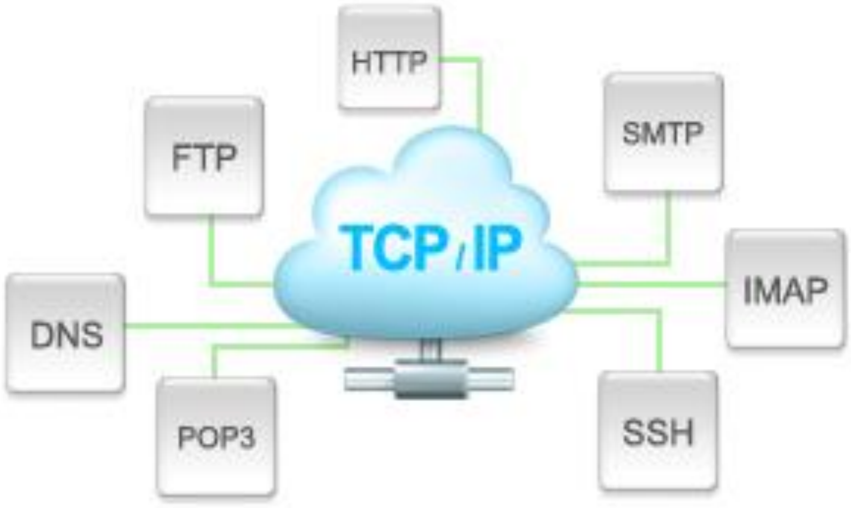
**Overview:**

Using the Amazon Web Services (AWS) Console, open the Virtual Private Cloud (VPC) Console. View the Network Access Control Lists (ACLs). The ACL's can be edited to allow specific protocols. FTP(20, 21), SSH(22), SMTP(25), TFTP(69), DCE(135), NetBios(137, 138, 139), DNS(53 inbound), RDP(3389) and VNC(5500) is a long list of protocols that an organization is generally worried will be exploited by attackers. The challenge with these protocols is that AWS focuses more including and excluding ports than it focuses on protocols.



**Discovery:**

In the Virtual Private Cloud (VPC) Console, make sure to view the security groups and determine if the instances are correctly being restricted. The rules can be added to each security group to distinguish the differences for each instance and allow traffic to and from its associated instances. These rules can be modified at anytime.

Connection Tracking

- Security groups use connection tracking to track all the information to and from the instances. The rules are applied depending on the state of the connection of the traffic. They may allow and deny the traffic from entering and leaving the instance at any point of time.

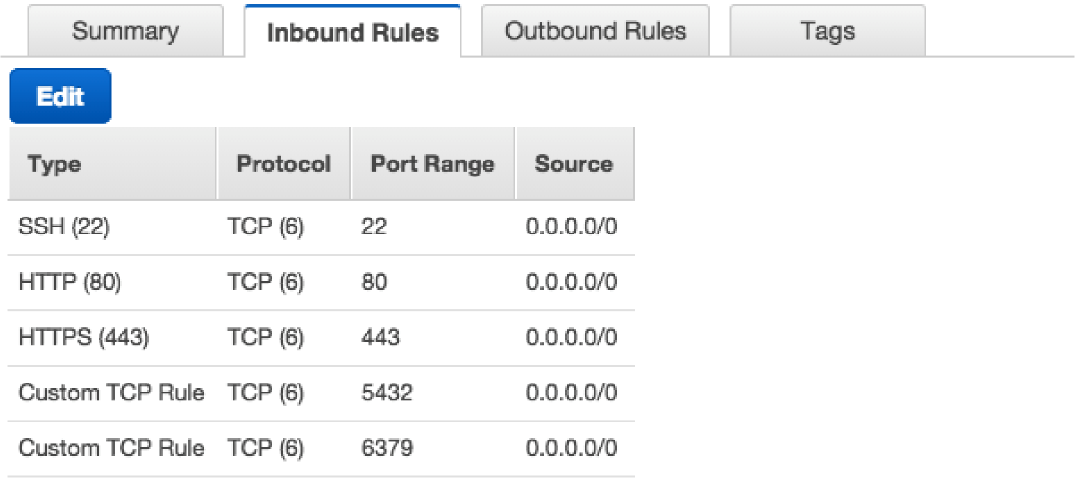
For more information about VPC:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html

For more information about Connection Tracking:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html#security-group-connection-tracking

**Remediation:**

To remediate attacks through a specific protocol, go to the VPC Console in Amazon Web Services Console. Visit all of the security groups and assign them strict rules. The security groups can be used to secure ports of TCP, UDP and ICMP protocols. Make sure to open only the ports that are necessary for the specific service to operate. Having too many ports open allows attackers various ways of gaining access to a system.

Below is an image of the protocols used for the inbound rules

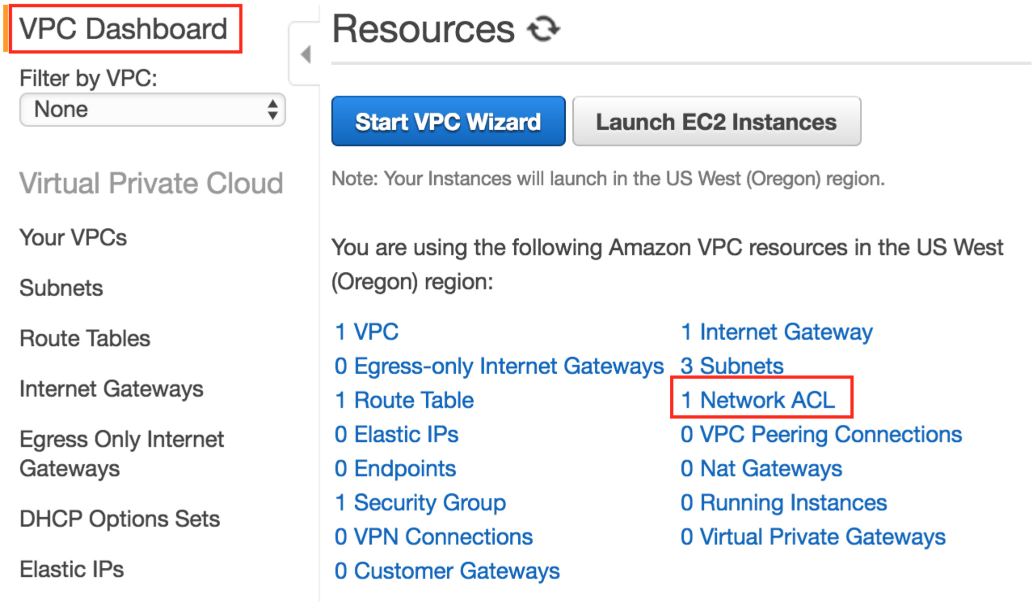


| Type | Protocol | Port Range | Source |
|---|---|---|---|
| SSH (22) | TCP (6) | 22 | 0.0.0.0/0 |
| HTTP (80) | TCP (6) | 80 | 0.0.0.0/0 |
| HTTPS (443) | TCP (6) | 443 | 0.0.0.0/0 |
| Custom TCP Rule | TCP (6) | 5432 | 0.0.0.0/0 |
| Custom TCP Rule | TCP (6) | 6379 | 0.0.0.0/0 |

**Prevention:**

Amazon S3 Access Control Lists (ACLs) enables the user to manage access to the buckets and objects. Each bucket and object has an ACL attached to its subresources. When a request is received against a resource, Amazon S3 checks the corresponding ACL to verify the requester has the necessary access permissions. ACLs limit the amount of people with capabilities to edit security groups and stand up instances on a network. The security groups will restrict permissions and capabilities of each of the instances. As mentioned earlier, having too many ports open allows attackers various ways of attaining access to a system. It is easier to keep track of who accesses a fewer number of ports versus all of the available ports.

Go to the Network ACL Dashboard through the VPC Dashboard in AWS

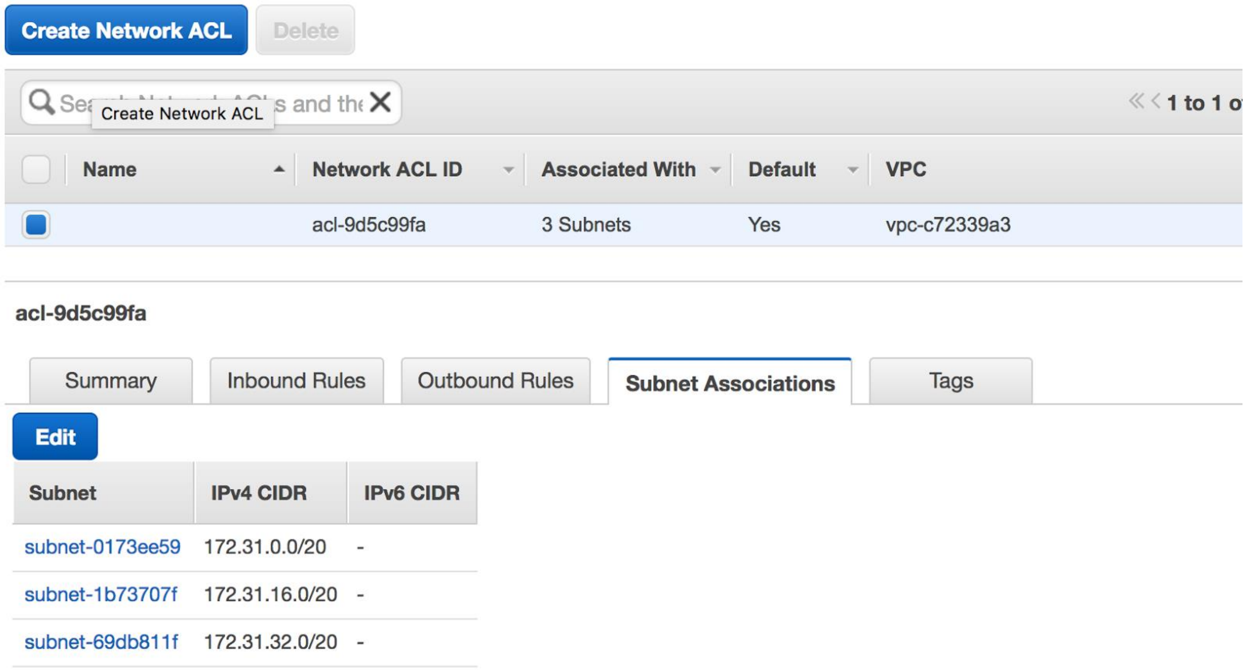


Note: Your Instances will launch in the US West (Oregon) region.

You are using the following Amazon VPC resources in the US West (Oregon) region:

| | |
|---|---|
| 1 VPC | 1 Internet Gateway |
| 0 Egress-only Internet Gateways | 3 Subnets |
| 1 Route Table | 1 Network ACL |
| 0 Elastic IPs | 0 VPC Peering Connections |
| 0 Endpoints | 0 Nat Gateways |
| 1 Security Group | 0 Running Instances |
| 0 VPN Connections | 0 Virtual Private Gateways |
| 0 Customer Gateways | |

Edit the inbound and outbound rules for each Network ACL



| | Name | Network ACL ID | Associated With | Default | VPC |
|---|---|---|---|---|---|
| ■ | | acl-9d5c99fa | 3 Subnets | Yes | vpc-c72339a3 |

acl-9d5c99fa

| Subnet | IPv4 CIDR | IPv6 CIDR |
|---|---|---|
| subnet-0173ee59 | 172.31.0.0/20 | - |
| subnet-1b73707f | 172.31.16.0/20 | - |
| subnet-69db811f | 172.31.32.0/20 | - |