CSCI 6706 Network Design and Management

Term Project Report

Course Professor: Dr. Nur Zincir-Heywood

Prepared By

Yansong Li B00755354

Table of Contents

Section 1: Introduction

In this project, an experiment of penetration testing was conducted using the "Blue team vs. Red team" game pattern. The experiment was carried out on a Kali system virtual machine, so the Kali VM was treated as the attacker and a target network in GNS3 was used as the defender.

The GNS3 network provides web service, and the client's web pages require authentication for access. For the penetration testing, there were three activities involved, which are the Dos attack, password-cracking, and getting the root access of the server. There are two tools applied when it comes to the defense parts, which are the access control list (ACL) and Snort for port scan detection. Also, the penetration testing was conducted in two different set-ups, which are no-security set-up and as secure as possible set-up

In the rest of the report, I will illustrate the details of web service configuration in section 2. Then I will demonstrate the results that I got from the penetration activities and defense activities in section 3. Next, I will have an overall discussion about the results in section 4. Finally, I will make a conclusion about this project in section 5.

Section 2: Web Service Configuration

The target network was configured inside GNS3 using different appliances. Specifically, the network is composed of client-side dockers, a server-side docker, router dockers, switch dockers, and a Snort docker.

For the client-side docker, the Web term was initially used. The pulling of the image was successful, but it gave an internal server error while creating an instance from the template as shown in Figure 2.1. After several attempts, the client-side docker has been changed After

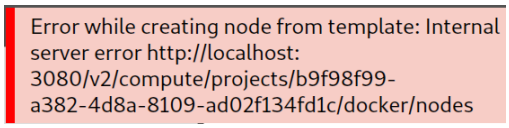

Figure 2.1 Web term error while creating a node.

several attempts, the client-side docker has been changed due to the unsolvable error. The Firefox appliance was used as an alternative for client-side docker. As the docker's name indicated, it has a web browser feature that can be accessed without any additional installation or configuration (jlesage, 2021). Unfortunately, this appliance gave an error while trying to start the docker, but the error was solved by adding *enable_hardware_acceleration = False*; *require_hardware_acceleration = False* two lines in gns3_server.conf file under [Qemu] section. Three Firefox dockers are used in the web service configuration with ipv4 addresses, 192.168.1.1, 192.168.1.2, and 192.168.1.3.

The Nginx appliance was deployed as the server-side docker. Nginx is an open-source reverse proxy server that supports HTTP protocol, which has a strong focus on high concurrency, high performance and low memory usage (NGINX, 2011). The docker was pulled and started successfully without any error. A single Nginx server was added to the web service configuration with ipv4 address 192.168.2.1.

Two types of routers are also used in the target network, which is the FRR router and the Cisco router. Except for connecting different sub-networks and forward packets, the primary functions of these two routers are keeping a copy of the monitored traffic by Snort and applying ACLs to the traffic. Two switches are also added to the configuration for dispatching appropriate packets to certain destinations, and a Snort docker for traffic monitoring. Besides, the network configuration also includes a Snort docker that is connected only with the FRR-router, which is prepared for port monitoring in defense activities.

After all the dockers are set and connected, and all the ports are configured with some additional routes added, the web service is initially completed with the following configuration shown in Figure 2.2. As a result, all three desktops can access the website that host by the webserver by

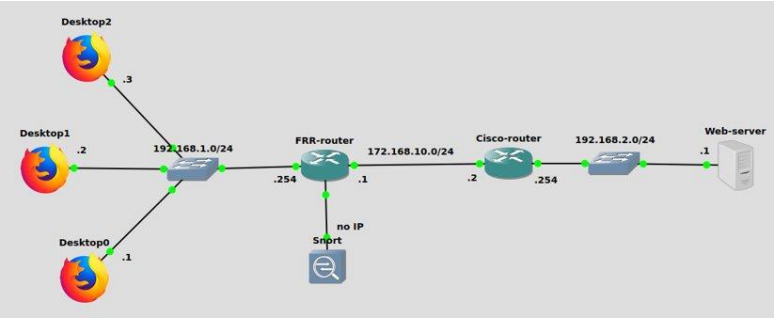

Figure 2.2 Web service configuration.

its ipv4 address192.169.2.1. In addition, three web pages that associate with each desktop was developed with password authentication using built-in functions of Nginx, which means each desktop can log in to their webpage using a unique username and password as shown in Figure 2.3(Quick Notepad Tutorial, 2016).

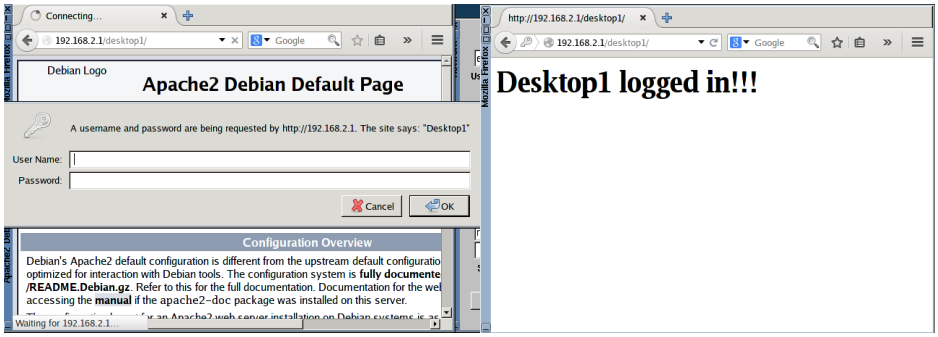

Figure 2.3 Authentication page and Homepage.

Another essential component of the network configuration is normal traffic generation because we have to make sure there are packets being delivered while penetration attacking and defencing. There were several different approaches applied for generating normal traffic during this experiment. More specifically, the primary method was logging in to the client's webpages from time to time using the Firefox browser; meanwhile, the webpage could also be accessed through the client's terminals using commands such as *curl* and *wget*. Besides, mutually pinging between clients and the server is also applied as one of the normal traffic generation approaches. Except for the manual approaches of generating normal traffic, *iperf* was also used as an automatic method for generating normal traffic from the server to clients. Since the web service configuration with authentication is completed and the generation of normal traffic is defined, the blue team vs. red team attack and defence game can be initiated.

Section 3: Results

The experiment was conducted in two phases, the first phase was Kali VM attacking the GNS3 web service configuration with no security set-up, and the second phase was Kali VM attacking the GNS3 web service configuration with the most security set-up that can be provided at this time. In this section, the results of both phases of penetration attacking and defending will be demonstrated.

Section 3.1 : Penetration Activies

There are three penetration activities that have been conducted during this experiment, which are password cracking of the client accounts, deny of service (Dos) attacking for blocking the server from clients, and obtaining the root access of the webserver using Nmap and Metasploit. The results of all these activities will be illustrated in detail under this section.

Password-cracking is a powerful approach to penetration, which can provide the access to a user's webpage without the permission of the user under the scenario of this experiment. Brute-force attacking is one of the most common methods of password-cracking, which is just attacking by submitting possible passwords relentlessly and hoping the correct password is one of the tried passwords. Meanwhile, the wordlist is usually an essential component of a brute-force attack, which is a collection of possible passwords in plain text. For instance, "rockyou.txt" is a wordlist that comes with the standard installation of Kali Linux, which contains millions of passwords that are commonly used. Hydra was applied as the tool for password-cracking due to its simplicity of input, the result of password cracking is as shown in Figure 3.1.1. As Figure 3.1 indicates, after 14,344,399 login tries, the password of client desktop0 is cracked with one

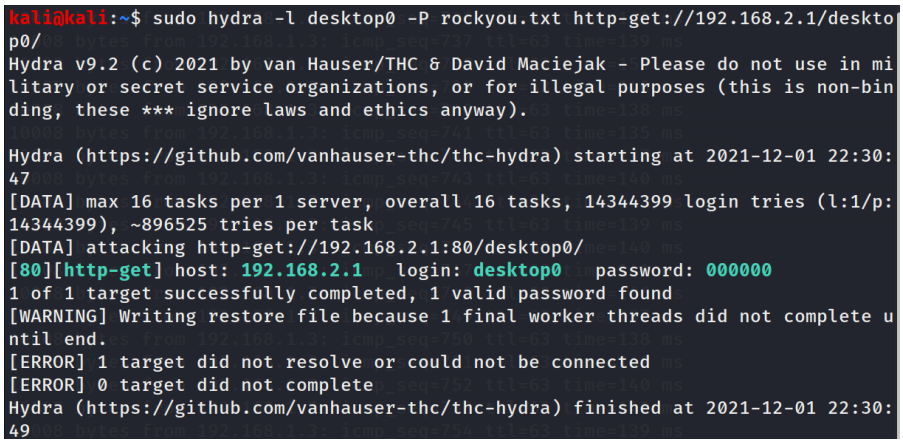

Figure 3.1.1: Password-cracking of desktop0 using Hydra.

potential password 000000. To verify the correctness of the password, *curl* commands was used, as shown in Figure 3.1.2, Kali VM was able to log in to the webpage of desktop0, which means the password-cracking is successful, Kali VM now can access desktop0's webpage without its permission. This penetration activity is succeeded.

Figure 3.1.2 The webpage of desktop0 accessed from Kali VM.

Dos attack was conducted as the second penetration activity, which is a scenario that "legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor" (CISA, 2019). In this experiment, *hping3* was used as the method to initiate the Dos attack. When the client desktops are under attack, they were unable to reach the server anymore either by ICMP echo message or HTTP message, and once the attack was terminated, several packets arrived at the server instantly with very high latencies as shown in Figure 3.1.3. For the performance of packet delivery of the other direction while Dos

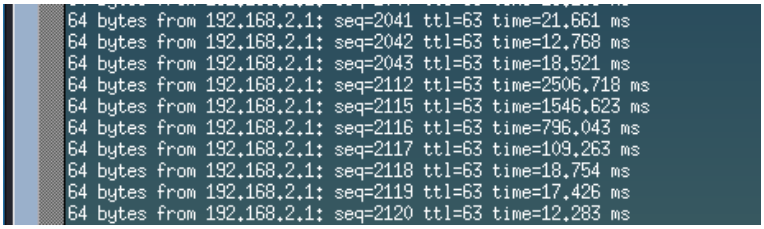

Figure 3.1.3: Packet delivers latencies caused by Dos attacks.

attacking, the server appeared unable to ping the client within a reasonable amount of latency as well or any other port that belongs to the network 192.168.1.0 such as the gate 192.168.1.254 on the FRR-router. However, it is interesting that the server can still reach the 192.168.2.0 network and the 172.16.10.0 network with latencies around 100ms, which is still quite noticeable compare to the normal traffic but much better than the latencies that take to the client's network.

Therefore, the Dos attack was launched successfully for both directions of packets deliveries in the network.

The last penetration activity that has been conducted in this experiment was obtaining the root accesses of the webserver. First, Nmap was used to find the open ports on the server, as shown in Figure 3.1.4 there are two open ports 80 with service HTTP and 8834 with service nessus-

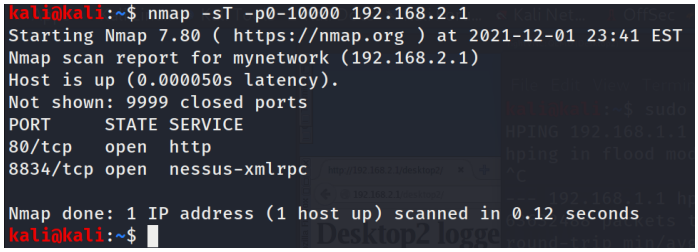

Figure 3.1.4: Open ports on webserver from port 0 to 10,000.

xmlrpc in the range of the first 10,000 ports. Once the open ports are confirmed, Metasploit was applied to exploit the open ports. When port 8834 was tested, it gave 1415 modules that can be exploited with different ranks. First, module 1415 with a rank of excellent was chosen to be exploited, the exploit was completed, but no session was created as shown in Figure 3.1.5. Next,

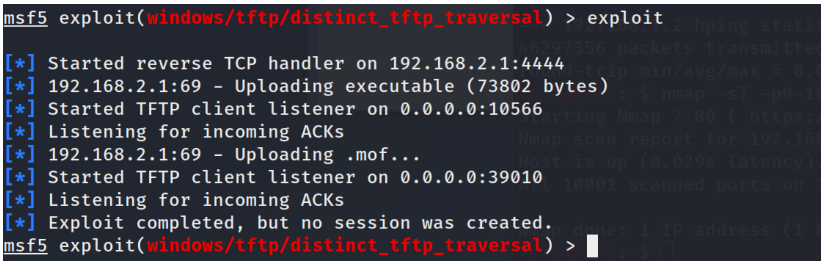

Figure 3.1.5: The exploit result of module 1415.

module 1384 with a rank of excellent was exploited but failed since the connection was refused by the remote host. Then, module 1303 with a rank of excellent was exploited as well and still failed because the permission was denied. In addition, port 80 was also tested as an alternative because of the failed exploitation of port 8834; however, after several modules with excellent ranks were exploited, there is still no available backdoor that can be found to gain the root access. As a result, the activity of obtaining the root access of the webserver was failed.

Section 3.2 : Defense Activies

Once the penetration testing with no security set-up is completed, the experiment moved to the second phase, which was building a secure set-up that can deal with penetrations. Two defence tools have been used for the defense activities, which are Snort and ACL. Snort is an open-source Intrusion Prevention System (IPS) that uses “a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users” (Roesch, 2021). In this experiment, Snort is used for Dos attack detection, ping detection, SSH connection detection, brute-force password cracking detection, and port scan detection. For ACL,

it “performs packet filtering to control the movement of packets through a network” (Cisco, 2018). In this experiment, standard ACLs are used for defending password cracking and Dos attacks and extended ACLs are used to deny traffic that uses a certain type of protocol to a particular direction.

For the deployment of snort, a snort docker is connected to the FRR-router with no Ip address. Also, before configuring the snort docker, the daemonlogger is applied on FRR-router for logging the network packets that go through the router, and the log file is named daemonlogger.pcap. Next, four rules were written in alerts.rules file for four types of detections that were planed respectively as shown in Figure 3.2.1. Meanwhile, the alerts.rules were added to the rules section

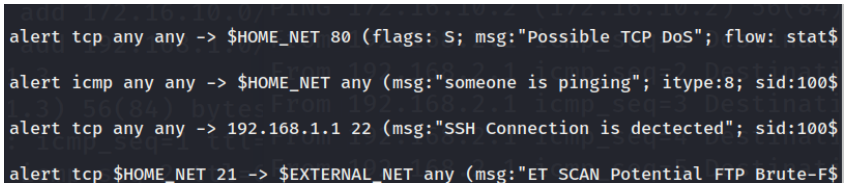

Figure 3.2.1: Rules of alert of Snort.

of snort.conf (the configuration file of Snort) for activating it. Besides, an extra line preprocessor *sfportscan: proto { all } scan_type { all } sense_level { high } logfile { portscan.log }* was also added to snort.conf for enable the Once the proposed rules have had added to the configuration of Snort, the traffic monitoring was started using the following command *snort -i eth0 -c /etc/snort/snort.conf -l /var/log/snort -A full*. While Snort is monitoring the traffic through the network, three penetration activities that were conducted in the first phase are repeated to verify the effectiveness of Snort. ICMP ping and SSH connection are also included as normal traffic.

After three penetration activities have been repeated and two normal connections have applied, the monitoring history was obtained from the alert file. As shown in Figure 3.2.2, Snort did catch the

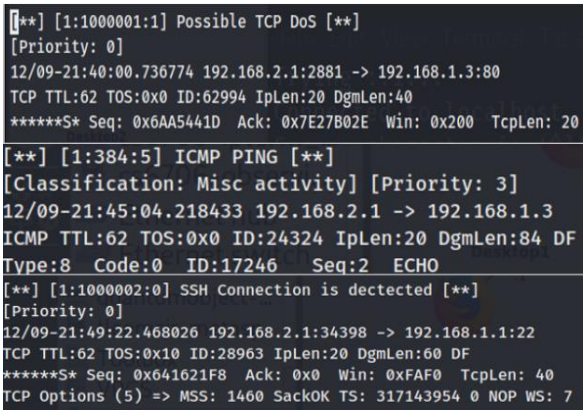

Figure 3.2.2: Alerts from Snort after penetration activities.

signs of Dos attack, ICMP ping and SSH connection in the traffic while there are penetration activities on-going, and since the amount of traffic was enormous during the Dos attack, the alerts that caused by it occupied most of the content of the alert file. However, there was no alert about possible brute-force attack in the alert file, which indicates the rule of potential FTP brute-

force attack was invalid, For the detection of the port scan, the results were collected in a different file called portscan.log as shown in Figure 3.2.3. As the log file indicates, there was a

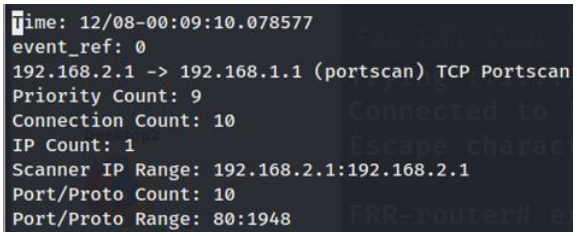

Figure 3.2.3: Monitored port scan using Snort.

port scan happened on 9:10 December 8th with scanner Ip 192.168.2.1 (Kali VM) and scanned Ip 192.168.1.1 (desktop). More specifically, there were 10 ports scanned with a port range from 80 to 1948. Therefore, the port scan there was conducted in phase 2 has been captured by Snort as well. Overall, there were five detections proposed, and four of them are successful.

While Snort was only able to detect malicious network activities, with the help of ACL, the GNS3 network will be able to block these malicious network packets. First, extended ACL was applied for blocking the ICMP echo message from the Kali VM to the clients. As a result, the Kali VM is no longer able to ICMP ping the clients anymore as shown in Figure 3.2.4. Meanwhile,

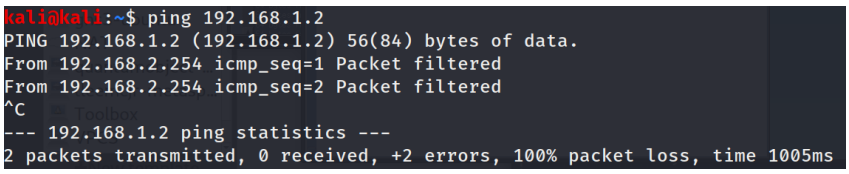

Figure 3.2.4: Blocked ICMP ping message from Kali VM to desktop1.

the clients are still able to ping the server since only one direction of ICMP ping has been blocked. Therefore, the extended ACL worked as expected.

The expectation of standard ACLs was to block the packets that either belong to Dos attack or brute-force attack because standard ACLs are meant to block all the traffic from a specific host. However, neither Dos attacks nor Hydra password cracking was successfully blocked by the standard ACL. Overall, the deployment of extended ACL was succeeded, but the standard ACL was failed.

Section 4: Discussion

Throughout this attack and defense game between Kali VM and GNS3, there were three penetration activities being conducted and two defense tools being deployed. During the phase of attacking with no security set-up, Kali VM was able to gain access to clients' webpages by cracking their passwords and blocking the network packets delivery between clients and the server using Dos attack, but getting the root permission of the server was failed.

More specifically, cracking the client's passwords used a brute-force attack by Hydra, which was just simply trying a huge number of possible passwords. The deployment of Hydra was straightforward forward and the cracking process was quick. On the one hand, the wordlist that was used by Hydra was collected and tested as the most possible passwords that have been used and. On the other hand, the client's password was set to be too easy (e.g., 000000) and allows unlimited attempts. Therefore, for security purposes, the passwords should be complex enough; for example, they must include at least one lower case letter, one upper case letter, one integer, and one special character. Also, two-factor authentication and a limited number of attempts would also be in favour of security.

Besides, the Dos attack was also launched successfully with an easy initiation process, and it affected the normal traffic significantly. For preventing or dealing with such an attack there are several options are available such as increasing the bandwidth, distinguishing and dropping from the identified source of the attack or malformed packets, and limiting the network broadcasting. Nevertheless, the obtaining root permission of the webserver was failed. The primary reason for that could be there were only two open ports found on the server, and only a few modules were exploited using Metasploit. Therefore, an efficient approach of choosing which module is going to be exploited should be developed because the number of open ports could be small but the number of modules that can be exploited could be large.

For the second phase attack with security set-up, two defense tools were applied, which are Snort and ACL. Snort successfully detected Dos attack, port scan, ping, and SSH connection but failed to detect FTP brute-force attack. Therefore, a more specific and valid rule that targets password cracking should be developed. As the ACLs, the extended ACL was able to block the ICMP ping packets from Kali VM to the clients and keep the ICMP ping packets transferring from the client to Kali VM. However, the standard ACLs were not able to filter the packets that belong to the Dos attack and FTP brute-force attack. Hence, other types of ACL could be applied to see if they can deal with Dos attack password-cracking attack.

Section 5: Conclusion

This attack and defense game was conducted using Kali VM as the attacker and a GNS3 network with web service as the defender. Three penetration activities and two defense tools were involved. When there is no security set-up, the Kali VM was able to crack the passwords of the webpages and access them, block the server from the client, and scan the open ports on the server and the network would not even know it. However, the root permission of the server was not successfully obtained due to insufficient open ports and unsuited modules exploitation. When there is security, the GNS3 network was able to detect the port scan, Dos attack, ping, and SSH connection from the Kali VM but not the password-cracking attack. Also, the network was only able to block ICMP ping from Kali VM but not Dos attack, nor password-cracking attack.

Overall, the penetration of the GNS3 network was successful but not completely successful. If Kali VM obtains root access, then it can have more control over the network such as manipulating the content of the web pages. Meanwhile, the defense of the network has progressed but not enough. With the help of Snort, most of the malicious packets can be detected but can not be blocked. For future work, it should be focused on how to optimize the deployment of Metasploit to obtain the root permission for attacking, and how to apply more various ACLs on the router to block the malicious packets. Furthermore, the deployment of firewalls should be considered as well for building a more secure network.

References

CISA. *Security tip (ST04-015)*. (2019, November 20). Retrieved December 11, 2021, from https://www.cisa.gov/uscert/ncas/tips/ST04-015.

Cisco. (2015, March 27). *Security configuration guide: ACLs, Cisco Ios XE release 3S - IP named ACLs [support]*. Retrieved December 13, 2021, from https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/xe-3s/sec-data-acl-xe-3s-book/sec-acl-named.html.

Jlesage. (2021, April 9). *Docker container for Firefox*. Docker Hub. Retrieved December 13, 2021, from https://hub.docker.com/r/jlesage/firefox.

NGINX. (2011). *Nginx*. Docker Hub. Retrieved December 9, 2021, from https://hub.docker.com/_/nginx.

NOURI, A. J. (2016, December 21). *Testing a container-based client-server web application in GNS3.* Gns3.com. Retrieved December 13, 2021, from https://gns3.com/managing-devices-with-ansibler-c.

Quick Notepad Tutorial. (2016, December 12). *How To Set Up Password Authentication with Nginx*. YouTube. Retrieved December 13, 2021, from https://www.youtube.com/watch?v=cXrhG81C4mg&t=65s.

Roesch, M. (2021). Snort. Retrieved December 13, 2021, from https://www.Snort.org/.

Appendix

Log files

The folder includes an alert file and a port scan log file.

https://dalu-my.sharepoint.com/:f:/g/personal/yn315233_dal_ca/EgK2UvYZG25DrD9vx6oDMB8BKPOFmL4JPsSVJdJZ5gaGgg?e=CSJ1CO