Sapienza
Università di Roma

Faculty of Information Engineering, Informatics and statistics

Practical Network Defense

Assignment 1

Students:
Natnael Solomon Fantu,
Elizaveta Lapiga,
Chiara Menghini,
Alessio Scanu

Academic Year 2023/2024

1 Initial Brainstorming

We made the decision to work on each objective one at a time using a sequential method. We executed the required configurations by concentrating on each bullet item separately, and we modified the rules as we went along. Through this approach, it was made sure that all aspects of the security policy was thoroughly examined and reinforced.

2 Evaluation of the security policy and Policy implementation

- **All the ACME hosts must use the internal DNS Server as a DNS resolver.**
  Initially, we configured *dnsmasq* on the DNS server by following a video guide from the Classroom. We updated the */etc/hosts* file with the provided IP addresses. Once the *dnsmasq* service is running, we need to inform the other hosts in the network to use the DNS server at IP address 100.100.1.2 as their DNS resolver. For the firewalls (FWs), we used the GUI to configure the DNS resolver by navigating to System > Settings > General. Since modifications to *resolv.conf* are not retained after a reboot, we decided to use a *cronjob* to modify the content of */etc/resolv.conf* at startup:

  ```
  @reboot echo "nameserver 100.100.1.2" > /etc/resolv.conf;
  echo "nameserver 2001:470:b5b8:1981:720b:3ef7:1acd:d9da"
  >> /etc/resolv.conf
  ```

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, TCP/ UDP	any	any	100.100.1.2/24	53
PASS, IN	IPv6, TCP/ UDP	any	any	2001:470:b5b8:1981: 720b:3ef7:1acd:d9da/64	53

Table 1: Internal FW. External interface.

- **The HTTP/HTTPS service provided in the DMZ has to be accessible from the Internet.**
  To make websserver be accessible from Internet we applied the following rules on the WAN interface of Main FW:

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, TCP/ UDP	any	any	100.100.6.2/24	80
PASS, IN	IPv6, TCP/ UDP	any	any	2001:470:b5b8:1906: 40fa:57ff:fe4a:2073/64	80
PASS, IN	IPv4, TCP/ UDP	any	any	100.100.6.2/24	443
PASS, IN	IPv6, TCP/ UDP	any	any	2001:470:b5b8:1906: 40fa:57ff:fe4a:2073/64	443

Table 2: Main FW. WAN interface.

- **The proxy service provided in the DMZ has to be accessible by the hosts of the ACME network and from the Internet.**
  Since proxyserver should be accessible only from Internet and ACME network we applied the following rules on the External interface of Main FW:

Policy	Protocol	Source	Src_port	Destination	Dst_port
BLOCK, IN	IPv4	any	any	100.100.6.3/24	any
BLOCK, IN	IPv6	any	any	2001:470:b5b8:1906: 74da:b5ff:fed2:952a/64	any

Table 3: Main FW. External interface.

We already set up the rules for DMZ at previous step, so there is no need to add anything else.

- **Besides the DNS resolver, the other services in the Internal server network must be accessible only to hosts of the Client and DMZ networks.**
  With these rules, we allowed the connection to Internal servers network from DMZ and Client network.

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4	DMZ net	any	100.100.1.0/24	any
PASS, IN	IPv6	DMZ net	any	2001:470:b5b8:1981::/64	any

Table 4: Main FW. DMZ interface.

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4	100.100.1.0/24	any	DMZ net	any
PASS, IN	IPv6	2001:470: b5b8:1981::/64	any	DMZ net	any

Table 5: Main FW. Internal interface.

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4	100.100.1.0/24	any	CLIENTS net	any
PASS, IN	IPv6	2001:470: b5b8:1981::/64	any	CLIENTS net	any

Table 6: Internal FW. Clients interface.

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4+6	SERVERS net	any	CLIENTS net	any

Table 7: Internal FW. Servers interface.

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, TCP/ UDP	100.100.6.0/24	any	SERVERS net	any
PASS, IN	IPv6, TCP/ UDP	2001:470: b5b8:1906::/64	any	SERVERS net	any

Table 8: Internal FW. External interface.

- **All the hosts (but the Client network hosts) have to use the syslog and the log collector services on the Log server (syslog) and Graylog server.**
  We have configured remote logging for the following systems:

- FWs;
- Hosts in the DMZ;
- DNS Server.

Logserver Configuration The Logserver is set up to use UDP on the standard port for the RSYSLOG protocol (514). We modified the */etc/rsyslog.conf* file with the following lines:

```
$template RemoteLogs,"/var/log/%HOSTNAME%/RemoteLog.log"
.* ?RemoteLogs
& stop
```

We also uncommented these lines to enable the UDP module and set the port:

```
module(load="imudp")
input(type="imudp" port="514")
```

Graylog Host Configuration
For the Graylog host, we used the GUI to add three different inputs that use UDP on three different ports:

- Port 1514 for the FWs;
- Port 1515 for the DMZ hosts;
- Port 1516 for the DNS server.

FW Configuration
For the two FWs, we configured remote logging through the GUI:

1. Navigate to System > Settings > Logging/Targets;
2. Add two remote log servers:
    - Logserver on port 514;
    - Graylog on port 1514.

DMZ and DNS Server Hosts Configuration
For the hosts in the DMZ and the DNS server, we used the terminal to configure remote logging. We created configuration files in the */etc/rsyslog.d* directory.
Logserver Configuration (Common for DMZ and DNS)
Create a file named *logserver.conf* with the following content:

```
$PreserveFQDN on
*.* @100.100.1.3:514:RSYSLOG_SyslogProtocol23Format
```

The *graylog.conf* file for the DMZ hosts is the following:

```
$PreserveFQDN on
*.* @100.100.1.10:1515:RSYSLOG_SyslogProtocol23Format
```

The *graylog.conf* file for the DNS server hosts is the following:

```
$PreserveFQDN on
*.* @100.100.1.10:1516:RSYSLOG_SyslogProtocol23Format
```

This setup ensures that all logs are appropriately directed to both the Logserver and the Graylog server, categorized by the source and type of host.

- **The Greenbone server must be able to scan all the network hosts.** Since we used the 'allow any connection' method we did not add any FW rule.

- **All network hosts must be managed via SSH only from hosts within the Client network.**

  We enabled the SSH service on each host within the ACME network by modifying the */etc/ssh/sshd_conf* file, specifically uncommenting and setting several key parameters. Additionally, we configured FW rules to restrict SSH connections, allowing access only from the Client network hosts.

  The parameters we uncommented and set in the */etc/ssh/sshd_conf* file are as follows:

  – Port 22

  – PermitRootLogin yes (changed from PermitRootLogin Prohibit_Password)

  – PasswordAuthentication yes

  SSH access is restricted from the following sources:

  – WAN (the internet);

  – External services;

  – Internal services (the servers).

  These measures ensure that SSH access is controlled and limited to authorized client network hosts only.

  At first we set an internal FW rules to allow SSH connection from Client network and then we blocked any other connection on all FWs' interfaces by implementing the following rules:

- **The Client network hosts have only access to external web services (HTTP/HTTPS) through the proxy server in the DMZ.**

  We set up the proxy server by installing Squid and on the other hosts we execute:

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4+6	any	any	any	22

Table 9: Internal FW. Client interface.

Policy	Protocol	Source	Src_port	Destination	Dst_port
BLOCK, IN	IPv4+6	any	any	any	any

Table 10: Every interface of every FW.

```
export http_proxy=proxyIP:3128
export https_proxy=proxyIP:3128
```

The Clients Network hosts can only access web services (HTTP/HTTPS) in the External Services Network (e.g., fantasticcoffee) through the proxy server. To enforce this rule, we added the following FW rules on the Clients interface:

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, TCP/UDP	any	any	100.100.6.3/24	3128
PASS, IN	IPv6, TCP/UDP	any	any	2001:470:b5b8:1906:74da:b5ff:fed2:952a/64	3128

Table 11: Main FW. Internal interface.

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, TCP/UDP	100.100.2.0/24	any	100.100.6.3/24	any
PASS, IN	IPv6, TCP/UDP	2001:470:b5b8:1982::/64	any	2001:470:b5b8:1906:74da:b5ff:fed2:952a/64	any

Table 12: Main FW. Internal interface.

- **Any packet the Main FW receives on port 65432 should be redirected to port 80 of the proxy host.**
  On the Main FW, we added two NAT rules:

Policy	Protocol	Source	Src_port	Destination	Dst_port	NAT IP	NAT port
DMZ, EXTERNAL, INTERNAL, WAN	TCP/ UDP	any	any	This FW	65432	100.100.6.3	80
DMZ, EXTERNAL, INTERNAL, WAN	TCP/ UDP	any	any	This FW	any	2001:470: b5b8:1906: 74da:b5ff: fed2:952a	80

Table 13: Main FW. All interfaces.

- **All the internal hosts should use the public IP address of the Main FW to exit towards the Internet.**
  Since the existing rule was already configured, we did not add additional NAT rules.

- **All the hosts of the ACME network should be able to ping (and receive replies of) the other hosts and the Internet hosts.**
  We added the following rules for each interface (excluding the external interface of the main FW):

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, ICMP_Echo_Req/Rep	any	any	any	any
PASS, IN	IPv6, ICMP_Echo_Req/Rep	any	any	any	any

Table 14: Main and internal FWs. All interfaces (besides external of main FW).

- **Only hosts in the DMZ should be reachable using the ping and traceroute tools from the Internet.**
  We have implemented a restriction on all incoming ICMP echo requests from the internet, except for those originating from the internet and directed towards the DMZ. For tracerouting we allowed the UDP traffic to DMZ network.

- **ICMP redirect packets should not cross any network.**
  We already implemented the necessary rules to complete this point (rule: BLOCK ALL). They can be seen in the next point, where we block all connections that are not explicitly specified.

- **Anything that is not explicitly allowed has to be denied.**
  Like we explained in SSH configuration section with the following rule we deny

Policy	Protocol	Source	Src_port	Destination	Dst_port
PASS, IN	IPv4, ICMP_Echo_Req	any	any	DMZ net	any
PASS, IN	IPv6, ICMP_Echo_Req	any	any	2001:470:b5b8: 1906::/64	any
PASS, IN	IPv4, ICMP_Echo_Rep	any	any	any	any
PASS, IN	IPv6, ICMP_Echo_Rep	any	any	any	any
PASS, IN	IPv4+6, UDP	any	any	DMZ net	any

Table 15: Main FW. WAN interfaces.

any other connection:

Policy	Protocol	Source	Src_port	Destination	Dst_port
BLOCK, IN	IPv4+6	any	any	any	any

Table 16: Every interface of every FW.

3 Tests of Configuration

- **All the ACME hosts must use the internal DNS Server as a DNS resolver.**
  To test this configuration we attempted to resolve the machine names using the ping command for each host. For example, on *client-ext1*, we used the following commands:

  ```
  ping webserver
  ping dnsserver
  ping proxyserver
  ```

- **The HTTP/HTTPS service provided in the DMZ has to be accessible from the Internet.**
  From the WAN network, we managed to successfully ping the webserver (100.100.6.2) located within our DMZ network.

- **The proxy service provided in the DMZ has to be accessible by the hosts of the ACME network and from the Internet.** We verified the ability to establish a connection from WAN and External hosts with the proxy server by using the ping command.

- **Besides the DNS resolver, the other services in the Internal server network must be accessible only to hosts of the Client and DMZ networks.** From the External Services hosts and our machine, we unsuccessfully ping hosts within the Internal servers network. Additionally, we verified if the DNS server was capable of resolving the hostnames. For example, from *client-ext1* we tried to do:

  ```
  ping graylog
  ```

  Although the ping itself failed, but the host successfully resolved the name *graylog*.

- **All the hosts (but the Client network hosts) have to use the syslog and the log collector services on the Log server (syslog) and Graylog server.**
  To check the logs, we used the command `logger` and send some message. For Graylog, we confirmed the log activity by reviewing the messages received by the Inputs in the web GUI. For the Logserver, we confirmed the log activity by examining the directories within the */var/log/* folder.

- **The Greenbone server must be able to scan all the network hosts.**
  We utilized the GUI of Greenbone to perform some host vulnerability assessments, and everything went smoothly.

- **All network hosts must be managed via SSH only from hosts within the Client network.**
  We attempted to establish SSH connections using the ssh command line from each host in the lab. As expected, SSH connections are only functional if they are started by Client Network hosts.

- **The Client network hosts have only access to external web services (HTTP/HTTPS) through the proxy server in the DMZ.**
  From a client machine, we tried to access the external website directly. The connection failed, indicating direct access is blocked. In the browser's settings, we set the proxyserver's IP address and port.
  Then we tried to access the external website and this time it was working. To check the blocked traffic we went to the OPNsense web interface, open Firewall > Log Files > Live View. We filtered the logs to verify direct HTTP/HTTPS traffic is blocked and traffic to the proxyserver is allowed.

- **Any packet the Main Firewall receives on port 65432 should be redirected to port 80 of the proxy host.**
  We tried to connect to each interface of the Main FW on the port 65432, with the browser(ex: 100.100.0.2:65432) or with Wget/curl, and we observed that the 'rendered' page is the fantasticcoffee's home page.

- **All the internal hosts should use the public IP address of the Main Firewall to exit towards the Internet.**
  To test this rule, we used tcpdump on our pc (that is connected from the Internet through the WAN interface). To connect inside the lab, we used the VPN. We started tcpdump on the interface created by the VPN on our machine with the command:

  ```
  sudo tcpdump -ni tap0
  ```

  Next, we pinged our machine (with the address 100.101.0.2) from the web server (100.100.6.2) and observed the capture from tcpdump. In the capture, the address 100.100.6.2 did not appear; instead, it was replaced by the address 100.100.0.2 (the address of the WAN interface).

- **All the hosts of the ACME network should be able to ping (and receive replies of) the other hosts and the Internet hosts.**
  To evaluate the configuration, we conducted a series of ping tests between all hosts and also included a host from the WAN network. For example, from Arpwatch we ping proxyserver and our machine:

  ```
  ping 100.100.6.3
  ping 100.101.0.2
  ```

- **Only hosts in the DMZ should be reachable using the ping and traceroute tools from the Internet.**
  As in previous test, we also conducted the trecerouting and ping the webserver and proxyserver from the internet. For example, from our machine we used the following commands:

  ```
  ping 100.100.6.3
  traceroute 100.100.6.2
  ```

- **ICMP redirect packets should not cross any network.**
  To test this configuration we performed the following commands on both FWs:

  ```
  sysctl net.ipv4.conf.all.send_redirects
  sysctl net.ipv6.conf.all.send_redirects
  ```

Since ICMP redirection was restricted, the outcome should have been 0, indicating that ICMP packets are not permitted to cross any FW interfaces. However, this test was not performed successfully, because `systmctl` command was not set up properly for test this configuration.

- **Anything that is not explicitly allowed has to be denied.**
  To check this configuration, we decided to check whether it is possible to perform a TCP traceroute from the WAN network to the proxy server using the following command:

  ```
  sudo tcptraceroute 100.100.6.3
  ```

  As a result, we were unable to successfully execute the command, although it was still possible to perform ping tests between the same hosts.

4 Final Remarks

We want to highlight an issue we noticed while working in the OPNsense environment: the web interface does not always indicate the reasons for service failures. For example, if a service is misconfigured, OPNsense may fail to start the service without displaying any error messages. While the interface is fairly user-friendly, referring to OPNsense's documentation is essential for understanding its behavior and properly configuring certain services.