

Assignment 1 Report

Group 10

Tommaso De Nicola - 2006686
Lorenzo Colombini - 1973692
Mattia Romano - 1982886
Riccardo Capobianco - 1884636

Initial Brainstorming

We initially decided to complete every step of the assignment following the order given by the professor.

At this stage we decided not to implement IPv6 Rules on Firewalls because it wasn't clearly requested and all the tasks could be done through the use of IPv4.

To facilitate creating firewall rules we created some aliases on the 2 routers, in order to group up the networks and to keep the work cleaner.

Evaluation and Implementation of Security policies of ACME co.

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

The internal dns server has IP 100.100.1.2 and resides in the Internal Servers Network. First, we had to configure dnsmasq in order to use such DNS, in particular: we used the file hosts to specify all the entries.

The next step was to configure in every hosts the file /etc/resolv.conf as follows:

```
root@webserver:~# cat /etc/resolv.conf
# --- BEGIN PVE ---
#search acme.corp
#nameserver 151.100.4.13

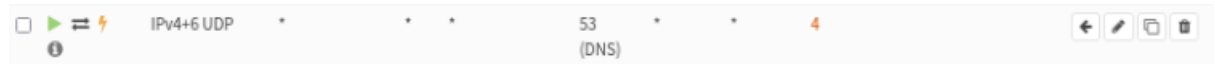
search acme-10.test
nameserver 100.100.1.2
nameserver 2001:470:b5b8:a81:1b8:8db5:c93:cf12

# --- END PVE ---
root@webserver:~#
```

This made every server use the DNS to resolve hostnames, *in the tests section at the end of the document we verify such claim.*

2. The DNS Server should be able to answer DNS requests coming from the Internet.

In order to make every server use the DNS we implemented an initial firewall rule, which was then edited to include this point, allowing requests coming from the internet.



3. The HTTP/HTTPS service provided in the DMZ has to be accessible from the Internet. It can also be accessed using the WAN address of the main firewall.

To allow this, we set up two port forwarding NAT rules from the main firewall to the DMZ as follows:



As well as the following firewall rules, which we implemented through the use of an alias for the IP of the web server.



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

The Proxy server installed is Squid Proxy, and we configured it to proxy all the HTTP and HTTPS traffic, with port as default at 3128.

The hosts which we configured to make use of such server have 2 new environment variables within their respective .bashrc / .zshrc files:

```
http_proxy=100.100.6.3:3128
https_proxy=100.100.6.3:3128
```

Then we added the following rule to allow traffic towards the proxy itself:



Finally, in order to let the proxy perform HTTP/HTTPS requests to the internet, we added the following rule:



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

We implemented this through the following set of rules:

<input type="checkbox"/>		IPv4+6 TCP	CLIENTS net	*	graylog	80 (HTTP)	*	*			
<input type="checkbox"/>		IPv4+6 TCP	CLIENTS net	*	graylog	9200	*	*			
<input type="checkbox"/>		IPv4+6 TCP	CLIENTS net	*	greenbone	9392	*	*			
<input type="checkbox"/>		IPv4+6 TCP	DMZ_net	*	graylog	80 (HTTP)	*	*			
<input type="checkbox"/>		IPv4+6 TCP	DMZ_net	*	graylog	9200	*	*			
<input type="checkbox"/>		IPv4+6 TCP	DMZ_net	*	greenbone	9392	*	*			

While the DNS already has the previously implemented rule to allow traffic on port 53.

- 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 implement this, we configured the `/etc/rsyslog.conf` in every host, appending the 2 lines within the image on the right:

```
*. * @graylog:1514
*. * @logserver:514
```

We then set up the correct firewall rules to make such logging possible:

<input type="checkbox"/>		IPv4+6 UDP	*	*	graylog	1514	*	*	3			
<input type="checkbox"/>		IPv4+6 UDP	*	*	syslog_server	514	*	*	3			

Finally, we inserted a listening (Input) rule in through the graylog interface (using the kali machine) and configured the machine to listen and log UDP traffic on the log server, through the `/etc/rsyslog.conf` file:

The screenshot shows the Graylog web interface at `100.100.1.10/system/inputs`. A new input named "Logging UDP" (Syslog UDP) is being configured. The configuration details are as follows:

- Global inputs:** 1 configured
- Logging UDP (Syslog UDP):** (6811dd02be5b9947f849eff)
 - Configuration:**

```
allow_override_date: true
bind_address: 0.0.0.0
charset_name: UTF-8
expand_structured_data: false
force_rdns: false
number_worker_threads: 2
override_source: <empty>
port: 1514
recv_buffer_size: 262144
store_full_message: false
timezone: NotSet
```
 - Throughput / Metrics:**
 - 1 minute average rate: 22,468 msg/s
 - Network IO: 3.7MiB (total: 4.7GiB)
 - Empty messages discarded: 0
- Local inputs:** 0 configured

```
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")
```

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

To implement this, we added this rule to both firewalls:



8. All network hosts must be managed via SSH only from hosts within the Client network, so be sure that all the hosts have the SSH service up and running.

To allow ssh traffic only from the Client network, we inserted the following rule in both firewalls:



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

We already implemented this feature when taking on task number 4. We tested the correct behavior of the clients in the final section of this document.

10. Any packet the Main Firewall receives on port 65000 should be redirected to port 80 of the proxy host.

To accomplish this task we created a forwarding rule for the WAN interface as follows:



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

We inserted the following NAT outward rules in the main firewall:

<input type="checkbox"/>		WAN	CLIENTS_net	*	*	*	WAN address	*	NO
<input type="checkbox"/>		WAN	SERVERS_net	*	*	*	WAN address	*	NO
<input type="checkbox"/>		WAN	DMZ net	*	*	*	WAN address	*	NO

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

We implemented this by inserting these rules on both firewalls:

<input type="checkbox"/>			IPv4 ICMP	*	*	*	*	*	*	3	Request
<input type="checkbox"/>			IPv6 ICMP	*	*	*	*	*	*	3	Request
<input type="checkbox"/>			IPv4 ICMP	*	*	*	*	*	*	3	Reply
<input type="checkbox"/>			IPv6 ICMP	*	*	*	*	*	*	3	Reply

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

To accomplish this we added the following egress rules::

<input type="checkbox"/>		IPv4 ICMP	*	*	DMZ net	*	*	*	Request			
<input type="checkbox"/>		IPv6 ICMP	*	*	DMZ net	*	*	*	Request			
<input type="checkbox"/>		IPv4 ICMP	*	*	DMZ net	*	*	*	Reply			
<input type="checkbox"/>		IPv6 ICMP	*	*	DMZ net	*	*	*	Reply			
<input type="checkbox"/>		IPv4 ICMP	*	*	DMZ net	*	*	*	Destination Unreachable			
<input type="checkbox"/>		IPv6 ICMP	*	*	DMZ net	*	*	*	Destination Unreachable			
<input type="checkbox"/>		IPv4 ICMP	*	*	DMZ net	*	*	*	Time Exceeded			
<input type="checkbox"/>		IPv6 ICMP	*	*	DMZ net	*	*	*	Time Exceeded			

- ICMP redirect packets should not cross any network.

We denied ICMP redirection on both routers by creating a floating rule for every interface that explicitly denies the ICMP Redirect traffic.

<input type="checkbox"/>		IPv4 ICMP	*	*	*	*	*	*	Redirect			
<input type="checkbox"/>		IPv6 ICMP	*	*	*	*	*	*	Redirect			

Also we noticed that every device in our network blocks by default this type of packet by having the secure redirects settings in the kernel already enabled.

```
(user@pci)-[~]
$ sysctl -a | grep secure_redirects
sysctl: permission denied on key 'kernel.apparmor_display_secid_mode'
sysctl: permission denied on key 'kernel.apparmor_restrict_unprivileged_unconfined'
sysctl: permission denied on key 'kernel.cad_pid'
sysctl: permission denied on key 'kernel.unprivileged_usersns_apparmor_policy'
sysctl: permission denied on key 'kernel.usermodehelper.bset'
sysctl: permission denied on key 'kernel.usermodehelper.inheritable'
sysctl: permission denied on key 'net.core.bpf_jit_harden'
sysctl: permission denied on key 'net.core.bpf_jit_kallsyms'
sysctl: permission denied on key 'net.core.bpf_jit_limit'
net.ipv4.conf.all.secure_redirects = 1
net.ipv4.conf.default.secure_redirects = 1
net.ipv4.conf.eth0.secure_redirects = 1
net.ipv4.conf.lo.secure_redirects = 1
sysctl: permission denied on key 'net.ipv4.tcp_fastopen_key'
sysctl: permission denied on key 'net.ipv6.conf.all.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.default.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.eth0.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.lo.stable_secret'
sysctl: permission denied on key 'vm.mmap_rnd_bits'
sysctl: permission denied on key 'vm.mmap_rnd_compat_bits'
sysctl: permission denied on key 'vm.stat_refresh'
```

- Anything that is not explicitly allowed has to be denied

Finally, we denied all the not specifically allowed traffic

		IPv4+6	*	*	*	*	*	*	Default deny / state violation rule			
--	--	--------	---	---	---	---	---	---	-------------------------------------	--	--	--

Tests of the security Policies

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

```
root@webserver:~# host logserver
logserver.acme-10.test has address 100.100.1.3
logserver.acme-10.test has IPv6 address 2001:470:b5b8:a81:282e:3ed2:707e:f853
root@webserver:~# ping logserver
PING logserver(logserver.acme-10.test (2001:470:b5b8:a81:282e:3ed2:707e:f853)) 56 data bytes
64 bytes from logserver.acme-10.test (2001:470:b5b8:a81:282e:3ed2:707e:f853): icmp_seq=1 ttl=62 time=1.46 ms
64 bytes from logserver.acme-10.test (2001:470:b5b8:a81:282e:3ed2:707e:f853): icmp_seq=2 ttl=62 time=1.37 ms
64 bytes from logserver.acme-10.test (2001:470:b5b8:a81:282e:3ed2:707e:f853): icmp_seq=3 ttl=62 time=1.21 ms
^C
--- logserver ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.211/1.347/1.463/0.103 ms
root@webserver:~#
```

2. The DNS Server should be able to answer DNS requests coming from the Internet.

```
user@kathara24:~/Desktop/HW$ nslookup webserver
Server:      100.100.1.2
Address:     100.100.1.2#53

Name:   webserver
Address: 100.100.6.2
Name:   webserver
Address: 2001:470:b5b8:a06:f03:b78f:7d56:7591

user@kathara24:~/Desktop/HW$ nslookup google.com
Server:      100.100.1.2
Address:     100.100.1.2#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.180.174
Name:   google.com
Address: 2a00:1450:4002:403::200e
```

3. The HTTP/HTTPS service provided in the DMZ has to be accessible from the Internet. It can also be accessed using the WAN address of the main firewall.

We started a web server on the given machine::

```
root@webserver:~# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

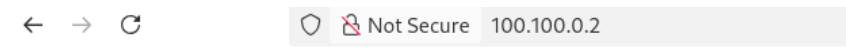
Then connected from client-ext-1...

← → ↻  Not Secure 100.100.6.2

Directory listing for /

- [.bash_history](#)
- [.bashrc](#)
- [.local/](#)
- [.profile](#)
- [.ssh/](#)
- [.viminfo](#)

...and from a personal PC outside of the ACME network, using the firewall interface.



Directory listing for /

- [.bash_history](#)
- [.bashrc](#)
- [.local/](#)
- [.profile](#)
- [.ssh/](#)
- [.viminfo](#)

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

From the ACME Network (client-ext-1):

```
(user@pci)~$ wget webservice
Prepended http:// to 'webservice'
Prepended http:// to '100.100.6.3:3128'
--2025-04-28 11:56:11-- http://webservice/ Backups
Connecting to 100.100.6.3:3128... connected.
Proxy request sent, awaiting response... 200 OK
Length: 6795 (6.6K) [text/html]
Saving to: 'index.html'

index.html          100%[=====] 6.64K
--2025-04-28 11:56:11 (30.5 MB/s) - 'index.html' saved [6795/6795]
```

From the Internet (external PC)

```
$ curl -v http://webservice
* Uses proxy env variable http_proxy == '100.100.6.3:3128'
* Trying 100.100.6.3:3128...
* Connected to (nil) (100.100.6.3) port 3128 (#0)
> GET http://webservice/ HTTP/1.1
> Host: webservice
> User-Agent: curl/7.81.0
> Accept: */*
> Proxy-Connection: Keep-Alive
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: SimpleHTTP/0.6 Python/3.9.2
< Date: Mon, 28 Apr 2025 10:02:18 GMT
< Content-Type: text/html; charset=utf-8
```

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

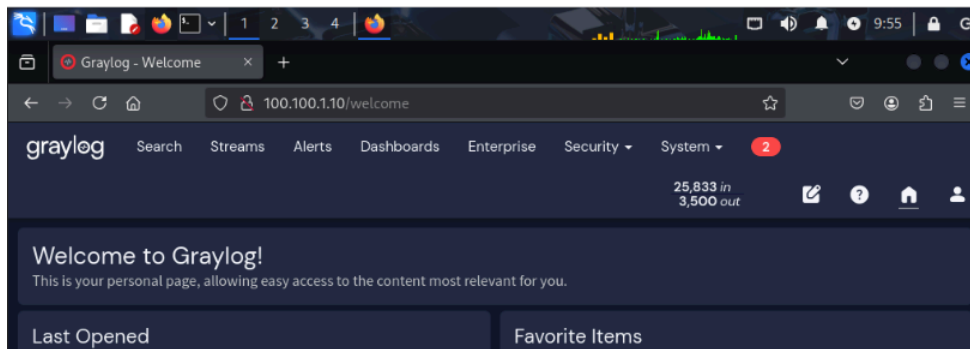
From the DMZ - Graylog Service on port 9200

```
root@webservice:~# wget http://100.100.1.10:9200
--2025-05-02 07:49:19-- http://100.100.1.10:9200/
Connecting to 100.100.6.3:3128... connected.
Proxy request sent, awaiting response... 200 OK
Length: 556 [application/json]
Saving to: 'index.html'

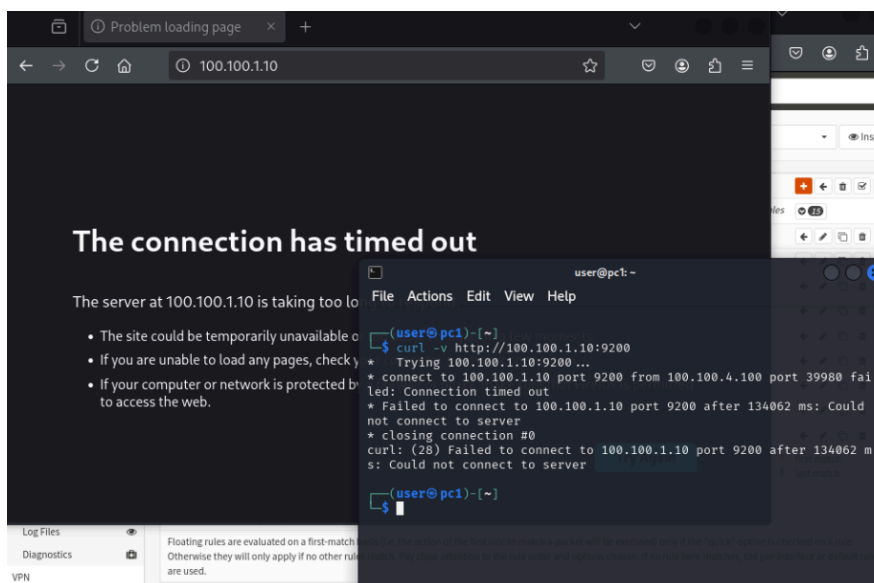
index.html          100%[=====]
--2025-05-02 07:49:19 (72.5 MB/s) - 'index.html' saved [556/556]

root@webservice:~#
```

From the Clients - Graylog Web from kali

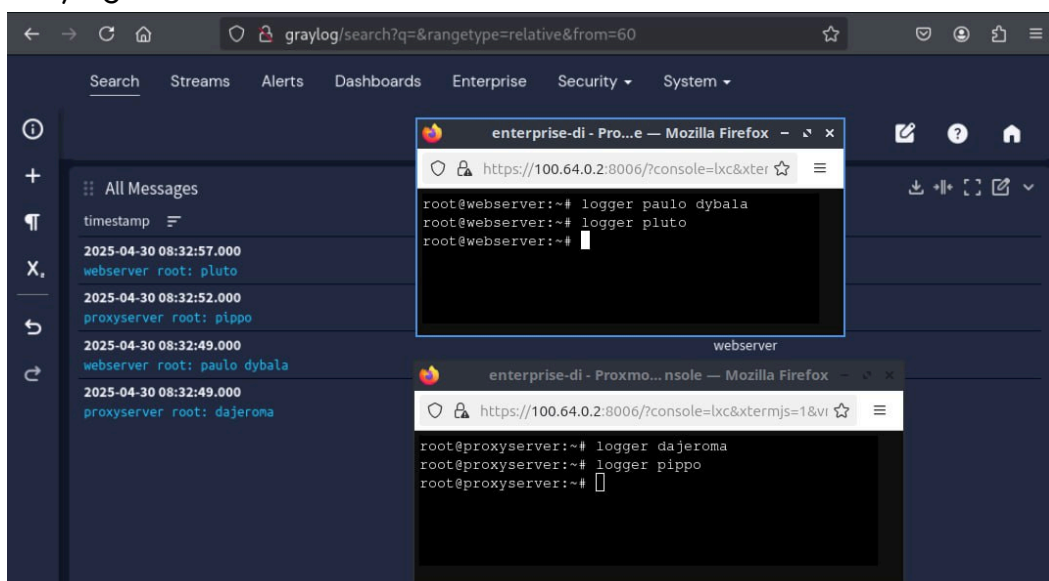


From client-ext-1, in a non-authorized network:



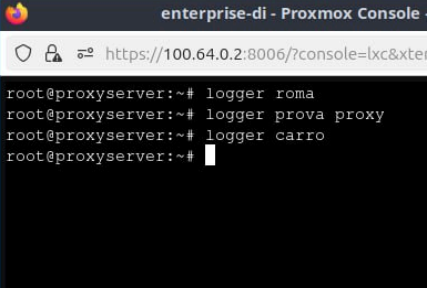

6. All the hosts (but the Client network hosts) have to use the syslog and the log collector services on the Log server (syslog) and Graylog server.

Graylog test



log server test

```
root@logserver:/var/log# ls
acme_network_sources.log  auth.log.3.gz  daemon.log.4.gz  m
alternatives.log         auth.log.4.gz  dpkg.log         m
alternatives.log.1       bttmp         dpkg.log.1       m
alternatives.log.2.gz    bttmp.1       dpkg.log.2.gz    m
apt                      daemon.log     journal          m
auth.log                 daemon.log.1   lastlog          m
auth.log.1              daemon.log.2.gz mail.info         m
auth.log.2.gz           daemon.log.3.gz mail.info.1       m
root@logserver:/var/log# cat acme_network_sources.log
Apr 30 09:19:57 proxyserver root: roma
Apr 30 09:20:07 proxyserver root: prova proxy
Apr 30 09:20:11 webserver root: prova webserver
Apr 30 09:20:13 webserver root: daje
Apr 30 09:20:16 proxyserver root: carro
root@logserver:/var/log#
```



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

```
root@greenbone:~# nmap -sn 100.100.1.0/24 100.100.2.0/24 100.100.6.0/24 100.100.4.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-30 09:37 UTC
Nmap scan report for 100.100.1.1
Host is up (0.0014s latency).
MAC Address: E6:F5:18:8B:B3:E5 (Unknown)
Nmap scan report for dnsserver.acme-10.test (100.100.1.2)
Host is up (0.000061s latency).
MAC Address: EE:6C:35:C7:77:6F (Unknown)
Nmap scan report for logserver.acme-10.test (100.100.1.3)
Host is up (0.000051s latency).
MAC Address: 16:CD:D6:00:4E:4C (Unknown)
Nmap scan report for graylog.acme-10.test (100.100.1.10)
Host is up (0.00016s latency).
MAC Address: 02:8C:10:8C:C9:54 (Unknown)
Nmap scan report for greenbone.acme-10.test (100.100.1.4)
Host is up.
Nmap scan report for 100.100.2.1
Host is up (0.00066s latency).
Nmap scan report for kali.acme-10.test (100.100.2.100)
Host is up (0.0032s latency).
Nmap scan report for 100.100.6.1
Host is up (0.0023s latency).
Nmap scan report for webserver.acme-10.test (100.100.6.2)
Host is up (0.0028s latency).
Nmap scan report for proxyserver.acme-10.test (100.100.6.3)
Host is up (0.0013s latency).
Nmap scan report for fantastic-coffee.acme-10.test (100.100.4.1)
Host is up (0.0022s latency).
Nmap scan report for client-ext-1.acme-10.test (100.100.4.100)
Host is up (0.0062s latency).
Nmap done: 1024 IP addresses (12 hosts up) scanned in 11.45 seconds
root@greenbone:~#
```

8. All network hosts must be managed via SSH only from hosts within the Client network, so be sure that all the hosts have the SSH service up and running.

From arpwatich client to logserver

```
root@arpwatch-clients:~# ssh -i .ssh/logserver_id_rsa root@logserver
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: Wed Apr 30 07:47:25 2025
root@logserver:~#
```

From webserver to logserver

```
root@webserver:~# ssh root@logserver
ssh: connect to host logserver port 22: Connection timed out
root@webserver:~#
```

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

After opening an HTTP server on our PC -> http://100.101.0.3:80
Without the proxy:

```
root@arpwatch-clients:~# tail -n 2 .bashrc
# export http_proxy=100.100.6.3:3128
# export https_proxy=100.100.6.3:3128
root@arpwatch-clients:~# wget http://100.101.0.3:80
--2025-05-02 08:08:08-- http://100.101.0.3/
Connecting to 100.101.0.3:80... failed: Connection timed out.
Retrying.

--2025-05-02 08:10:20-- (try: 2) http://100.101.0.3/
Connecting to 100.101.0.3:80...
```

With the proxy

```
root@arpwatch-clients:~# tail -n 2 .bashrc
export http_proxy=100.100.6.3:3128
export https_proxy=100.100.6.3:3128
root@arpwatch-clients:~# wget http://100.101.0.3:80
--2025-05-02 08:05:32-- http://100.101.0.3/
Connecting to 100.100.6.3:3128... connected.
Proxy request sent, awaiting response... 200 OK
Length: 777 [text/html]
Saving to: 'index.html.1'

index.html.1                                100% [=====]

2025-05-02 08:05:32 (1.06 MB/s) - 'index.html.1' saved [777/777]

root@arpwatch-clients:~#
```

10. Any packet the Main Firewall receives on port 65000 should be redirected to port 80 of the proxy host.

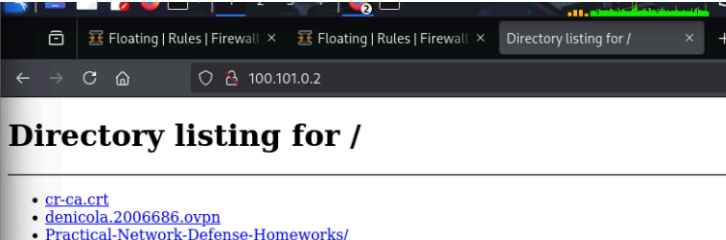
```
$ nc 100.100.0.2 65000
GET /

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

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

```
kali:/mnt/c/Users/tommy/Desktop/acme$ sudo python3 -m http.server 80

Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
100.100.0.2 - - [30/Apr/2025 12:06:43] "GET / HTTP/1.1" 200 -
100.100.0.2 - - [30/Apr/2025 12:06:43] code 404, message File not found
100.100.0.2 - - [30/Apr/2025 12:06:43] "GET /favicon.ico HTTP/1.1" 404 -
100.100.0.2 - - [30/Apr/2025 12:06:44] "GET / HTTP/1.1" 200 -
```



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

```
(user@pc1)-[~]
$ ping 100.100.1.2 -c 1
PING 100.100.1.2 (100.100.1.2) 56(84) bytes of data.
64 bytes from 100.100.1.2: icmp_seq=1 ttl=62 time=1.58 ms

--- 100.100.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.579/1.579/1.579/0.000 ms

(user@pc1)-[~]
$ ping 100.100.2.100 -c 1
PING 100.100.2.100 (100.100.2.100) 56(84) bytes of data.
64 bytes from 100.100.2.100: icmp_seq=1 ttl=62 time=1.64 ms

--- 100.100.2.100 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.636/1.636/1.636/0.000 ms

(user@pc1)-[~]
$ ping 100.100.6.3 -c 1
PING 100.100.6.3 (100.100.6.3) 56(84) bytes of data.
64 bytes from 100.100.6.3: icmp_seq=1 ttl=63 time=1.08 ms

--- 100.100.6.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.080/1.080/1.080/0.000 ms

(user@pc1)-[~]
$ ping 100.101.0.3 -c 1
PING 100.101.0.3 (100.101.0.3) 56(84) bytes of data.
64 bytes from 100.101.0.3: icmp_seq=1 ttl=62 time=56.0 ms

--- 100.101.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 56.048/56.048/56.048/0.000 ms
```

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

```
kali:/mnt/c/Users/tommy/Desktop/acme$ traceroute -I 100.100.6.3
traceroute to 100.100.6.3 (100.100.6.3), 64 hops max
 1  100.101.0.1  16.984ms  68.733ms  12.222ms
 2  100.100.0.2  10.102ms  9.634ms  9.726ms
 3  100.100.6.3  10.453ms  9.565ms  9.081ms
kali:/mnt/c/Users/tommy/Desktop/acme$ ping 100.100.6.3 -c 1
PING 100.100.6.3 (100.100.6.3) 56(84) bytes of data.
64 bytes from 100.100.6.3: icmp_seq=1 ttl=62 time=11.1 ms

--- 100.100.6.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 11.115/11.115/11.115/0.000 ms
kali:/mnt/c/Users/tommy/Desktop/acme$ ping 100.100.4.100 -c 1
PING 100.100.4.100 (100.100.4.100) 56(84) bytes of data.

--- 100.100.4.100 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

14. ICMP redirect packets should not cross any network.

There are no tests required for this point, as it's evident that if all ICMP redirects are blocked by both firewalls, none can traverse the networks.

15. Anything that is not explicitly allowed has to be denied

As we can see in other tests (like in point 8 or 5), by default everything that is not allowed is denied, so we don't need to worry.