Practical Network Defense

Second Assignment - University "La Sapienza"

Group 27: Nicola Bartoloni ******* - Valerio Trenta 1856471

2020-13-06

1 Scope and Initial Considerations

The scope of this assignment comprehends the whole target network: a series of services provided by different machines in several subnetworks in the **ACME** environment are listed, and we focus on how to enable these services to the clients of our network by exposing them on a controlled manner through firewall rules specified in the **OPNSense** service provided by the two main routers of the network.
By catching a quick glimpse of the policy proposed by the assignemnt, we can tell the scope is to provide our network of a series of *white-listed* services that we want to be enabled and reachable from certain machines in the network itself, thus reducing the *attack surface* of the network by also excluding from this white list every other service that is not of interest.
Our analysis should begin with the services that we want to provide in the target network, and then on the firewall rules that make it possible to reach each service on each different machine.
Please notice that, since IPv6 addresses are not part of the scope of this assignment, we will ignore them during the configuration of the firewall rules in section 4.

2 Infrastructure Setup

We identify four main services which require machine-to-machine communication and, thus, are susceptible to the firewall rules that we'll define:

- an **Apache** web server reachable through **TCP** ports **80** (HTTP) and **443** (HTTPS) at **100.100.6.2**;
- a **DNS service** reachable through standard **UDP** port **53** at **100.100.1.2**;
- a **syslog server** reachable through standard **UDP** port **514** at **100.100.1.3**;
- a **proxy server** reachable through port **3128** at **100.100.6.3**, to be configured later on Assignemnt 3.

We thus identify two new subnetworks: the **Internal Servers** subnetwork **100.100.1.0/24** and the **DMZ** subnetwork **100.100.6.0/24**, and for the sake of defining the scope of this assignment and the setup of the whole infrastructure, we should also consider two additional details:

- Two of the four services - web server and syslog server - are already setup and running in the system: they do not need to be configured or launched;
- the **proxy server** is to be configured on the next assignment: for now, knowing from the **zentyal** portal that it is provided on port 3128 is enough.

This leaves us with the only **DNS service** to be configured at the moment. This step has been performed by following the instructions provided with the assignment: by accessing the **zentyal** portal at **100.100.1.2** on port **8843** - credentials have been changed - we added as forwarders the suggested IP addresses and specified the domain name - **acme.group27**. Then, the file located at **/etc/zentyal/dns.conf** on the aforementioned machine was modified including the target subnetworks (Clients and DMZ, that are the ones which will exploit the service).
At this point, pairs of IP addresses and hostnames were specified in the portal, so that the well-known machines of the network can now be associated with the following names:

- **kali**.acme.group27;
- **watchdog**.acme.group27;
- **dc**.acme.group27;
- **web**.acme.group27;
- **proxy**.acme.group27;
- **coffee**.acme.group27;

also, as suggested in the assignment, the external services machines were provided with an external DNS service such as 8.8.8.8 in their DHCPv4 configuration. Please notice also that the *logserver* was not given a name since it is only meant to be accessed by the SPOCK environment or by SSH and is not offering any browser-related service - and the same reasoning could be applied to the first two machines, *kali* and *arpwatch*, which we could exclude from this pairing list.
Last step was to actually modify the DHCPv4 settings in the two main routers to set the **dc** machine as the **DNS server** in the Internal Servers network, after having disabled the **Service Unbound DNS** in both routers.
Keep also in mind that some of the machines with predefined IP addresses (fixed, not assigned by DHCP service) had either to be configured externally, or had to have their */etc/resolv.conf* file re-configured by modifying the IP address of their nameserver (for instance, Proxy and Kali machines).
The **DNS service** configuration is tested in section 5.

3 Policy Evaluation

The proposed policy targets the four services listed in the previous paragraph and a series of machines which will either exploit or provide the corresponding services.
The best way of interpreting and understanding this policy is given by its fifth line:

- *"Anything that is not specifically allowed has to be denied"*;

which suggests the *white-listed* approach we have to adopt when defining the firewall rules: the policy is a list of **PASS** rules - meaning, rules that when matched will let the packet pass and continue its journey - while everything that doesn't match the rule has to be rejected, i.e. the packet must be dropped. Thus, we can devise a list of rules to apply at each of the two target routers and their corresponding effects on the network.

Internal Router:

- **Clients Interface**: only accept incoming packets on this interface with destination ports 80(HTTP), 443(HTTPS), 22(SSH), 53(UDP-DNS) and 3128(Proxy Service). Anything else will be discarder, since clients are not supposed to perform different actions and exploit different protocols than the aforementioned ones. For practical reasons, also packets on port 8443 (for **zentyal** panel service) are allowed - but this might be changed;

- **Servers Interface**: only accept incoming packets on this interface with destination port 53(UDP-DNS) or 514(UDP-SYSLOG, only if coming from **DMZ subnet**) and on port 22(SSH) only if coming from the **Clients subnet**, otherwise discard. This means that the two servers can only be reached by Clients or DMZ subnetworks, and only for the services that they provide (SYSLOG, DNS, SSH);

Main Router:

- **Internal Interface:** only accept incoming packets on this interface on ports 80(HTTP) and 443(HTTPS) between **Clients subnet** and **External services subnet**, or ports 22(SSH) and 3182(Proxy) between **Clients subnet** and **DMZ subnet**, or ports 53(UDP-DNS) and 514(UDP-SYSLOG) between **DMZ subnet** and **Internal services subnet**, otherwise discard. This means the Client host will only be able to reach the external web services, or the proxy to actually reach the WAN, and won't be able to do it directly by itself. Furthermore, SSH is eanbled for Client hosts to reach the **DMZ subnet** and the services offered by the **Internal servers** are reachable from the machines in the DMZ;

- **DMZ Interface:** we want the proxy server to be able to reach the internet via HTTP/HTTPS protocol, so we specify that this interface has to accept incoming packets specifically from the **Proxy server** and with **any** destination address on ports HTTP/HTTPS, and the same goes (but reversed) for the **web server** which we want to be accessible on ports HTTP/HTTPS from the Internet. However, the

proxy service itself must be only available for client hosts in the **Clients subnet**, so we should also specify that this interface should accept incoming packets on port 3182 of the Proxy machine with source address in the **Clients subnet**. The aforementioned rules for DNS/SYSLOG services in the Internal Services apply also here on this interface, while every other packet which is not *white-listed* must be rejected;

- **WAN Interface:** this interface should be the first line of prevention against intruders from the Internet, so it only has to accept - as specified by the policy - connections with destination port 80 or 443 on the **web server** machine in DMZ and connections to any other machine in the Internet on port 80 or 443 if they are initiated by the **Proxy** machine in DMZ: every other packet incoming on this interface should be rejected.

If implemented correctly, this policy should allow the Internal services to be exploited and reachable only by the **Clients or DMZ networks**, and it should also allow the client hosts in the **Clients** network to either reach the **external web services** or the Internet via the **Proxy service** to be configured in the next assignment, being this service in the **DMZ** the only one able to actually initiate connections with Internet, while the **web server** should be the only machine which can accept connections initiated by someone else outside our target network.
Notice that this may not be the only possible setup to actually implement the policy, and we will deal with this in the next paragraph.

4 Policy Implementation

While evaluating the policy in the previous paragraph, we have already proposed a high-level description of the rules to be implemented at each interface of the two firewalls. Notice that, for isntance, the rules we applied at the **Internal interface** of the **Main Router**, could have also been applied, instead, at the **External interface** of the **Internal Router** or, to have a complete in-depth defence, the same rules could have been applied at both interfaces - there is an option, indeed, to group rules applied to multiple interfaces in OPNSense - which could be the best solution in order to prevent the firewall to be taken down when one of the two machines is not working properly.
Also, there is another interesting fact about OPNSense firewall configuration: it tracks connections by default, so that every packet in a TCP connection which has already been *established* and has been accepted, is automatically accepted by default. This means that if we set a rule to allow machine A to ping machine B through ICMP protocol, machine B's response will be automatically accepted too by the firewall, while if machine B initiates a new connection by pinging machine A, the firewall will not allow it - and we can verify this on the testing paragraph.

5 Tests

We performed some basic tests for the DNS service and the new policy that were enabled and implemented in this assignment.
To perform thorough tests on the firewall rules we implemented, the WAN interface on the **Main router** was enabled, through the option provided in OPNSense, to accept private connections, so that it could be reached from external hosts (Internet, or our local machines).

5.1 Testing the DNS Service

5.2 Testing the policy

6 Final remarks

References

[1] *Just a placeholder.*