

SUNY POLYTECHNIC INSTITUTE

NCS 450

NETWORK SECURITY

Lab 7: ASA VPN

Author:

Jess YANARELLA

Tony PEREZ

Professor:

Ronny BULL

April 2015

1 Introduction

In this lab we setup a encrypted tunnel between two pods in our classroom network. Both our pods are different networks and it will show that we can transfer data and secure it by encrypting it over an open network.

2 Diagram

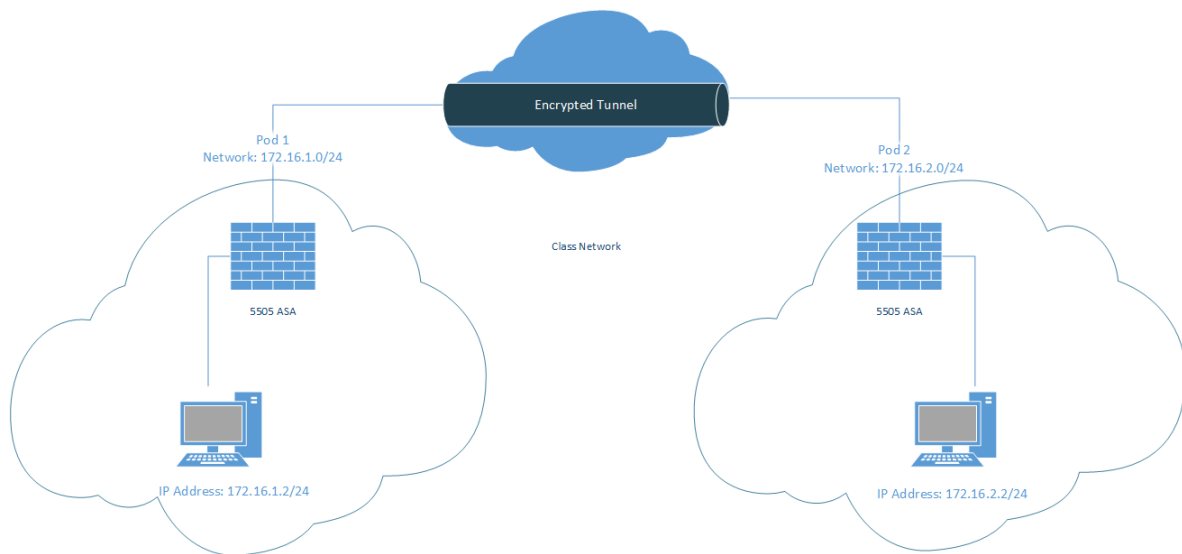


Figure 1: Network Setup

3 Bring the ASA Online:

3.1 Set host name and password:

On our ASA, we need to set up the basics for the configuration and give it a hostname and password.

```
(config)# hostname ciscoasa
(config)# enable password dh1240
```

```
hostname ciscoasa
domain-name cs.sunyit.edu
enable password I8utyYWFySgj7DF encrypted
passwd 2KFQnbNIIdI.2KYOU encrypted
```

Figure 2: Hostname

3.2 Attach to the network:

The first steps in setting up the network properly is to attach the external interface and vlan to the classroom network. First, we need to remove vlan1 from the ASA. This is done because we don't want to use a default vlan for the tunnel.

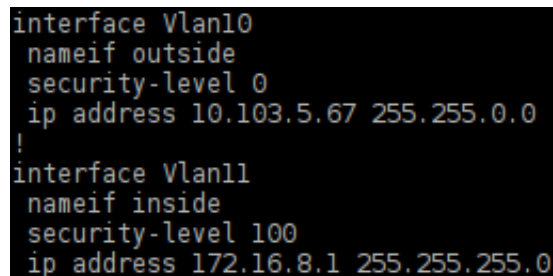
```
(config)# no int vlan1
```

We will be using vlan10 for our external vlan so we create it and assign it an interface. It will be our outside interface with the IP address 10.103.5.67/16.

```
(config)# int vlan10
(config)# nameif outside
(config)# ip address 10.103.5.65 255.255.0.0
```

Once the setup is configured, we can enter the interface for the actual physical Ethernet port. Our vlan10 will be assigned to this physical port and must be enabled.

```
(config)# int Ethernet 0/0
(config)# switchport access vlan 10
(config)# no shutdown
```



```
interface Vlan10
 nameif outside
 security-level 0
 ip address 10.103.5.67 255.255.0.0
!
interface Vlan11
 nameif inside
 security-level 100
 ip address 172.16.8.1 255.255.255.0
```

Figure 3: Vlan10 Configuration

3.3 Verify network connectivity and set default route:

To test that our network is active, we will ping the atlantis server.

```
# ping 10.103.0.25
```

Our ping was successful so we can now assign it a default route to the ASA.

```
(config)# route outside 0 0 10.103.0.1
```

With this route set up, we can now ping further in the network including Fang.

```
# ping 150.156.192.11
```

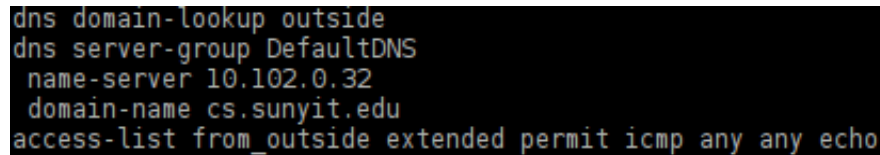
3.4 Configure DNS:

DNS is set up on the ASA so we can communicate by host names instead of using the IP addresses. We will enable DNS lookup using the outside interface and create a DNS server-group.

```
# dns domain-lookup outside
# dns server-group DefaultDNS
```

Also, we need to set the DNS server for queriers and set the system domain name, which is cs.sunyit.edu

```
# name server 150.156.192.72
# domain-name cs.sunyit.edu
```



```
dns domain-lookup outside
dns server-group DefaultDNS
name-server 10.102.0.32
domain-name cs.sunyit.edu
access-list from_outside extended permit icmp any any echo
```

Figure 4: Domain

Now, we can ping atlantis without typing the IP address.

```
# ping atlantis
```

4 Configure ASA internal network:

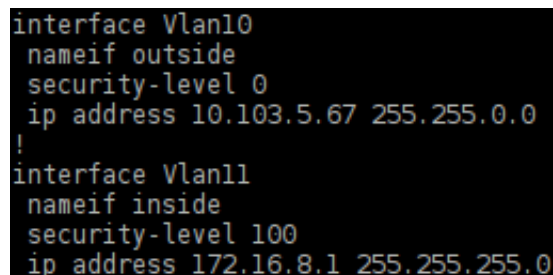
4.1 Create an internal VLAN:

We already created vlan10 earlier, now we will create vlan11. This will be our internal vlan and it needs an IP address as well.

```
(config-if)# ip address 172.16.8.1 255.255.255.0
```

This vlan will be assigned to our interface Ethernet 0/1.

```
(config-if)# switchport access vlan11
```



```
interface Vlan10
nameif outside
security-level 0
ip address 10.103.5.67 255.255.0.0
!
interface Vlan11
nameif inside
security-level 100
ip address 172.16.8.1 255.255.255.0
```

Figure 5: Vlan 11

4.2 Configure internal NAT:

We need to enable NAT for traffic on the internal subnet and utilize the outside interface.

```
(config)# nat (inside) 1 172.16.2.0 255.255.255.0
```

```
(config)# global (outside 1 interface
```

```

interface Ethernet0/0
  switchport access vlan 10
!
interface Ethernet0/1
  switchport access vlan 11

```

Figure 6: Ethernet Interfaces

4.3 Move your Pod PC to the Pod LAN:

We need to connect our pod PC to ASA port 1 and assign our PC an IP address 172.16.2.2. We also need to give it a default gateway.

```
route add default gw 172.16.2.1
```

```

yanarej@tequila ~ $ sudo ifconfig eth0 172.16.2.2 netmask 255.255.255.0
yanarej@tequila ~ $ sudo route add default gw 172.16.2.1
yanarej@tequila ~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.2.2 netmask 255.255.255.0 broadcast 172.16.2.255
    inet6 fe80::82c1:6eff:febd:28a1 prefixlen 64 scopeid 0x20<link>
    ether 80:c1:6e:fd:28:a1 txqueuelen 1000 (Ethernet)
    RX packets 15039 bytes 12848365 (12.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6054 bytes 647030 (631.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xfe700000-fe720000

```

Figure 7: Pod PC Config

4.4 Enable ICMP traffic through the ASA:

We need to make sure our ASA will allow ping tests from the internal network. First, we will create an access list to permit the traffic. Next we will need to enter policy_map config mode and add the default inspection list. Also, we must instruct the ASA to globally inspect ICMP traffic.

```

(config)# access-list from_outside extended permit icmp any any echo
(config)# policy-map global_policy
(config)# class inspection_default
(config)# inspect icmp

```

4.5 Verify internal Pod PC traffic:

To test our Pod LAN PC, we will ping fang.

```
yanarej@tequila ~ $ ping fang
PING fang.cs.sunyit.edu (150.156.192.11) 56(84) bytes of data:
64 bytes from fang.cs.sunyit.edu (150.156.192.11): icmp_seq=1 ttl=61 time=2.01 ms
64 bytes from fang.cs.sunyit.edu (150.156.192.11): icmp_seq=2 ttl=61 time=1.28 ms
64 bytes from fang.cs.sunyit.edu (150.156.192.11): icmp_seq=3 ttl=61 time=1.27 ms
^C
--- fang.cs.sunyit.edu ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.276/1.525/2.018/0.349 ms
```

Figure 8: Pod PC fang

5 Build the encrypted tunnel:

5.1 Enable ISAKMP on the outside Interface:

For the encrypted tunnel to work, the protocol ISAKMP must be enabled. This will be set up on our ASA's outside interface. It is used for establishing security associations and managing security keys.

```
(config)# crypto isakmp enable outside
```

```
pod2asa# show crypto isakmp sa

  Active SA: 1
  Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)
Total IKE SA: 1

1  IKE Peer: 10.103.5.51
   Type    : L2L           Role    : responder
   Rekey    : no           State   : MM ACTIVE
```

Figure 9: ISAKMP

5.2 Build the tunnel group and tunnel ACL:

The tunnel group will be used to transport traffic between the two ASA end points. It is created to identify the outside interface of the remote ASA.

```
tunnel-group 10.103.5.67 type ipsec-l2l
```

Now, we need to specify the address of the remote ASA outside interface.

```
tunnel-group 10.103.5.51 ipsec-attributes
```

For the connection to be authenticated, the same pre shared key must be created on both ASA's.

```
pre-shared-key cisco
```

Next, we need to specify the number of seconds that the peer is allowed to idle before starting keepalive monitoring. We will be using 2 seconds for the retry parameter.

```
isakmp keepalive threshold 10 retry 2
```

Lastly in this section, we need to create an access control list to identify traffic flow from inside our subnet. This traffic will then be sent through our encrypted tunnel when created.

```
access-list outside_tunnel extended permit ip 172.16.2.0 255.255.255.0
172.16.1.0 255.255.255.0
```

5.3 Phase 1:

This phase will allow us to authenticate the IPsec peer end points for the secure channel IKE to be established. This connection will set the encryption algorithm, hash algorithm, authentication method, and Diffie-Hellman group to be used. After this phase, both peers will have authentication of the matching shared keys. We will first need to create an ISAKMP policy and set up AES encryption on it. The hashing algorithm we need for the ASA's will be SHA-1. Also, we need to specify the authentication for it to take place.

```
(config)# crypto isakmp policy 10
(config)# encryption aes
(config)# hash sha
(config)# authentication pre-share
```

The Diffie-Hellman group that we specified earlier, now needs to be assigned to a group. We will use Group 2, which generates a 1024 bit shared secret. The secret keys can be negotiated after a certain number seconds and for this example, we will use 86400.

```
(config)# group 2
(config)# lifetime 86400
```

5.4 Phase 2:

In this phase we will use the IKE peers that authenticated with each other and are now using a secret key which they have agreed on. Using this secure channel we will negotiate an IPSec security association. Using the secret keys, we will create a IPSec tunnel between the two peers which are comprised of two unidirectional security associations. This is the encrypted tunnel we will be using to transfer data.

First we create an IPSec policy to specify the encryption and hashing algorithms to use for phase 2 security. This transform set will use AES encrypted Encapsulating Security Payload(ESP) with SHA-1 hashing. ESP provides authentication of the remote peer, encryption of the data, and anti-replay protection.

```
crypto ipsec transform-set ESP-AES-SHA esp-aes esp-sha-hmac
```

Now we make a crypto map entry to identify the traffic flows to be secured. We will use be referring to our "outside_tunnel" ACL which we created and set previously to only permit

our network to the remote pod network.

```
crypto map outside_map 20 match address outside_tunnel
```

Next we will identify the IP address of the remote pod's external ASA interface.

```
crypto map outside_map 20 set peer 10.103.5.1
```

We specify the previously created transform set to apply to the tunnel.

```
crypto map outside_map 20 set transform-set ESP-AES-SHA
```

Now we will apply the crypto map to the outside named interface on our ASA.

```
crypto map outside_map interface outside
```

5.5 Disable NAT for Tunnel Traffic

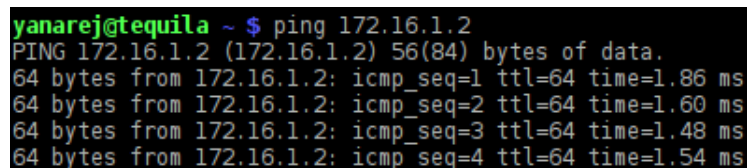
We will make a NAT statement to prevent traffic identified by the "outside_tunnel" ACL from being NAT'ed. With the encrypted tunnel we have made a direct route from our local network to the remote pod's local network.

```
nat (inside) 0 access-list outside_tunnel
```

6 Test Encrypted Tunnel

To test the communication from our local network to the remote network we used our Pod PC to see if it could talk to the remote Pod PC.

We will ping Pod 1's PC with IP: 172.16.1.2/24 from our Pod PC with IP: 172.16.2.2/24.



```
yanarej@tequila ~ $ ping 172.16.1.2
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data:
64 bytes from 172.16.1.2: icmp_seq=1 ttl=64 time=1.86 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=64 time=1.60 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=64 time=1.48 ms
64 bytes from 172.16.1.2: icmp_seq=4 ttl=64 time=1.54 ms
```

Figure 10: Remote PC ping Test

Next we will attempt to ssh into the remote Pod PC.

This shows us a successful ssh login. This also shows us the hostname of that PC, "morris". Next, we will see the eth0 interface configuration.


```

yanarej@tequila ~ $ ssh yanarej@172.16.1.2
The authenticity of host '172.16.1.2 (172.16.1.2)' can't be established.
ECDSA key fingerprint is 84:6d:0f:a9:4d:16:3a:fd:6f:73:60:64:2c:cf:5f:ae.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.1.2' (ECDSA) to the list of known hosts.
Password:
Creating directory '/home/undergrad/yanarej'.
(morris:~) yanarej>

```

Figure 11: SSH Test

```

(morris:~) yanarej> ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.1.2 netmask 255.255.255.0 broadcast 172.16.1.255
    inet6 fe80::82c1:6eff:febd:3288 prefixlen 64 scopeid 0x20<link>
    ether 80:c1:6e:fd:32:88 txqueuelen 1000 (Ethernet)
    RX packets 255594 bytes 67001736 (63.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 52748 bytes 10581803 (10.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xfe700000-fe720000

```

Figure 12: Remote Host Network Information

We can also see it's routes.

```

(morris:~) yanarej> route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 172.16.1.1 0.0.0.0 UG 0 0 0 eth0
loopback * 255.0.0.0 U 0 0 0 lo
loopback localhost 255.0.0.0 UG 0 0 0 lo
172.16.1.0 * 255.255.255.0 U 0 0 0 eth0

```

Figure 13: Remote PC's Routes

6.1 Verify Encryption

Now that we have logged in to the remote Pod PC, we can verify that the data sent through the tunnel is actually encrypted in the ASA.

```

pod2asa# show crypto isakmp sa

Active SA: 1
Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)
Total IKE SA: 1

1 IKE Peer: 10.103.5.51
  Type    : L2L           Role    : responder
  Rekey    : no           State   : MM ACTIVE

```

Figure 14: Show the IKE Peer we are connected to

```

pod2asa# show crypto ipsec sa
interface: outside
  Crypto map tag: outside_map, seq num: 20, local addr: 10.103.5.67

  access-list outside_tunnel extended permit ip 172.16.2.0 255.255.255.0 172.16.1.0 255.255.255.0
  local ident (addr/mask/prot/port): (172.16.2.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port): (172.16.1.0/255.255.255.0/0/0)
  current_peer: 10.103.5.51

  #pkts encaps: 12, #pkts encrypt: 12, #pkts digest: 12
  #pkts decaps: 12, #pkts decrypt: 12, #pkts verify: 12
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 12, #pkts comp failed: 0, #pkts decomp failed: 0
  #pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0
  #PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0
  #send errors: 0, #recv errors: 0

  local crypto endpt.: 10.103.5.67, remote crypto endpt.: 10.103.5.51

  path mtu 1500, ipsec overhead 74, media mtu 1500
  current outbound spi: B1841E5B
  current inbound spi : B651BD8E

inbound esp sas:
  spi: 0xB651BD8E (3058810254)
    transform: esp-aes esp-sha-hmac no compression
    in use settings = {L2L, Tunnel, }
    slot: 0, conn_id: 4096, crypto-map: outside_map
    sa timing: remaining key lifetime (kB/sec): (4373982/28119)
    IV size: 16 bytes
    replay detection support: Y
    Anti replay bitmap:
      0xFFFFFFFF 0xFFFFFFFF
outbound esp sas:
  spi: 0xB1841E5B (2978225755)
    transform: esp-aes esp-sha-hmac no compression
    in use settings = {L2L, Tunnel, }
    slot: 0, conn_id: 4096, crypto-map: outside_map
    sa timing: remaining key lifetime (kB/sec): (4373982/28119)
    IV size: 16 bytes
    replay detection support: Y
    Anti replay bitmap:
      0x00000000 0x00000001

```

Figure 15: Show the number of packets encapsulated and encrypted