbps in/Kbps out)
1.	827	3.1	0.29/0.90
2.	827	0.0	0.06/0.45
3.	827	3.2	0.06/0.45
4.	827	0.0	0.41/1.78
5.	827	0.0	0.24/0.92
6.	827	0.0	0.00/0.00
7.	827	0.0	0.06/2.11
8.	827	0.0	0.00/0.00
9.	827	3.2	0.28/0.88
Average (Metric/9)	827Mi/1.9Gi	1.06%	0.156Kbps in /0.832Kbps out

Preliminary DDoS attacks were carried out for each attack in Table 1 above to validate the operationalization of the implemented architecture in Figure 2. The implementation involved creating custom Wazuh detection rules to utilize Suricata's network flow logging mechanism.[29] Once the architecture was configured and validated, the DDoS attacks were executed with the XDR running in blocking mode. Table 3 below shows the results of the DDoS attack on the web server in open and XDR blocking modes, respectively.

Table 3: RMM of the web server under attack at Stage 2, Stage 3, and 4.

Attack	Duration	Open mode (No blocking action)			Blocking mode (Detect and respond to attacks)		
		MU (MiB)	CPU (%)	Net. IO (Kbps)	MU (Mi)	CPU (%)	Net. IO (Kbps)
UDP Flood	120s	978.11	44.01	2034.817/ 144.201	1109.24	39.62	1225.321/ 120.414
ICMP flood	120s	994.11	33.12	1666.952/ 85.833	1,103.64	28.8	1080.371/ 88.631
SYN flood	120s	992.11	44.09	1562.153/ 1324.659	1126.4	32.96	1297.526/ 87.813
Network bandwidth exhaustion	120s	953.89	47.28	15203.363/ 1227.666	1137.78	24.33	8657.417/ 56.021
SSL/TLS Exhaustion	120s	1027.78	58.87	314.163/ 1019.283	1217.42	11.43	22.466/ 63.582
Malformed Packets	120s	964	1.04	160.804/ 0.326	1126.4	2.63	112.672/ 0.514
HTTP/S Flood	120s	1339.02	53.38	288.328/ 7159.614	1,308.44	6.311	92.22/ 329.363
Slowloris	120s	1228.8	2.08	3.293/ 5.868	974.1	4.3	3.599/ 23.359
HTTP GET/POST attack	120s	1297.07	56.63	189.669/ 2256.697	997.1	20.2	57.4/ 845.223

VII. Discussion

A. Statistical Tests

Based on the results, statistical tests were performed to determine the functionality of our proposed solutions and implementations.

1. The average values were as follows:

$$average = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Table 4: Average values of the RMMs.

	Normal operation average			
	MU (MiB)	CPU (%)	Net. IN (Kbps)	Net. OUT (Kbps)
Normal operation	827	1.06	0.156	0.832
Under DDoS attack Open mode (No blocking action)	1086.1	37.83	2380.39	1469.35
Under DDoS attack Blocking mode (Detect and respond to attacks)	1122.28	18.95	1394.33	179.44

B. Memory usage results

The statistical analysis shows that the average memory usage under regular operation was 827 MiB. Compared to the average of 1086.1 MiB in open mode and 1122.28 MiB in Blocking mode during attack operations, this shows a significant increase in memory consumption during the DDoS attacks. Memory consumption climbs progressively across the research stages during the attacks, with the average highest consumption occurring when the IDS, XDR, and custom blocking are operational. This leads us to conclude that the extra system operations required to execute the detection and blocking of the attacks cause an increase in the memory usage of the web server.

C. CPU usage results

The average CPU usage during normal operations was 1.06%. This means the CPU was near idle during the web server's regular operation. However, during the DDoS attack operations, the average CPU usage in open mode was 37.83% and 18.95% in blocking mode. The spike in CPU usage in open mode can be attributed to the various resource-consuming DDoS attacks that are not responded to, which causes the CPU to perform more computation and processing of network traffic. This is evidenced by SSL/TLS exhaustion attacks, which are resource-consuming attacks with the highest CPU consumption rate of 58.87%. Another contributor to the high CPU usage during open mode is the IDS operations to receive and send logs to the XDR in order to perform malicious activity detection, but no blocking. Given that the consumption of the IDS and XDR operating in open mode tripled the consumption of idle mode, we can assume that under active attack, the CPU consumption of the IDS and XDR is approximately 35%. The CPU metric in blocking mode was reduced to 18.95%. This indicates that the custom detection and the blocking using the IDS and XDR were successful, as the CPU consumption is less than the CPU consumption when processing huge packets.

D. Network In/Out (IO) results

From the research measurement metrics retrieved, and the results of our statistical analysis, we observed that the average network IO during regular operation was 0.156Kbps/0.832Kbps. During the DDoS attack operations, the average network IO in open mode was 2621.081Kbps/251.86Kbps with the IDS, XDR, and our custom detection rules running. In blocking mode, the average network IO was 1394.33Kbps/179.44Kbps. This indicates that the DDoS attacks, on average, flooded the web server with network packets as is expected, with the network bandwidth exhaustion attack having by far the highest values for network IN because it is a resource exhaustion attack, and HTTP/S flood attack having the highest average values for network OUT as the HTTP/S flood packets are receiving replies. However, while the network IN for open and blocking modes was close, the network out for blocking mode was significantly lower. This indicates that the custom detection rules triggering the blocking action work efficiently to prevent and suppress DDoS attacks.

VIII. Conclusions

This study investigated the effects of integrating an Intrusion Detection System (IDS) and Extended Detection and Response (XDR) with custom detection rules for detecting and responding to Distributed Denial of Service (DDoS) attacks. The study found that memory usage, CPU usage, and network IO were low during normal operations. However, during DDoS attack operations, there was a significant increase in memory consumption and CPU usage, especially when the IDS and XDR detections using custom rules were operational, but blocking offending IP addresses was not enabled. When the blocking of IP addresses was enabled, the CPU consumption was reduced. The network In/Out (IO) was measured. It was determined that although DDoS attacks flooded the web server with network packets, the custom detection rules combined with the Wazuh XDR blocking capability could suppress the attack and reduce the network IO to about half of the network IO when blocking was not enabled.

The strength of this research lies in the fact that it extended the detections an IDS provides by integrating it with an XDR and using custom detection rules to enhance the detection of DDoS attacks, then subsequently providing a mechanism to respond to those attacks. Additionally, practical tests were done, and resource monitoring was implemented to determine the implications of implementing the

proposed solution on the performance of the underlying infrastructure.

A. Limitations of the study

This study is limited by the infrastructure used in its execution. Some infrastructural limitations included a limited number of bots in the DDoS swarm, as we could only work with a randomized source IP because this research was executed in a virtual environment. Also, some DDoS attacks, such as amplification attacks, could not be executed because they would need to be executed against a DNS server, which may cause legal problems. Finally, the attack, detection, and responses have been observed for only 120 seconds. More research may be needed to determine how our proposal would work under a sustained onslaught for hours.

Availability of Data and Material

The data will be made available on reasonable request.

Funding

This research was self-funded by the authors.

Acknowledgments

Not applicable.

References

1. Deshmukh, R.V. and Devadkar, K.K. (2015). Understanding DDoS Attack & its Effect in Cloud Environment. Procedia Computer Science, 49, pp.202–210. doi:https://doi.org/10.1016/j.procs.2015.04.245.

2. Jabez, J. and Muthukumar, B. (2015). Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach. Procedia Computer Science, [online] 48, pp.338–346. doi:https://doi.org/10.1016/j.procs.2015.04.191.

3. George, A. S., George, H., Baskar, T., and Pandey, D. (2021). XDR: The Evolution of Endpoint Security Solutions - Superior Extensibility and Analytics to Satisfy the Organizational Needs of the Future. International Journal of Advanced Research in Science Communication and Technology (IJARSCT), 8(1), 493–501. https://doi.org/10.5281/zenodo.7028219

4. Prakash, A., Satish, M., Bhargav, T.S.S. and Bhalaji, N. (2016). Detection and Mitigation of Denial of Service Attacks Using Stratified Architecture. Procedia Computer Science, 87, pp.275–280. doi:https://doi.org/10.1016/j.procs.2016.05.161.

5. Bahashwan, A.A.; Anbar, M.; Hanshi, S.M. Overview of IPv6 Based DDoS and DoS Attacks Detection Mechanisms. In Communications in Computer and Information Science; Springer: Singapore, 2020; Volume 1132 CCIS, pp. 153–167.

6. Brandao, P. R., and Nunes, J. (2021). Extended Detection and Response Importance of Events Context. Kriativ-Tech, I(9). doi:https://doi.org/10.31112/kriativ-tech-2021-10-58.

7. Çakmakçı, S. D., Hutschenreuter, H., Maeder, C., and Kemmerich, T. (2021), A Framework For Intelligent DDoS Attack Detection and Response using SIEM and Ontology. IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 2021, pp. 1-6, doi: 10.1109/ICCWorkshops50388.2021.9473869.

8. crowdstrike.com. (2022). Open XDR vs Native XDR: Key Differences - CrowdStrike. [online] Available at: https://www.crowdstrike.com/cybersecurity-101/what-is-xdr/open-xdr-vs-native-xdr/ [Accessed 9 Apr. 2023].

9. Dantas Silva, F.S., Silva, E., Neto, E.P., Lemos, M., Venancio Neto, A.J. and Esposito, F. (2020). A Taxonomy of DDoS Attack Mitigation Approaches Featured by SDN Technologies in IoT Scenarios. Sensors, 20(11), p.3078. doi:https://doi.org/10.3390/s20113078.

10. Dayal, N., Maity, P., Srivastava, S. and Khondoker, R. (2016). Research Trends in Security and DDoS in SDN. Security and Communication Networks, 9(18), pp.6386–6411. doi:https://doi.org/10.1002/sec.1759.

11. Elliott, J. (2000). Distributed denial of service attacks and the zombie ant effect. IT Professional, 2(2), pp.55–57. doi:https://doi.org/10.1109/mitp.2000.839372.

12. Feinstein, L., Schnackenberg, D., Balupari, R. and Kindred, D. (2023). Statistical approaches to DDoS attack detection and response. Proceedings DARPA Information Survivability Conference and Exposition. doi:https://doi.org/10.1109/discex.2003.1194894.

13. Fekolkin, R. (2015). Intrusion detection & prevention system: overview of snort & suricata. Internet Security, A7011N, Lulea University of Technology. (Jan. 2015), 1-4.

14. Garg, K. and Chawla, R. (2011). Detection of DDoS attacks using Data Mining, International Journal of Computing and Business Research (IJCBR), 2(1), pp. 2229-6166.

15. Gómez, J., Gil, C., Padilla, N., Baños, R., & Jiménez, C. (2009). Design of a snort-based hybrid intrusion detection system. Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. Springer Berlin. pp.515-522.

16. Gupta, B. B., Badve, O. P. (2017). Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. Neural Computing and Applications, 28(12), pp.3655-3682.

17. Iudica, R. (2022). A monitoring system for embedded devices widely distributed. Webthesis Polito.it.

doi:https://webthesis.biblio.polito.it/secure/24599/1/tesi.pdf.

18. Jasmeen Kaur Chahal, Bhandari, A. and Behal, S. (2019). Distributed Denial of Service Attacks: A Threat or Challenge. New Review of Information Networking 24(2019) pp.31-103. doi:10.1080/13614576.2019.1611468.

19. Kali.org. (2022). siege | Kali Linux Tools. Available at: https://www.kali.org/tools/siege/#:~:text=Siege%20is%20an%20regression%20test,%2C%20concurrency%2C%20and%20return%20status. [Accessed 15 Apr. 2023].

20. Owasp.org. (2013). OWASP Mutillidae II, OWASP Foundation. [online] Available at: https://owasp.org/www-project-mutillidae-ii/ [Accessed 15 Apr. 2023].

21. Roopak, M., Tian, G. Y. and Chambers, J. (2020), An Intrusion Detection System Against DDoS Attacks in IoT Networks. 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0562-0567, doi: 10.1109/CCWC47524.2020.9031206.

22. Shiaeles, S. (2013). Real time detection and response of distributed denial of service attacks for web services. Doctoral Thesis, Democritus University of Thrace.

23. Slate, S. (2018). Endpoint Security: An Overview and a Look into the Future, Latin American Political History, doi: 10.4324/9780429499340-15.

24. Sousa Araújo, T. E., Matos, F. M. and Moreira, J. A. (2017). Intrusion detection systems' performance for distributed denial-of-service attack. 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Pucon, Chile, 2017, pp. 1-6, doi: 10.1109/CHILECON.2017.8229519.

25. Vishwakarma, R. and Jain, A.K. (2019). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommunication Systems, [online] 73(1), pp.3–25. doi:https://doi.org/10.1007/s11235-019-00599-z.

26. Wazuh (2023). Wazuh · The Open Source Security Platform. [online] Wazuh. Available at: https://wazuh.com/ [Accessed 16 Apr. 2023].

27. GitHub. (2024). xrisbarney/wazuh-suricata-ddos. [online] Available at: https://github.com/xrisbarney/wazuh-suricata-ddos [Accessed 1 Jun. 2024].

Alaa Q. Raheema. (2023). Threat Analysis in IOT Network Using Evolutionary Sparse Convolute Network Intrusion Detection System. *International Journal of Online and Biomedical Engineering (iJOE)*, 19(03), pp. 18–33. https://doi.org/10.3991/ijoe.v19i03.37571