

Instituto Superior Técnico



## Laboratory 2 – Firewalls and AAA

Master's in computer science and Engineering

Network Advanced Security and Architecture

Group 2



98678, Bruno Freitas



98742, Alexandru Pena

# Table of Contents

1. Classical firewalls versus Zone Based Policy Firewalls .....	4
The Network.....	4
Network Initial Configuration .....	4
Classical firewall.....	4
Private Zones .....	4
DMZ Zone.....	6
OUT Zone.....	7
Mitigating Some attacks with ACLs .....	8
Zone Based Policy Firewall .....	9
CBAC vs ZBPF .....	10
Test.....	11
2. Defense against DoS attacks .....	14
Direct connection between attacker and victim .....	14
Defence using a ZBPF .....	15
SYN Flood .....	15
ICMP Flood.....	16
3. AAA.....	17
TACACS+ .....	17
RADIUS .....	19
802.11x .....	20
4. ASA Firewall.....	26
SSH Configuration.....	26
DNS Doctoring .....	27

# Table of Figures

Figure 1 - Network for the exercise 3.1 .....	4
Figure 2 - Network to test exercise 3.1 .....	11
Figure 3 - Out network filtering .....	11
Figure 4 - DNS Resolution .....	12
Figure 5 - SMTP Test.....	12
Figure 6 - SSH Test .....	13
Figure 7 - Network for exercise 3.2 .....	14
Figure 8 - Attack failed .....	14
Figure 9 - Attack successful with source IP randomized .....	14
Figure 10 - Protection with ZBPF firewall topology .....	15
Figure 11 - SYN Flood attack stopped .....	15

Figure 12 - Police before attack .....	16
Figure 13 - Police after attack .....	16
Figure 14 - TACACS Authentication Process .....	17
Figure 15 - TACACS Authentication Process Wireshark .....	17
Figure 16 - TACACS Authentication PASS .....	18
Figure 17 - TACACS Accounting Message .....	18
Figure 18 - TACACS Authorization Request by saarShow user .....	18
Figure 19 - TACACS Authorization Request fail by saarShow user.....	19
Figure 20 - RADIUS Protocol .....	19
Figure 21 - RADIUS Protocol Wireshark .....	19
Figure 22 - RADIUS Authorization Attempt .....	20
Figure 23 - RADIUS showBrief user.....	20
Figure 24 - RADIUS saarAdmin commands.....	20
Figure 25 - 802.11x message Exchange.....	21
Figure 26 - EAP Identity Response .....	21
Figure 27 - EAP messages between supplicant and switch .....	22
Figure 28 - RADIUS messages between switch and RADIUS server.....	22
Figure 29 - EAP Authenticated.....	22
Figure 30 - SSH Topology .....	26
Figure 31 - SSH Limited Algorithms .....	27
Figure 32 - SSH Working .....	27
Figure 33 - DNS Doctoring Topology .....	28

# 1. Classical firewalls versus Zone Based Policy Firewalls

## The Network

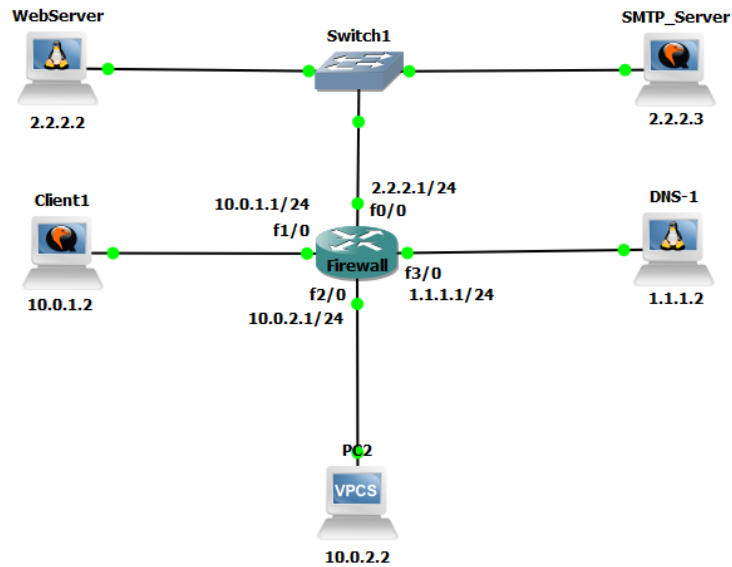


Figure 1 - Network for the exercise 3.1

## Network Initial Configuration

All Network initial configuration is in Appendix chapter.

## Classical firewall

All four Zones will have the following rules applied to Inbound:

- 1-ip inspect name RETURN-TRAFFIC udp
- 2-ip inspect name RETURN-TRAFFIC tcp
- 3-ip inspect name RETURN-TRAFFIC icmp

Objective: Return traffic (TCP, UDP or ICMP) from Outside to Inside will be allowed through because the traffic was initiated from Inside.

## Private Zones

For the Private Zones (PR1 and PR2) were implemented the following ACLs:

- 1-no ip access-list extended IN\_PRIVATES
- 2-ip access-list extended IN\_PRIVATES
- 3-deny ip any 10.0.0.0 0.0.3.255
- 4-permit tcp 10.0.0.0 0.0.3.255 host 10.0.1.1 eq 22
- 5-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.2 eq 80
- 6-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.2 eq 443
- 7-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq pop3
- 8-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq 993
- 9-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq smtp

```
10-permit icmp 10.0.0.0 0.0.3.255 host 2.2.2.2
11-permit icmp 10.0.0.0 0.0.3.255 host 2.2.2.3
12-Deny ip 10.0.0.0 0.0.3.255 2.2.2.0 0.0.0.255
13-permit udp 10.0.0.0 0.0.3.255 any eq 53
14-permit tcp 10.0.0.0 0.0.3.255 any eq 80
15-permit tcp 10.0.0.0 0.0.3.255 any eq 443
16-permit icmp 10.0.0.0 0.0.3.255 any
17-deny ip any any
18-exit
19-
20-no ip access-list extended OUT_PRIVATES
21-ip access-list extended OUT_PRIVATES
22-deny ip any any
23-exit
24-
25-int f1/0
26-ip inspect RETURN-TRAFFIC in
27-ip access-group IN_PRIVATES in
28-ip access-group OUT_PRIVATES out
29-exit
30-int f2/0
31-ip inspect RETURN-TRAFFIC in
32-ip access-group IN_PRIVATES in
33-ip access-group OUT_PRIVATES out
43-exit
```

Policies to be implemented:

- R1 cannot communicate with PR2  
This is made with the ACL in line 3 which blocks all traffic between the Private Zones.
- PR1 and PR2 cannot be accessed from other zones  
This is made with the OUT\_PRIVATES ACCESS-LIST that is applied to OUTbound. This tell us that all traffic will be blocked (except the one that was initiated from Inside because of ip inspect).
- PR1 and PR2 can make web requests (HTTP and HTTPS) to the Out Zone  
This is made with the ACLs in lines 14 and 15 which permit HTTP and HTTPS traffic to all zones but because the Private Zones cannot communicate with each other no HTTP and HTTPS traffic is allowed between them.
- PR1 and PR2 are allowed to ICMP to Out Zone  
This is made with the ACL in line 16 which permits ICMP traffic to all zones but because the Private Zones cannot communicate with each other no ICMP traffic is allowed between them.
- PR1 and PR2 are allowed to DNS to Out Zone  
This is made with the ACL in line 13 which permits DNS traffic to all zones but because the Private Zones cannot communicate with each other no DNS traffic is allowed between them. Note that DNS will also not be possible to the DMZ Zone (this will be explained in the DMZ Zone Subsection).
- PR1 and PR2 can make web requests (HTTP and HTTPS) to the DMZ Zone  
This is made with the ACLs in lines 5 and 6 which permit HTTP and HTTPS traffic to the host 2.2.2.2 (which is the Web Server).
- PR1 and PR2 are allowed to ICMP to DMZ Zone  
This is made with the ACLs in lines 10 and 11 which permit ICMP traffic to the hosts 2.2.2.2(Web Server) and 2.2.2.3 (Email Server). NOTE: the ACL in line 16

described before, permits us to send ICMP traffic to other hosts in DMZ Zone (but this will not happen because the ACL line 12 which denies all other traffic to the DMZ Zone is applied before the line 16 one).

- PR1 and PR2 are allowed to Email (SMTP, POP3 and IMAP) to DMZ Zone, only for the two DMZ servers

This is made with the ACLs in lines 7, 8 and 9 which permit POP3, IMAP and SMTP traffic to the host 2.2.2.3 (which is the Email Server).

- PR1 and PR2 are allowed to SSH to the firewall

This is made with the ACL in line 4 which permits SSH traffic to the firewall (note: deny traffic to PR2 is implemented before this one, so no SSH is possible to the router through the PR2's interface).

### DMZ Zone

For the DMZ Zone was implemented the following ACLs

```

1-no ip access-list extended IN_DMZ
2-ip access-list extended IN_DMZ
3-permit udp 2.2.2.0 0.0.0.3 1.1.1.0 0.0.0.255 eq 53
4-permit tcp 2.2.2.0 0.0.0.3 1.1.1.0 0.0.0.255 eq smtp
5-permit icmp 2.2.2.0 0.0.0.3 1.1.1.0 0.0.0.255
6-permit tcp any host 2.2.2.1 eq 22
7-deny ip any any
8-exit
9-
10-no ip access-list extended OUT_DMZ
11-ip access-list extended OUT_DMZ
12-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq 993
13-permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq pop3
14-permit tcp any host 2.2.2.2 eq 80
15-permit tcp any host 2.2.2.2 eq 443
16-permit tcp any host 2.2.2.3 eq smtp
17-permit icmp any host 2.2.2.2
18-permit icmp any host 2.2.2.3
19-deny ip any any
20-exit
21-
22-int f0/0
23-ip inspect RETURN-TRAFFIC in
24-ip access-group IN_DMZ in
25-ip access-group OUT_DMZ out
26-exit

```

Policies to be implemented:

- Access to DMZ is restricted to the Web Server and Email Server  
This is made by specifying which hosts could be contacted in the OUT\_DMZ ACCESS-LIST and deny all traffic directed to other hosts in the DMZ's subnet.
- DMZ hosts are allowed to send DNS requests to the Out Zone  
This is made with the ACL in line 3 which permits DNS traffic to the OUT Zone.
- DMZ must allow Private Zones to make web requests (HTTP and HTTPS)  
This is made with the ACLs in lines 14 and 15 which permit HTTP and HTTPS traffic from all zones to the host 2.2.2.2 (Web Server)
- DMZ must allow Private Zones to send ICMP messages  
This is made with the ACLs in lines 17 and 18 which permit ICMP traffic from all zones to the hosts 2.2.2.2 (Web Server) and 2.2.2.3 (Email Server)

- DMZ must allow Private Zones to send Email traffic (POP3, IMAP, SMTP)  
This is made with the ACLs in lines 12, 13 and 16 which permit IMAP, POP3 traffic from Private Zones to the host 2.2.2.3 (Email Server) and SMTP traffic from all zones to the host 2.2.2.3 (Email Server).
- DMZ are allowed to ICMP to the Out Zone  
This is made with the ACL in line 5 which permits ICMP traffic to the OUT Zone.
- DMZ are allowed to SMTP to the Out Zone  
This is made with the ACL in line 4 which permits SMTP traffic to the OUT Zone.
- DMZ must allow Out Zone to send web requests (HTTP and HTTPS)  
This is made with the ACLs in lines 14 and 15 which permit HTTP and HTTPS traffic from all zones to the host 2.2.2.2 (Web Server)
- DMZ must allow Out Zone to send ICMP traffic  
This is made with the ACLs in lines 17 and 18 which permit ICMP traffic from all zones to the hosts 2.2.2.2 (Web Server) and 2.2.2.3 (Email Server)
- DMZ must allow Out Zone to send SMTP traffic  
This is made with the ACL in line 16 which permits SMTP traffic from all zones to the host 2.2.2.3 (Email Server).
- DMZ is allowed to SSH to the firewall  
This is made with the ACL in line 6 which permits SSH traffic to the firewall.

### OUT Zone

For the OUT Zone was implemented the following ACLs

```

1-no ip access-list extended IN_OUT
2-ip access-list extended IN_OUT
3-permit tcp any host 2.2.2.2 eq 80
4-permit tcp any host 2.2.2.2 eq 443
5-permit tcp any host 2.2.2.3 eq smtp
6-permit tcp any host 1.1.1.1 eq 22
7-permit icmp any host 2.2.2.2
8-permit icmp any host 2.2.2.3
9-deny ip any any
10-exit
11-
12-int f3/0
13-ip access-group IN_OUT in
14-ip inspect RETURN-TRAFFIC in
15-exit

```

Policies to be implemented:

- OUT Zone must allow Private Zones to Web browser (HTTP and HTTPS traffic)
- OUT Zone must allow Private Zones to ICMP
- OUT Zone must allow Private Zones to DNS
- OUT Zone must allow DMZ to ICMP
- OUT Zone must allow DMZ to SMTP
- OUT Zone must allow DMZ to DNS
- OUT Zone is allowed to make web requests (HTTP and HTTPS) to the DMZ  
This is made with the ACL in line 3 and 4 which permit HTTP and HTTPS traffic from all zones to the host 2.2.2.2 (Web Server).
- OUT Zone is allowed to ICMP to the DMZ two servers

This is made with the ACLs in lines 7 and 8 which permit ICMP traffic to the hosts 2.2.2.2 (Web Server) and 2.2.2.3 (Email Server)

- OUT Zone is allowed to SMTP to DMZ  
This is made with the ACL in lines 5 which permit SMTP traffic from all zones to the host 2.2.2.3 (Email Server).
- OUT Zone is allowed to SSH to the firewall  
This is made with the ACL in line 6 which permits SSH traffic to the firewall.

### Mitigating Some attacks with ACLs

The group followed strictly the policies given but knows that some policies could be improved to mitigate some attacks. So, here we specify some examples that could be revised.

### Mitigating Spoofing Attacks

To mitigate Spoofing attacks some specific ACLs could be added. These ACLs address some well-known IP address classes that should never be seen as source IP addresses for traffic entering an organization's network. To accomplish this, the interfaces from our company (Privates and DMZ) should only allow INbound packets with a source address from that network (which is the case). For the interface from the Out Zone in the INbound should be added

```
1-deny ip host 0.0.0.0 any (deny all zeros addresses)
2-deny ip 127.0.0.0 0.255.255.255 any (deny local host addresses (127.0.0.0/8))
3-deny ip 2.2.0.0 0.0.255.255 any (deny organization's reserved private address)
4-deny ip 10.0.0.0 0.255.255.255 any (deny organization's reserved private address and (RFC 1918))
5-deny ip 172.16.0.0 0.0.255.255 any (deny reserved private addresses (RFC 1918))
6-deny ip 192.168.0.0 0.0.255.255 any (deny reserved private addresses (RFC 1918))
7-deny ip host 255.255.255.255 any (deny broadcast address)
```

### Mitigating ICMP Abuse

Hackers can use ICMP packets to discover subnets and hosts, generate DOS flood attacks or to alter host routing tables (with ICMP Redirects for example). For a proper network operation, the following ICMP messages should be allowed into the internal network:

- Allow users to ping external hosts - Echo reply.
- Requests that the sender decrease the traffic rate of messages - Source quench.
- Send Unreachable for packets that are administratively denied by an ACL – Unreachable

Translating it to ACLs we get the following ACLs that should be applied in interfaces' INbound of OutZone:

```
1-permit icmp any any echo-reply
2-permit icmp any any source-quench
3-permit icmp any any unreachable
4-deny icmp any any
```

For a proper network operation, the following ICMP messages are required and should be allowed to exit the network:

- Allow users to ping external hosts - Echo.



- Informs the host of packet header problems - Parameter problem.
- Enables packet maximum transmission unit (MTU) discovery - Packet too big.
- Throttles down traffic when necessary - Source quench.

Translating it to ACLs we get the following ACLs that could be applied in interfaces' INbound of our organization (Private Zones and DMZ):

```
1-apply to Private Zones:
2-permit icmp 10.0.0.0 0.0.3.255 any echo
3-permit icmp 10.0.0.0 0.0.3.255 any parameter-problem
4-permit icmp 10.0.0.0 0.0.3.255 any packet-too-big
5-permit icmp 10.0.0.0 0.0.3.255 source-quench
6-apply to DMZ Zone:
7-permit icmp 2.2.2.0 0.0.0.255 any echo
8-permit icmp 2.2.2.0 0.0.0.255 any parameter-problem
9-permit icmp 2.2.2.0 0.0.0.255 any packet-too-big
10-permit icmp 2.2.2.0 0.0.0.255 source-quench
11-deny icmp any any
```

### *Mitigating SNMP Attacks*

"If SNMP is necessary, exploitation of SNMP vulnerabilities can be mitigated by applying interface ACLs to filter SNMP packets from non-authorized systems. An exploit may still be possible if the SNMP packet is sourced from an address that has been spoofed and is permitted by the ACL." (Cisco Academy) The most effective way to prevent such attacks is to disable the SNMP Server on IOS devices that does not require it.

### *Zone Based Policy Firewall*

To start the ZBPF firewall configuration we start by creating the zones DMZ, OUT, PRIVATE-1 and PRIVATE-2 and consequently attribute each of them to their respective firewall interface.

```
1-zone security PRIVATE-1
2-zone security PRIVATE-2
3-zone security OUT
4-zone security DMZ
5-exit
6-int f1/0
7-zone-member security PRIVATE-1
8-exit
9-int f2/0
10-zone-member security PRIVATE-2
11-exit
12-int f3/0
13-zone-member security OUT
14-exit
15-int f0/0
16-zone-member security DMZ
17-exit
```

At this point all communications will be automatically denied, therefore we proceed to create the class-maps, policy-maps and zone-pairs.

Here is an example of a class-map which will match all web browser traffic like HTTP and HTTPS:

```
1-class-map type inspect match-any HTTP-TRAFFIC
2-match protocol HTTP
3-match protocol HTTPS
```

Everything that does not match these protocols will be denied in this class-map. After creating a class-map we must then create a policy-map that will be used to create the zone-pair

```
1-policy-map type inspect PRIV-TO-DMZ-POLICY
2-class type inspect DMZ-SERVERS-ALLOWED
3-drop
4-class type inspect HTTP-TRAFFIC
5-inspect
6-class type inspect ICMP-TRAFFIC
7-inspect
8-class type inspect USER-EMAIL-TRAFFIC
9-inspect
10-class type inspect SMTP-TRAFFIC
11-inspect
12-class type inspect DNS-TRAFFIC
13-inspect
```

This is an example of policy-map configuration. It can have multiple class-maps to permit different traffic types. This separation allows for an easier read and understanding of the firewall rules, which opposed to the classic firewall can be pretty messy to read. This policy-map corresponds to the policies of the traffic that goes from the private zone to the DMZ. It will allow as specified in the policies HTTP, Email and DNS traffic (check each class-map in the annex file for the configuration of the other class-maps). Using this policy-map we can then create a zone-pair association by specifying the source and destination that the policy-map should be applied to:

```
1-zone-pair security PRIV-1-DMZ source PRIVATE-1 destination DMZ
2-service-policy type inspect PRIV-TO-DMZ-POLICY
```

### CBAC vs ZBPF

We start by saying that for the work assigned, both profiles work and can be used. The biggest difference between them, which can be more of an opinion, is that ZBPF policies are a lot easier to read and understand, therefore making the work easier and less error prone.

The CBAC policies are less verbose and for its configuration the total number of lines was around 20 lines less, but the group noticed, during the testing part, that the firewall rules based on the CBAC profile were more likely to fail and have errors. This because of how much attention is necessary to pay to the IP, IP wildcard and if the policies are going to be applied to ingress or outgress which can lead to confusion.

The ZBPF profile avoids this confusion by having to define the rules more "verbosely" with the source destination configurations in the zone-pair.

## Test

To test all configurations, we used the following structure:

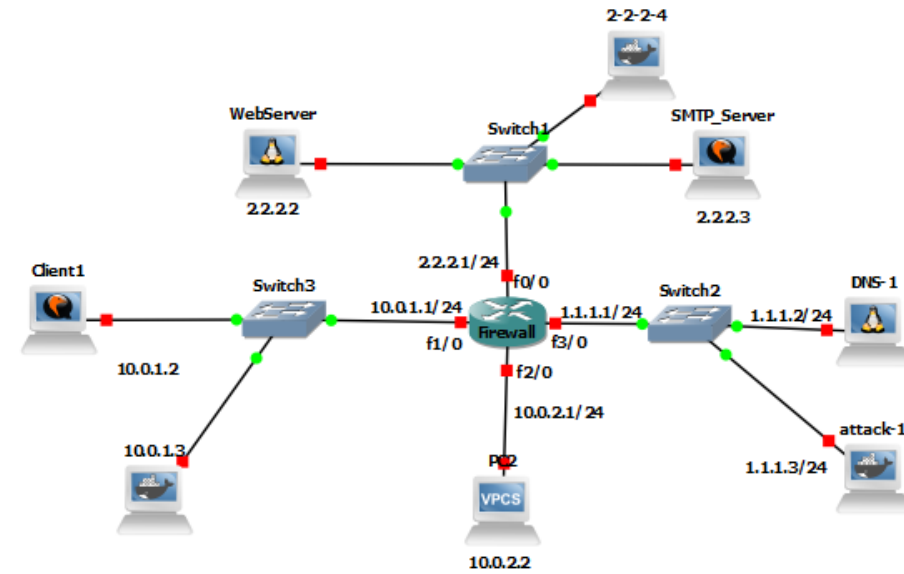


Figure 2 - Network to test exercise 3.1

The structure contains a couple more hosts and attack machines than the original, to check if everything is working as expected.

We started by testing if the communication between the different private networks was working correctly, and not allowing some sort of communication. Then by checking if any of the outer zones could initiate communications with the private zones in any protocol. To realize these tests, we used pings from different locations to test ICMP, telnet in the Kali machines to test TCP protocols and HPING3 for other tests.

We then tried to contact the web server 2.2.2.4, added by the own group, to test if only the SMTP server and web server were exposed. All other hosts in that subnet should not be able to be contacted by neither the private zone nor out zone.

```

root@attack-1:/home# nping 10.0.2.1

Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2021-03-29 12:59 UTC
SENT (0.0044s) ICMP [1.1.1.3 > 10.0.2.1 Echo request (type=8/code=0) id=15937 seq=1] IP [ttl=64 id=358 iplen=28 ]
RCVD (0.0252s) ICMP [10.0.2.1 > 1.1.1.3 Communication administratively prohibited by filtering (type=3/code=13) ] IP [ttl=255 id=1118 iplen=56 ]
SENT (1.0182s) ICMP [1.1.1.3 > 10.0.2.1 Echo request (type=8/code=0) id=15937 seq=2] IP [ttl=64 id=358 iplen=28 ]
RCVD (1.0311s) ICMP [10.0.2.1 > 1.1.1.3 Communication administratively prohibited by filtering (type=3/code=13) ] IP [ttl=255 id=1119 iplen=56 ]
^C
Max rtt: 17.561ms | Min rtt: 12.385ms | Avg rtt: 14.973ms
Raw packets sent: 2 (568) | Rcvd: 2 (1128) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.17 seconds
root@attack-1:/home# ping 2.2.2.2
bash: ping: command not found
root@attack-1:/home# nping 2.2.2.2

Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2021-03-29 13:00 UTC
SENT (0.0023s) ICMP [1.1.1.3 > 2.2.2.2 Echo request (type=8/code=0) id=16729 seq=1] IP [ttl=64 id=14500 iplen=28 ]
RCVD (0.0241s) ICMP [2.2.2.2 > 1.1.1.3 Echo reply (type=0/code=0) id=16729 seq=1] IP [ttl=63 id=60062 iplen=28 ]
SENT (1.0032s) ICMP [1.1.1.3 > 2.2.2.2 Echo request (type=8/code=0) id=16729 seq=2] IP [ttl=64 id=14500 iplen=28 ]
RCVD (1.0232s) ICMP [2.2.2.2 > 1.1.1.3 Echo reply (type=0/code=0) id=16729 seq=2] IP [ttl=63 id=60259 iplen=28 ]
^C
Max rtt: 21.814ms | Min rtt: 19.979ms | Avg rtt: 20.896ms
Raw packets sent: 2 (568) | Rcvd: 2 (568) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.24 seconds
root@attack-1:/home# nping 2.2.2.5

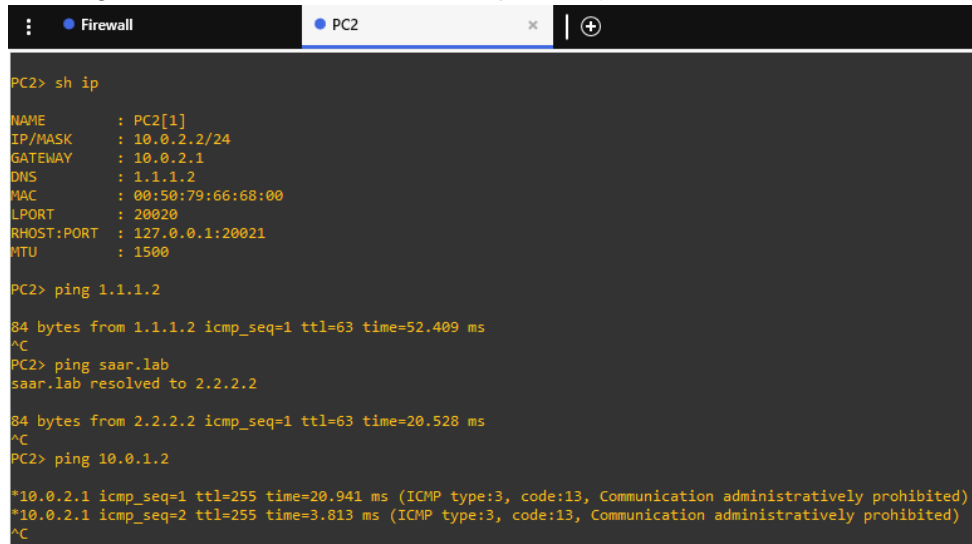
Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2021-03-29 13:00 UTC
SENT (0.0046s) ICMP [1.1.1.3 > 2.2.2.5 Echo request (type=8/code=0) id=41859 seq=1] IP [ttl=64 id=60799 iplen=28 ]
RCVD (0.0109s) ICMP [1.1.1.1 > 1.1.1.3 Communication administratively prohibited by filtering (type=3/code=13) ] IP [ttl=255 id=1120 iplen=56 ]

```

Figure 3 - Out network filtering

We can see in the figure, that the host attack-1 located in the out network can't contact any of the hosts in the private zone, the communication is administratively prohibited. The same for any other server in the DMZ zone, connection prohibited. Only the web server 2.2.2.2 and 2.2.2.3 are allowed.

We then tested DNS by making pings to the host "saar.lab" and seeing the name resolution in place working (for the DNS server the group used an appliance found on the GNS3 market which only requires the edition of the file "/etc/hosts", but this appliance behaviour can be a bit unstable and requires the restart of it a couple times before it starts working, in case this architecture is replicated).



```

PC2> sh ip
NAME       : PC2[1]
IP/MASK    : 10.0.2.2/24
GATEWAY    : 10.0.2.1
DNS        : 1.1.1.2
MAC        : 00:50:79:66:68:00
LPORT      : 20020
RHOST:PORT : 127.0.0.1:20021
MTU        : 1500

PC2> ping 1.1.1.2
64 bytes from 1.1.1.2 icmp_seq=1 ttl=63 time=52.409 ms
AC

PC2> ping saar.lab
saar.lab resolved to 2.2.2.2

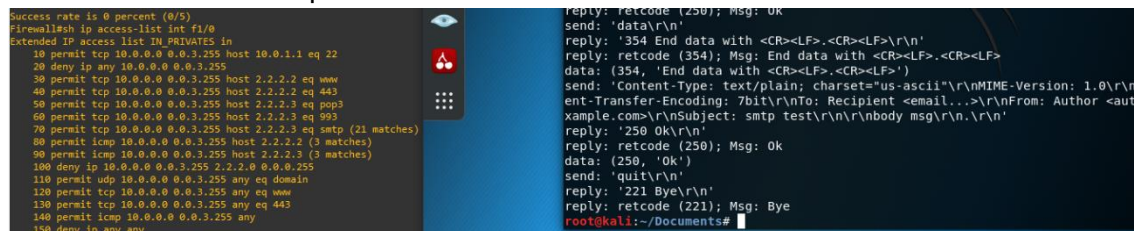
64 bytes from 2.2.2.2 icmp_seq=1 ttl=63 time=20.528 ms
AC

PC2> ping 10.0.1.2
*10.0.2.1 icmp_seq=1 ttl=255 time=20.941 ms (ICMP type:3, code:13, Communication administratively prohibited)
*10.0.2.1 icmp_seq=2 ttl=255 time=3.813 ms (ICMP type:3, code:13, Communication administratively prohibited)
AC

```

Figure 4 - DNS Resolution

After testing DNS, the group proceeded to test the SMTP service. The SMTP server and client are both python scripts. The server uses the module smtpd to create a custom SMTP server. Both scripts can be found in annex.



```

Success rate is 0 percent (0/5)
Firewall#sh ip access-list int f1/0
Extended IP access list IN PRIVATES in
 10 permit tcp 10.0.0.0 0.0.3.255 host 10.0.1.1 eq 22
 20 deny ip any 10.0.0.0 0.0.3.255
 30 permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.2 eq www
 40 permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.2 eq 443
 50 permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq pop3
 60 permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq 993
 70 permit tcp 10.0.0.0 0.0.3.255 host 2.2.2.3 eq smtp (21 matches)
 80 permit icmp 10.0.0.0 0.0.3.255 host 2.2.2.2 (3 matches)
 90 permit icmp 10.0.0.0 0.0.3.255 host 2.2.2.3 (3 matches)
100 deny ip 10.0.0.0 0.0.3.255 2.2.2.0 0.0.0.255
110 permit udp 10.0.0.0 0.0.3.255 any eq domain
120 permit tcp 10.0.0.0 0.0.3.255 any eq www
130 permit tcp 10.0.0.0 0.0.3.255 any eq 443
140 permit icmp 10.0.0.0 0.0.3.255 any
150 deny ip any any

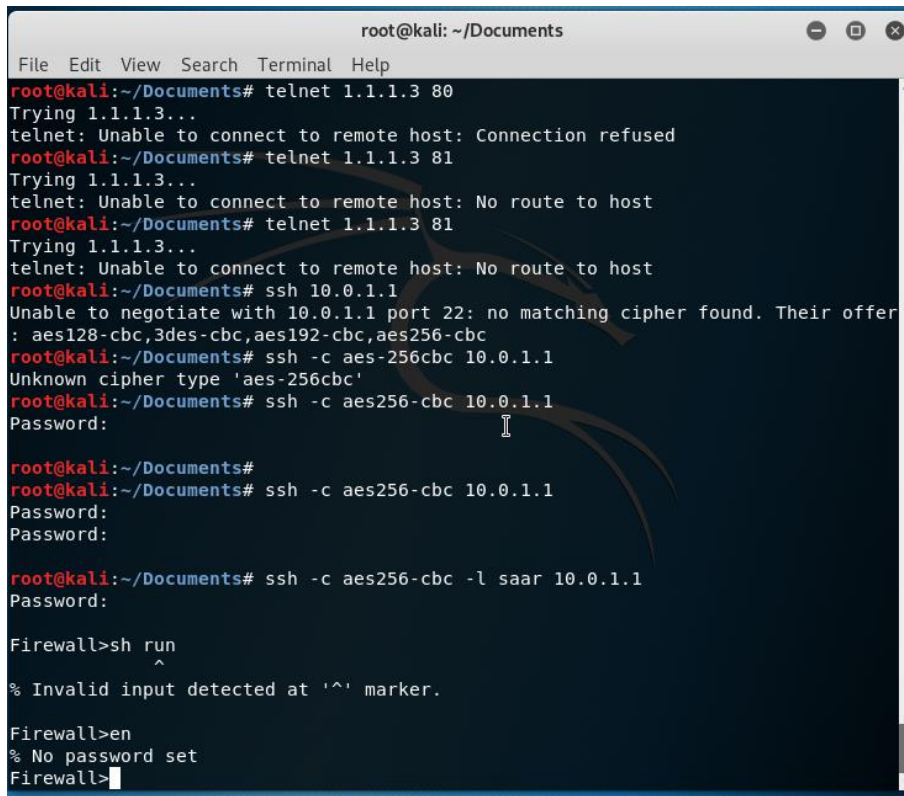
reply: retcode (250); Msg: OK
send: 'data\r\n'
reply: '354 End data with <CR><LF>.<CR><LF>\r\n'
data: (354, 'End data with <CR><LF>.<CR><LF>')
send: 'Content-Type: text/plain; charset="us-ascii"\r\nMIME-Version: 1.0\r\nContent-Transfer-Encoding: 7bit\r\nTo: Recipient <email...>\r\nFrom: Author <author@example.com>\r\nSubject: smtp test\r\n\r\nbody msg\r\n\r\n'
reply: '250 Ok\r\n'
reply: retcode (250); Msg: OK
data: (250, 'Ok')
send: 'quit\r\n'
reply: '221 Bye\r\n'
reply: retcode (221); Msg: Bye
root@kali:~/Documents#

```

Figure 5 - SMTP Test

After a couple more connectivity and traffic tests, we completed by testing the SSH connectivity to the firewall.

The user saar with password saar was configured in the SSH firewall service. Also because the SSH cryptographic algorithms supported by the router are pretty old we forced the host to use them in order to establish ssh keys and create the secure tunnels.

A terminal window titled 'root@kali: ~/Documents' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a series of commands and their outputs. First, 'telnet 1.1.1.3 80' is run, resulting in 'Trying 1.1.1.3...' and 'telnet: Unable to connect to remote host: Connection refused'. Then, 'telnet 1.1.1.3 81' is run, resulting in 'Trying 1.1.1.3...' and 'telnet: Unable to connect to remote host: No route to host'. Next, 'ssh 10.0.1.1' is run, resulting in 'Unable to negotiate with 10.0.1.1 port 22: no matching cipher found. Their offer: aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc'. Then, 'ssh -c aes-256cbc 10.0.1.1' is run, resulting in 'Unknown cipher type 'aes-256cbc''. Finally, 'ssh -c aes256-cbc 10.0.1.1' is run, resulting in 'Password:'. Below this, 'ssh -c aes256-cbc 10.0.1.1' is run again, resulting in 'Password:' and 'Password:'. Then, 'ssh -c aes256-cbc -l saar 10.0.1.1' is run, resulting in 'Password:'. Finally, 'Firewall>sh run' is run, resulting in 'Firewall>sh run', '^', and '% Invalid input detected at '^' marker.'. Then, 'Firewall>en' is run, resulting in 'Firewall>en', '% No password set', and 'Firewall>'.

```
root@kali: ~/Documents
File Edit View Search Terminal Help
root@kali:~/Documents# telnet 1.1.1.3 80
Trying 1.1.1.3...
telnet: Unable to connect to remote host: Connection refused
root@kali:~/Documents# telnet 1.1.1.3 81
Trying 1.1.1.3...
telnet: Unable to connect to remote host: No route to host
root@kali:~/Documents# telnet 1.1.1.3 81
Trying 1.1.1.3...
telnet: Unable to connect to remote host: No route to host
root@kali:~/Documents# ssh 10.0.1.1
Unable to negotiate with 10.0.1.1 port 22: no matching cipher found. Their offer
: aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
root@kali:~/Documents# ssh -c aes-256cbc 10.0.1.1
Unknown cipher type 'aes-256cbc'
root@kali:~/Documents# ssh -c aes256-cbc 10.0.1.1
Password:
root@kali:~/Documents#
root@kali:~/Documents# ssh -c aes256-cbc 10.0.1.1
Password:
Password:
root@kali:~/Documents# ssh -c aes256-cbc -l saar 10.0.1.1
Password:

Firewall>sh run
^
% Invalid input detected at '^' marker.

Firewall>en
% No password set
Firewall>
```

*Figure 6 - SSH Test*

A small note, in some tests the image may say "connection refused" or "no route to host". The "connection refused" error occurs when there is a path to the host, the traffic can flow but the host has no open service in that port, the own operative system of the host rejects the request. When a "no route to host" error happens, indicates that the firewall dropped the packets and the target is unreachable, example trying to connect to some port protocol that is not allowed by the firewall. This can be observed in figure 1.6 of the SSH test, the client tried to make a web request to the host 1.1.1.3 in the out zone, which is allowed but has no web server running in port 80, but if we try the same test with port 81, which is a not allowed protocol, the packets will be filtered.

## 2. Defense against DoS attacks

In this section we learn about 2 different types of denial-of-service attacks, SYN Flood and ICMP flood. The SYN flood attack is executed by starting many TCP handshakes with the SYN flag enabled in the TCP packets, but never completing the handshake, which forces the server, in case there is no protection mechanism, to simply wait with the port occupied. In this time no other process will be able to use this port. In the ICMP flood attack we just overwhelm a victim with many ICMP requests packets.

### Direct connection between attacker and victim

In this part of the experiment, we will execute an SYN Flood attack without any defence mechanisms, so we establish a direct connection between the victim and attacker as observed in the figure:

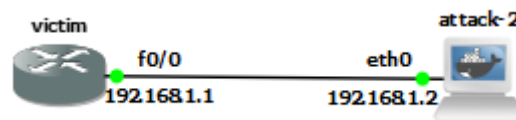


Figure 7 - Network for exercise 3.2

We then execute the attack by using the tool hping3, but the attack is not successful because as it can be seen in the picture of the Wireshark capture, the router/web server will answer with a TCP packet with the flags RST/ACK enabled telling the sender that the connection was closed. This happens because the source of all SYN packets sent by the attacker have the same source IP.

In the second experiment the switch "--rand-source" will randomize the source IP of all TCP packets, this makes it possible to start many SYN half connections as it will not be reset.

No.	Time	Source	Destination	Protocol	Length	Info
77314	1461.445247	192.168.1.2	192.168.1.1	TCP	54	3731 → 80 [SYN] Seq=0 Win=512 Len=0
77315	1461.450716	192.168.1.1	192.168.1.2	TCP	60	80 → 3724 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
77316	1461.455917	192.168.1.2	192.168.1.1	TCP	54	3732 → 80 [SYN] Seq=0 Win=512 Len=0
77317	1461.461245	192.168.1.1	192.168.1.2	TCP	60	80 → 3725 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
77318	1461.466533	192.168.1.2	192.168.1.1	TCP	54	3733 → 80 [SYN] Seq=0 Win=512 Len=0
77319	1461.471658	192.168.1.1	192.168.1.2	TCP	60	80 → 3726 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
77320	1461.477067	192.168.1.2	192.168.1.1	TCP	54	3734 → 80 [SYN] Seq=0 Win=512 Len=0
77321	1461.482037	192.168.1.1	192.168.1.2	TCP	60	80 → 3727 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figure 8 - Attack failed

No.	Time	Source	Destination	Protocol	Length	Info
3359	548.635215	66.219.198.228	192.168.1.1	TCP	54	6082 → 80 [SYN] Seq=0 Win=512 Len=0
3360	548.646205	180.176.77.96	192.168.1.1	TCP	54	6083 → 80 [SYN] Seq=0 Win=512 Len=0
3361	548.657081	39.123.23.145	192.168.1.1	TCP	54	6084 → 80 [SYN] Seq=0 Win=512 Len=0
3362	548.667299	250.13.145.120	192.168.1.1	TCP	54	6085 → 80 [SYN] Seq=0 Win=512 Len=0
3363	548.678138	77.66.233.227	192.168.1.1	TCP	54	6086 → 80 [SYN] Seq=0 Win=512 Len=0
3364	548.688922	49.164.181.225	192.168.1.1	TCP	54	6087 → 80 [SYN] Seq=0 Win=512 Len=0
3365	548.699019	120.201.184.159	192.168.1.1	TCP	54	6088 → 80 [SYN] Seq=0 Win=512 Len=0
3366	548.709766	66.114.143.237	192.168.1.1	TCP	54	6089 → 80 [SYN] Seq=0 Win=512 Len=0

Figure 9 - Attack successful with source IP randomized



## Defence using a ZBPF

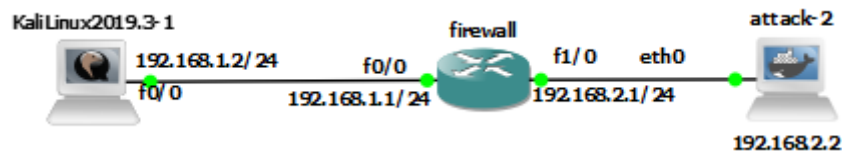


Figure 10 - Protection with ZBPF firewall topology

In this section we execute both previously mentioned attacks, but we will use an ZBPF firewall policy to defend against both attacks. To defend against SYN Flood we will use a parameter-map with the configuration "tcp synwait-time 3" to instruct the firewall that if the TCP connection is not established in 3 seconds, then close it by sending to the victim a RST packet. To defend against ICMP flood attacks we will apply some quality-of-service policies with the command police in the policy-map. The full ZBPF firewall configuration can be found in annex to this report.

### SYN Flood

As said, this attack is stopped by creating a parameter map and applying it to the policy map. The firewall will send as observed in figure 2.5, TCP packets with the RST flag to instruct the victim to terminate the connections.

```
1-parameter-map type inspect PARAMAP
2-tcp synwait-time 3
3-
4-policy-map type inspect OUT-TO-IN-POLICY
5-class type inspect HTTP-SERVER-ACCESS-ONLY
6-drop
7-class type inspect ICMP-TRAFFIC
8-inspect
9-class type inspect HTTP-TRAFFIC
10-inspect PARAMAP
```

No.	Time	Source	Destination	Protocol	Length	Info
23465	52.173712	157.206.147.197	192.168.1.2	TCP	54	9641 → 80 [SYN] Seq=0 Win=512 Len=0
23466	52.174045	192.168.1.2	157.206.147.197	TCP	60	80 → 9641 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23467	52.184649	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23468	52.184668	21.240.199.153	192.168.1.2	TCP	54	9642 → 80 [SYN] Seq=0 Win=512 Len=0
23469	52.184945	192.168.1.2	21.240.199.153	TCP	60	80 → 9642 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23470	52.195531	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23471	52.195561	242.127.242.235	192.168.1.2	TCP	54	9643 → 80 [SYN] Seq=0 Win=512 Len=0
23472	52.195913	192.168.1.2	242.127.242.235	TCP	60	80 → 9643 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23473	52.206367	147.225.180.114	192.168.1.2	TCP	54	9644 → 80 [SYN] Seq=0 Win=512 Len=0
23474	52.207182	192.168.1.2	147.225.180.114	TCP	60	80 → 9644 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23475	52.216816	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23476	52.216852	125.254.229.75	192.168.1.2	TCP	54	9645 → 80 [SYN] Seq=0 Win=512 Len=0
23477	52.217280	192.168.1.2	125.254.229.75	TCP	60	80 → 9645 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23478	52.227219	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23479	52.227244	17.236.153.40	192.168.1.2	TCP	54	9646 → 80 [SYN] Seq=0 Win=512 Len=0
23480	52.227598	192.168.1.2	17.236.153.40	TCP	60	80 → 9646 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
23481	52.238158	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23482	52.238192	43.167.195.228	192.168.1.2	TCP	60	9263 → 80 [RST] Seq=1 Win=0 Len=0
23483	52.238198	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23484	52.238204	153.184.153.247	192.168.1.2	TCP	60	9264 → 80 [RST] Seq=1 Win=0 Len=0
23485	52.238209	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23486	52.238215	1.126.234.43	192.168.1.2	TCP	60	9265 → 80 [RST] Seq=1 Win=0 Len=0
23487	52.238220	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23488	52.238238	118.202.108.20	192.168.1.2	TCP	60	9266 → 80 [RST] Seq=1 Win=0 Len=0
23489	52.238244	235.184.245.220	192.168.1.2	TCP	60	9267 → 80 [RST] Seq=1 Win=0 Len=0
23490	52.238249	192.168.1.1	192.168.1.2	ICMP	70	Destination unreachable (Host unreachable)
23491	52.238254	33.170.35.105	192.168.1.2	TCP	60	9268 → 80 [RST] Seq=1 Win=0 Len=0

> Frame 23198: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface -, id 0

Figure 11 - SYN Flood attack stopped

## ICMP Flood

The ICMP flood attack will be stopped with the configuration "police rate 128000 burst 8000" in the policy-map.

```
1-policy-map type inspect OUT-TO-IN-POLICY
2-class type inspect HTTP-SERVER-ACCESS-ONLY
3-drop
4-class type inspect ICMP-TRAFFIC
5-inspect
6-police rate 128000 burst 8000
7-class type inspect HTTP-TRAFFIC
8-inspect PARMAP
9-police rate 128000 burst 8000
```

This command applies a QoS policy, which indicates what is considered a "normal" traffic rate and how much traffic burst (fast incoming flux of packets) it can consider "OK" before deciding that too much traffic is coming, and some must be buffered and other dropped.

The parameter "rate 128000" is the average bits per second that is considered normal, anything below this rate is not anomalous.

The "burst 8000" parameter indicates, without entering in much detail about the bucket size of tokens, how much traffic coming in a short temporal range it will be able to handle before dropping.

This is successful in stopping the ICMP flood attack because the attack relies on a fast flux of data to overwhelm the victim, and this will not happen because traffic will start being dropped. We can observe that in the figures "2.6" and "2.7".

```
Police
 rate 128000 bps,8000 limit
 conformed 0 packets, 0 bytes; actions: transmit
 exceeded 0 packets, 0 bytes; actions: drop
 conformed 0 bps, exceed 0 bps

Class-map: class-default (match-any)
 Match: any
 Drop
  0 packets, 0 bytes
firewall#
```

Figure 12 - Police before attack

```
Inspect
 Packet inspection statistics [process switch:fast switch]
 icmp packets: [0:10920]

Session creations since subsystem startup or last reset 2
Current session counts (estab/half-open/terminating) [1:0:0]
Maxever session counts (estab/half-open/terminating) [1:1:0]
Last session created 00:00:29
Last statistic reset never
Last session creation rate 1
Maxever session creation rate 1
Last half-open session total 0
TCP reassembly statistics
received 0 packets out-of-order; dropped 0
peak memory usage 0 KB; current usage: 0 KB
peak queue length 0

Police
 rate 128000 bps,8000 limit
 conformed 10920 packets, 480042 bytes; actions: transmit
 exceeded 114339 packets, 4955922 bytes; actions: drop
 conformed 15000 bps, exceed 107000 bps
```

Figure 13 - Police after attack



### 3. AAA

#### TACACS+

TACACS+ authentication starts when the configured router receives a connection attempt by a user. It will first send an "Authentication" packet to the TACACS+ server. This packet contains information like the action, which in the first case will be "Inbound login", privilege level, authentication type and the accessed service (login). This packet will also contain the address of the remote user trying to access the router. The server will answer with the same message type to instruct the router to prompt the user with a username and password. For the user to be accepted, the credentials must correspond to an user entry in the TACACS+ server database.

After a successful login the router will trigger accounting messages to be sent to the TACACS+ server, because the configuration defined it to happen when the "exec" service is executed (router shell opening) with the instruction "aaa accounting exec default start-stop group tacacs+".

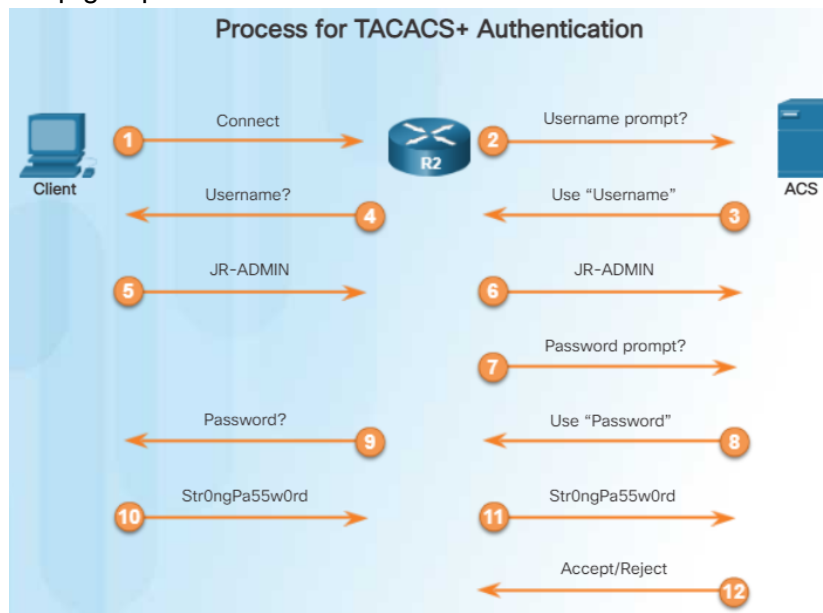


Figure 14 - TACACS Authentication Process

No.	Time	Source	Destination	Protocol	Length	Info
51	35.962312	192.168.1.2	192.168.1.1	TCP	109	[TCP Retransmission] 49 → 62358 [PSH, ACK] Seq=1 Ack=30 Win=64211 Len=55
52	35.966265	192.168.1.1	192.168.1.2	TCP	60	62358 → 49 [ACK] Seq=30 Ack=56 Win=4073 Len=0
53	44.778553	192.168.1.1	192.168.1.2	TACACS+	80	Q: Authentication
54	44.778752	192.168.1.2	192.168.1.1	TCP	54	49 → 62358 [ACK] Seq=56 Ack=56 Win=64185 Len=0
55	44.778761	192.168.1.2	192.168.1.1	TACACS+	82	R: Authentication
56	45.071992	192.168.1.1	192.168.1.2	TCP	60	62358 → 49 [ACK] Seq=56 Ack=84 Win=4045 Len=0
57	47.559685	192.168.1.1	192.168.1.2	TACACS+	80	Q: Authentication
58	47.560145	192.168.1.2	192.168.1.1	TCP	54	49 → 62358 [ACK] Seq=84 Ack=82 Win=64159 Len=0
59	47.560157	192.168.1.2	192.168.1.1	TACACS+	72	R: Authentication
60	47.560161	192.168.1.2	192.168.1.1	TCP	54	49 → 62358 [FIN, ACK] Seq=102 Ack=82 Win=64159 Len=0
61	47.570728	192.168.1.1	192.168.1.2	TCP	60	62358 → 49 [ACK] Seq=82 Ack=103 Win=4027 Len=0
62	47.570777	192.168.1.1	192.168.1.2	TCP	60	62358 → 49 [FIN, PSH, ACK] Seq=82 Ack=103 Win=4027 Len=0
63	47.570785	192.168.1.1	192.168.1.2	TCP	60	46339 → 49 [SYN] Seq=0 Win=4128 Len=0 MSS=1460
64	47.570932	192.168.1.2	192.168.1.1	TCP	54	49 → 62358 [ACK] Seq=103 Ack=83 Win=64159 Len=0
65	47.570941	192.168.1.2	192.168.1.1	TCP	58	49 → 46339 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
66	47.581718	192.168.1.1	192.168.1.2	TCP	60	46339 → 49 [ACK] Seq=1 Ack=1 Win=4128 Len=0
67	47.581777	192.168.1.1	192.168.1.2	TACACS+	130	Q: Accounting
68	47.582585	192.168.1.2	192.168.1.1	TCP	54	49 → 46339 [ACK] Seq=1 Ack=77 Win=64164 Len=0
69	47.582596	192.168.1.2	192.168.1.1	TACACS+	71	R: Accounting
70	47.582600	192.168.1.2	192.168.1.1	TCP	54	49 → 46339 [FIN, ACK] Seq=18 Ack=77 Win=64164 Len=0
71	47.592772	192.168.1.1	192.168.1.2	TCP	60	46339 → 49 [ACK] Seq=77 Ack=19 Win=4111 Len=0
72	47.592833	192.168.1.1	192.168.1.2	TCP	60	46339 → 49 [FIN, PSH, ACK] Seq=77 Ack=19 Win=4111 Len=0
73	47.592955	192.168.1.2	192.168.1.1	TCP	54	49 → 46339 [ACK] Seq=19 Ack=78 Win=64164 Len=0

Figure 15 - TACACS Authentication Process Wireshark

```

Transmission Control Protocol, Src Port: 42, Dst Port: 50743, Seq: 87, Len: 69
  TACACS+
    Major version: TACACS+
    Minor version: 0
    Type: Authentication (1)
    Sequence number: 6
    > Flags: 0x00 (Encrypted payload, Multiple Connections)
    Session ID: 1323196214
    Packet length: 6
    Encrypted Reply
  Decrypted Reply
    Status: Authentication Passed (0x01)
    Flags: 0x00
    Server message length: 0
    Data length: 0

```

Figure 16 - TACACS Authentication PASS

```

  TACACS+
    Major version: TACACS+
    Minor version: 0
    Type: Accounting (3)
    Sequence number: 1
    > Flags: 0x00 (Encrypted payload, Multiple Connections)
    Session ID: 3248555079
    Packet length: 69
    Encrypted Request
  Decrypted Request
    > Flags: 0x02
    Auth Method: TACACSPLUS (0x06)
    Privilege Level: 4
    Authentication type: ASCII (1)
    Service: Login (1)
    User len: 8
    User: saarShow
    Port len: 4

```

Figure 17 - TACACS Accounting Message

The Figure 17 - TACACS Accounting Message, demonstrates that there was a successful login made by the user "saarShow" with privilege level 4. The same information can be observed in the Figure 16 - TACACS Authentication PASS but the latter is to inform the router of the authentication decision.

The second important packet in this protocol is the "Authorization" packet. This type is used to query the TACACS+ server if the user is allowed to execute the action specified by the user. This packet will contain the user identifier, it's remote address, its privilege level and the command.

```

Admin-Router#systat
Command authorization failed.
Admin-Router#

```

Figure 18 - TACACS Authorization Request by saarShow user

This user as specified in the requirements, is only allowed to use the "show" command and the cmd value is the command "systat". The correct behaviour would be to have the command execution rejected, and this is what happens as it can be observed in the Figure 19 - TACACS Authorization Request fail by saarShow user This is only possible with TACACS+ as Cisco as not implemented authorization for the RADIUS protocol in order to make its protocol preferred.

```

▼ TACACS+
  Major version: TACACS+
  Minor version: 0
  Type: Authorization (2)
  Sequence number: 2
  > Flags: 0x00 (Encrypted payload, Multiple Connections)
  Session ID: 4192903335
  Packet length: 6
  Encrypted Reply
  ▼ Decrypted Reply
    Auth Status: FAIL (0x10)
    Server Msg length: 0
    Data length: 0
    Arg count: 0

```

Figure 19 - TACACS Authorization Request fail by saarShow user

## RADIUS

The RADIUS protocol works in a similar way in the sense of the router querying the RADIUS server for an authentication decision, but the router will only send the request to the server after prompting the user for the credentials. Therefore, minimizing the traffic generated between the router and authentication server.

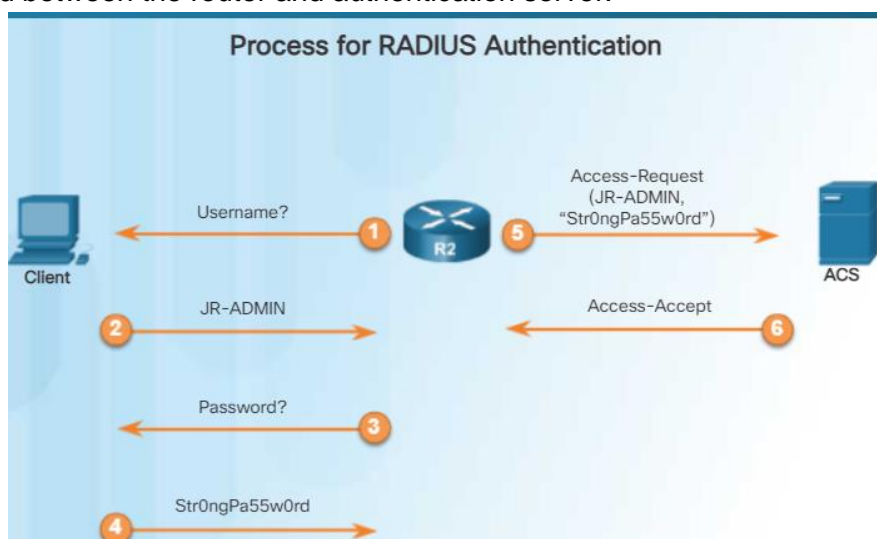


Figure 20 - RADIUS Protocol

40	293.129952	192.168.1.1	192.168.1.2	RADIUS	115	Access-Request id=3
41	293.130313	192.168.1.2	192.168.1.1	RADIUS	111	Access-Accept id=3
42	293.151976	192.168.1.1	192.168.1.2	RADIUS	131	Accounting-Request id=1
43	293.184749	192.168.1.2	192.168.1.1	RADIUS	62	Accounting-Response id=1

```

> Frame 40: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface -, id 0
> Ethernet II, Src: ca:03:04:64:00:00 (ca:03:04:64:00:00), Dst: 16:42:b3:c4:6f:9c (16:42:b3:c4:6f:9c)
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2
> User Datagram Protocol, Src Port: 1645, Dst Port: 1812
▼ RADIUS Protocol
  Code: Access-Request (1)
  Packet identifier: 0x3 (3)
  Length: 73
  Authenticator: ffd57330db133e4a510029b4034bf97
  [The response to this request is in frame 41]
  ▼ Attribute Value Pairs
    > AVP: t=User-Name(1) l=11 val=saarBrief
    > AVP: t=User-Password(2) l=18 val=Encrypted
    > AVP: t=NAS-Port(5) l=6 val=2
    > AVP: t=NAS-Port-Id(87) l=6 val=tty2
    > AVP: t=NAS-Port-Type(61) l=6 val=Virtual(5)
    > AVP: t=NAS-IP-Address(4) l=6 val=192.168.1.1

```

Figure 21 - RADIUS Protocol Wireshark

It can be observed in the Wireshark capture of the protocol that the username and password are sent in the same packet and triggering the same accounting messages.

Cisco router does not allow for an authorization mechanism like TACACS+, and as such, is not possible to limit which commands users can execute, but only to set their privilege level and forbidding them of entering a privileged shell with an enable password.

```
Admin-Router(config)#aaa authorization commands 1 default group radius
Admin-Router(config)#exit
*Apr 12 14:18:34.407: %AAAA-4-SERVNOTACPLUS: The server-group "radius" is not a tacacs+ server group. Please define "radius" as a tacacs+ server group.
Admin-Router(config)#exit
```

Figure 22 - RADIUS Authorization Attempt

```
root@Telnet-Client:~# telnet 192.168.2.1
Trying 192.168.2.1...
Connected to 192.168.2.1.
Escape character is '^]'.

User Access Verification

Username: saarBrief
Password:
Welcome saarBrief

Admin-Router#show privilege
Current privilege level is 2
Admin-Router#en
Password:
% Error in authentication.

Admin-Router#conf t
^
% Invalid input detected at '^' marker.

Admin-Router#sysstat
      Line      User      Host(s)      Idle      Location
    *  0 con 0    saarAdmin  idle        00:01:52
    *  2 vty 0    saarBrief  idle        00:00:00  192.168.2.2

Interface      User      Mode      Idle      Peer Address
```

Figure 23 - RADIUS showBrief user

No.	Time	Source	Destination	Protocol	Length	Info
80	538.492570	192.168.1.1	192.168.1.2	RADIUS	115	Access-Request id=4
81	538.493535	192.168.1.2	192.168.1.1	RADIUS	112	Access-Accept id=4
82	538.513994	192.168.1.1	192.168.1.2	RADIUS	131	Accounting-Request id=3
83	538.515314	192.168.1.2	192.168.1.1	RADIUS	62	Accounting-Response id=3

> Frame 80: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface -, id 0

> Ethernet II, Src: ca:03:04:64:00:00 (ca:03:04:64:00:00), Dst: 16:42:b3:c4:6f:9c (16:42:b3:c4:6f:9c)

> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2

> User Datagram Protocol, Src Port: 1645, Dst Port: 1812

▼ RADIUS Protocol

Code: Access-Request (1)

Packet Identifier: 0x4 (4)

Length: 73

Authenticator: be05782bd5f4bc7b012bf47c88f172c8

[The response to this request is in frame 81]

▼ Attribute Value Pairs

- > AVP: t=User-Name(1) l=11 val=saarAdmin
- > AVP: t=User-Password(2) l=18 val=Encrypted
- > AVP: t=NAS-Port(5) l=6 val=2
- > AVP: t=NAS-Port-Id(87) l=6 val=tty2
- > AVP: t=NAS-Port-Type(61) l=6 val=Virtual(5)
- > AVP: t=NAS-IP-Address(4) l=6 val=192.168.1.1

```
clns
CLNS network information
Admin-Router#sh bgp
% BGP not active
Admin-Router#sh run
^
% Invalid input detected at '^' marker.
Admin-Router#sh run
^
% Invalid input detected at '^' marker.
Admin-Router#exit
Connection closed by foreign host.
root@Telnet-Client:~# telnet 192.168.2.1
Trying 192.168.2.1...
Connected to 192.168.2.1.
Escape character is '^]'.

User Access Verification

Username: saarAdmin
Password:
Welcome saarAdmin

Admin-Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Admin-Router(config)#
```

Figure 24 - RADIUS saarAdmin commands

802.11x

The 802.11x protocol operates on switch ports. A supplicant must first authenticate himself by talking with an "Authenticator" when connecting to the port, which in turn will communicate with the Authentication Server thorough the RADIUS protocol.

The messages between the supplicant and authenticator will use the EAPOL encapsulation technique for communication between the two entities and EAP to do the authentication.

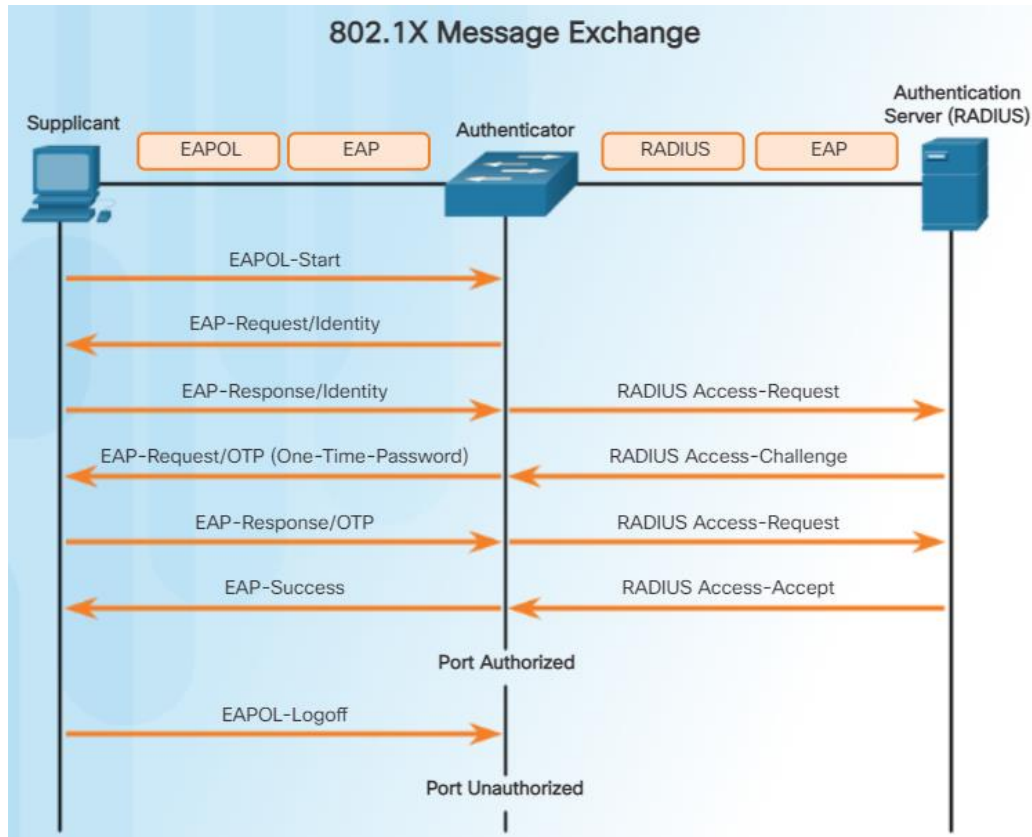


Figure 25 - 802.11x message Exchange

The process starts with the supplicant sending an EAPOL-Start message. The switch will send an EAP-Request Identity message and the supplicant will answer with an EAP-Response Identity message.

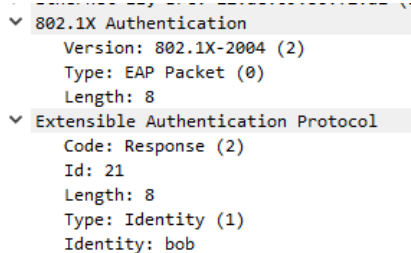


Figure 26 - EAP Identity Response

The authenticator will then communicate with the authentication server by sending a RADIUS Access-Request message. The RADIUS server will then select an access challenge and send it back to the authenticator which sends it to the supplicant. In this case as it can be observed in the Wireshark captures that the challenge is "EAP-MD5-CHALLENGE". If the challenge response is correct then the supplicant will be authenticated and allowed to access other resources through the switch, bypassing the authentication the next time.

```

14 25.037683 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... EAPOL 18 Start
15 25.045370 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... EAP 60 Request, Identity
16 25.045763 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... EAP 26 Response, Identity
17 25.063983 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... EAP 60 Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
18 25.064432 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... EAP 24 Response, Legacy Nak (Response Only)
19 25.083188 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... EAP 60 Request, Protected EAP (EAP-PEAP)
20 25.083789 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... TLSv1.2 218 Client Hello
21 25.095314 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... EAP 1022 Request, Protected EAP (EAP-PEAP)
22 25.095399 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... EAP 24 Response, Protected EAP (EAP-PEAP)
23 25.105571 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... TLSv1.2 200 Server Hello, Certificate, Server Key Exchange, Server Hello Done
24 25.106618 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... TLSv1.2 154 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
25 25.117492 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... TLSv1.2 75 Change Cipher Spec, Encrypted Handshake Message
26 25.117701 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... EAP 24 Response, Protected EAP (EAP-PEAP)
27 25.127317 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... TLSv1.2 60 Application Data
28 25.127407 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... TLSv1.2 57 Application Data
29 25.138028 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... TLSv1.2 92 Application Data
30 25.138225 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... TLSv1.2 111 Application Data
31 25.147213 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... TLSv1.2 100 Application Data
32 25.147608 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... TLSv1.2 55 Application Data
33 25.157279 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... TLSv1.2 64 Application Data
34 25.157580 12:ae:09:86:f2:a1 Nearest-non-TPMR-br... TLSv1.2 64 Application Data
35 25.173166 0c:ce:ce:b3:86:00 Nearest-non-TPMR-br... EAP 60 Success

```

Figure 27 - EAP messages between supplicant and switch

130	182.228629	192.168.1.3	192.168.1.4	RADIUS	281 Access-Request id=21
131	182.229000	192.168.1.4	192.168.1.3	RADIUS	122 Access-Challenge id=21
132	182.246394	192.168.1.3	192.168.1.4	RADIUS	297 Access-Request id=22
133	182.246769	192.168.1.4	192.168.1.3	RADIUS	118 Access-Challenge id=22
134	182.261398	192.168.1.3	192.168.1.4	RADIUS	491 Access-Request id=23
135	182.262905	192.168.1.4	192.168.1.3	RADIUS	1110 Access-Challenge id=23
136	182.272625	192.168.1.3	192.168.1.4	RADIUS	297 Access-Request id=24
137	182.272857	192.168.1.4	192.168.1.3	RADIUS	282 Access-Challenge id=24
138	182.283439	192.168.1.3	192.168.1.4	RADIUS	427 Access-Request id=25
139	182.283847	192.168.1.4	192.168.1.3	RADIUS	157 Access-Challenge id=25
140	182.294063	192.168.1.3	192.168.1.4	RADIUS	297 Access-Request id=26
141	182.294287	192.168.1.4	192.168.1.3	RADIUS	140 Access-Challenge id=26
142	182.303712	192.168.1.3	192.168.1.4	RADIUS	330 Access-Request id=27
143	182.304315	192.168.1.4	192.168.1.3	RADIUS	174 Access-Challenge id=27
144	182.314436	192.168.1.3	192.168.1.4	RADIUS	384 Access-Request id=28
145	182.314767	192.168.1.4	192.168.1.3	RADIUS	182 Access-Challenge id=28
146	182.324843	192.168.1.3	192.168.1.4	RADIUS	328 Access-Request id=29
147	182.325108	192.168.1.4	192.168.1.3	RADIUS	146 Access-Challenge id=29
148	182.334284	192.168.1.3	192.168.1.4	RADIUS	337 Access-Request id=30
149	182.334542	192.168.1.4	192.168.1.3	RADIUS	207 Access-Accept id=30

Figure 28 - RADIUS messages between switch and RADIUS server

```

root@attacker-latest-1:/home# cat wpa.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=2
ap_scan=0
network={
    key_mgmt=IEEE8021X
    eap=PEAP
    identity="bob"
    password="gns3"
    phase2="authchap=MSCHAPV2"
    eapol_flags=0
}
root@attacker-latest-1:/home# wpa_supplicant -Dwired -ieth0 -c /home/wpa.conf &
[1] 66
root@attacker-latest-1:/home# Successfully initialized wpa_supplicant
eth0: Associated with 01:80:c2:00:00:03
eth0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
eth0: CTRL-EVENT-EAP-STARTED EAP authentication started
eth0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=4 -> NAK
eth0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
eth0: CTRL-EVENT-EAP-METHOD EAP vendor 0 method 25 (PEAP) selected
eth0: CTRL-EVENT-EAP-PEER-CERT depth=0 subject='/CN=3569def2f404' hash=391181dd6d949826bfcdea70129596f52b9bc97a487e59d3cd7d6dd9487780f2
eth0: CTRL-EVENT-EAP-PEER-ALT depth=0 DNS:3569def2f404
eth0: CTRL-EVENT-EAP-PEER-CERT depth=0 subject='/CN=3569def2f404' hash=391181dd6d949826bfcdea70129596f52b9bc97a487e59d3cd7d6dd9487780f2
eth0: CTRL-EVENT-EAP-PEER-ALT depth=0 DNS:3569def2f404
EAP-MSCHAPV2: Authentication succeeded
EAP-TLV: TLV Result - Success - EAP-TLV/Phase2 Completed
eth0: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
eth0: CTRL-EVENT-CONNECTED - Connection to 01:80:c2:00:00:03 completed [id=0 id_str=]

```

Figure 29 - EAP Authenticated

Telnet Client:  
ip 192.168.2.2/24

C7200 Administrated Router:  
conf t  
int f1/0  
no shut  
ip address 192.168.2.1 255.255.255.0  
exit  
int f0/0  
no shut  
ip address 192.168.1.1 255.255.255.0  
exit  
username admin-router secret admin-router  
aaa new-model  
aaa authentication login default group tacacs+ local  
tacacs server TACACS-AUTH  
address ipv4 192.168.1.2  
single-connection  
key gns3  
exit  
aaa authorization exec default group tacacs+  
aaa authorization commands 1 default group tacacs+  
aaa accounting exec default start-stop group tacacs+  
aaa accounting network default start-stop group tacacs+  
enable secret admin-router  
exit

AAA Apliance - AAA Server:  
key = saar

users & pass's:  
saarAdmin:saarAdmin (privilege 15 - must have permission to use all IOS commands)  
saarShow:saarShow (privilege 4 - can only show commands)  
saarBrief:saarBrief (privilege 2 - can only use "show ip int brief")

/etc/tacacs+/tac\_plus.conf: (enable persistence)  
accounting file = /var/log/tac\_plus.acct  
key = gns3

```
user = saarAdmin {
    name = "saarAdmin"
    login = cleartext saarAdmin
    member = admin
}
```

# Grupo admin default do ficheiro tac\_plus

```
user = saarShow {
    name = "saarShow"
    login = cleartext saarShow

    service = exec {
        priv-lvl = 4
    }

    cmd = show {
        permit .*
    }
}
```

```

user = saarBrief {
    name = "saarBrief"
    login = cleartext saarBrief

    service = exec {
        priv-lvl = 2
    }

    cmd = show {
        permit ip int br
        deny .*
    }
}

```

Telnet Client:  
ip 192.168.2.2/24

C7200 Administrated Router:

```

conf t
int f1/0
no shut
ip address 192.168.2.1 255.255.255.0
exit
int f0/0
no shut
ip address 192.168.1.1 255.255.255.0
exit
username admin-router secret admin-router
aaa new-model
aaa authentication login default group radius local
radius server RADIUS-AUTH
address ipv4 192.168.1.2 auth-port 1812 acct-port 1813
key gns3
exit
aaa authorization exec default group radius local
aaa authorization network default group radius local
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
enable secret admin-router
exit

```

AAA Apliance - AAA Server:  
secret = gns3

```

/etc/freeradius/3.0/users:
saarAdmin    Cleartext-Password := "saarAdmin"
              Reply-Message = "Welcome %{User-Name}",
              Service-Type = Administrative-User,
              Cisco-AVPair += "shell:priv-lvl=15"

saarShow     Cleartext-Password := "saarShow"
              Reply-Message = "Welcome %{User-Name}",
              Service-Type = NAS-Prompt-User,
              Cisco-AVPair += "shell:priv-lvl=4"

saarBrief    Cleartext-Password := "saarBrief"
              Reply-Message = "Welcome %{User-Name}",
              Service-Type = NAS-Prompt-User,
              Cisco-AVPair += "shell:priv-lvl=2"

```



Docker attacker:  
ip 192.168.1.2/24

IOSvL2 switch:  
en  
conf t  
int vlan 1  
ip address 192.168.1.3 255.255.255.0  
no shut  
exit  
aaa new-model  
radius server 8021X-RADIUS  
address ipv4 192.168.1.4 auth-port 1812 acct-port 1813  
key gns3  
exit  
aaa authentication dot1x default group radius  
dot1x system-auth-control  
interface gi0/0  
description Access Port  
switchport mode access  
authentication port-control auto  
dot1x pae authenticator  
exit

AAA Appliance - RADIUS Server:  
ip 192.168.1.4/24

/etc/freeradius/3.0/users:  
bob:gns3  
alice:gns3  
(users default)

## 4. ASA Firewall

This section has the goal of exploring a couple functionalities of the Cisco ASA virtual appliance. The Cisco ASA integrates firewall technology, intrusion prevention system, high-performance VPN's and fault tolerance and many other functionalities. The additional exercise chosen to test some of those functionalities were DNS Doctoring and SSH Cipher mode configuration to disable the CBC encryption mode of AES.

### SSH Configuration

The purpose of this functionality is to configure which algorithms the ASA supports. This can help prevent vulnerabilities and increase security. In this exercise we decided to disable the CBC encryption mode and permit only CTR mode, as it was found that it can be possible under certain conditions to abuse SSH and allow an attacker to recover the plaintext associated with a block of ciphertext (<https://www.kb.cert.org/vuls/id/958563>). In both exercises the group decided to use the terminal to implement the configurations instead of ASDM.

We used the following topology to implement the configurations:



Figure 30 - SSH Topology

We used the following configuration:

```

Attacker Outside:
192.168.1.2/24
gw 192.168.1.1/24

Client Inside:
192.168.2.2/24
gw 192.168.2.1/24

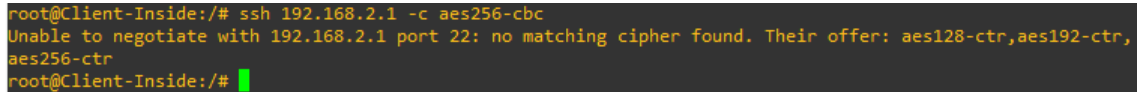
ASA:
en
conf t
int gi0/0
nameif inside
security-level 100
ip address 192.168.2.1 255.255.255.0
no shut
exit
int gi0/1
nameif outside
security-level 0
ip address 192.168.1.1 255.255.255.0
no shut
exit
policy-map global_policy
class inspection_default
inspect icmp
exit
exit
enable password saar
  
```

```

username saar password saar
aaa authentication ssh console LOCAL
crypto key generate rsa modulus 1024
ssh 192.168.2.2 255.255.255.0 trust
ssh cipher encryption custom aes128-ctr:aes192-ctr:aes256-ctr
exit

```

When trying now to establish an SSH connection from the client to the ASA device we receive the following alert:



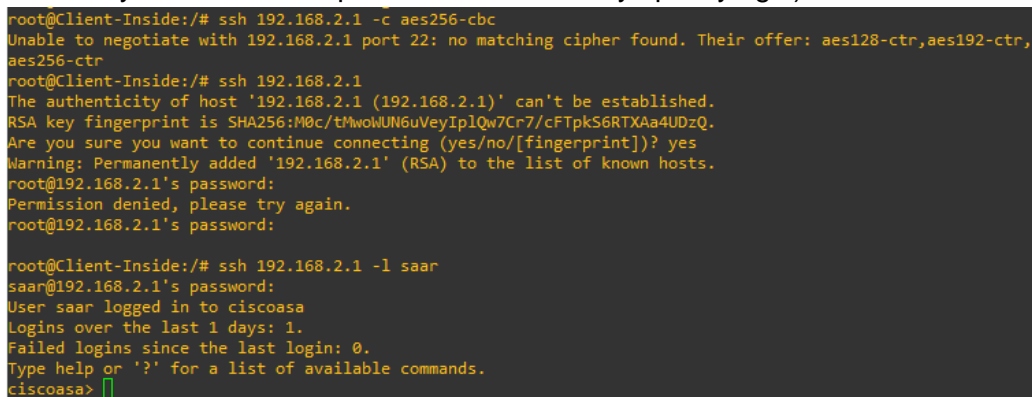
```

root@Client-Inside:/# ssh 192.168.2.1 -c aes256-cbc
Unable to negotiate with 192.168.2.1 port 22: no matching cipher found. Their offer: aes128-ctr,aes192-ctr,
aes256-ctr
root@Client-Inside:/#

```

Figure 31 - SSH Limited Algorithms

So we must use the correct algorithms to establish SSH (which are negotiated automatically and doesn't require the user manually specifying it)



```

root@Client-Inside:/# ssh 192.168.2.1 -c aes256-cbc
Unable to negotiate with 192.168.2.1 port 22: no matching cipher found. Their offer: aes128-ctr,aes192-ctr,
aes256-ctr
root@Client-Inside:/# ssh 192.168.2.1
The authenticity of host '192.168.2.1 (192.168.2.1)' can't be established.
RSA key fingerprint is SHA256:M0c/tMw0WUN6uVeyIp1Qw7Cr7/cFTpkS6RTXAa4UDzQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.1' (RSA) to the list of known hosts.
root@192.168.2.1's password:
Permission denied, please try again.
root@192.168.2.1's password:

root@Client-Inside:/# ssh 192.168.2.1 -l saar
saar@192.168.2.1's password:
User saar logged in to ciscoasa
Logins over the last 1 days: 1.
Failed logins since the last login: 0.
Type help or '?' for a list of available commands.
ciscoasa>

```

Figure 32 - SSH Working

## DNS Doctoring

This functionality is used when there is a DNS server that resolves hostnames to IP's that require NAT translations. This problem has 2 different perspectives. When the DNS server is in Internet and announces public IPs to clients that are in the same network segment as the server and therefore actually need the private IP, or, when the DNS server is in the private network and is announcing the private IP of the server, which is not routable from outside. Both solutions to the two different sides are implemented by configuring DNS Doctoring, which alters the embedded IPs of DNS responses.

To implement this functionality the group decided to use the DNS server in the private network, and therefore having to configure DNS Doctoring to translate the private IP address of the DNS response to the associated public IP, for example when the External-Client in the image below sends an DNS Query.

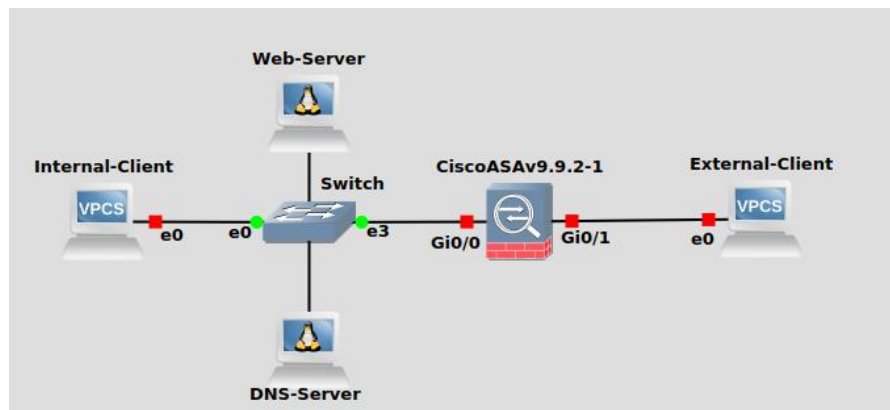


Figure 33 - DNS Doctoring Topology

The DNS Doctoring configuration is implemented by adding "dns inspect" to the class inspection\_default of the global policy-map and making a NAT Translation of private IP to public IP by creating a NAT network object:

```
1-policy-map global_policy
2-class inspection_default
3-inspect icmp
4-exit
5-exit
6-object network dns-server
7-host 192.168.1.4
8-nat (inside,outside) source static 200.0.0.10
```

The group does not present the full configuration of the network, as unfortunately it was not able to successfully implement this functionality for lack of time and AAA configurations by the ASA which mysteriously dropped communications from the "Outside" interface to the "Inside" interface even with the same security-level (100) and ACLs to allow the communication.