AWS Network Firewall for Ingress/Egress Traffic

**SPL-TF-300-NWNFIE-1 - Version 1.0.5**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

As a network engineer, your role encompasses monitoring and troubleshooting of the network. Understanding the flow of traffic through the network, being able to identify where traffic is coming from and ensuring that unauthorized access attempts are unsuccessful. This lab demonstrates the key concepts of AWS Network Firewall. This lab will demonstrate the use of AWS Network Firewall to filter outbound web traffic using resources that are provisioned as part of this lab. You will learn to use Amazon Route 53 DNS Firewall to block certain domains and apply it to the VPCs. Next you will use Network Firewall Engine to inspect traffic and to configure stateless and stateful rule group to mitigate threat against AWS workload. You will set up CloudWatch Log Groups to monitor your network and detect anomalies. Based on the finding displayed in CloudWatch, you investigate and mitigate the suspected threats.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Configure the AWS Network Firewall using the Network Firewall rule groups to filter outbound web traffic.
* Create DNS Firewall Domain List, rule groups and rules using your custom domain lists.
* Use Route 53 DNS Firewall to filter and secure DNS traffic.
* Use CloudWatch Log Groups to filter queries for auditing and identifying potential threats.
* Configure stateful rule groups in AWS Network Firewall that follow Suricata-compatible intrusion prevention system (IPS) rule specifications.
* Apply stateful and stateless firewall rules to detect suspicious network traffic.
* Configure Firewall Policy and monitoring for AWS Network Firewall to hunt for suspicious network activity.
* Stop malicious activities identified through monitoring and investigation of security alerts.

TECHNICAL KNOWLEDGE PREREQUISITES

Familiarity with routing and DNS are recommended. You should also be comfortable working with the Command Line Interface (CLI) in a Linux environment.

PREREQUISITES

This lab requires:

* Access to a computer with Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat).
* A modern internet browser such as Chrome or Firefox.

DURATION

This lab requires approximately 90 minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to certain aspects of the guide. The following list explains the purpose for each one:

* **Command:** A command that you must run.
* **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.
* **Additional information:** Information of special interest or importance.
* **Note:** The note icon specifies important hints, tips, guidance, or advice.
* **Caution:** Calls attention to information of special interest or importance. Failure to read the note does not result in breaking the service or losing any data, but could result in the need to repeat certain steps.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Conclusion:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

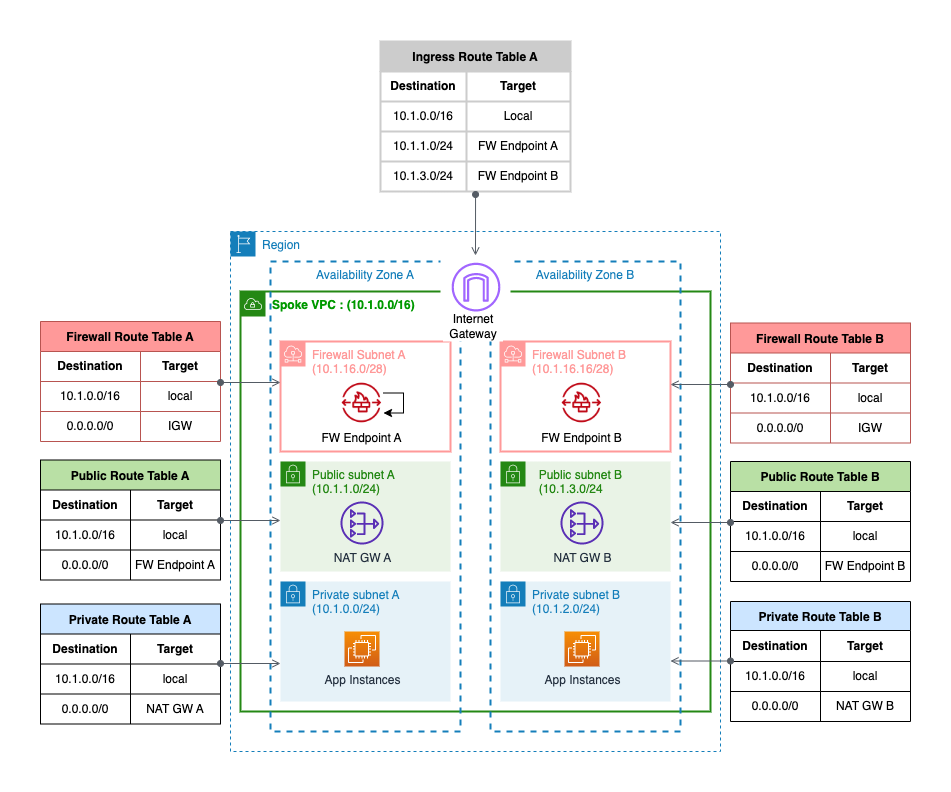
**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

LAB ENVIRONMENT

This lab deploys two VPCs one being the SpokeVPC and the other is MaliciousVPC. As shown in the below architecture diagram, SpokeVPC has two private instances with NAT Gateway deployed in dedicated public subnet allowing the private instances to communicate with resources on the internet. AWS Network Firewall is deployed into the Spoke VPC which requires protection. The two VPCs do not require connectivity to each other. AWS Network Firewall is deployed in a dedicated Firewall subnet which has access to Internet Gateway (IGW). Returning traffic from Internet Gateway (IGW) is returned back to firewall Elastic Network Interface (ENI) using Ingress Routing table attached to the IGW. This ensures the traffic is symmetric for full inspection. This lab is pre-populated with a MaliciousInstance that is deployed in MaliciousVPC, performing a variety of activities that will be detected by you as you configure security controls and monitoring. The MaliciousVPC and its resources are not shown in the below architecture diagram.



*In the preceding diagram, two VPCs are deployed which are SpokeVPC & MaliciousVPC where SpokeVPC has two private instances in separate Availability Zones with NAT Gateway deployed in dedicated public subnet allowing the private instances to communicate with resources on the internet. AWS Network Firewall is deployed into the Spoke VPC which requires protection.*

SERVICES USED IN THIS LAB

**Amazon VPC (Virtual Private Cloud)**

Amazon VPC is a service that enables users to create and manage isolated virtual networks within the AWS infrastructure. With Amazon VPC, customers can configure their own IP address range, create subnets, and set up routing tables and network gateways. This allows for greater control over network security, traffic routing, and resource access within an organization’s cloud environment.

**Amazon EC2 (Elastic Compute Cloud)**

Amazon EC2 is a scalable compute service that allows users to run virtual servers in the AWS cloud. With EC2, customers can quickly provision and configure instances to meet their specific needs, including choosing the operating system, storage, and network settings. The service offers a variety of instance types and pricing options, making it suitable for a wide range of applications, from small-scale web hosting to large-scale data processing and analytics.

**Amazon CloudWatch**

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, and set alarms. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

**AWS Network Firewall**

AWS Network Firewall is a highly available, managed network firewall service for your Amazon Virtual Private Cloud (Amazon VPC). It enables you to easily deploy and manage stateful inspection, intrusion prevention and detection, and web filtering to protect your virtual networks on AWS. AWS Network Firewall automatically scales with your traffic, ensuring high availability with no additional customer investment in security infrastructure. While AWS Network Firewall secures the network traffic to your applications, Route 53 DNS firewall complements it by performing filtering and securing DNS query traffic to the Route 53 resolver.

There are 3 key components of AWS Network Firewall.

* Rule Groups: Holds a reusable collection of criteria for inspecting traffic and for handling packets and traffic flows that match the inspection criteria.
* Policy: Defines a reusable set of stateless and stateful rule groups, along with some policy-level behavior settings.
* Firewall: Enforces the inspection rules in the firewall policy to the VPC that the rules protect. Each firewall requires one firewall policy. The firewall additionally defines settings like how to log information about your network traffic and the firewall’s stateful traffic filtering.

**Amazon Route 53 Resolver DNS Firewall (DNS Firewall)**

Amazon Route 53 Resolver DNS Firewall (DNS Firewall) helps you block DNS queries that are made for known malicious domains, while allowing DNS queries to trusted domains. A primary use of DNS Firewall protections is to help prevent DNS exfiltration of your data. DNS exfiltration can happen when a bad actor compromises an application instance in your VPC and then uses DNS lookup to send data out of the VPC to a domain that they control. DNS Firewall has a simple deployment model that makes it straightforward for you to start protecting your VPCs by using managed domain lists, as well as custom domain lists. With DNS Firewall, you can filter and regulate outbound DNS requests. The service inspects DNS requests that are handled by Route 53 Resolver and applies actions that you define to allow or block requests.

There are 3 components of Route 53 Resolver DNS Firewall

* Domain Lists: Defines a named, reusable collection of domain specifications for use in DNS filtering. Each rule in a rule group requires a single domain list.
* Rule Groups: Defines a named, reusable collection of DNS Firewall rules for filtering DNS queries. You populate the rule group with the filtering rules, then associate the rule group with one or more VPCs.
* Rule: Defines a filtering rule for DNS queries in a DNS Firewall rule group. Each rule specifies one domain list and an action to take on DNS queries whose domains match the domain specifications in the list. You can allow, block, or alert on matching queries. You can also define custom responses for blocked queries.

**How DNS Firewall works with AWS Network Firewall**

DNS Firewall and AWS Network Firewall both offer domain name filtering, but for different types of traffic. With DNS Firewall and AWS Network Firewall together, you can configure domain-based filtering for application layer traffic over two different network paths.

DNS Firewall provides filtering for outbound DNS queries that pass through the Route 53 Resolver from applications within your VPCs. You can also configure DNS Firewall to send custom responses for queries to blocked domain names.

AWS Network Firewall provides filtering for both network and application layer traffic, but does not have visibility into queries made to the Route 53 Resolver.

**Task 1: Verify firewall resources**

Since an AWS Network Firewall has already been provisioned as part of this lab, your first task is to verify firewall resources.

TASK 1.1 - LIST FIREWALLS

Verify the Firewall policy and rule groups provisioned as part of this lab.

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **Network Firewall**, choose **Firewalls**.

Located the pre-configured Firewall:

* **SpokeVPC-InspectionFirewall**

TASK 1.2 - FIREWALL DETAILS

1. Choose **SpokeVPC-InspectionFirewall** Firewall to see more details.
2. At this step, review the following information:

i. What is the current status of the Firewall?

ii. What policy has been associated with the Firewall?

iii. What is the default Stateless action?

iv. What Stateful rules are configured in the policy mentioned above?

1. Choose the **Firewall details** tab and this tab provides the following details:

* Details of Firewall endpoints & status.
* Logging configuration displaying Flow/Alert type and CloudWatch Log Group configurations.
* Tags.

TASK 1.3 - FIREWALL MONITORING DETAILS

1. Choose the **Monitoring** tab and this tab provides the following details on Firewall metrics for example:

* How many packets are received/passed/dropped by Firewall Stateless Inspection?
* How many packets are received/passed/dropped by Firewall Stateful Inspection?

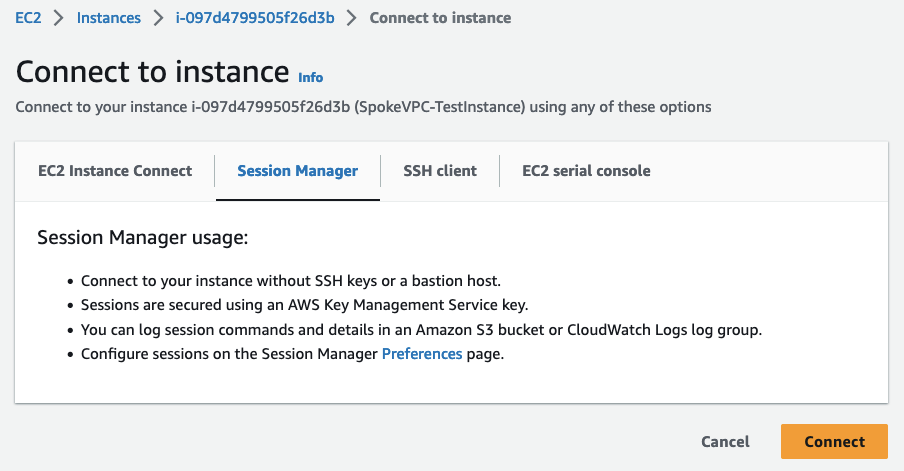
TASK 1.4 - LIST AND VERIFY EC2 INSTANCE

1. At the top of the page, in the unified search bar, search for and choose

EC2

.

1. In the left navigation pane, under **Instances**, choose **Instances**.
2. Select the instance **SpokeVPC-TestInstance1**.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .



 Terminal session should open in a new browser tab. You are now connected to EC2 instance. You can terminate the session (browser tab) by choosing **Terminate** on the top right corner.

**Task complete:** You have successfully verified all the VPC and firewall resources.

**Task 2: Egress web filtering**

In this task, you use AWS Network Firewall to filter outbound web traffic using resources previously provisioned as part of this lab.

TASK 2.1 - LIST FIREWALLS & RULE GROUPS

In this task, verify the rule group **NetworkFirewall-DomainAllow-RuleGroup**.

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
2. Choose **NetworkFirewall-DomainAllow-RuleGroup** to see more details.

 You can observe that the Stateful rule group **NetworkFirewall-DomainAllow-RuleGroup** is allowing traffic matching domain ***.amazon.com*** and ***.amazonaws.com***. You can add further domains here for testing purposes or leave it as default.

TASK 2.2 - LIST & VERIFY EC2 INSTANCES

As part of this lab setup, an EC2 instance named **SpokeVPC-TestInstance1** and **SpokeVPC-TestInstance2** are pre-created in a Private Subnets of the **SpokeVPC**. In this task, you verify EC2 instance details.

1. At the top of the page, in the unified search bar, search for and choose

EC2

.

1. In the left navigation pane, under **Instances**, choose **Instances**.
2. Select the instance **SpokeVPC-TestInstance1**.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

Now you are ready to test the firewall policy.

TASK 2.3 - VERIFY DOMAIN FILTERING

1. On the **SpokeVPC-TestInstance1** instance, run the below command:

curl https://aws.amazon.com --max-time 5

**Sample output:** You observe the output for the curl command. Notice that the curl command is completed successfully.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ curl https://aws.amazon.com --max-time 5

<!doctype html>

<html class="no-js aws-lng-en\_US aws-with-target" lang="en-US" data-static-assets="https://a0.awsstatic.com" data-js-version="1.0.545" data-css-version="1.0.485">

<head>

<meta property="og:type" content="company">

<meta property="og:url" content="https://aws.amazon.com/">

<meta property="og:image" content="https://a0.awsstatic.com/libra-css/images/logos/aws\_logo\_smile\_1200x630.png">

<meta property="og:site\_name" content="Amazon Web Services, Inc.">

<meta name="google-site-verification" content="XHghG81ulgiW-3EylGcF48sG28tBW5EH0bNUhgo\_DrU">

<meta name="msvalidate.01" content="6F92E52A288E266E30C2797ECB5FCCF3">

<link rel="stylesheet" href="https://a0.awsstatic.com/libra-css/css/1.0.485/style-awsm.css">

<script type="application/json" id="aws-page-settings">

Now try testing this command again with another domain which is not permitted by default.

1. On the **SpokeVPC-TestInstance1** instance, run the below command:

curl -vvv https://google.com -o /dev/null --max-time 5

**Sample output:** Notice that the curl command times out eventually without any data returning.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ curl -vvv https://google.com -o /dev/null --max-time 5

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0\* Trying 142.251.33.110:443...

\* Connected to google.com (142.251.33.110) port 443 (#0)

\* ALPN: offers h2,http/1.1

\* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

} [5 bytes data]

\* TLSv1.2 (OUT), TLS handshake, Client hello (1):

} [512 bytes data]

\* CAfile: /etc/pki/tls/certs/ca-bundle.crt

\* CApath: none

0 0 0 0 0 0 0 0 --:--:-- 0:00:04 --:--:-- 0\* Connection timed out after 5000 milliseconds

0 0 0 0 0 0 0 0 --:--:-- 0:00:05 --:--:-- 0

\* Closing connection 0

curl: (28) Connection timed out after 5000 milliseconds

TASK 2.4 - UPDATE DOMAIN RULE GROUP

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
2. Select the **NetworkFirewall-DomainAllow-RuleGroup** rule group.
3. Scroll down to **Domains (2)** section and choose **Edit** .
4. In the **Edit domains for NetworkFirewall-DomainAllow-RuleGroup** pop-up window, under Domain names, add

.google.com

.

1. Choose **Save** .

TASK 2.5 - VERIFY UPDATED DOMAIN SUCCESS

Now you have 3 domains that are allowed in this Rule Group.

1. Run the below command again on the **SpokeVPC-TestInstance1** instance:

curl -vvv https://google.com -o /dev/null --max-time 5

**Sample output:** Notice that the curl command completes successfully now that you have permitted the domain in Rule Group.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ curl -vvv https://google.com -o /dev/null --max-time 5

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0\* Trying 142.251.33.78:443...

\* Connected to google.com (142.251.33.78) port 443 (#0)

\* ALPN: offers h2,http/1.1

\* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

} [5 bytes data]

\* TLSv1.2 (OUT), TLS handshake, Client hello (1):

} [512 bytes data]

\* CAfile: /etc/pki/tls/certs/ca-bundle.crt

\* CApath: none

{ [5 bytes data]

\* TLSv1.2 (IN), TLS handshake, Server hello (2):

{ [96 bytes data]

\* TLSv1.2 (IN), TLS handshake, Certificate (11):

{ [6482 bytes data]

\* TLSv1.2 (IN), TLS handshake, Server key exchange (12):

{ [149 bytes data]

\* TLSv1.2 (IN), TLS handshake, Server finished (14):

{ [4 bytes data]

\* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):

} [70 bytes data]

\* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):

} [1 bytes data]

\* TLSv1.2 (OUT), TLS handshake, Finished (20):

} [16 bytes data]

\* TLSv1.2 (IN), TLS change cipher, Change cipher spec (1):

{ [1 bytes data]

\* TLSv1.2 (IN), TLS handshake, Finished (20):

{ [16 bytes data]

\* SSL connection using TLSv1.2 / ECDHE-ECDSA-AES128-GCM-SHA256

\* ALPN: server accepted h2

\* Server certificate:

\* subject: CN=\*.google.com

\* start date: May 22 08:17:22 2023 GMT

\* expire date: Aug 14 08:17:21 2023 GMT

\* subjectAltName: host "google.com" matched cert's "google.com"

\* issuer: C=US; O=Google Trust Services LLC; CN=GTS CA 1C3

\* SSL certificate verify ok.

} [5 bytes data]

\* using HTTP/2

\* h2h3 [:method: GET]

\* h2h3 [:path: /]

\* h2h3 [:scheme: https]

\* h2h3 [:authority: google.com]

\* h2h3 [user-agent: curl/8.0.1]

\* h2h3 [accept: \*/\*]

\* Using Stream ID: 1 (easy handle 0x864180)

} [5 bytes data]

> GET / HTTP/2

> Host: google.com

> user-agent: curl/8.0.1

> accept: \*/\*

>

{ [5 bytes data]

< HTTP/2 301

< location: https://www.google.com/

< content-type: text/html; charset=UTF-8

< content-security-policy-report-only: object-src 'none';base-uri 'self';script-src 'nonce-MNvpA-8Q\_2WZmYfQzKH8cA' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp

< date: Fri, 16 Jun 2023 19:05:38 GMT

< expires: Sun, 16 Jul 2023 19:05:38 GMT

< cache-control: public, max-age=2592000

< server: gws

< content-length: 220

< x-xss-protection: 0

< x-frame-options: SAMEORIGIN

< alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

<

{ [5 bytes data]

100 220 100 220 0 0 3470 0 --:--:-- --:--:-- --:--:-- 3492

\* Connection #0 to host google.com left intact

TASK 2.6 - VERIFY ALERT LOGS CAPTURED IN CLOUDWATCH

 It can take 5 to 10 minutes for logs to show up after the event.

1. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

1. In the left navigation pane, under **Logs**, choose **Log groups**.
2. Select **/NetworkFirewall/Alert** Log group.
3. Select the latest Log streams **/aws/network-firewall/alert/SpokeVPC-InspectionFirewall\_\*** .

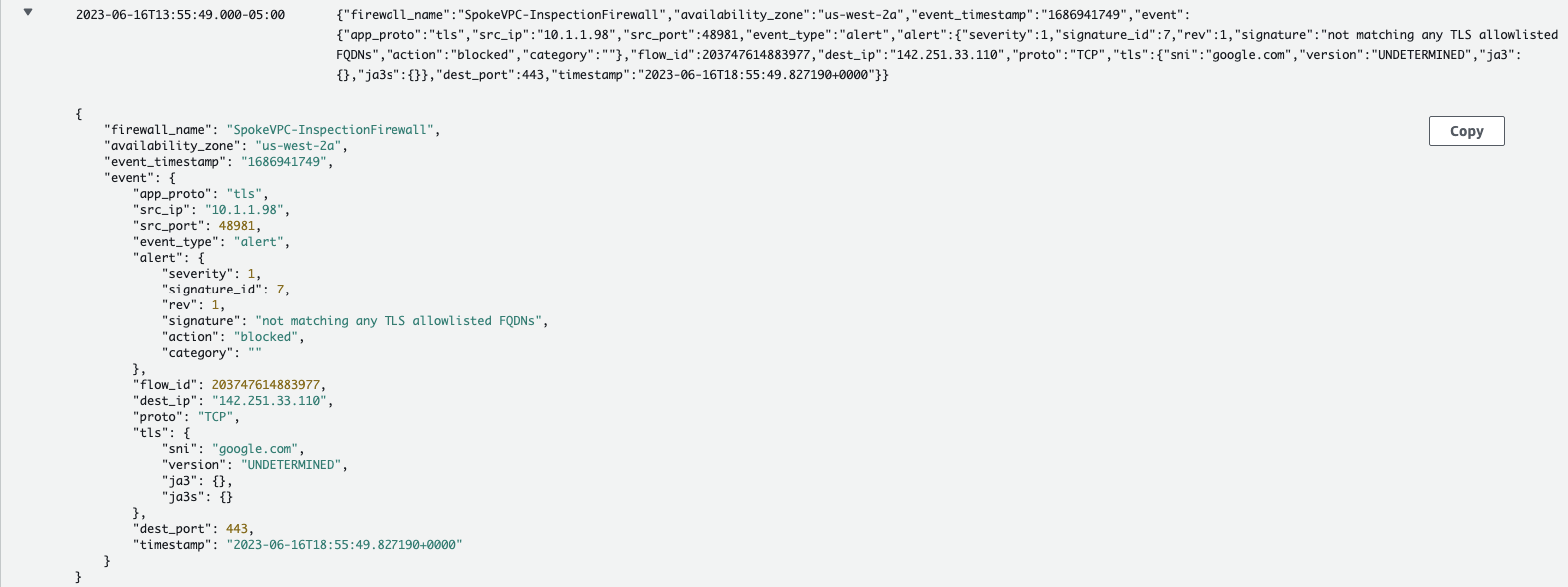
 You observe alerts being captured with all details under the log group.

1. In the Filter events dialog box type

google.com

 and look for the Log event where google.com is getting blocked.

**Sample output:** A sample screen shot of the log event is show below:



**Optional - Verify ICMP alerts**

As you may have noticed under the Network Firewall rule groups, the lab has also provisioned another Rule Group **NetworkFirewall-IcmpAlert-RuleGroup**. This allows all ICMP traffic to pass through but logs the traffic in logs.

1. To test this Rule Group, on your EC2 instance, run the below command:

ping 1.1.1.1 -c 5

**Sample output:** A sample screen shot for the ping command is shown below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ ping 1.1.1.1 -c 5

PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.

64 bytes from 1.1.1.1: icmp\_seq=1 ttl=31 time=10.3 ms

64 bytes from 1.1.1.1: icmp\_seq=2 ttl=31 time=9.15 ms

64 bytes from 1.1.1.1: icmp\_seq=3 ttl=31 time=8.98 ms

64 bytes from 1.1.1.1: icmp\_seq=4 ttl=31 time=9.01 ms

64 bytes from 1.1.1.1: icmp\_seq=5 ttl=31 time=18.8 ms

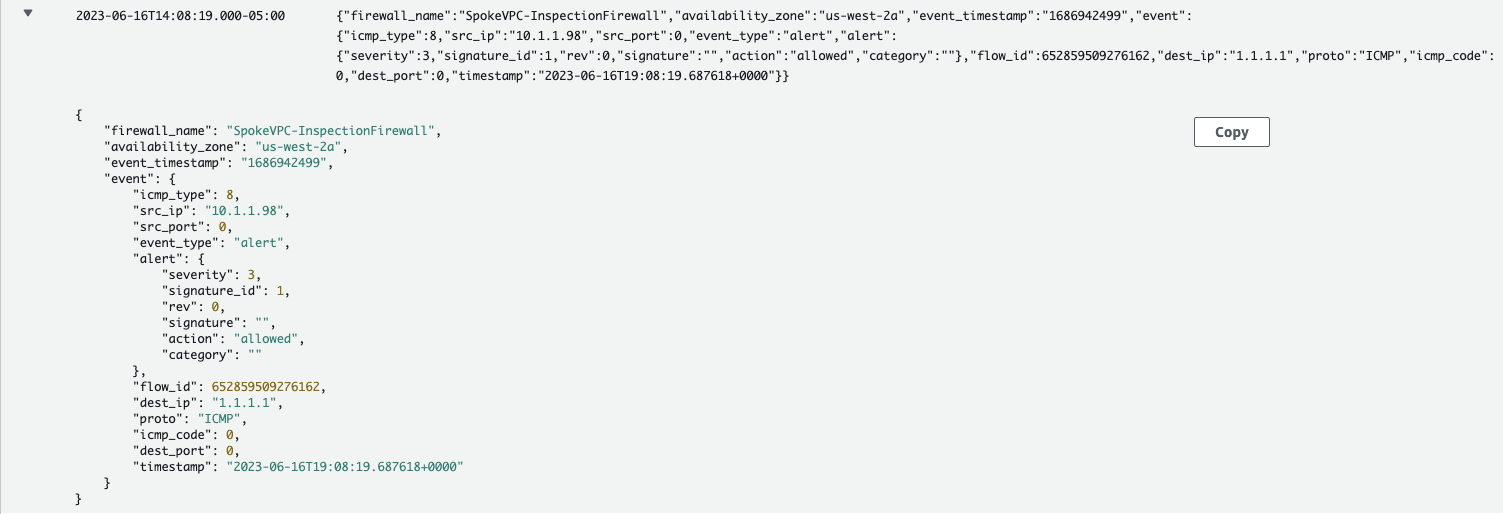
--- 1.1.1.1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4005ms

rtt min/avg/max/mdev = 8.985/11.276/18.838/3.816 ms

1. Use CloudWatch to view the logs. You can use the steps provided earlier in this task on how to navigate to Cloudwatch and view the logs.

**Sample output:** A sample screen shot of the log event is show below:



**Task complete:** You have successfully used AWS Network Firewall to filter and verify outbound web traffic in this task.

**Task 3: Egress DNS query filtering**

In previous task, you used AWS Network Firewall to allow access to certain domains. AWS Network firewall performs the inspection of the network traffic of application and performs the actions of allowing, rejecting or alerting on the traffic. This is done by inspecting the protocol headers of HTTP, TCP/UDP or IP. However it does not inspect DNS queries to the domains which are handled by the Route 53 resolver. In order to provide more comprehensive security, you would also want to prevent DNS resolution of certain domains. This can be done using Route 53 DNS Firewall.

In this task, you use Amazon Route 53 DNS Firewall to block certain domains and apply it to the VPCs previously provisioned as part of the lab setup.

TASK 3.1 - CREATE A DNS FIREWALL DOMAIN LIST

In this task you create the domain list. This domain list will specify the domains for which you want to block DNS queries for.

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **DNS firewall**, choose **Domain lists**.

The **Domain lists** page shows you:

* **Owned domain lists:** AWS provides an ability to create your own domain list.
* **AWS managed domain lists:** Managed domain lists contain domain names that are associated with malicious activity or other potential threats. AWS maintains these lists to enable Route 53 Resolver customers to check outbound DNS queries against them for free when using DNS Firewall.

1. Under **Owned domain lists** section, choose **Add domain list** and configure the following:

* **Domain list name:**

NetworkFirewall-blocked-domains-list

 .

* **Enter one domain per line:**

\*.google.com

 and

google.com

 .

1. Choose **Add domain list** .

TASK 3.2 - CREATE A RULE GROUP

1. In the left navigation pane, under **DNS firewall**, choose **Rule groups**.
2. On the splash screen, choose **Create rule group** .
3. On the **Add rule group** page, under **Rule group details**, specify

NetworkFirewall-block-domains-rule-group

 for Name.

1. Choose **Next** .
2. On the **Add rules -** ***optional*** page, choose **Add rule**.
3. On the **Add rules -** ***optional*** page:

* For **Name**, enter

NetworkFirewall-block-domains-rule

 .

* For **Domain list**, choose **Add my own domain list**.
* For **Choose or create a new domain list**, select **NetworkFirewall-blocked-domains-list** from the drop-down menu.
* For **Choose an action to take when a DNS query fits the matches**, select **BLOCK** from the drop-down menu.
* For **Select a response to send for the BLOCK action**, choose **NXDOMAIN**.

1. Choose **Add rule** .
2. Choose **Next** .

 On **Set rule priority -** ***optional*** page, you can set the Rule priority, however since you have only 1 rule, go with the default.

1. Choose **Next** .
2. On **Add tags -** ***optional*** page, choose **Next** .
3. Choose **Create rule group** .

TASK 3.3 - ASSOCIATE THE RULE GROUP WITH YOUR VPC

1. Select the **NetworkFirewall-block-domains-rule-group** and scroll down to the **Associated VPCs** tab.
2. Choose **Associate VPC** .
3. From the drop-down menu, select **vpc-**\*\*\*\*\*\*\* **(SpokeVPC)**.
4. Choose **Associate** .

 This process will take a few minutes and you can see the progress in the Status column. Once the Status is  **Complete** proceed to the next task.

TASK 3.4 - CONFIGURE DNS QUERY LOGGING

To start logging the DNS queries that are filtered by DNS Firewall rules that originate in your VPCs, you need to configure Query Logging in Route 53.

1. At the top of the page, in the unified search bar, search for and choose

Route 53

.

1. In the left navigation pane, under **Resolver**, choose **Query logging**.

 By default Route 53 > Resolver > Query logging opens in ***N. Virginia (us-east-1)*** region. **Make sure you are accessing Query logging from the region where the lab is running.** If appropriate region is not selected from the drop down, depending on the access permission, you might run into an error.

1. **Optional-** In the previous step, if you had to change the region because the lab was not launched in ***N. Virginia (us-east-1)*** region, then you need to choose **Query logging** again from the left navigation pane.
2. Choose **Configure query logging** .
3. On the **Configure query logging** page:

* For **Query logging configuration name**, enter

NetworkFirewall-dnsfw-queries

.

* For **Destination for query logs**, choose **CloudWatch Logs log group**.
* For **CloudWatch Logs log groups**, select **Create log group**.
* For **New log group name**, enter

/aws/route53/NetworkFirewall-dnsfw-queries-loggroup

.

* For **VPCs to log queries for -** *optional*
  + Choose **Add VPC** .
  + Select **SpokeVPC**.
  + Choose **Add** .

1. Choose **Configure query logging** .

TASK 3.5 - VERIFY DOMAIN NAME QUERY RESOLUTION

1. At the top of the page, in the unified search bar, search for and choose

EC2

.

1. In the left navigation pane, under **Instances**, choose **Instances**.
2. Select the instance **SpokeVPC-TestInstance1**.
3. Choose **Connect** from the navigation bar.

 This instance is part of the **SpokeVPC**. In Task 3.3, you associated this VPC with the DNS resolver.

1. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. On the **SpokeVPC-TestInstance1** instance, run the below command:

nslookup www.google.com

**Sample output:** A sample screen shot for the nslookup command on www.google.com is shown below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ nslookup www.google.com

Server: 10.1.0.2

Address: 10.1.0.2#53

\*\* server can't find www.google.com: NXDOMAIN

sh-4.2$

1. On the **SpokeVPC-TestInstance1** instance, run the below curl command:

curl -vvv https://google.com -o /dev/null --max-time 5

**Sample output:** A sample screen shot for the curl command is shown below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ curl -vvv https://google.com -o /dev/null --max-time 5

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0\* Could not resolve host: google.com

\* Closing connection 0

curl: (6) Could not resolve host: google.com

sh-4.2$

1. Test again with another domain which is not part of blocked domain.

nslookup www.amazon.com

**Sample output:** A sample screen shot for the nslookup command on www.amazon.com is shown below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sh-4.2$ nslookup www.amazon.com

Server: 10.1.0.2

Address: 10.1.0.2#53

Non-authoritative answer:

www.amazon.com canonical name = tp.47cf2c8c9-frontier.amazon.com.

tp.47cf2c8c9-frontier.amazon.com canonical name = d3ag4hukkh62yn.cloudfront.net.

Name: d3ag4hukkh62yn.cloudfront.net

Address: 99.84.79.137

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:c000:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:ca00:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:d600:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:e00:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:2400:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:3400:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:9600:7:49a5:5fd2:8621

Name: d3ag4hukkh62yn.cloudfront.net

Address: 2600:9000:2163:b200:7:49a5:5fd2:8621

TASK 3.6 - VERIFY DNS QUERY LOGS IN CLOUDWATCH

 It can take 5 to 10 minutes for logs to show up after the event.

1. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

1. In the left navigation pane, under **Logs**, choose **Log groups**.
2. Select **/aws/route53/NetworkFirewall-dnsfw-queries-loggroup** Log group.

 You observe alerts being captured with all details under the log group.

1. Select the latest Log streams.
2. In the Filter events dialog box type

www.google.com

 and look for the Log event where www.google.com is getting blocked.

 You see a filtered list of entries pertaining to the queries made to the domain.

 It may take few minutes for the queries to show up. Wait for some time to see the entries pertaining to your query.

**Sample output:** A sample screen shot of the log event is show below:



Through this task you can see how Route 53 DNS Firewall complements your AWS Network Firewall deployment. It ensures that you are securing traffic at the DNS as well as at the application level.

**Task complete:** You have successfully used Amazon Route 53 DNS Firewall to block certain DNS queries and thereby prevent access to those domains.

**Task 4: Using open source rules with AWS Network Firewall**

In this task, take a look at how you can utilize Open Source/Suricata compatible rules in AWS Network Firewall. For this example, choose Suricata specific rules from the community such as Proofpoint’s OPEN ruleset found [here](https://rules.emergingthreats.net/open/). For more details on open-source and commercial rules from Proofpoint, check [Additional Resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-300-NWNFIE-1%3A1.0.5-78686cb2/en-US#resources).

AWS Network Firewall can be setup in various deployment models depending on the requirements. To get more details on the deployment models and how to setup firewalls, check [Additional Resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-300-NWNFIE-1%3A1.0.5-78686cb2/en-US#resources).

To use open-source rules with AWS Network Firewall, use the following steps:

TASK 4.1 - DOWNLOAD/CLONE RULES

For this lab, you need to choose “User-Agents” category from Proofpoint’s OPEN rules which can be used to detect suspicious user-agents. Rules for this category are available [here](https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-user_agents.rules).

1. To download the rules, follow **ONE of the below options**:

* You can right-click > Save Link As… and save the rule file on your local machine.

OR

* Use **wget** to your local machine using the following command:

wget https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-user\_agents.rules -O emerging-user-agents.rules

OR

* Use **curl** to you local machine using the following command:

curl https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-user\_agents.rules -o emerging-user-agents.rules

OR

* On a Windows machine, you can use Invoke-WebRequest command:

Invoke-WebRequest https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-user\_agents.rules -OutFile $env:USERPROFILE\Downloads\emerging-user-agents.rules

Once downloaded, note down the location where the rules are saved so you can use it in next steps.

TASK 4.2 - CREATE RULE GROUP WITH SURICATA-COMPATIBLE RULES

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **Network Firewall**, choose **Create rule group**.
2. Choose **Create rule group** .
3. On the **Choose rule group type** page, in the **Rule group type** section:

* For **Rule group type**, select **Stateful rule group**.
* For **Rule group format**, select **Suricata compatible rule string** from the drop-down menu.
* For **Rule evaluation order**, select **Action order**.

1. Choose **Next** .
2. On the **Describe rule group** page, in the **Rule group details** section:

* For **Name**, enter

NetworkFirewall-Emerging-User-Agents-Rules-RuleGroup

.

* For **Description - optional**, enter

Emerging user agents rules rule group

.

* For **Capacity**, enter

300

.

1. Choose **Next** .
2. On the **Configure rules** page, in the **Rule variables - optional** section:

* In the **IP set variables** section:
  + For **Variable name**, enter

HOME\_NET

.

* + For **Values**, enter

10.0.0.0/8

.

 The HOME\_NET rule group variable is used to define source IP range eligible for processing in the Stateful Domain list and optionally Suricata compatible IPS Rule Groups. By default, it is set to the VPC CIDR where firewall endpoints are deployed.

* In the **Suricata compatible rule string** section, in the text input field,  copy and paste the rules that you downloaded in Task 4.1 which contains the Emerging Threats User-Agents rules.

1. Choose **Next** .
2. On the **Configure advanced settings - optional** page, leave everything default and choose **Next** .
3. On the **Add tags - optional** page, leave everything default and choose **Next** .
4. On the **Review and create** page, choose **Create rule group** .

TASK 4.3 - MODIFY FIREWALL POLICY TO ADD AND FORWARD TRAFFIC TO STATEFUL RULE GROUPS

1. In the left navigation pane, under **Network Firewall**, choose **Firewall policies**.
2. Choose **SpokeVPC-InspectionFirewall-Policy** to view details.

Before you add the new rule group that you created in the previous task, first remove the existing rule groups.

1. In the **Stateful rule groups** section, select these two rules:

* **NetworkFirewall-DomainAllow-RuleGroup**
* **NetworkFirewall-IcmpAlert-RuleGroup**

1. From the **Actions** drop-down menu, select **Disassociate from policy** and choose **Disassociate from policy** .

Now add the new rule group that you created in the previous task.

1. In the **Stateful rule groups** section, from the **Actions** drop-down menu, select **Add unmanaged stateful rule groups**.
2. Select **NetworkFirewall-Emerging-User-Agents-Rules-RuleGroup**.
3. Choose **Add stateful rule group** .

 To ensure traffic is forwarded to stateful inspection engine, it is required to add a custom defined stateless rule group which cover the interesting traffic or set a default action for all stateless traffic to be forwarded to stateful rule groups in the firewall policy. For this lab, you can leave the settings under **Stateless default actions** as ***Forward to stateful rule groups***.

TASK 4.4 - TEST AND MONITOR

To generate interesting traffic, you must have compute resources for example EC2 instances which are protected by AWS Network Firewall.

For this task, there is a pre-created EC2 instance to generate request against signature # 2029569 from the Proofpoint’s OPEN rule set imported in the previous task which detects the suspicious user-agents.

1. At the top of the page, in the unified search bar, search for and choose

EC2

.

1. In the left navigation pane, under **Instances**, choose **Instances**.
2. Select the instance **SpokeVPC-TestInstance1**.
3. Choose **Connect** from the navigation bar.
4. With **Session Manager** tab selected, choose **Connect** .

 Terminal session should open in a new browser tab.

1. On the **SpokeVPC-TestInstance1** instance, run the below command:

wget -U "easyhttp client" http://www.amazon.com -o /dev/null

 This command will generate a HTTP GET request with a user agent as **easyhttp client**. To check the logs, navigate to the

CloudWatch

 console.

1. At the top of the page, in the unified search bar, search for and choose

CloudWatch

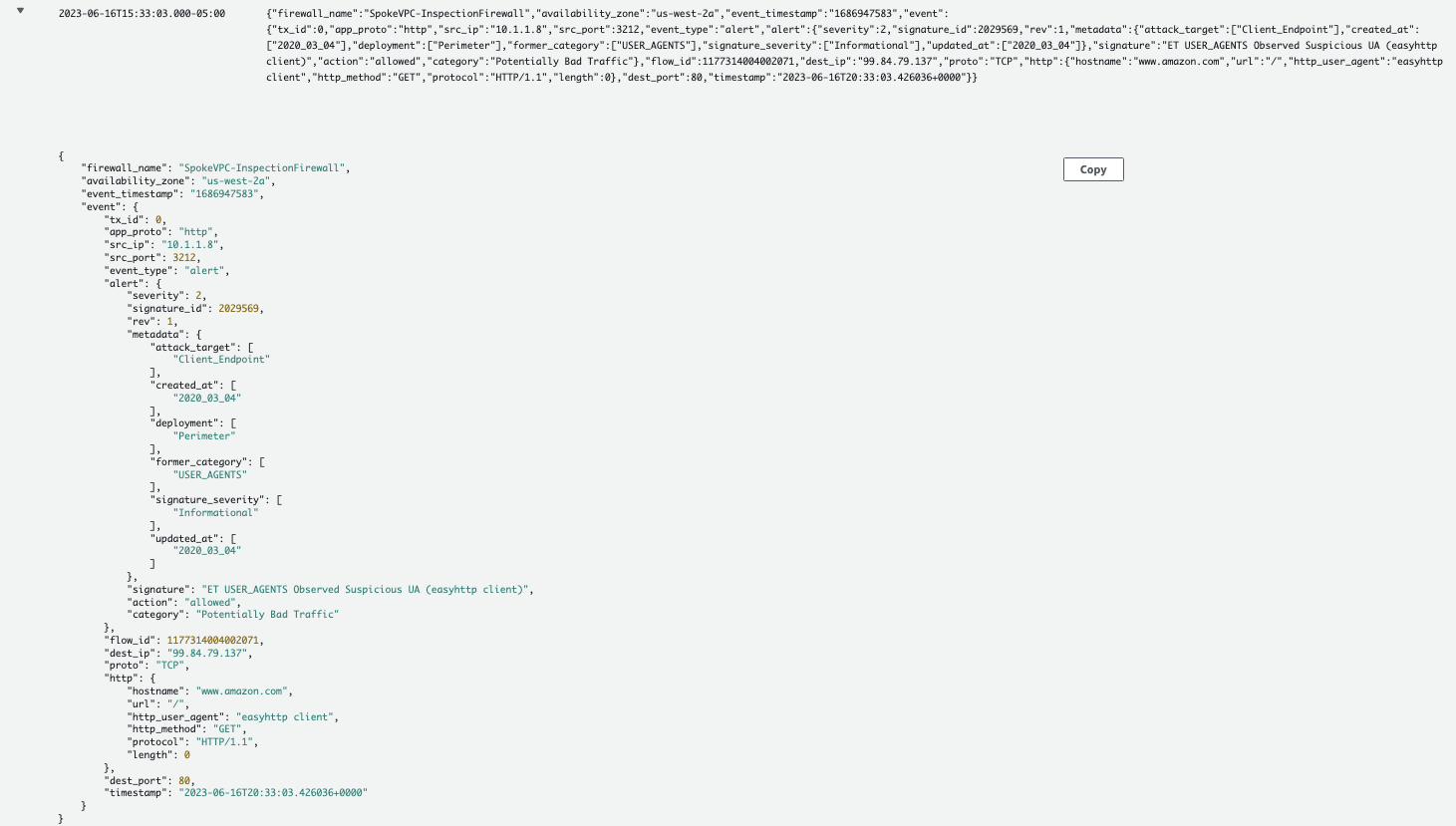
.

1. In the left navigation pane, under **Logs**, choose **Log groups**.
2. Select **/NetworkFirewall/Alert** Log group.
3. Select the latest Log streams **/aws/network-firewall/alert/SpokeVPC-InspectionFirewall\_\*** .
4. In the Filter events dialog box type

easyhttp client

 and look for the Log event with **“signature”: “ET USER\_AGENTS Observed Suspicious UA (easyhttp client)”**.

**Sample output:** A sample screen shot of the log event for the HTTP GET request with a user agent as **easyhttp client** is shown below:



This concludes how to import Suricata rule references into AWS Network Firewall.

Before moving on to next task, disassociate

NetworkFirewall-Emerging-User-Agents-Rules-RuleGroup

 from the firewall policy and associate/add

NetworkFirewall-DomainAllow-RuleGroup

 and

NetworkFirewall-IcmpAlert-RuleGroup

 back to the firewall policy by following the below steps:

1. To remove the rule group, select **NetworkFirewall-Emerging-User-Agents-Rules-RuleGroup**.
2. From the **Actions** drop-down menu, select **Disassociate from policy** and choose **Disassociate from policy** .
3. To add the new rule group, from the **Actions** drop-down menu, select **Add unmanaged stateful rule groups**.
4. Select the below rule groups under Stateful rule group:

* **NetworkFirewall-DomainAllow-RuleGroup**
* **NetworkFirewall-IcmpAlert-RuleGroup**

1. Choose **Add stateful rule group** .

**Task complete:** You have successfully imported and utilized Open Source/Suricata compatible rules in AWS Network Firewall to detect suspicious traffic.

**Task 5: Threat hunting with AWS Network Firewall**

This task shows you how to quickly use AWS Network Firewall to hunt for suspicious network activity that may indicate a compromise of your infrastructure. In this task look for non-TLS TCP traffic that is traversing over port 443 on your network, this type of network traffic is uncommon and should be investigated. To generate some interesting traffic for you to investigate, the lab has already deployed compute resources which are protected by AWS Network Firewall along with a malicious host.

TASK 5.1 - CREATE A FIREWALL RULE GROUP TO DETECT NON-TLS TRAFFIC OVER TLS PORTS

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
2. Choose **Create rule group** .
3. On the **Choose rule group type** page, in the **Rule group type** section:

* For **Rule group type**, select **Stateful rule group**.
* For **Rule group format**, select **Suricata compatible rule string** from the drop-down menu.
* For **Rule evaluation order**, select **Action order**.

1. Choose **Next** .
2. On the **Describe rule group** page, in the **Rule group details** section:

* For **Name**, enter

NetworkFirewall-Suricata-Detect-Non-TLS-Over-TLS-Ports-RuleGroup

.

* For **Description - optional**, enter

This rule will detect non-TLS traffic over ports 443

.

* For **Capacity**, enter

10

.

1. Choose **Next** .
2. On the **Configure rules** page, in the **Rule variables - optional** section:

* In the **IP set variables** section:
  + For **Variable name**, enter

HOME\_NET

.

* + For **Values**, enter

10.0.0.0/8

.

 The HOME\_NET rule group variable is used to define source IP range eligible for processing in the Stateful Domain list and optionally Suricata compatible IPS Rule Groups. By default, it is set to the VPC CIDR where firewall endpoints are deployed.

* In the **Suricata compatible rule string** section, in the text input field,  copy and paste the following Suricata rule:

alert tcp any any <> any 443 (msg:"SURICATA Port 443 but not TLS"; flow:to\_server,established; app-layer-protocol:!tls; sid:2271003; rev:1;)

 This rule will alert on TCP traffic from any interface in either direction <> over port 443 that is not !tls protocol.

1. Choose **Next** .
2. On the **Configure advanced settings - optional** page, leave everything default and choose **Next** .
3. On the **Add tags - optional** page, leave everything default and choose **Next** .
4. On the **Review and create** page, choose **Create rule group** .

TASK 5.2 - ADD THE NEWLY CREATED FIREWALL RULE GROUP TO THE EXISTING FIREWALL POLICY

1. In the left navigation pane, under **Network Firewall**, choose **Firewall policies**.
2. Choose **SpokeVPC-InspectionFirewall-Policy** to begin editing it.
3. Under **Stateful rule groups** section, from the **Actions** drop-down menu, select **Add unmanaged stateful rule groups**.
4. Select **NetworkFirewall-Suricata-Detect-Non-TLS-Over-TLS-Ports-RuleGroup**.
5. Choose **Add stateful rule group** .

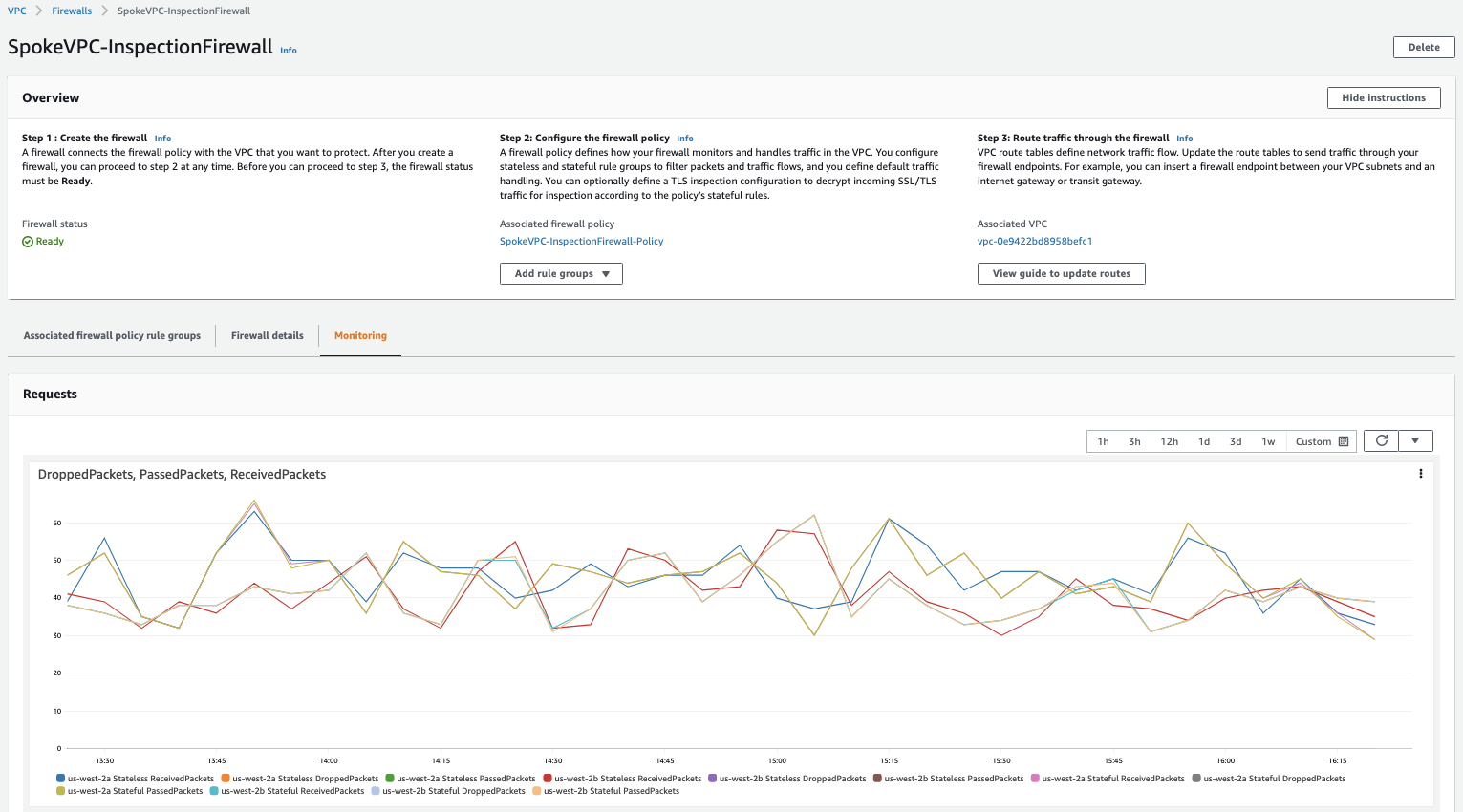
 To ensure traffic is forwarded to stateful inspection engine, it is required to add a custom defined stateless rule group which cover the interesting traffic or set a default action for all stateless traffic to be forwarded to stateful rule groups in the firewall policy. For this lab, you can leave the settings under **Stateless default actions** as ***Forward to stateful rule groups***.

TASK 5.3 - TIME TO INVESTIGATE

1. In the left navigation pane, under **Network Firewall**, choose **Firewalls**.
2. Choose **SpokeVPC-InspectionFirewall** to view details.
3. Select the **Monitoring** tab to check and see if traffic is flowing through the Network Firewall.

 It can take a few minutes to appear, hit the graph refresh button in the lower right until you see some traffic.

**Sample output:** A sample screen shot of the Monitoring graph is shown below:



1. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

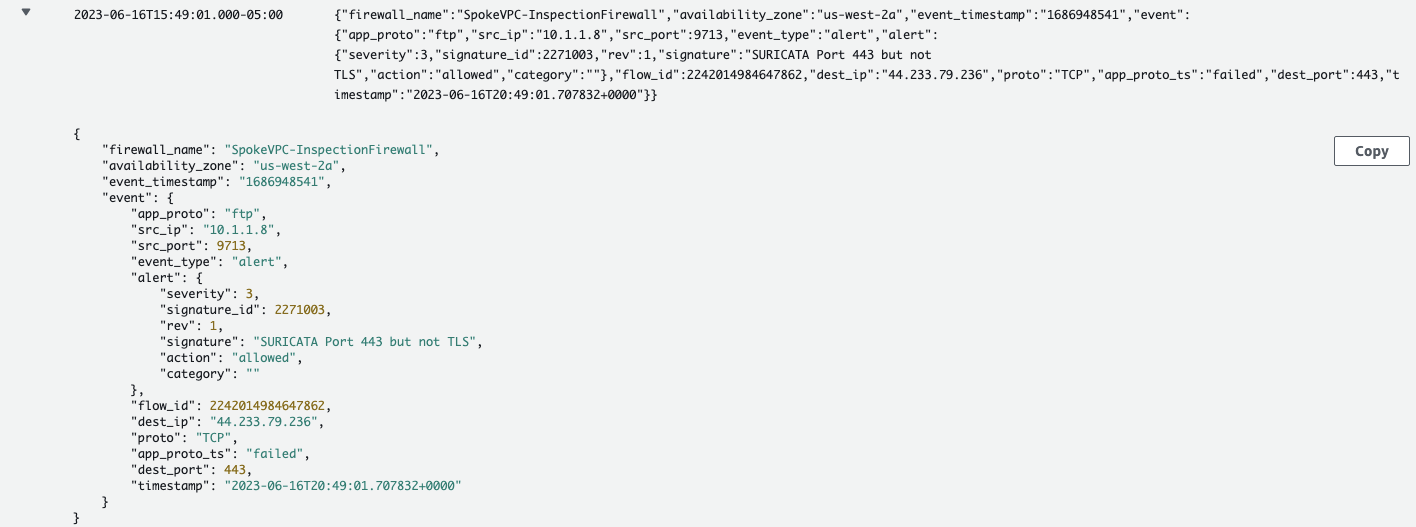
1. In the left navigation pane, under **Logs**, choose **Log groups**.
2. Select **/NetworkFirewall/Alert** Log group.
3. Select the latest Log streams **/aws/network-firewall/alert/SpokeVPC-InspectionFirewall\_\*** .
4. In the Filter events dialog box type

SURICATA

 and look for the Log event with signature set as **SURICATA Port 443 but not TLS** and action set as **allowed**.

 It may take 10-15 minutes before the logs show up in CloudWatch but when it does, you may see multiple log streams. Go into several of them and investigate what the alerts are about. Do you see anything suspicious? What protocol was actually in use?

**Sample output:** A sample screen shot of the log event is show below:



You discovered some instances are communicating non-TLS traffic over port 443, now what changes could you make to block that activity? Look at the Suricata rule you created to detect non-TLS traffic, what parameter change would drop the traffic?

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the left navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
2. Choose **NetworkFirewall-Suricata-Detect-Non-TLS-Over-TLS-Ports-RuleGroup** rule group.
3. Under the **Rules** section, choose **Edit** .
4. Modify the existing rule to the following:

drop tcp any any <> any 443 (msg:"SURICATA Port 443 but not TLS"; flow:to\_server,established; app-layer-protocol:!tls; sid:2271003; rev:1;)

1. Choose **Save rule group** .
2. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

1. In the left navigation pane, under **Logs**, choose **Log groups**.
2. Select **/NetworkFirewall/Alert** Log group.
3. Select the latest Log streams **/aws/network-firewall/alert/SpokeVPC-InspectionFirewall\_\*** .
4. In the latest Log stream, search for the Log event with signature set as **SURICATA Port 443 but not TLS** and action set as **blocked**.

**Sample output:** A sample screen shot of the log event is show below:



**Task complete:** You have successfully used AWS Network Firewall as an effective service for blocking unwanted network activity. Additionally, because of the integration of Suricata rules you have incredible flexibility and control over your AWS VPC network.

 It is important to test your rules, some protocols behave differently depending on the ‘flow’ parameter, such as

flow:to\_server

 vs.

flow:established

, read more about the topic under [Additional Resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-300-NWNFIE-1%3A1.0.5-78686cb2/en-US#resources).

**Conclusion**

You have successfully done the following:

* Configured the AWS Network Firewall using the Network Firewall rule groups to filter outbound web traffic.
* Created DNS Firewall Domain List, rule groups and rules using your custom domain lists.
* Used Route 53 DNS Firewall to filter and secure DNS traffic.
* Used CloudWatch Log Groups to filter queries for auditing and identifying potential threats.
* Configured stateful rule groups in AWS Network Firewall that follow Suricata-compatible intrusion prevention system (IPS) rule specifications.
* Applied stateful and stateless firewall rules to detect suspicious network traffic.
* Configured Firewall Policy and monitoring for AWS Network Firewall to hunt for suspicious network activity.
* Stoped malicious activities identified through monitoring and investigation of security alerts.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* For more information about AWS Network Firewall, see [AWS Network Firewall](https://docs.aws.amazon.com/network-firewall/latest/developerguide/what-is-aws-network-firewall.html).
* For more information about Route 53 Resolver DNS Firewall, see [Route 53 Resolver DNS Firewall](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver-dns-firewall.html).
* For more information about Deployment Model for AWS Network Firewall, see [Deployment Model for AWS Network Firewall (blog)](https://aws.amazon.com/blogs/networking-and-content-delivery/deployment-models-for-aws-network-firewall/).
* For more details on open-source and commercial rules from Proofpoint, see [ET Pro Ruleset](https://www.proofpoint.com/au/products/et-pro-ruleset).
* For more information about Flow Keywords, see [Flow Keywords](https://docs.suricata.io/en/suricata-6.0.0/rules/flow-keywords.html).

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).