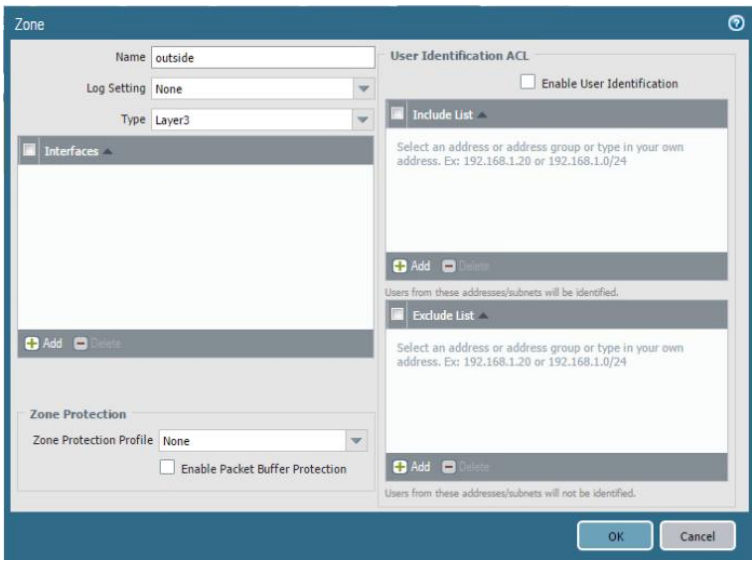
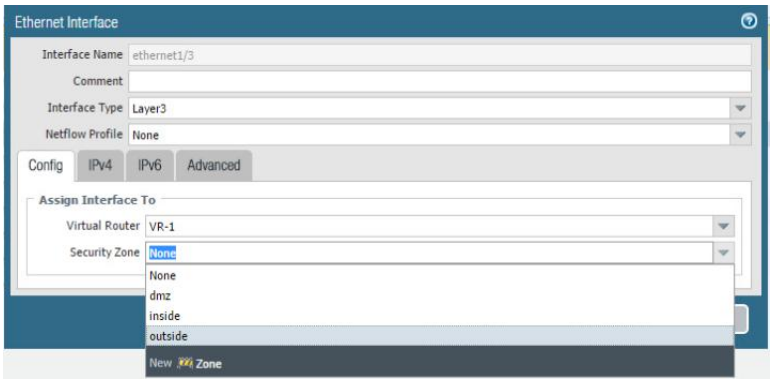
SIDDHANT GAHTORI
ESSENTIAL PROJECT 1

Module 1A (LAB 1): Creating a Zero Trust Environment

Summary: In this module, we created a zero day trust environment by creating zones, applying security policies and after that we tested it. We created 3 zones inside, outside & DMZ for different purposes. We also created NAT Policies for packets. After committing all the changes, we tested it by visiting a webpage and we can see all the traffic in Firewall Logs.



- Creating Zones



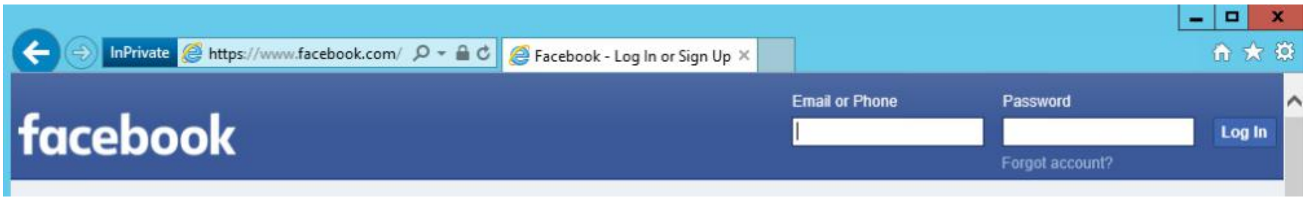
- Configuring Ethernet Interfaces

	Name	Tags	Type	Source Zone	Source Address	Source User	Source HIP Profile	Destination Zone	Destination Address	Application	Service	Action
1	Allow-Inside-Out	none	universal	inside	any	any	any	outside	any	any	application-default	Allow

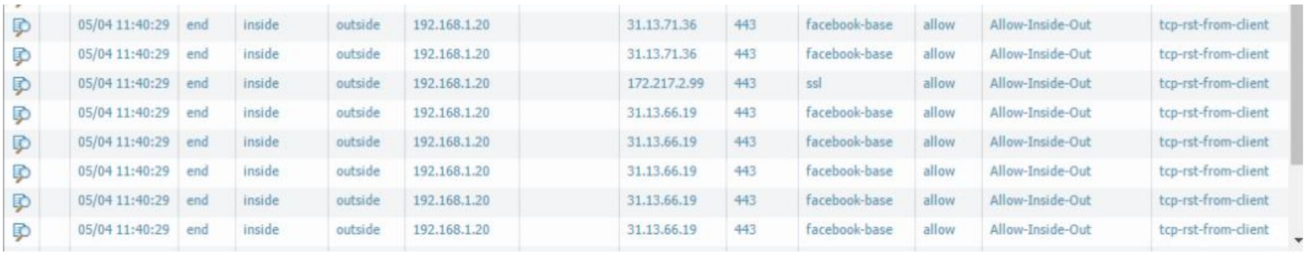
- Creation of Rules

			Original Packet						
	Name	Tags	Source Zone	Destination Zone	Destination Interface	Source Address	Destination Address	Service	Source Translation
1	Inside-NAT-Outside	none	inside	outside	any	any	any	any	dynamic-ip-and-port ethernet1/1 203.0.113.20/24

- NAT Configurations



- Testing Process (Visiting a Webpage)

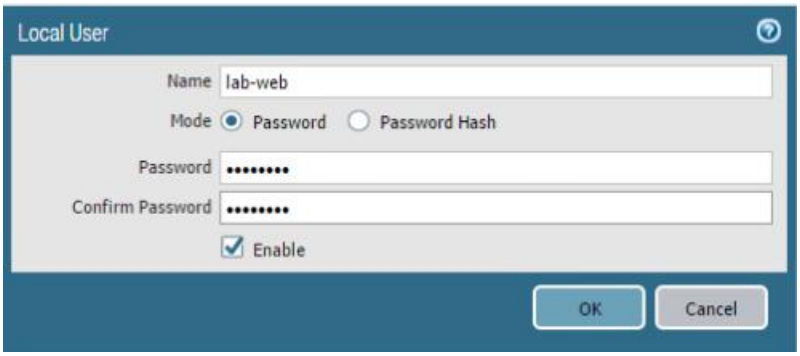


05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.71.36	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.71.36	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	172.217.2.99	443	ssl	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.66.19	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.66.19	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.66.19	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.66.19	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client
05/04 11:40:29	end	inside	outside	192.168.1.20	31.13.66.19	443	facebook-base	allow	Allow-Inside-Out	tcp-rst-from-client

- We can see detailed information about traffic in Logs.

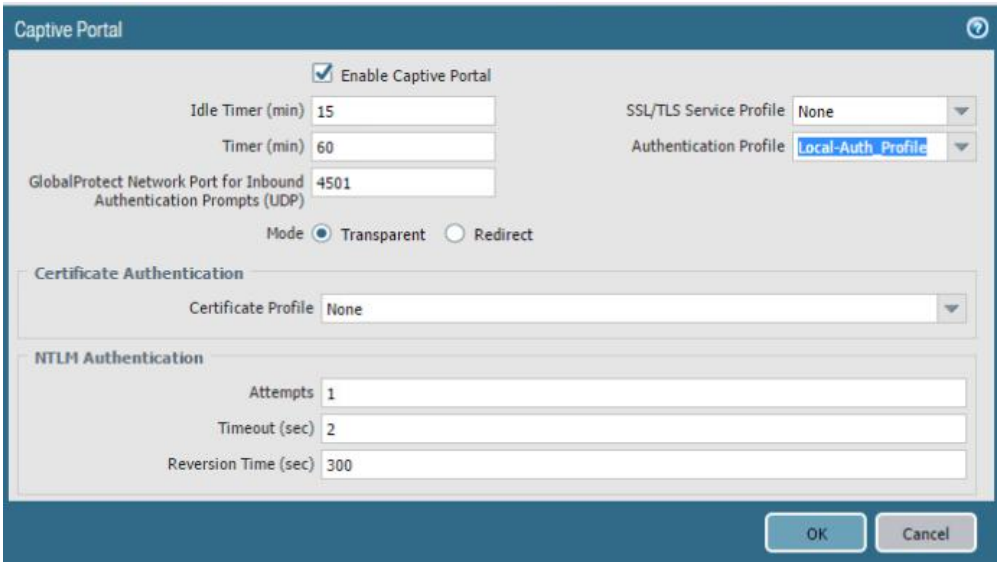
Module 1B (LAB 2): Configuring Authentication

Summary: In this module, we implemented a captive portal gateway for accessing web services in Palo Alto Firewall. We also created local user authentication for security and monitoring. We also analyzed logs of the user.



- Creating a Local User Account

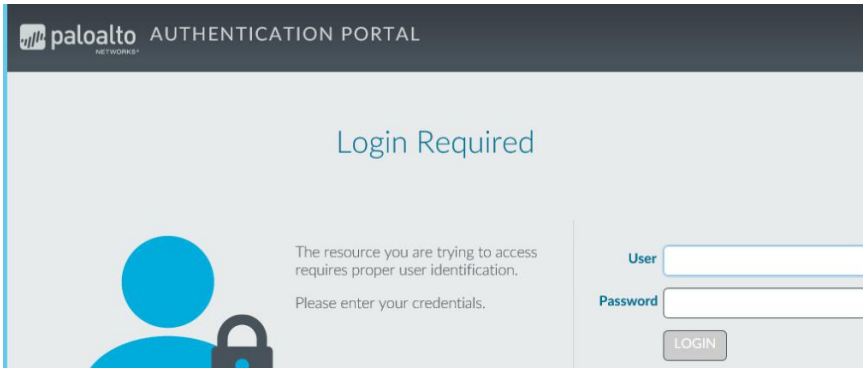
- Creating an authentication profile



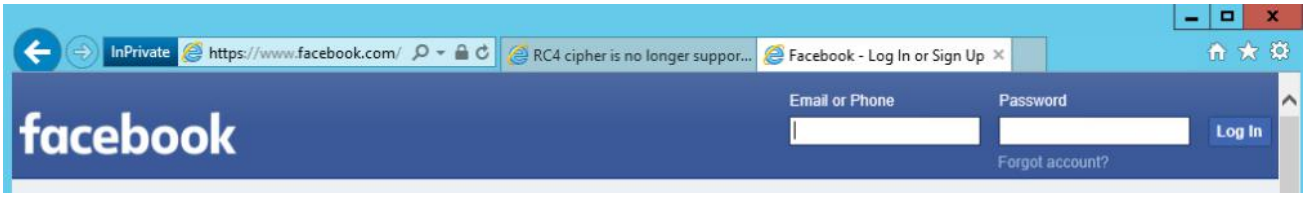
- Captive Portal Configurations

	Name	Tags	Source				Destination		Service	Authentication Enforcement
			Zone	Address	User	HIP Profile	Zone	Address		
1	web-form-policy	none	inside	any	any	any	outside	any	service-http service-https	default-web-form-

- Firewall Rules Configurations



- Our Captive Portal asking for User Information



- After Successful login user can access websites

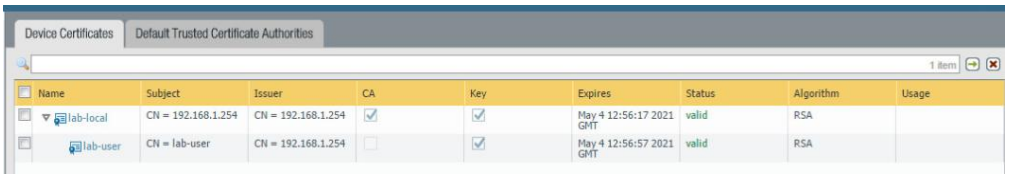
05/04 12:18:48	end	inside	outside	192.168.1.20	lab-web	8.8.8.8	53	dns	allow	Allow-Any	aged-out
05/04 12:18:48	end	inside	outside	192.168.1.20	lab-web	8.8.8.8	53	dns	allow	Allow-Any	aged-out
05/04 12:18:47	end	inside	inside	192.168.1.20	lab-web	192.168.1.255	138	netbios-dg	allow	Allow-Any	aged-out
05/04 12:18:47	end	inside	outside	192.168.1.20	lab-web	8.8.8.8	53	dns	allow	Allow-Any	aged-out
05/04 12:18:43	end	inside	outside	192.168.1.20	lab-web	8.8.8.8	53	dns	allow	Allow-Any	aged-out
05/04 12:18:43	end	inside	outside	192.168.1.20	lab-web	8.8.8.8	53	dns	allow	Allow-Any	aged-out
05/04 12:18:42	end	inside	outside	192.168.1.20	lab-web	8.8.8.8	53	dns	allow	Allow-Any	aged-out

1 2 3 4 5 6 7 8 9 10 | Resolve hostname | Highlight Policy Actions | Displaying logs 1 - 20 | 20 | per page | DESC

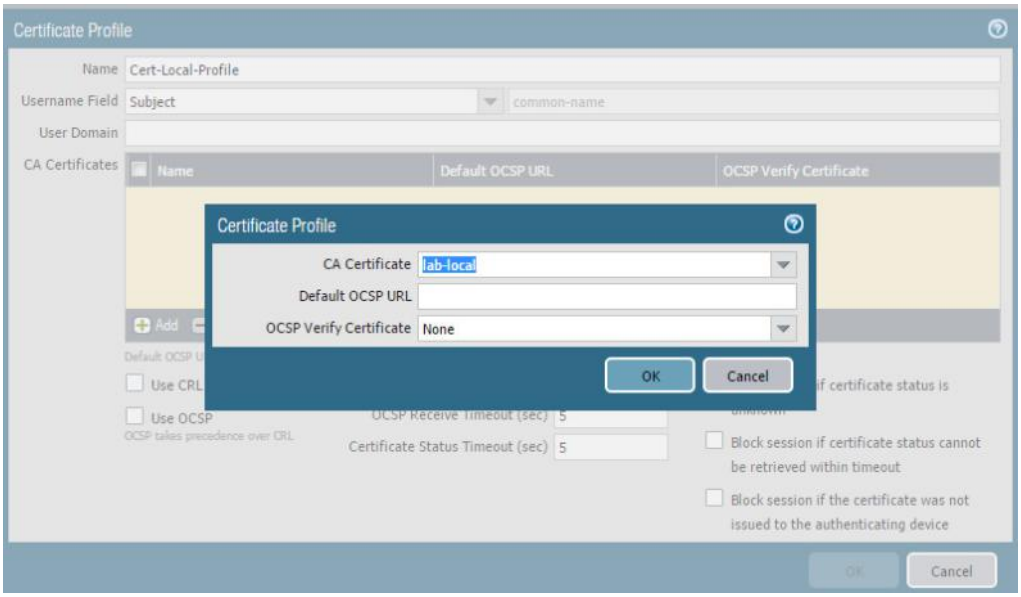
- We can monitor user’ s activity in the logs

Module 2A (LAB 3): Using 2 Factor Authentication to secure the Firewall

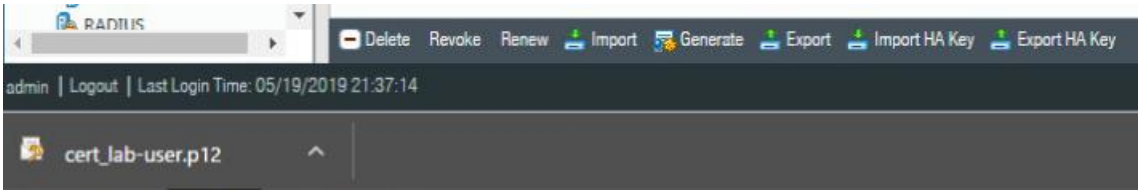
Summary: In this module, we enabled 2 Factor authentication for Palo Alto Firewall using a digital certificate.



- Creating User account for digital certificate.



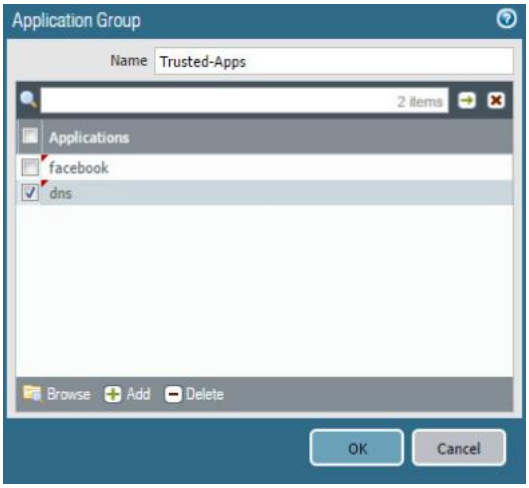
- Creating Certificate Profile



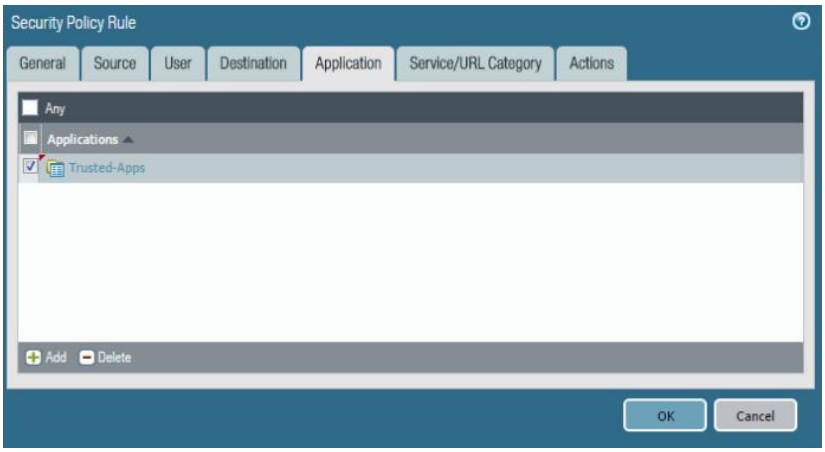
- Downloading Certificate for the User

Module 2B (LAB 4): Allowing only Trusted Applications

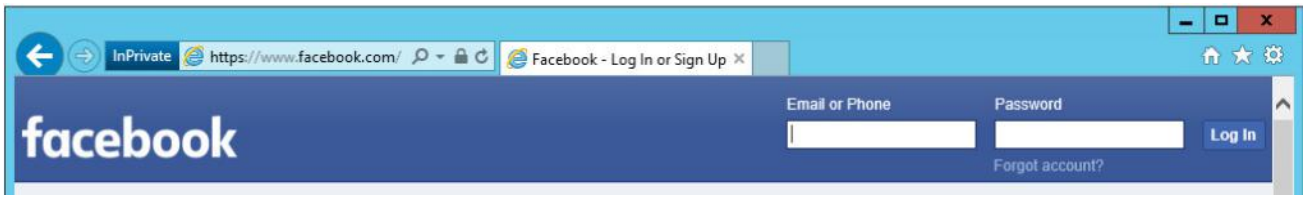
Summary: In this module, we configured our firewall to allow traffic from and to selected apps. First we made an application group and added the websites to it. Then we configured security policy rule for the same. Then we tested it and analyzed the results.



- Application Group Configurations



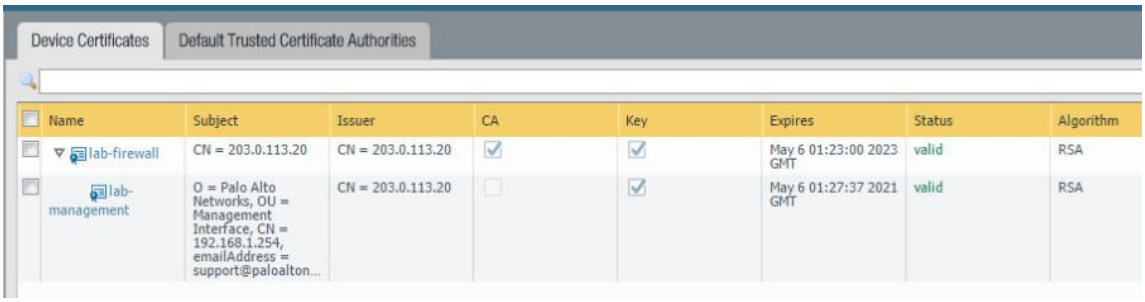
- Security Policy Rule Configurations



- Testing the rules and configurations

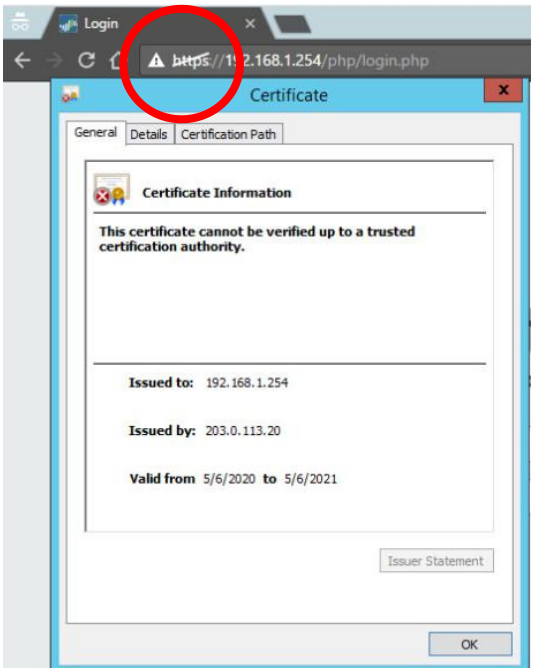
Module 3A (LAB 5): Managing Certificates

Summary: In this module, we generated a digital certificate for inbound management traffic. After that we exported tested and verified it.



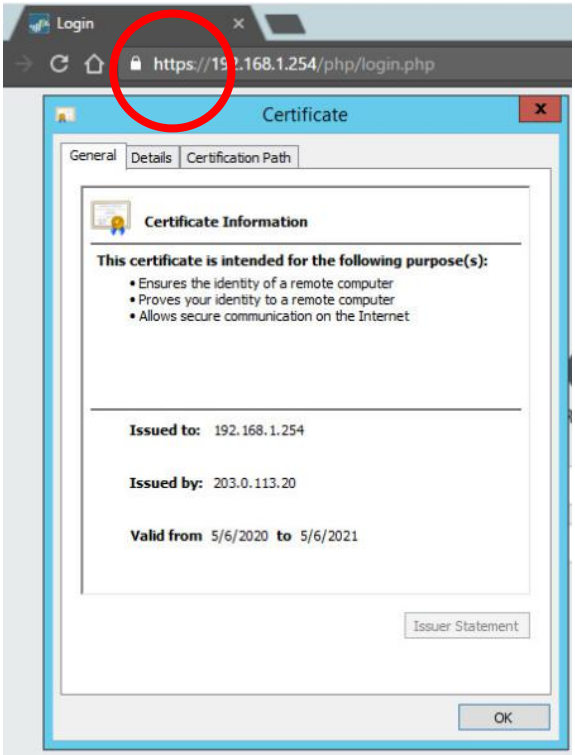
- Certificate Generation

- Exporting Certificate



- Testing it and we can see that it is not verified.

- After downloading it locally, we imported it in the to certificates.msc utility.



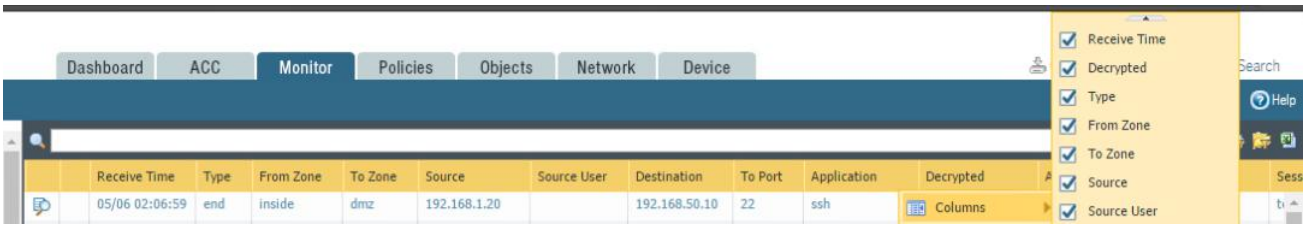
- Now if we try to test it, it is verified now.

Module 3B (LAB 6 & 7): Decrypting SSH Traffic

Summary: In this module, we decrypted some SSH Packets. First, we created a Policy for decryption from inside zone to DMZ. Then we generated SSH Traffic towards DMZ. After that we decrypted it using the same policy.

			Source			Destination				
	Name	Tags	Zone	Address	User	Zone	Address	URL Category	Service	Action
1	Decrypting SSH	none	inside	any	any	dmz	any	any	any	decrypt

- Creation of the Policy



- Decrypting traffic in logs

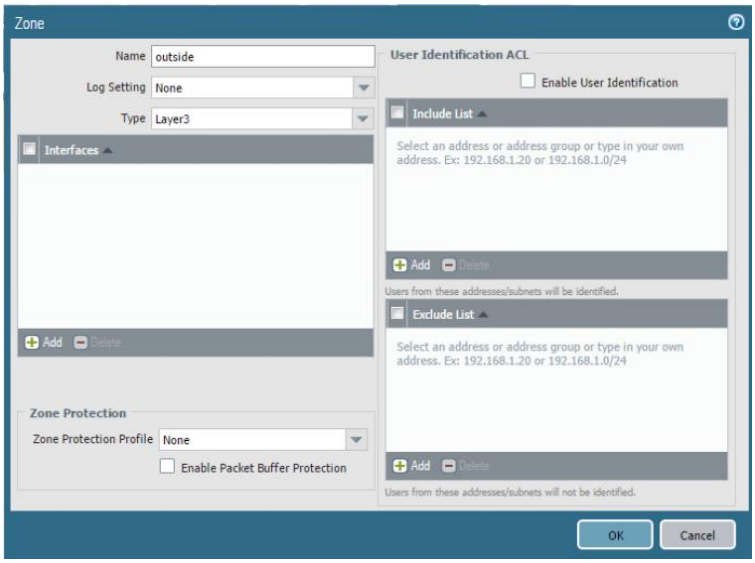
Project Introduction:

In this project, you will configure the firewall for a zero - trust environment.

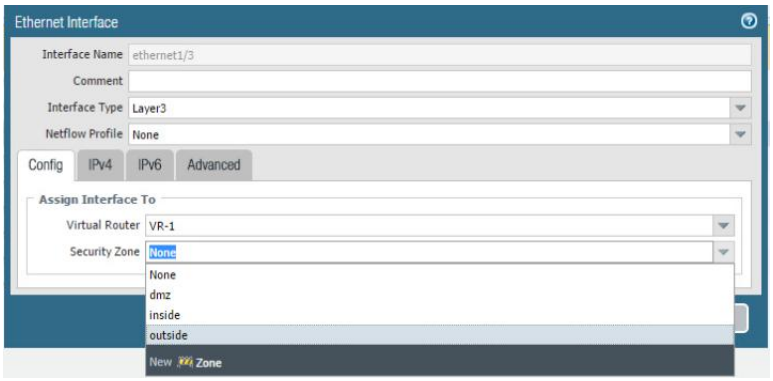
Objectives:

1. Create zones and associate the zones to interfaces
2. Create a Security Policy Rule
3. Create a NAT Policy.

Screenshot:



- Creating Zones



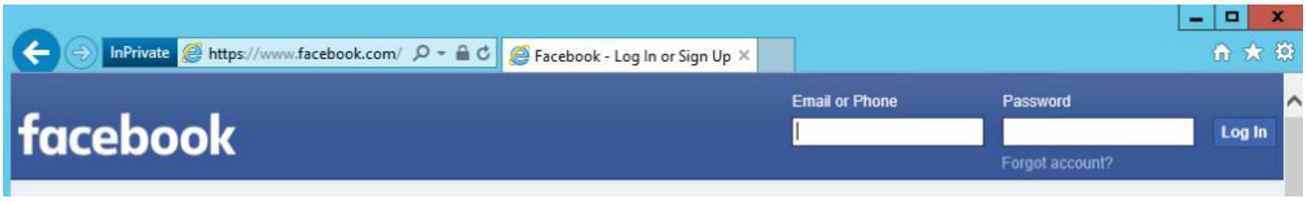
- Configuring Ethernet Interfaces

				Source				Destination				
	Name	Tags	Type	Zone	Address	User	HIP Profile	Zone	Address	Application	Service	Action
1	Allow-Inside-Out	none	universal	inside	any	any	any	outside	any	any	application-default	Allow

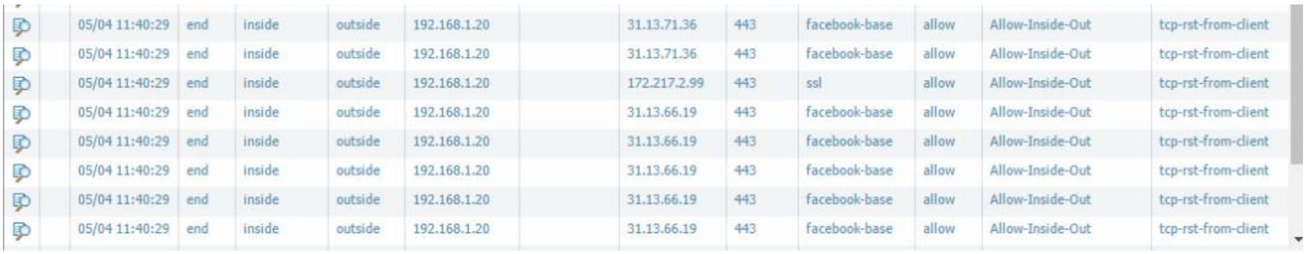
◆ Creation of Rules

			Original Packet						
	Name	Tags	Source Zone	Destination Zone	Destination Interface	Source Address	Destination Address	Service	Source Translation
1	Inside-NAT-Outside	none	inside	outside	any	any	any	any	dynamic-ip-and-port ethernet1/1 203.0.113.20/24

◆ NAT Configurations



◆ Testing (Visiting Facebook)



```
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.71.36  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.71.36  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  172.217.2.99  443  ssl           allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.66.19  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.66.19  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.66.19  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.66.19  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
05/04 11:40:29  end  inside  outside  192.168.1.20  31.13.66.19  443  facebook-base  allow  Allow-Inside-Out  tcp-rst-from-client
```

◆ We can see detailed information about traffic in Logs.

-- THE END --