ACME_19_a1_report

1 Team Info

Team Name: Guardians of the Gateway

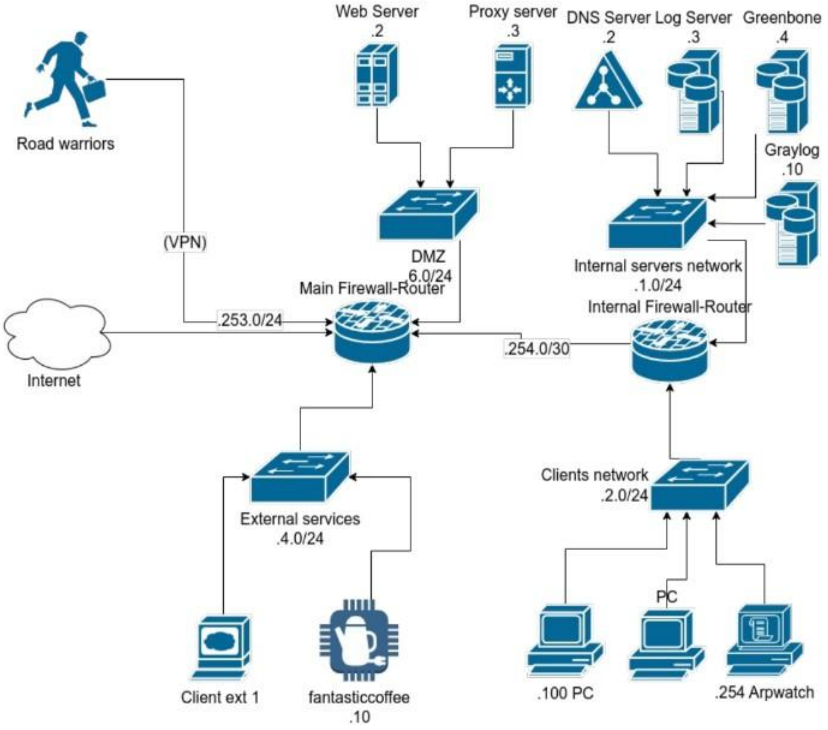
Team Number: 19 Team

Members:

- Simone Ciferri
- Matteo Concutelli
- Giorgio Pesce

2 Initial Approach

At the outset of this assignment, our team was not familiar with the OPNSense environment. However, as we progressed, we dedicated time to exploring the GUI of OPNSense. Through trial and error, and by referencing available documentation, we gradually became more confident in navigating the environment. We learned how to apply firewall rules, set up port forwarding rules, and configure various services required by the assignment. One of the most crucial discoveries was the "Live View" feature on the firewalls. This feature, along with the ability to log the rules, proved to be invaluable for debugging purposes.



After understanding the OPNSense environment and configuring the required services such as the web server, the proxy server, the DNS server, and others, we reviewed the firewall rules.

Based on the most critical rule of them all, "Anything that is not explicitly allowed has to be denied", we begun our firewall journey by essentially blocking everything, in order to apply the "least privilege" principle. In particular we implemented the following rules:

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	DMZ	out	any	any	IPv4/IPv6 - any	any	Block
Main	EXT. SERVICES	out	any	any	IPv4/IPv6 - any	any	Block
Internal	CLIENTS	out	any	any	IPv4/IPv6 - any	any	Block
Internal	SERVERS	out	any	any	IPv4/IPv6 - any	any	Block

Table 1: Block everything going inside every subnet

Notice that we didn't put any rule on the *main-INTERNAL* interface and on the *internal-EXTERNAL* interface, because we want the two firewalls to comunicate freely. Also, every interface, including the WAN interface (the Internet) has a "*Default deny/state violation rule*", that blocks everything going in the interface. So now all the traffic is completely blocked. All this rules are applied as last match. This means that every rule we're going to add will be applied *before* the blocking rule. After blocking all the traffic, we allow the comunication between the two firewalls:

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	INTERNAL	in	any	any	IPv4/IPv6 - any	any	Pass
Internal	EXTERNAL	in	any	any	IPv4/IPv6 - any	any	Pass

Table 2: Allow the communication between the gateways

We will now proceed to add the necessary rules to allow packets according to the defined policies.

3 Implementation

The security policy provided for the ACME network is comprehensive and covers essential aspects of network security. It specifies clear rules for various services such as the web server, DNS, syslog, and proxy server, ensuring that only authorized traffic is allowed.
However, the policy does not specify other important security measures, such as the use of authentication, network hardening, or the implementation of VPNs. These aspects will be addressed in subsequent assignments.

3.1 All the ACME hosts must use the internal DNS Server as a DNS resolver

To ensure a centralized and secure method of resolving DNS queries within the ACME network, we need to allow all ACME hosts to use the internal DNS Server as their DNS resolver.

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Internal	SERVERS	out	any	DNS	TCP/UDP - IPv4/IPv6	53	Pass
Main	DMZ	in	DMZ net	DNS	TCP/UDP - IPv4/IPv6	53	Pass
Main	EXT. SERVICES	in	EXT. SERVICES net	DNS	TCP/UDP - IPv4/IPv6	53	Pass
Internal	CLIENTS	in	CLIENTS net	DNS	TCP/UDP - IPv4/IPv6	53	Pass

Table 3: Allow ACME hosts to DNS

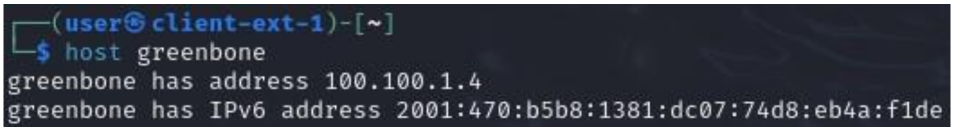
We don't need any firewall rule on the SERVERS interface of the internal firewall, since all the hosts are in the same network of the dnsserver.

In order to allow the DNS server to recursively contact external DNS servers, we add the following rule:

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Internal	SERVERS	in	DNS	any	TCP/UDP - IPv4/IPv6	53	Pass

Table 4: Allow recursive DNS requests

Testing the rules To test this rule, it's sufficient to perform the *host* command on any host of the network. For example on client-ex1:



```
┌──(user㉿client-ext-1)-[~]
└─$ host greenbone
greenbone has address 100.100.1.4
greenbone has IPv6 address 2001:470:b5b8:1381:dc07:74d8:eb4a:f1de
```

From the Internet, the command '*host greenbone 100.100.1.2*' doesn't work, since it's not requested by the policy.

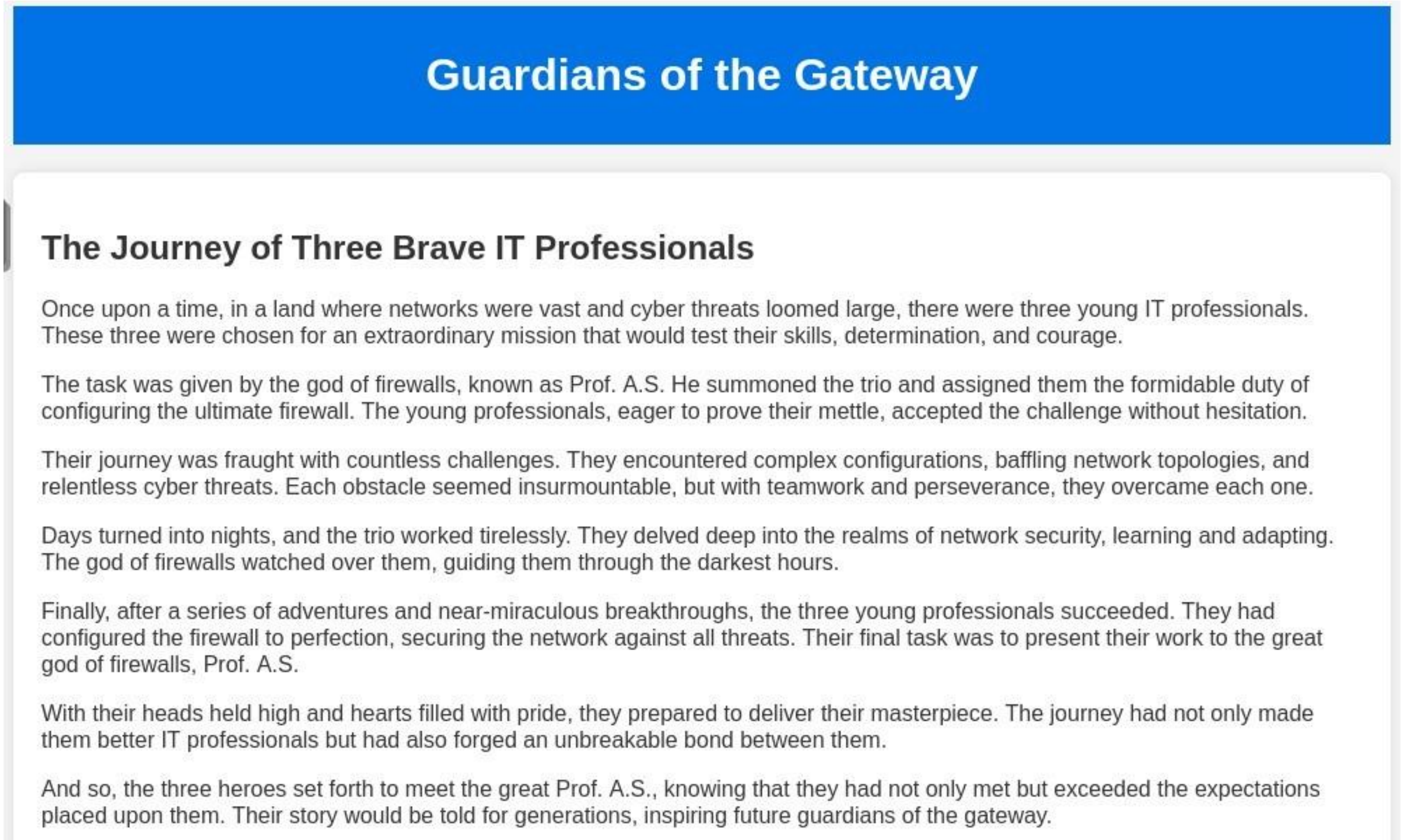
3.2 The HTTP/HTTPS service provided in the DMZ has to be accessible from the Internet

With the following rules we allow HTTP/S requests coming from the Internet to the Webserver, both on the WAN interface and on the DMZ interface of the main firewall.

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	WAN	in	any	Web server	TCP - IPv4/IPv6	80,443	Pass
Main	DMZ	out	any	Web server	TCP - IPv4/IPv6	80,443	Pass

Table 5: Allow Internet to Webserver

Testing the rules To test the correctness of this rules we can just try to access the web server from our katharà machine:



Guardians of the Gateway

The Journey of Three Brave IT Professionals

Once upon a time, in a land where networks were vast and cyber threats loomed large, there were three young IT professionals. These three were chosen for an extraordinary mission that would test their skills, determination, and courage.

The task was given by the god of firewalls, known as Prof. A.S. He summoned the trio and assigned them the formidable duty of configuring the ultimate firewall. The young professionals, eager to prove their mettle, accepted the challenge without hesitation.

Their journey was fraught with countless challenges. They encountered complex configurations, baffling network topologies, and relentless cyber threats. Each obstacle seemed insurmountable, but with teamwork and perseverance, they overcame each one.

Days turned into nights, and the trio worked tirelessly. They delved deep into the realms of network security, learning and adapting. The god of firewalls watched over them, guiding them through the darkest hours.

Finally, after a series of adventures and near-miraculous breakthroughs, the three young professionals succeeded. They had configured the firewall to perfection, securing the network against all threats. Their final task was to present their work to the great god of firewalls, Prof. A.S.

With their heads held high and hearts filled with pride, they prepared to deliver their masterpiece. The journey had not only made them better IT professionals but had also forged an unbreakable bond between them.

And so, the three heroes set forth to meet the great Prof. A.S., knowing that they had not only met but exceeded the expectations placed upon them. Their story would be told for generations, inspiring future guardians of the gateway.

3.3 The proxy service provided in the DMZ has to be accessible by the hosts of the ACME network and from the Internet

First of all, we allow the communication between the ACME hosts and the proxy server. Then we also allow the communication between Internet and the proxy server. Since we configured the proxy using squid, we open the proxy service on port 3128.

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	DMZ	out	any	Proxy server	TCP - IPv4	3128	Pass
Main	EXT. SERVICES	in	EXT. SERVICES net	Proxy server	TCP - IPv4	3128	Pass
Internal	CLIENTS	in	CLIENTS net	Proxy server	TCP - IPv4	3128	Pass
Internal	SERVERS	in	SERVERS net	Proxy server	TCP - IPv4	3128	Pass
Main	WAN	in	any	Proxy server	TCP - IPv4	3128	Pass

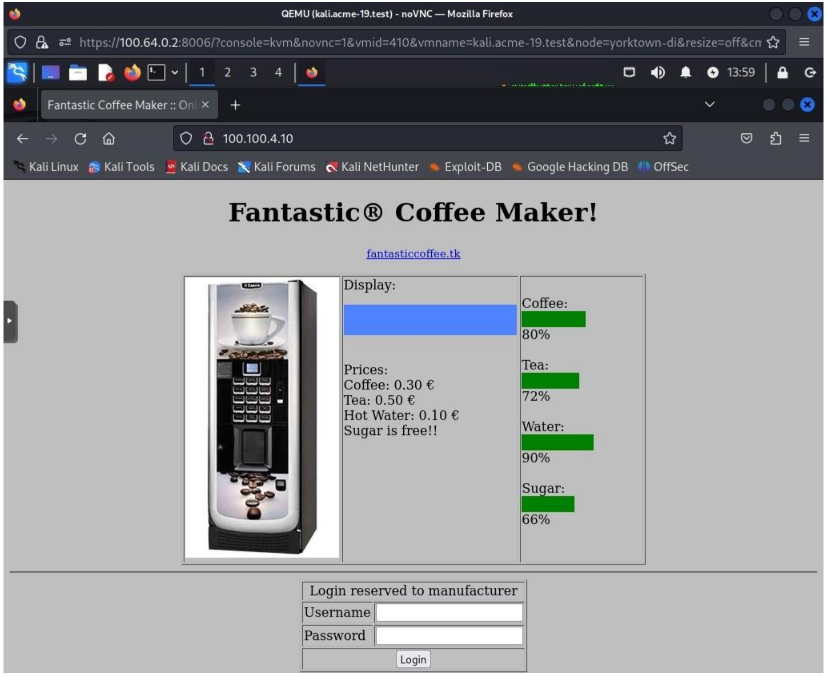
Table 6: Allow ACME hosts to Proxy server

Then, in order to have a properly working proxy service, we need to allow the proxy server to forward HTTP/S requests to the Internet and to the fantasticcoffee web service.

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	DMZ	in	Proxy server	any	TCP - IPv4/IPv6	80,443	Pass

Table 7: Allow Proxy server to forward HTTP/S requests

Testing the rules To test the correctness of these rules we can just try to access the *fantasticcoffee* web service, after properly configure the hosts to use the proxy(directly from firefox if you have a GUI, or by modifying the /etc/profile file). From Kali:


QEMU (kali.acme-19.test) - noVNC — Mozilla Firefox
https://100.64.0.2:8006/?console=kvm&novnc=1&vmid=410&vmname=kali.acme-19.test&node=yorktown-di&resize=off&cn
1
2
3
4
13:59
Fantastic Coffee Maker :: On
100.100.4.10
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec

Fantastic® Coffee Maker!

fantasticcoffee.tk

Display:

Prices:
Coffee: 0.30 €
Tea: 0.50 €
Hot Water: 0.10 €
Sugar is free!!

Coffee: 80%
Tea: 72%
Water: 90%
Sugar: 66%

Login reserved to manufacturer
Username
Password
Login

3.4 1) Besides the DNS resolver, the other services in the Internal server network must be accessible only to hosts of the Client and DMZ networks

2) 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

Since these two rules overlap a bit, we decided to describe them toghether. The services in the internal servers network that must be reachable, besides the DNS, are the log server and the graylog server.

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Internal	SERVER	out	DMZ net	Log server	UDP - IPv6	514	Pass
Internal	SERVER	out	EXT. SERVICES net	Log server	UDP - IPv6	514	Pass
Internal	SERVER	out	DMZ net	Graylog server	UDP - IPv4	514	Pass
Internal	SERVER	out	EXT. SERVICES net	Graylog server	UDP - IPv4	514	Pass
Main	DMZ	in	DMZ net	Log server	UDP - IPv6	514	Pass
Main	DMZ	in	DMZ net	Graylog server	UDP - IPv4	514	Pass
Main	EXT. SERVICES	in	EXT.SERVICES net	Log server	UDP - IPv6	514	Pass
Main	EXT. SERVICES	in	EXT. SERVICES net	Graylog server	UDP - IPv4	514	Pass

Table 8: Allow DMZ and EXT.SERVICES to Log and Graylog servers

Furthermore, we allow the Kali host to access *Greenbone*, since it's the only host in the policy that has a Graphical User Interface:

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Internal	SERVERS	out	Kali	Greenbone	TCP - IPv4	9392	Pass

Internal	CLIENTS	in	Kali	Greenbone	TCP - IPv4	9392	Pass

Table 9: Allow Kali to Greenbone web interface

We decided to allow only kali to the greenbone web interface, and not the proxy, even if rule 3.7 says that CLIENTS net hosts have only access to external web services(HTTP/S) through the proxy. We did this to restrict access to greenbone to every host that uses the proxy. Without this foresight, everyone also from the Internet would be able to access the Greenbone web interface, because of rule 3.3.

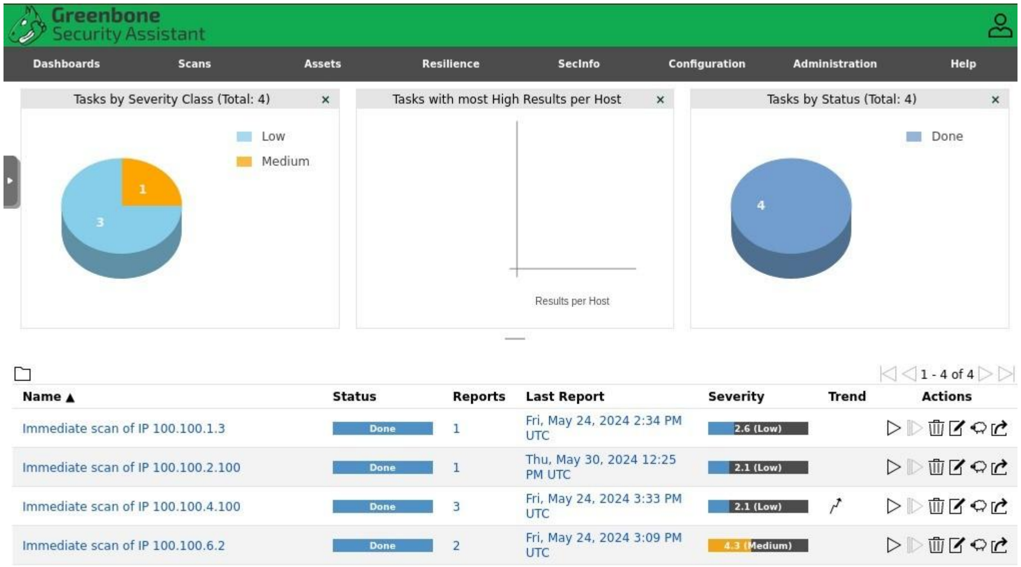
Testing the rules In order to test the correctness of these rules, it's sufficient to perform the command '*logger "some string"'* in the hosts of the DMZ and EXT. SERVICES networks. If the rules are correct, we should see the "*some string*" message inside */var/log/syslog* of the log server.

3.5 The Greenbone server must be able to scan all the network hosts

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Internal	SERVERS	in	Greenbone	any	any - IPv4/IPv6	any	Pass
Internal	CLIENTS	out	Greenbone	CLIENTS net	any - IPv4/IPv6	any	Pass
Main	DMZ	out	Greenbone	DMZ net	any - IPv4/IPv6	any	Pass
Main	EXT. SERVICES	out	Greenbone	EXT. SERVICES net	any - IPv4/IPv6	any	Pass

Table 10: Allow Greenbone to scan everything

Testing the rules We try to scan every network from the Greenbone web interface:



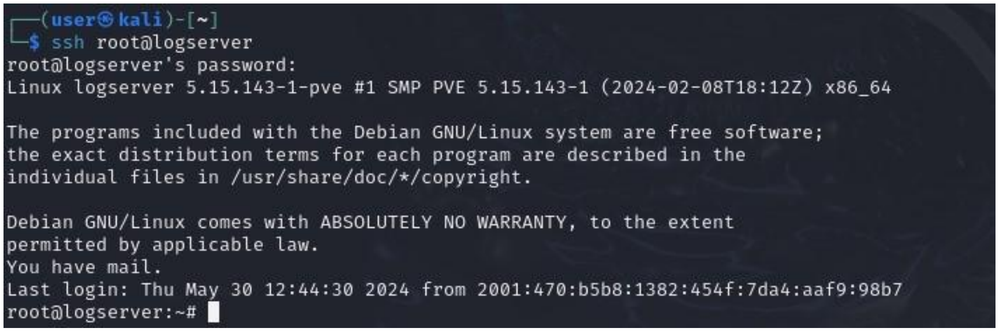
Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
Immediate scan of IP 100.100.1.3	Done	1	Fri, May 24, 2024 2:34 PM UTC	2.6 (Low)		
Immediate scan of IP 100.100.2.100	Done	1	Thu, May 30, 2024 12:25 PM UTC	2.1 (Low)		
Immediate scan of IP 100.100.4.100	Done	3	Fri, May 24, 2024 3:33 PM UTC	2.1 (Low)		
Immediate scan of IP 100.100.6.2	Done	2	Fri, May 24, 2024 3:09 PM UTC	4.3 (Medium)		

3.6 All network hosts must be managed via SSH only from hosts within the Client network

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Internal	CLIENTS	in	CLIENTS net	any	TCP - IPv4/IPv6	22	Pass
Internal	SERVERS	out	CLIENTS net	SERVERS net	TCP - IPv4/IPv6	22	Pass
Main	DMZ	out	CLIENTS net	DMZ net	TCP - IPv4/IPv6	22	Pass
Main	EXT. SERVICES	out	CLIENTS net	EXT. SERVICES net	TCP - IPv4/IPv6	22	Pass

Table 11: Allow CLIENTS to SSH everyone

Testing the rules To test the correctness of there rules we can try to access from kali to any host of the network, after correctly configuring the ssh deamon of the different hosts. For example:



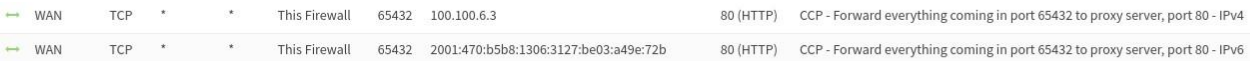
```
┌──(user㉿kali)-[~]
└─$ ssh root@logserver
root@logserver's password:
Linux logserver 5.15.143-1-pve #1 SMP PVE 5.15.143-1 (2024-02-08T18:12Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
Last login: Thu May 30 12:44:30 2024 from 2001:470:b5b8:1382:454f:7da4:aaf9:98b7
root@logserver:~# 
```

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

This rule is already inherently implemented. Since we don’t allow anything coming from CLIENTS net to port 80/443, but only to the proxy server on port 3128, the only way for the CLIENTS net to use HTTP/S services is to pass through the proxy. This is a case of FORWARD proxy.

3.8 Any packet the Main Firewall receives on port 65432 should be redirected to port 80 of the proxy host

To implement this policy in opnsense we used the ’NAT’ section. In particular we implemented two rules, one for IPv4 and one for IPv6, saying that everything that the main firewalls receives on port 65432 is redirected on port 80 of the proxy server.

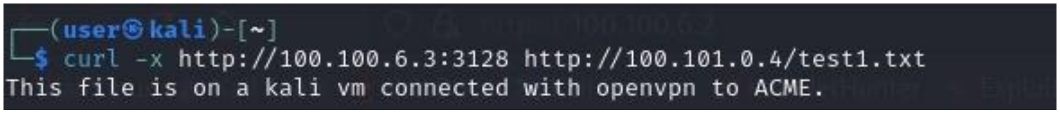


↔	WAN	TCP	*	*	This Firewall	65432	100.100.6.3	80 (HTTP)	CCP - Forward everything coming in port 65432 to proxy server, port 80 - IPv4
↔	WAN	TCP	*	*	This Firewall	65432	2001:470:b5b8:1306:3127:be03:a49e:72b	80 (HTTP)	CCP - Forward everything coming in port 65432 to proxy server, port 80 - IPv6

3.9 All the internal hosts should use the public IP address of the Main Firewall to exit towards the Internet

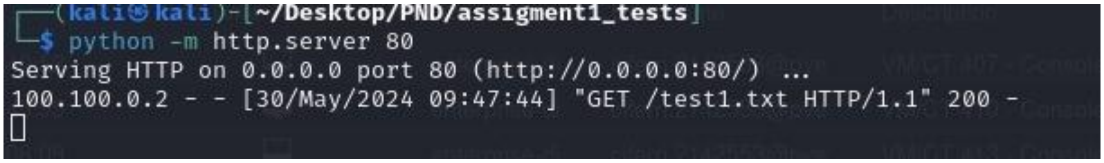
This rule is already implemented by the opnsense default policies.

Testing the rules To test this, we open a web server on our katharà machine(the Internet) with IP 100.101.0.4, and we try to connect to our webserver through the Kali machine, using the proxy server. On Kali:



```
┌──(user㉿kali)-[~]
└─$ curl -x http://100.100.6.3:3128 http://100.101.0.4/test1.txt
This file is on a kali vm connected with openvpn to ACME.
```

On Katharà:



```
┌──(kali㉿kali)-[~/Desktop/PND/assigment1_tests]
└─$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
100.100.0.2 - - [30/May/2024 09:47:44] "GET /test1.txt HTTP/1.1" 200 -
```

We can see that the web server receives a request from 100.100.0.2, which is the IP address of the Main Firewall in the WAN interface. This means that the NAT mechanism is working.

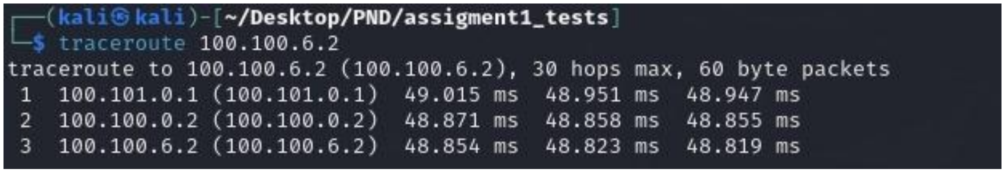
3.10 Only hosts in the DMZ should be reachable using the ping and traceroute tools from the Internet

Since our traceroute uses UDP on very high ports (33434-33534), we define the rules accordingly.

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	DMZ	out	any	DMZ net	ICMP Echo Request - IPv4/IPv6	-	Pass
Main	DMZ	out	any	DMZ net	UDP - IPv4/Ipv6	33434-33534	Pass
Main	WAN	in	any	DMZ net	ICMP Echo Request - IPv4/IPv6	-	Pass
Main	WAN	in	any	DMZ net	UDP - IPv4/Ipv6	33434-33534	Pass

Table 12: Allow Ping and Traceroute from the Internet to DMZ

Testing the rules To test this, we perform a traceroute from our katharà machine to the webserver:



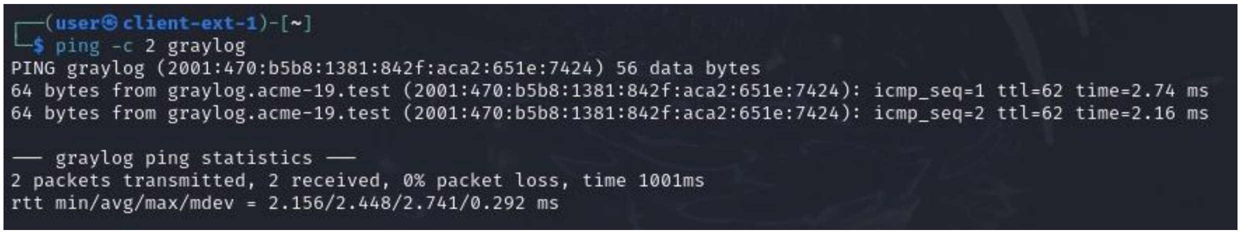
```
┌──(kali㉿kali)-[~/Desktop/PND/assigment1_tests]
└─$ traceroute 100.100.6.2
traceroute to 100.100.6.2 (100.100.6.2), 30 hops max, 60 byte packets
 1  100.101.0.1 (100.101.0.1)  49.015 ms  48.951 ms  48.947 ms
 2  100.100.0.2 (100.100.0.2)  48.871 ms  48.858 ms  48.855 ms
 3  100.100.6.2 (100.100.6.2)  48.854 ms  48.823 ms  48.819 ms
```

3.11 All the hosts of the ACME network should be able to ping (and receive replies of) the other hosts and the Internet hosts

Firewall	Interface	Direction	Source	Destination	Protocol	Port	Action
Main	DMZ	in	DMZ net	any	ICMP Echo Req. - IPv4/IPv6	-	Pass
Main	EXT. SERV.	in	EXT. SERV. net	any	ICMP Echo Req. - IPv4/IPv6	-	Pass
Main	EXT. SERV.	out	DMZ net	EXT. SERV. net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Main	EXT. SERV.	out	CLIENTS net	EXT. SERV. net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Main	EXT. SERV.	out	SERVERS net	EXT. SERV. net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	CLIENTS	in	CLIENTS net	any	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	CLIENTS	out	DMZ net	CLIENTS net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	CLIENTS	out	EXT. SERV. net	CLIENTS net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	CLIENTS	out	SERVERS net	CLIENTS net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	SERVERS	in	SERVERS net	any	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	SERVERS	out	DMZ net	SERVERS net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	SERVERS	out	EXT. SERV. net	SERVERS net	ICMP Echo Req. - IPv4/IPv6	-	Pass
Internal	SERVERS	out	CLIENTS net	SERVERS net	ICMP Echo Req. - IPv4/IPv6	-	Pass

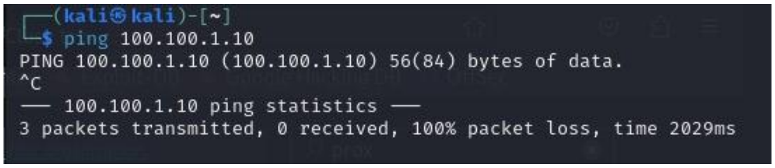
Table 13: Allow ACME network to ping other hosts and the Internet

Testing the rules To test the rules, we can try to ping from any ACME host to anywhere, for example from *client-ext-1* to *graylog*:



```
┌──(user㉿client-ext-1)-[~]
└─$ ping -c 2 graylog
PING graylog (2001:470:b5b8:1381:842f:aca2:651e:7424) 56 data bytes
64 bytes from graylog.acme-19.test (2001:470:b5b8:1381:842f:aca2:651e:7424): icmp_seq=1 ttl=62 time=2.74 ms
64 bytes from graylog.acme-19.test (2001:470:b5b8:1381:842f:aca2:651e:7424): icmp_seq=2 ttl=62 time=2.16 ms

── graylog ping statistics ──
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 2.156/2.448/2.741/0.292 ms
```

But if, from the internet, we try to ping some host(other than the DMZ), it doesn't work, as requested by the policy:



```
┌──(kali㉿kali)-[~]
└─$ ping 100.100.1.10
PING 100.100.1.10 (100.100.1.10) 56(84) bytes of data.
^C
── 100.100.1.10 ping statistics ──
3 packets transmitted, 0 received, 100% packet loss, time 2029ms
```

3.12 ICMP redirect packets should not cross any network

Since we never explicitely allow ICMP redirects packets anywhere on the firewalls, this policy is already implemented.

3.13 Anything that is not explicitly allowed has to be denied

This rule is inherenlty satisfied thanks to the default policy we set at the beginning.

4 Final Remarks

The key takeaways from our configuration include:

- Comprehensive Coverage: The firewall rules are designed to cover all necessary interfaces and directions.
- Segmentation and Security: By segmenting the network into DMZ, External Services, Clients, and Servers, we isolate different network zones, reducing the risk of unauthorized access and limiting the impact of potential security breaches.
- Flexibility and Scalability: The current rule set is very strict, yet flexible enough to accommodate future network expansions.