******Swinburne University of Technology**

*Faculty of Science, Engineering and Technology*

**ASSIGNMENT AND PROJECT COVER SHEET**

Unit Code: COS3005 Unit Title: IT Security Assignment number and title: Practical Project (Assignment 1) Due date: 05/09

Lab group: Tutor: Yasas Akurudda Liyanage Don Lecturer:

Family name: Minh Hoang Identity no: 104487115

Other names: Duong

**To be completed if this is an INDIVIDUAL ASSIGNMENT**

I declare that this assignment is my individual work. I have not worked collaboratively, nor have I copied from any other student’s work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for me by another person.



Signature:



**To be completed if this is a GROUP ASSIGNMENT**

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number Name Signature

Marker's comments:

Total Mark:

**Extension certification:**

This assignment has been given an extension and is now due on

Signature of Convener: Date: / 2024

**COS3005 – IT Security**

*Practical Assignment (Assignment 1)*

**Topic**: *Denial of Service (DoS)*

**Student**: *Minh Hoang Duong (104487115)*

**Student Name**: *Minh Hoang Duong*

**Student ID**: *104487115*

**Word Count:** *(excluding reference, coversheet, title page, and multiple titles)*

**Due Date**: *05/09/2024*

**Submission Date**: *05/09/2024*

1. **Criteria 1: Planning and Justification**
2. **Overview**:

**Denial-of-Service (DoS)[1]** is a kind of malicious cyber threat in which the threat actor's main objective is to render computers, devices, network infrastructure, or websites unavailable to legitimate users. Taking advantage of the limited capacities that apply to any type of network, or computing infrastructures, the attackers flood the infrastructure resources with traffic or requests, making the system unable to process legitimate traffic or simply crashing due to running out of computation resources. As a result, the targeted service became unresponsive or significantly slower, causing disruptions in user experience and multiple damages to the targeted organization and owners.

**Distributed Denial-of-Service (DDOS)[1]** is a type of DoS intrusion that originates from many distributed sources. The attackers often operate multiple botnets (compromised devices) to conduct a big-scale DoS attack on the target.

1. **Type of DoS attacks**:

While every type of DoS attack has the same objective is to disrupt the target’s service, each type of DoS attack has its unique characteristics and methods. In terms of categorizing, there are three types of DoS attacks [2]:

* **Volume-based attack [2]**: a kind of DoS attack that includes flooding one server’s bandwidth with requests or traffic. For instance, ICMP/ping flood [2].
* **Protocol attack [2]**: a kind of DoS attack that takes advantage of the nature of the set of rules defined in internet protocol, often works at layer 2 or 3 of the OSI model. For example, SYN flood, SYN-ACK flood [2].
* **Application layer attack [2]**: a type of DoS attack that focuses on the application layer (layer 7) and its protocol. For instance, HTTP flood [2].

Even though multiple types of DoS attacks could be classified into multiple categories, the threat actors could combine multiple DoS methods, and create multiple attacks, which renders many difficulties for the defender.

1. **Threat Justification, Impacts, and Case Study**:

The impacts of the DoS attack could range from temporary service disruption to severe physical damage (high heat in computational devices) leading to financial losses, affecting user experience and the owner's, organization's reputation. On one hand, a DoS attack can also act as a distraction while other types of malicious activities are being carried out, leading to further damage [6].

Due to the fact that this type of malicious cyber-attack exploits the natural weakness that is applied to any network, or computational infrastructure, and is arguably easy to execute as there are many tools that support DoS and stress-testing features (LOIC, HOIC, T50, h3ping,…), this type of malicious intrusion is highly disruptive and popular in the field, ranging from highly skilled malicious attackers, penetration testers to hacktivist, cyber-criminal and script kiddies, reflecting through the following famous cases in the past.

**The 2018 GitHub Attack [9]:** On Wednesday, February 28th, 2018 the GitHub.com website, one of the largest source code management tools was rendered unavailable from 5:21 pm to 5:26 pm. The attackers took advantage of the Memcached instances, which are accessible through the public internet and support UDP communications, flooding them requests with spoofed IP addresses. By spoofing the IP address, the attacker could allow memcached’s responses to be diverted to the Github.com IP address and send more data toward that server, amplifying up to 51000 times[9].

**The 2016 Dyn Attack [11]:** On October 21st, 2016**,** a botnet DDoS attack was launched toward Dyn (now Oracle), a company that serves the majority of DNS hosting infrastructure, resulting in a severe outage of many of largest services including Paypal[12], Netflix[12], Reddit [12], Twitter [12],…The threat actors utilized the “Mirai Botnet”, which infected up to 100,000 IoT devices[11]  and sent multiple requests.

**The 2007 Estonia Attack [13]:** InApril 2007, a series of cyber-attacks were launched on websites of Estonian organizations and services including the Estonian Government, Banks, and Ministries. The intrusion is a multiple vector attack ranging from ping floods to botnet and DDoS attacks.

In addition, there has been an increase in DoS in 2024 as Cloudflare reported mitigating 4.5 million cases in Q1 and 4 million in Q2 [4], compared with 14 million DDoS attacks for the whole of 2023 [4], which is considered a 20% year-over-year increase [4]. According to Netscout analysis, there have been 7,035,170 attacks at the time of writing (7:38 pm, 29/08/2024) [5] with multiple attacking vectors such as SYN flooding, ACK flooding, UDP flooding, RIPv1 AMP [5].

Therefore, even though DoS is an arguably simple type of malicious attack, it still plays a crucial role in the open-vast cyber security world. As a result, it has been chosen to be the topic of this practical report.

1. **Tools Evaluation and Justification**

**Offensive tool**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool/Activities** | **Ease of Install** | **Amount of documentation** | **Community Activity** | **Available Features** |
| Slowloris [7] | * Simple installation. * Required pre-installed python | * Fairly documented. | * Open-source. | * Opening and maintaining many simultaneous HTTP connections. |
| T50 [8] | * Simple installation. * Pre-installed in Kali Linux. * Available in APT and YUM for Linux systems. | * Well documented * Fairly large official Documentation base. | * Open-source. * Barely active community. | * Multiple-protocol packet injector. * Supporting TCP, UDP, ICMP, IGMPv2,..[8] |
| Low Orbit Ion Cannon (LOIC) [9] | * Simple installation. * Cross-platform. | * Well documented. * Many user-based guides. | * Open-source * Barely active community. | * User Interface. * TCP, UDP, HTTP floods. |

Due to ease of use, simple installation compared to LOIC, and capabilities with the attacking method the chosen tool for this testing scenario is **Slowloris.**

**Defensive tool**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool/Activities** | **Ease of Install** | **Amount of documentation** | **Community Activity** | **Available Features** |
| Snort [14] | * Required installation. | * Well documented. * Public official documentation based. * Many user-based guides | * Open-source. * Highly active community. * Frequent update. | * Packet sniffing. * Snort rules. * Network intrusion prevention. |
| iptables [15] | * Part of the Linux system. * Based on Firewall. | * Well documented. * Large public official documentation based. * Many user-based guides | * Open-source. * Fairly active community. | * Managing incoming, and outgoing packets. * Blocking, allowing traffic. |
| Wireshark [16] | * Pre-installed in Kali Linux * Simple installation | * Well documented. * Large public documentation * Many user-based guides | * Open-source. * Highly active community. | * Packet sniffing, tracking, analyzing. * User-interface. |

In the scenario, **Iptables** is also used for mitigation due to the ease of usage and availability. In addition, **Wireshark** is used for monitoring, detection, and protocol, packet tracking due to its availability, ease of installation, and ease of usage on the defender sides.

1. **Scenario proposal**

|  |
| --- |
| The scenario is based on a type of Layer 7 DoS attack known as **Slowloris Attack**.  **Slowloris Attack[17]** is a type of layer 7 DoS attack in which the attackers exploit the behavior of HTTP communication. After establishing a reliable connection (TCP) between the host and the user, the user’s machine will send a request host’s server, and the host’s server will open a thread for each incoming request. Exploiting this process, the attacker will send a partial request without ending it, keeping the thread open and maintaining a simultaneous connection between the host and the attacker. To prevent the server from timing out, the attacker will periodically send an HTTP request header to keep the server up and prevent the target from handling legitimate requests. The attacks were originally developed by Robert Hansen (RSnake) in 2009[20], and demonstrated by Sam Bowne at DEFCON 17 [18].  Based on the definition explained above, the attacker and defender aims are categorized as below:   * **The attacker (VM1):** Establishing HTTP communication with the host (VM2), keeping the connection up, and preventing the host (VM2) from processing requests from the observer (VM3). * **The defender (VM2):** Restoring the connection from the observer (VM3).   The proposed scenario is carried out in an isolated environment which is created with VMware Workstation Pro, assuming each other has known their IP address and there will be no IP address spoofing.  VM1: 192.168.100.100.200/24  VM2: 192.168.100.183/24, VM3: 192.168.100.129/24 |
| **Pre-requisites**:   * VMware Workstation on a host machine. * Isolated connection between virtual machines. * VM1 (Attacker): Kali Linux, Slowloris. * VM2 (Target/Defender): Linux (Kali Linux), Wireshark, iptables. * VM3 (Observer): Any OS (XPPro), Web browser (Internet Explorer), Wireshark. * All the previously proposed tools are either pre-installed with the OS or installed before being isolated from the internet/host machine.   **Steps:**   * Startup VM1, VM2 and VM3. * Perform a Slowloris DoS attack from VM1 against VM2. * Observe the effect in VM2 and VM3 (expecting an indefinitely slow connection between VM3’s browser and the website hosted on VM2, multiple packets in Wireshark). * Apply iptables rules to mitigate the attack. * Observe the effect in VM2 and VM3 (expecting a recovered fast connection between the browsers). |

1. **Criteria 2: Application and Documentation**
2. **Attacker perspective (VM1)**

* **Initiating the attack process:**
  + The chosen attacking tool implements the original Slowloris attack developed by Robert Hansen (RSnake) in 2009 [20]. The tool is written by Gokberk Yaltirakli in 2015 [10].
  + By running “python3 slowloris.py <target IPv4>”, the attacking machine will execute the script in Python and start initiating the attack toward the target.
  + The tool also features optional flags while running the command which can be easily found in the official GitHub repository [10]. In this scenario, there will be 2 additional flags used.
  + “-s <number of sockets>” is the flag to specify the number of sockets used in the test. The intended sockets to be used in the test is 500.
  + **A screenshot of a computer

    Description automatically generated**“-v” is the flag to enable “verbose mode” or to display more information about the attack.

**A screenshot of a computer

Description automatically generated**Figure 2.1.1: Executing “python3 slowloris.py <target IPv4> -s <number of socket> -v”

Figure 2.1.2: There were 500 sockets (0-499) being used in the attack.

* + After establishing the sockets, the program will send the first “keeping alive” HTTP header before entering a 15s “sleeping” period as the explained tactic in the scenario proposal parts.
  + While running, after every 15 seconds, the attacking script will display a report about the number of remaining sockets and send out another pack of “keeping alive” HTTP headers.
* **The attacking process after applying the defense mechanism**
* **A screenshot of a computer

  Description automatically generated**After applying the defense mechanism from the defender's perspective, the attacking script started sending out “fail to create new socket: timed out” messages, and eventually, the sockets count number dropped to 20 (the connection limit specified below) as the program tried to create another 480 sockets for connection in the total of 500 sockets.

Figure 2.1.2: Connection timing out, creating 480 (500 – 20 = 480) new sockets message from the attacking script.

1. **Defender perspective (VM2)**

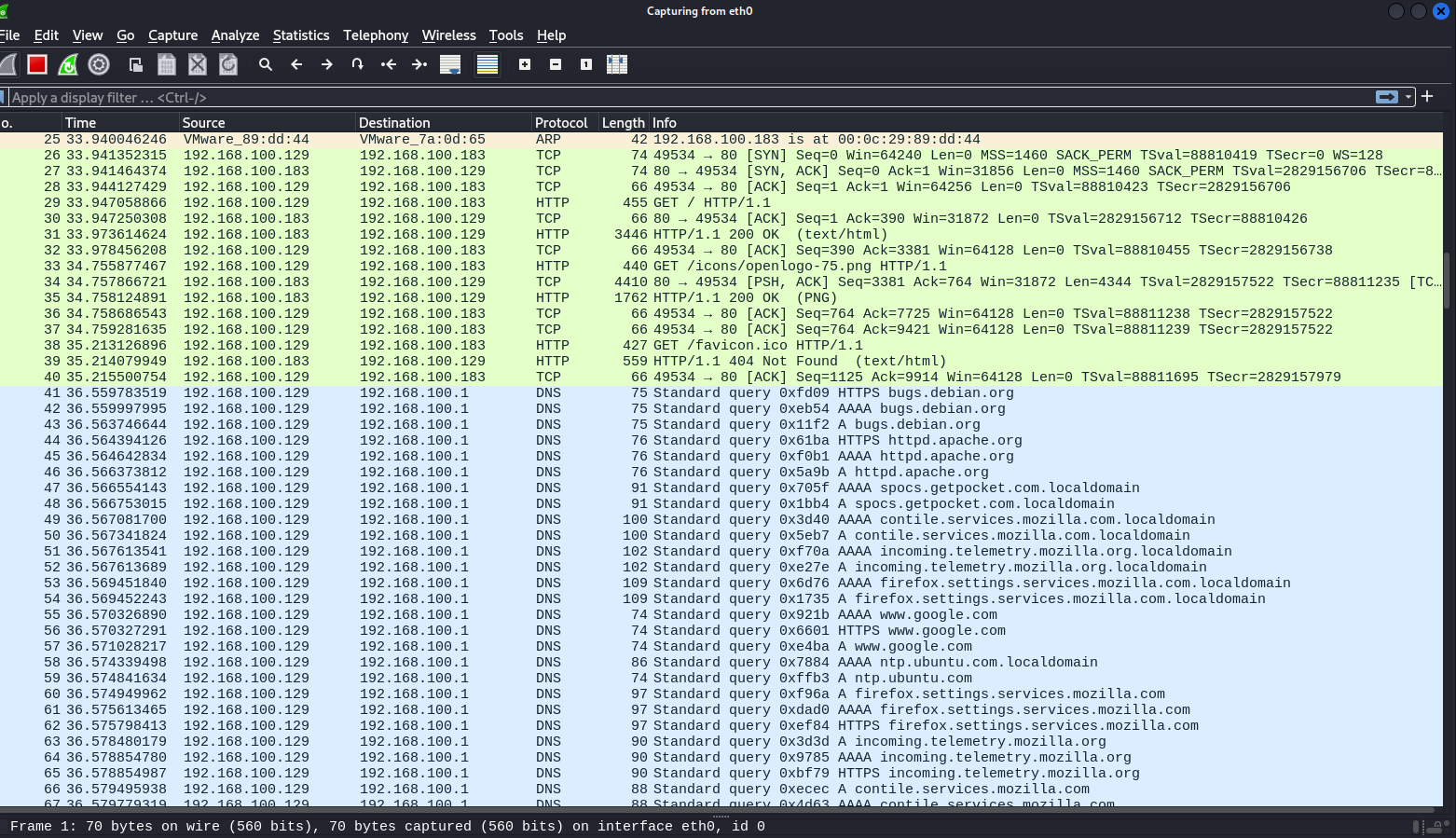
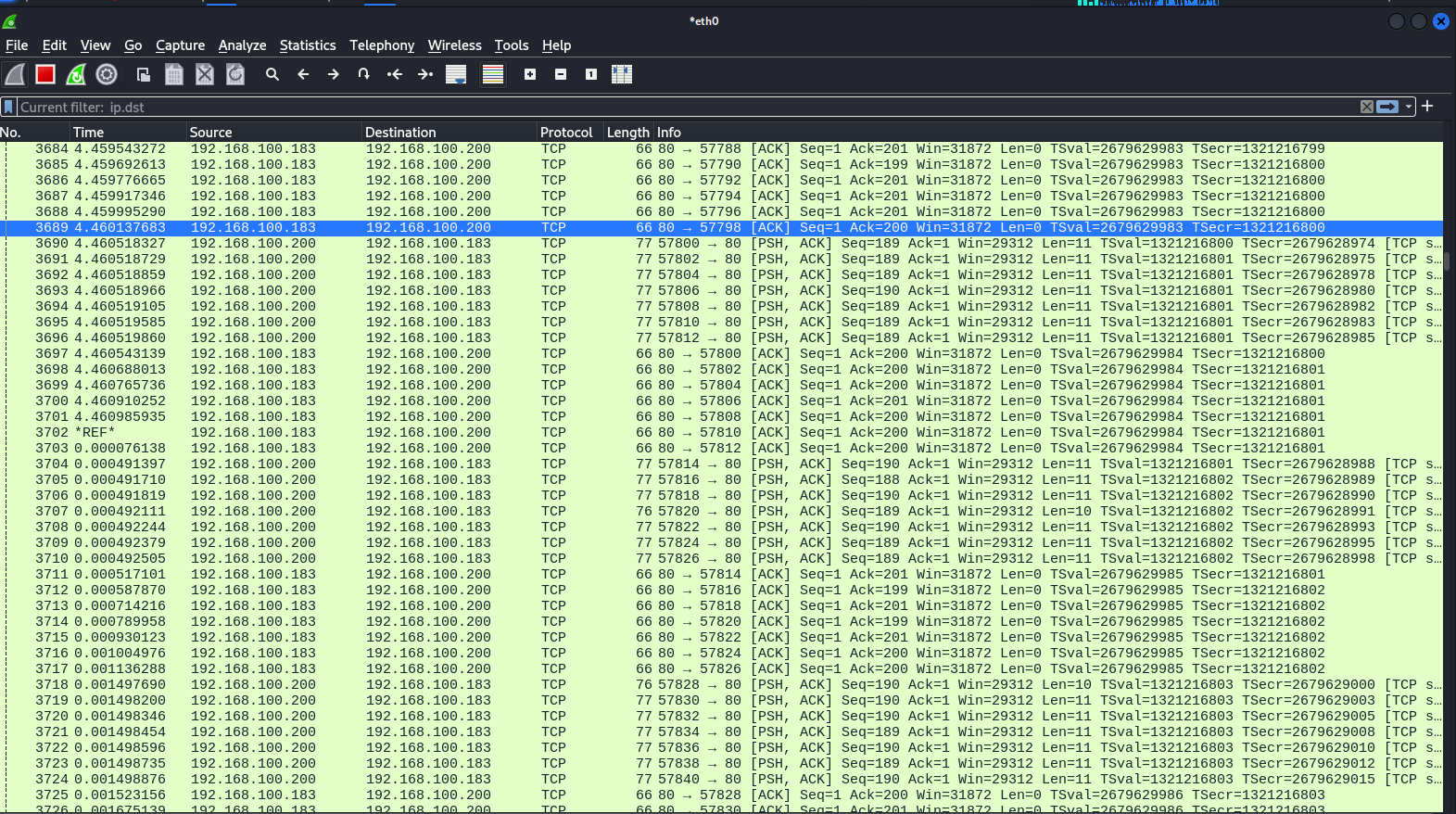
* **Wireshark observation before the attacks.**
  + The traffic remained normal with the TCP connects establishment and HTTP request from the observer (VM3), determined by the source and destination IPv4 address.

Figure 2.2.1: Traffic between the observer (VM3) and the defender (VM2).

* **Wireshark observation during the attacks, before applying defense mitigation.**
  + During the first stage of the attack, Wireshark displayed numerous TCP establishment packets between the host (VM2) and the attacker (VM1).

Figure 2.2.2: TCP establishment between each the attacker (VM1: 192.168.100.200) sockets and the host (VM2: 192.168.100.183)

* + After that is HTTP timeout requests which is an effort of the defender to disconnect from the attacker as explained in the scenario proposal.
  + Wireshark also captured the HTTP GET method which is mentioned in the “keep alive” headers, as an effort of the attacker to keep the connection up against network timing out.

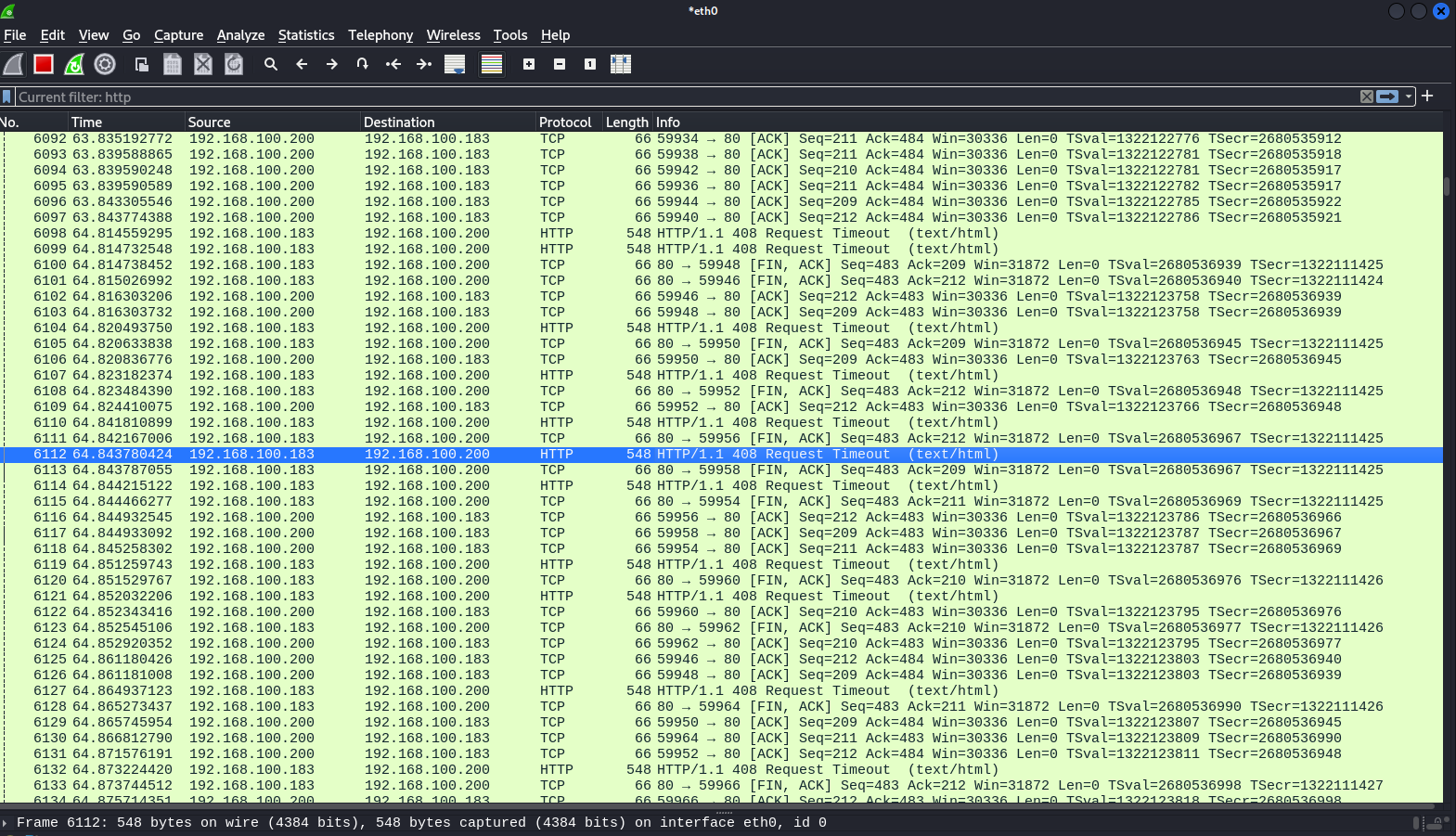


Figure 2.2.3: a mix of HTTP and TCP protocol packets in the traffic between the attacker (VM1: 192.168.100.200) sockets and the host (VM2: 192.168.100.183).

A screenshot of a computer

Description automatically generatedFigure 2.2.4: HTTP GET packet detected by Wireshark being sent from the attacker (VM1)

* **Applying defense mechanism**
  + Iptables is a firewall-based tool that is pre-installed in many Linux distributions. It allows for determining firewall rules that will control the tr incoming, outgoing, and forwarding traffic, from and to that machine.
  + Since this required modifying the firewall, the command must be run with the highest privilege (root).
  + The mitigation tactic is limiting the connection establishment packets between each IP address and dropping all packets with invalid TCP Flags, focusing mostly on the TCP establishment stages of the attacks.

**A screenshot of a computer

Description automatically generated**The default setting, which applies during the first attack, allows all inbound, outbound, and forwarding traffic to be processed. This could potentially be harmful to the machine if a cyber-attack occurs.

Figure 2.2.5: Default Iptables’ rules, allowing all INPUT, OUTPUT, and FORWARD traffic to be processed.

* The mitigation technique is applied through the following command:
  + iptables -A INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 -j DROP
  + iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
* The first command is to limit all incoming connections. The connection will be dropped if an IP address establishes more than 20 connections (20 SYN packets) to port 80 simultaneously.
* A screenshot of a computer

  Description automatically generatedThe second command is to drop all the TCP packets with no flags (bogus TCP packets).

Figure 2.2.6: Applying Defense Mechanism to Iptables.

1. **Observer perspective (VM3)**

* **Establishing connection between VM2 and VM3**
  + A computer screen shot of a computer screen

    Description automatically generatedUsing the Internet explorer browser, an HTTP connection between VM3 and VM2 was established with the address of “http://192.168.100.183”. Before the attack, the connection remained stable, displaying the default page of the Apache2 web-hosting system.

Figure 2.3.1: The default Apache2 webpage.

* **Establishing connection during the attack**
  + A screenshot of a computer

    Description automatically generatedWhen establishing a connection while the server is under attack, the browser remained loading for an undoubtedly long period of time, displaying the message “Website found, waiting for reply”, proving that the server is under Slowloris attack and the attacker had successfully kept the server “busy” from replying to the legitimate request.

Figure 2.3.2: “Website found, waiting for reply” message at the bottom left.

* A screenshot of a computer

  Description automatically generatedOn the other hand, if the observer (VM3) tried pinging (ICMP) the host (VM2) while the host was under attack and the webpage was waiting for a reply, the host machine still replied to the ping message, proving this type of attack only affect on the application layer (7).

Figure 2.3.3: ICMP ping works while the webpage is “waiting for reply”

* **Establishing connection after VM2 applied defense mechanism** 
  + A computer screen shot of a computer screen

    Description automatically generatedThe connection is recovered.

Figure 2.3.4: The website is recovered.

1. **Criteria 3: Analysis.**
2. **Scenario Analysis**

This documentation demonstrates one of many attack techniques that could be used in a cyber intrusion. During the first stage of the attack before applying the defense mechanism, Wireshark (from the defender’s perspective) recorded a massive number of TCP-establishing messages, which is the first step of this type of attack described in scenario proposal, Criteria 1. After that Wireshark (from the defender's perspective) also captured some HTTP request time-out packets and HTTP GET methods in the HTTP header which respectively the host's effort to disconnect from the attack and the attack’s effort to keep the connection up. Therefore, before the defender applies any defense mechanism, the attacker successfully achieves the aim mentioned in the Scenario proposal. Realistically, these are considered legitimate traffic, compared to SYN flood attacks in which the attacker sends many bogus TCP packets and could be detected through Wireshark, this type of attack cannot be detected by Wireshark observation.

On the other side, from the observer (VM3) viewpoint, the connection has been considerably slowed down, displaying “website found, waiting for a reply”, means the observer has successfully connected the host but it had to wait for the host to reply, carrying the website’s content. In addition, if the observer tries to ping the host, there are replies, that indicate the attack only affects the application layer (layer 7).

After applying the IP table rules to limit the number of connections to the host, the server starts rejecting massive amounts of simultaneous connections, which is reflected through the indication on the terminal from the running attack script, giving “space” for the host to process legitimate requests. The connection from the observer is also recovered. Therefore, the defender has successfully mitigated the attack, achieving the defender’s aim previously mentioned in the scenario proposal.

1. **Criteria 4: Evaluation**
2. **Attacking Evaluation**

In this scenario, The Slowloris has been successfully demonstrated as an effective method against HTTP web servers. This attack works by establishing many complete TCP connections between the attacker and the defender's port 80 (HTTP) of the defender. After that, the attacker will start sending partial HTTP requests to keep the connection up for an extended period, preventing it from responding to legitimate traffic.

The attacking method is highly effective in causing serious delays in response time, is low in bandwidth [17], and is hard to detect (as explained above in Criteria 3). Even though the attacking script used in the demonstration is not the original script written by Robert Hansen (RSnake), the tool is relatively more accessible to many cyber enthusiasts given that it was written Python.

On the other hand, the effectiveness of the attack still depends on several factors including the server’s configuration and technical specification. For example, a server that is configured to limit the number of simultaneous connections and connection time is less vulnerable. In addition, the attack is considered less effective against modern, well-configured servers that are designed to handle large amounts of requests.

1. **MITRE TTPs**

According to MITRE ATT&CK Matrix for enterprise, DoS and DDoS attacks are classified as the tactic of Impact (TA0040)[3] followed by many types of techniques including Endpoint Denial of Service (T1499)[3] and Network Denial of Service(T1498)[3]

The Slowloris attack could be classified as **Endpoint Denial of Service: Application Exhaustion Flood (T1499.003)[3].**

**Tactic**

**Impact (TA0040)**

* **Objective:** The primary goal of the attack is to disrupt the availability and deny operation of the service.
* **Context:** By maintaining multiple connections with the target, the attacker can keep the host server from processing legitimate or any other type of request.

**Technique**

**Endpoint Denial of Service: Application Exhaustion Flood (T1499.003)**

* **Description:** Established complete TCP connection. Sending partial legitimate HTTP request to keep the connection up, but the HTTP request is never finished.
* **Context:** Each thread-based web server hosting software such as Apache can only handle a certain amount of requests and use a timeout when they wait for an incomplete HTTP request, but it is set to 300 seconds by default. Therefore, establishing a large amount of connection and keeping the connection up will prevent it from processing other legitimate requests, causing a Denial of Service.

1. **Defending/Mitigation Evaluation**

The defense strategy implemented in this scenario is considered effective as limiting and restricting the number of concurrent access and requests is considered a realistic approach for DoS attack mitigation. In addition, as the mitigation method is applied through iptables into the firewall, the set-out rule is applied nearly immediately, reducing damage for the defender, therefore, even though the defender is overwhelmed before applying the security mechanism, **it is a win for the defender** as the attacker’s target has been blocked.

On the other hand, although iptables has successfully mitigated the impact of the slowloris attack, the application may not be optimal for all environments. Realistically, many defense methods could be used against the Slowloris attack including using well-known legitimate cloud cyber security providers like Cloudflare or implementing event-based web server hosting software like nginx which is considered immune to this type of cyber intrusion [21].

1. **Reference**
2. “Understanding Denial-of-Service Attacks | CISA,” *Cybersecurity and Infrastructure Security Agency CISA*, Feb. 01, 2021. <https://www.cisa.gov/news-events/news/understanding-denial-service-attacks>
3. Sharadin, G. (2023, December 20). DDOS attack types & mitigation Methods | Imperva. Learning Center. <https://www.imperva.com/learn/ddos/ddos-attacks/>
4. MITRE Corporation, "MITRE ATT&CK," <https://attack.mitre.org> (accessed Sep. 5, 2024).
5. “DDoS threat report for 2024 Q2,” *The Cloudflare Blog*, Aug. 27, 2024. <https://blog.cloudflare.com/ddos-threat-report-for-2024-q2/>
6. Netscout, “Global DDOS Threat Intelligence Reports | NETSCOUT,” *Latest Cyber Threat Intelligence Report*, Apr. 12, 2024. <https://www.netscout.com/threatreport/global-highlights/>
7. C. S. E. Canada, “Protecting your organization against denial of service attacks - ITSAP.80.100 - Canadian Centre for Cyber Security,” *Canadian Centre for Cyber Security*, Jul. 29, 2022. <https://www.cyber.gc.ca/en/guidance/protecting-your-organization-against-denial-service-attcks-itsap80100>
8. GeeksforGeeks, “Slowloris DDOS attack tool in Kali Linux,” *GeeksforGeeks*, Nov. 25, 2022. https://www.geeksforgeeks.org/slowloris-ddos-attack-tool-in-kali-linux/?ref=oin\_asr1
9. “t50 | Kali Linux Tools,” *Kali Linux*. <https://www.kali.org/tools/t50/>
10. S. Kottler, “February 28th DDOS incident report - the GitHub blog,” *The GitHub Blog*, Mar. 01, 2018. https://github.blog/news-insights/company-news/ddos-incident-report/
11. G. Yaltirakli [gkbrk], “GitHub - gkbrk/slowloris: Low bandwidth DOS tool. Slowloris rewrite in Python.,” *GitHub*, 2015. https://github.com/gkbrk/slowloris
12. “The 2016 DYN attack and its lessons for IoT security,” *MS&E 238 Blog, Standford Management Science and Engineering*, Jul. 30, 2017. https://mse238blog.stanford.edu/2018/07/clairemw/the-2016-dyn-attack-and-its-lessons-for-iot-security/ (accessed Aug. 30, 2024).
13. W. Turton, “This is why half the internet shut down today,” *Gizmodo*, Oct. 21, 2016. https://gizmodo.com/this-is-probably-why-half-the-internet-shut-down-today-1788062835
14. “StrATCOM | NATO Strategic Communications Centre of Excellence Riga, Latvia.” <https://stratcomcoe.org/publications/hybrid-threats-2007-cyber-attacks-on-estonia/86>
15. CrowdStrike and Lenaerts-Bergmans Bart, “Snort explained: Understanding Snort rules and use cases,” *crowdstrike.com*, Apr. 24, 2023. https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/snort-rules/ (accessed Aug. 30, 2024).
16. Khess, “Sysadmin tools: How to use iptables,” *Enable Sysadmin*, Jan. 12, 2023. <https://www.redhat.com/sysadmin/iptables>
17. R. Sharpe, E. Warnicke, and U. Lamping, “Wireshark User’s Guide,” *wireshark.org*. <https://www.wireshark.org/docs/wsug_html_chunked/>
18. V. Markova, “The Slowloris Attack: How it works and how to protect your website - ClouDNS blog,” *ClouDNS Blog*, Jan. 24, 4AD. <https://www.cloudns.net/blog/the-slowloris-attack-how-it-works-and-how-to-protect-your-website/>
19. Bowne, S., 2009. Hijacking Web 2.0 Sites with SSLstrip and SlowLoris -- Sam Bowne and RSnake (Robert Hansen) at Defcon 17. [video] Vimeo. Available at: <https://vimeo.com/7618090> [Accessed 4 September 2024].
20. R. Hansen [RSnake], “GitHub - XCHADXFAQ77X/SLOWLORIS: Slowloris HTTP DoS RSnake,” *GitHub*. https://github.com/XCHADXFAQ77X/SLOWLORIS?tab=readme-ov-file
21. Ivicti. (2023, May 29). *Slowloris attack*. <https://www.invicti.com/learn/slowloris-attack>
22. I. Muscat, “Mitigate slow HTTP GET/POST vulnerabilities in the Apache HTTP server | Acunetix,” *Acunetix*, Jun. 06, 2019. <https://www.acunetix.com/blog/articles/slow-http-dos-attacks-mitigate-apache-http-server/>