Amazon Virtual Private Cloud

Network Access Analyzer

Amazon Virtual Private Cloud: Network Access Analyzer

Table of Contents

What is Network Access Analyzer 1
- Network Access Analyzer concepts 1
- Working with Network Access Analyzer 2
- Pricing 2

How Network Access Analyzer works 3
- Supported source and destination resources in findings 3
- Supported path resources in findings 3
- Limitations 4
- Findings 5

Getting started 6
- Step 1: Analyze your network using an Amazon created Network Access Scope 6
- Step 2: Review your findings 7
- Create your own Network Access Scopes 11
- Duplicate a Network Access Scope 13
- Delete a Network Access Scope 13

Getting started using the CLI 14
- Step 1: Create a Network Access Scope 14
- Step 2: Analyze a Network Access Scope 15
- Step 3: Get the results of a Network Access Scope analysis 16

Working with Network Access Scopes 20
- ResourceStatements 20
- PacketHeaderStatements 21
- MatchPaths entries 21
- ExcludePaths entries 22
- Example Network Access Scopes 22

Identity and access management 28
- How Network Access Analyzer works with IAM 28
  - Network Access Analyzer identity-based policies 28
  - Authorization based on Network Access Analyzer tags 30
  - Network Access Analyzer IAM roles 30
- Allow IAM users to access Network Access Analyzer 30
  - Create an IAM policy 31
- Required API permissions 32

Quotas 35
- Analysis runtime 35

Document history 36

What is Network Access Analyzer?

Network Access Analyzer is a feature that identifies unintended network access to your resources on AWS. You can use Network Access Analyzer to specify your network access requirements and to identify potential network paths that do not meet your specified requirements. You can use Network Access Analyzer to:

- **Understand, verify, and improve your network security posture** – Network Access Analyzer helps you identify unintended network access relative to your security and compliance requirements, enabling you to take steps to improve your network security.
- **Demonstrate compliance** – Network Access Analyzer helps you demonstrate that your network on AWS meets your compliance requirements.

Network Access Analyzer can help you verify the following example requirements:

- **Network segmentation** – Verify that your production environment VPCs and development environment VPCs are isolated from one another. Likewise, you can verify that a separate logical network is used for systems that process credit card information, and that it's isolated from the rest of your environment.
- **Internet accessibility** – Identify resources in your environment that can be accessed from internet gateways, and verify that they are limited to only those resources that have a legitimate need to be accessible from the internet.
- **Trusted network paths** – Verify that you have appropriate network controls such as network firewalls and NAT gateways on all network paths between your resources and internet gateways.
- **Trusted network access** – Verify that your resources have network access only from a trusted IP address range, over specific ports and protocols. You can specify your network access requirements in terms of:
  - Resource IDs (for example, `vpc-1234567890abcdef0`)
  - Resource types (for example, `AWS::EC2::InternetGateway`)
  - Resource tags
  - IP address ranges, port ranges, and traffic protocols

Network Access Analyzer concepts

The following are the key concepts for Network Access Analyzer.

Network Access Scopes

You can specify your network access requirements as Network Access Scopes, which determine the types of findings that the analysis produces. You add entries to **MatchPaths** to specify the types of network paths to identify. You add entries to **ExcludePaths** to specify the types of network paths to exclude.

- **MatchPaths** – Specifies the types of network paths that an analysis produces. Typically, you specify network paths that you consider to be a violation of your security or compliance requirements. For example, if you don't network paths that starting from VPC A and end in VPC B, specify VPC A as a source and VPC B as a destination. When you analyze this Network Access Scope, you would see any findings that indicate any potential network paths that start from a network interface in VPC A and end at a network interface in VPC B.

- **ExcludePaths** – Prevents certain network paths from appearing in your findings. Typically, you specify network paths that you consider to be a legitimate exception to your network security or compliance requirements. For example, to identify all network interfaces that are reachable from an internet gateway except for your web servers, specify the relevant paths using **MatchPaths**, and then exclude any path with your web servers as a destination using **ExcludePaths**. When you analyze this Network Access Scope, you would see any network paths that originate from an internet gateway and end at a network interface, except for any paths that end at your web servers.

Findings

Findings are potential paths in your network that match any of the **MatchPaths** entries in your Network Access Scope, but do not match any of the **ExcludePaths** entries in your Network Access Scope.

Working with Network Access Analyzer

You can use any of the following interfaces to work with Network Access Analyzer:

- **AWS Management Console** – Provides a web interface that you can use to create and manage Network Access Analyzer resources.
- **AWS Command Line Interface (AWS CLI)** – Provides commands for AWS services including Network Access Analyzer. The AWS CLI is supported on Windows, macOS, and Linux. For more information, see the AWS Command Line Interface User Guide.
- **AWS CloudFormation** – Create templates to provision and manage AWS resources as a single unit. For more information, see AWS::EC2::NetworkInsightsAccessScope and AWS::EC2::NetworkInsightsAccessScopeAnalysis.
- **AWS SDKs** – Provides language-specific APIs and takes care of many of the connection details, such as calculating signatures, and handling request retries and errors. For more information, see Tools to build on AWS.
- **Query API** – Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Network Access Analyzer. However, the Query API requires your application to handle low-level details such as generating the hash to sign the request and handling errors. For more information, see Amazon VPC actions in the *Amazon EC2 API Reference*.

Pricing

When you run a Network Access Analyzer analysis, you are charged based on the number of network interfaces that are analyzed. For more information, see Pricing.

How Network Access Analyzer works

Network Access Analyzer uses automated reasoning algorithms to analyze the network paths that a packet can take between resources in an AWS network. It then produces findings for paths that match a customer defined Network Access Scope. Network Access Analyzer performs a static analysis of a network configuration, meaning that no packets are transmitted in the network as part of this analysis. Because Network Access Analyzer only considers the state of the network as described in the network configuration, packet loss that's due to transient network interruptions or service failures is not considered in this analysis.

Not all AWS network configurations are supported by Network Access Analyzer. The following sections describe the types of network paths that Network Access Analyzer produces as findings. To see what resources you can reference in Network Access Scopes, see *Working with Network Access Scopes* (p. 20).

Contents

- Supported source and destination resources in findings (p. 3)
- Supported path resources in findings (p. 3)
- Limitations (p. 4)
- Findings (p. 5)

Supported source and destination resources in findings

A Network Access Analyzer finding is a network path that a packet can take in a network. Network Access Analyzer can only produce findings for network paths that start or end at the following types of resources:

- Internet gateways
- Network interfaces
- Transit gateway attachments
- VPC interface endpoints
- VPC gateway endpoints
- VPC gateway load balancer endpoints
- VPC service endpoints
- VPC peering connections
- Virtual private gateways

Supported path resources in findings

A Network Access Analyzer network path can pass through multiple resources from the start to the end of the network path. Only the following resource types are supported as resources on network paths in Network Access Analyzer findings:

- Internet gateways
- Load balancers

- NAT gateways
- Network ACLs
- Network firewalls
- Network interfaces
- VPC route tables
- Security groups
- Target groups
- Transit gateway route tables
- Transit gateway attachments
- VPC interface endpoints
- VPC gateway endpoints
- VPC gateway load balancer endpoints
- VPC endpoints services
- VPC peering connections
- Virtual private gateways

Limitations

There are some types of network paths that Network Access Analyzer does not report. Network Access Analyzer does not report paths that contain resources in accounts or Regions other than the account or Region being analyzed. In particular, Network Access Analyzer does not produce paths containing resources in subnets shared from other accounts, or to resources connected by VPC peering connections, virtual private gateways, internet gateways, or transit gateways to resources in other accounts or in different Regions.

The paths that Network Access Analyzer reports contain a bounded number of resources. Network Access Analyzer does not produce arbitrary length network paths through atypical network configurations, such as load balancers that target themselves, or NAT gateways that send packets immediately to another NAT gateway.

The following network configurations are not supported by Network Access Analyzer.

Internet gateways and virtual private gateways

- Network Access Analyzer supports internet gateways and virtual private gateways at the beginning or end of a path, but does not report paths that pass through internet gateways or virtual private gateways. For example, Network Access Analyzer does not produce paths that start in one VPC, pass through the internet, and end in a second Amazon VPC after passing through an internet gateway. These resources are outside of AWS networking and therefore out of scope.
- Network Access Analyzer does not support NAT reflection at internet gateways. For example, Network Access Analyzer does not report network paths from one network interface to another network interface that are addressed to the second network interface's public IPV4 address.

Application Load Balancers

- Network Access Analyzer does not support advanced forwarding rules.

Gateway Load Balancers

- Packet transformations applied by Gateway Load Balancer targets are ignored. A packet is reflected from the targets back to the Gateway Load Balancer untouched.

- Findings through a Gateway Load Balancer must start at a Gateway Load Balancer endpoint service.

Gateway Load Balancer endpoints

- Paths through a Gateway Load Balancer endpoint do not include the load balancer and its targets.

Network interfaces

- Network Access Analyzer does not produce findings that start or terminate at network interfaces that belong to a NAT gateway or Network Load Balancer.

Network Load Balancers

- Network Access Analyzer does not support instance targets without client IP preservation enabled.

Network firewalls

- Network Access Analyzer does not analyze network firewall rules. Paths as reported in findings containing network firewalls may be spurious if the firewall on the path is configured with rules that would otherwise block the reported network traffic.

Transit gateways

- Network Access Analyzer does not support paths through AWS Transit Gateway peering connections to other regions or accounts, or AWS Transit Gateway direct connections.

IPv4 only

- The analysis that Network Access Analyzer performs is limited to IPv4, using UDP or TCP.

Unsupported resources

Network Access Analyzer does not produce network paths through resources that are associated with Amazon API Gateway, AWS Global Accelerator, Traffic Mirroring, AWS Wavelength, or AWS Direct Connect. Network Access Analyzer can't produce network paths containing customer managed Amazon EC2 instances that modify packet forwarding behavior.

Findings

A single Network Access Analyzer scope analysis produces at most 100 findings. Network Access Analyzer makes a best effort attempt to return a diverse, representative set of findings from among all possible findings.

Network Access Analyzer does not ensure that the same findings are produced if you analyze the same Network Access Scope in the same network. Network Access Analyzer might produce new findings for existing Network Access Scope analyses if new configurations are supported in the future.

A running analysis times out after 1 hour and 30 minutes.

Getting started with Network Access Analyzer

You can use Network Access Analyzer to understand network access to resources in your virtual private clouds (VPCs). You can get started with Network Access Analyzer using one of the Amazon created Network Access Scopes.

Tasks

- Step 1: Analyze your network using an Amazon created Network Access Scope (p. 6)
- Step 2: Review your findings (p. 7)
- Create your own Network Access Scopes (p. 11)
- Duplicate a Network Access Scope (p. 13)
- Delete a Network Access Scope (p. 13)

> **Note**
> Network Access Analyzer evaluates network paths only within the account and Region from which you run the analysis.

Step 1: Analyze your network using an Amazon created Network Access Scope

The analysis can take a few minutes to complete.

To analyze a Network Access Scope

1. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/.
2. In the navigation pane, choose **Network Access Analyzer**.
3. If you are using Network Access Analyzer for the first time, choose **Get Started**.
4. Select one of the Amazon created Network Access Scopes:

   All-IGW-Ingress (Amazon created): This Network Access Scope identifies network paths from internet gateways to all network interfaces in your account.

   All-IGW-Egress (Amazon created): This Network Access Scope identifies network paths from all network interfaces to internet gateways in your account.

   All-VPC-Ingress (Amazon created): This Network Access Scope identifies inbound (ingress) paths from internet gateways, peering connections, VPC endpoints, VPNs, and transit gateways to all VPCs in your account.

   All-VPC-Egress (Amazon created): This Network Access Scope identifies outbound (egress) paths to internet gateways, peering connections, VPC endpoints, VPNs, and transit gateways from all VPCs in your account.

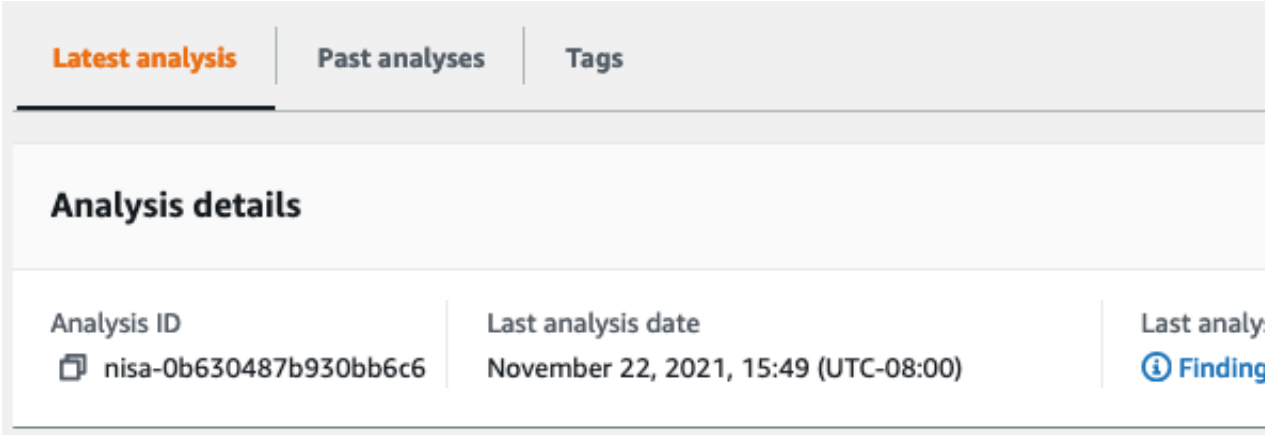
5. Choose **Analyze**.
6. Wait for the analysis to complete. Your screen will refresh with the results of the findings when the analysis is complete.

Step 2: Review your findings

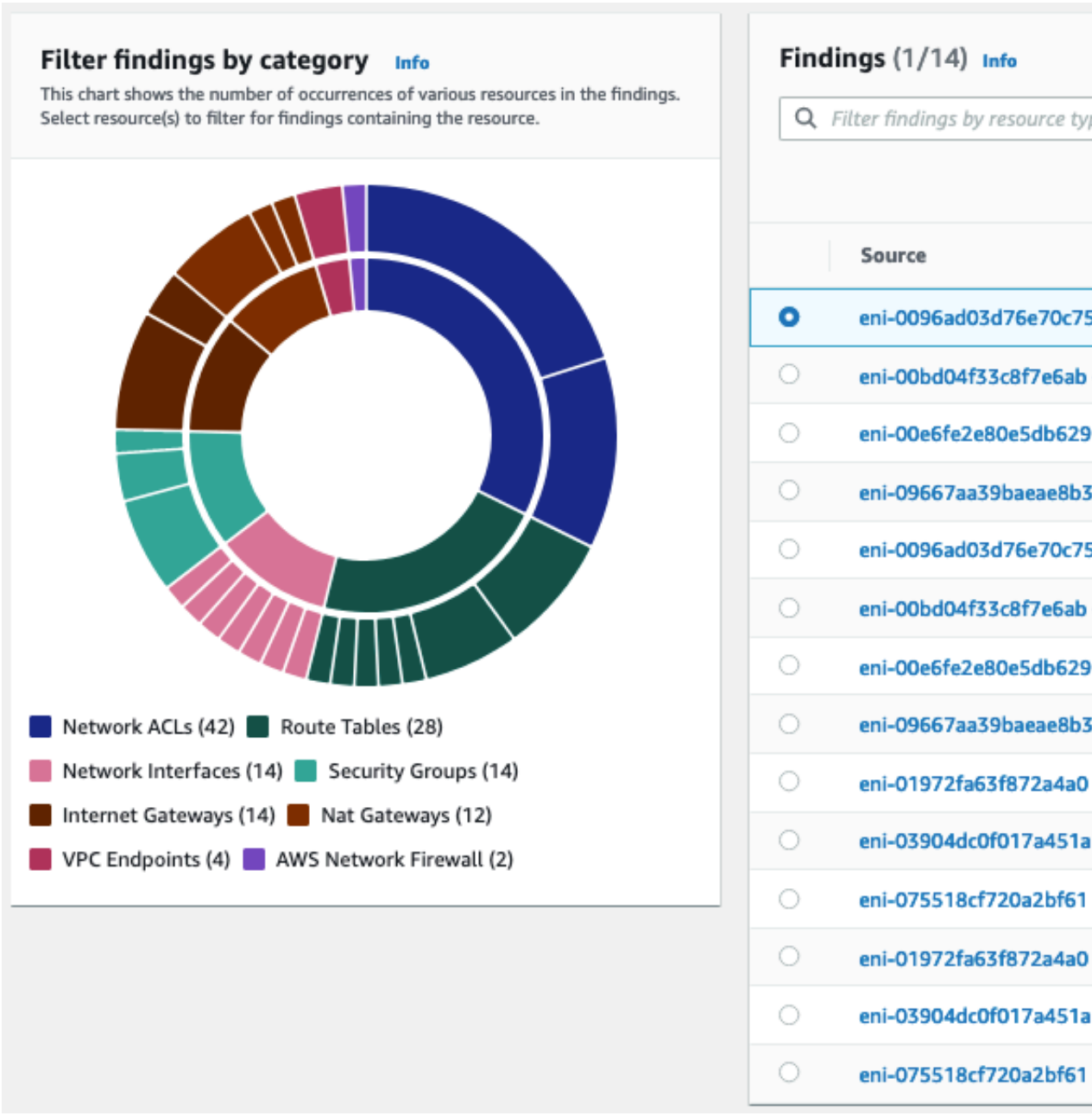
After your analysis is complete, you can review its results.

To review your findings

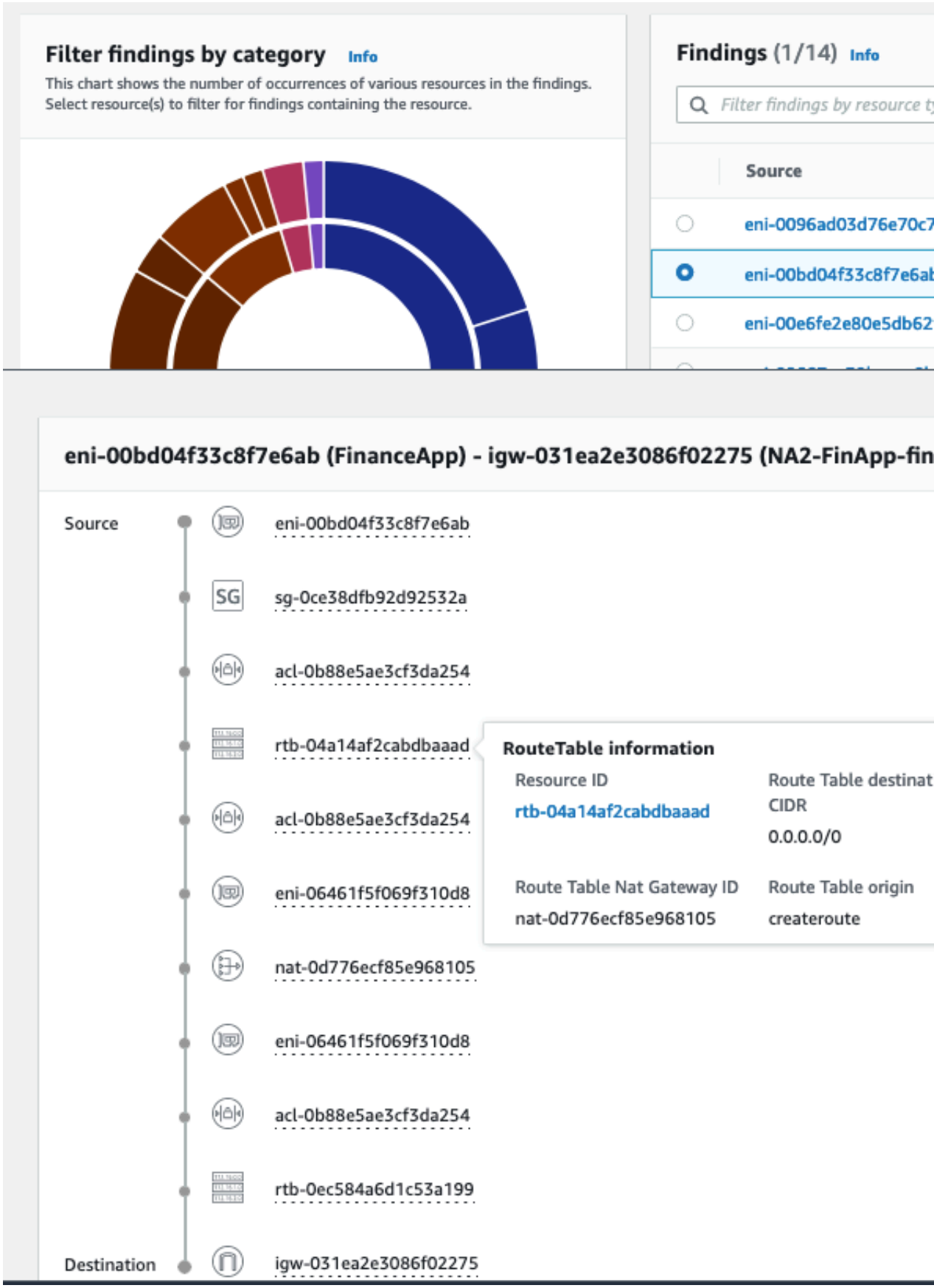
1. If your Network Access Scope analysis produces any findings, **Last analysis result** on the **Latest analysis** tab shows **Findings detected**, as shown in the following figure. Otherwise, **Last analysis result** shows **No findings detected**.



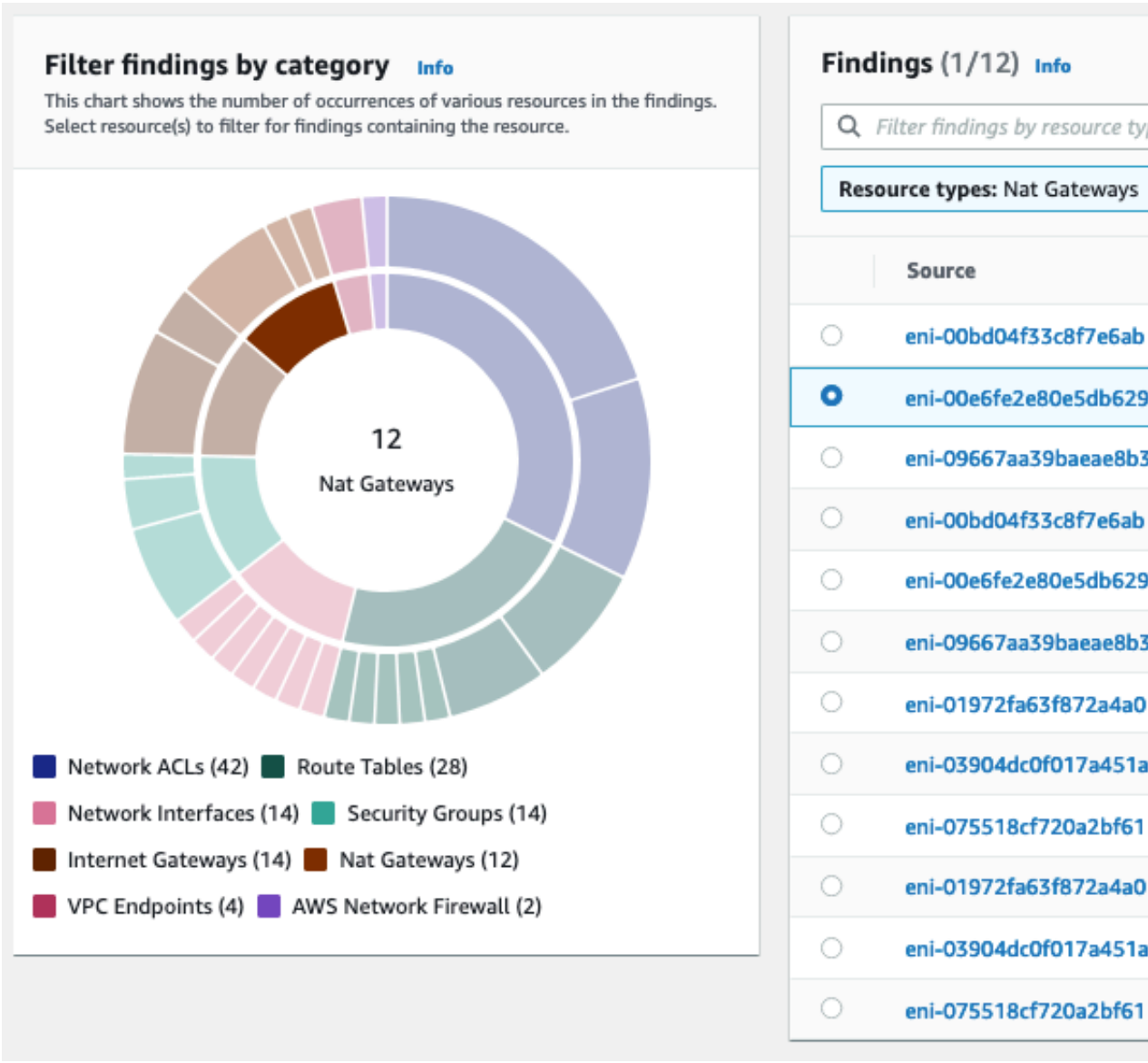
2. If you have findings present in your analysis, you can see the various network paths identified by your Network Access Scope, as shown in the following figure. For example, if you chose the **All-IGW-Egress (Amazon created)** Network Access Scope, the findings demonstrate network paths from all network interfaces to internet gateways in your account and Region.



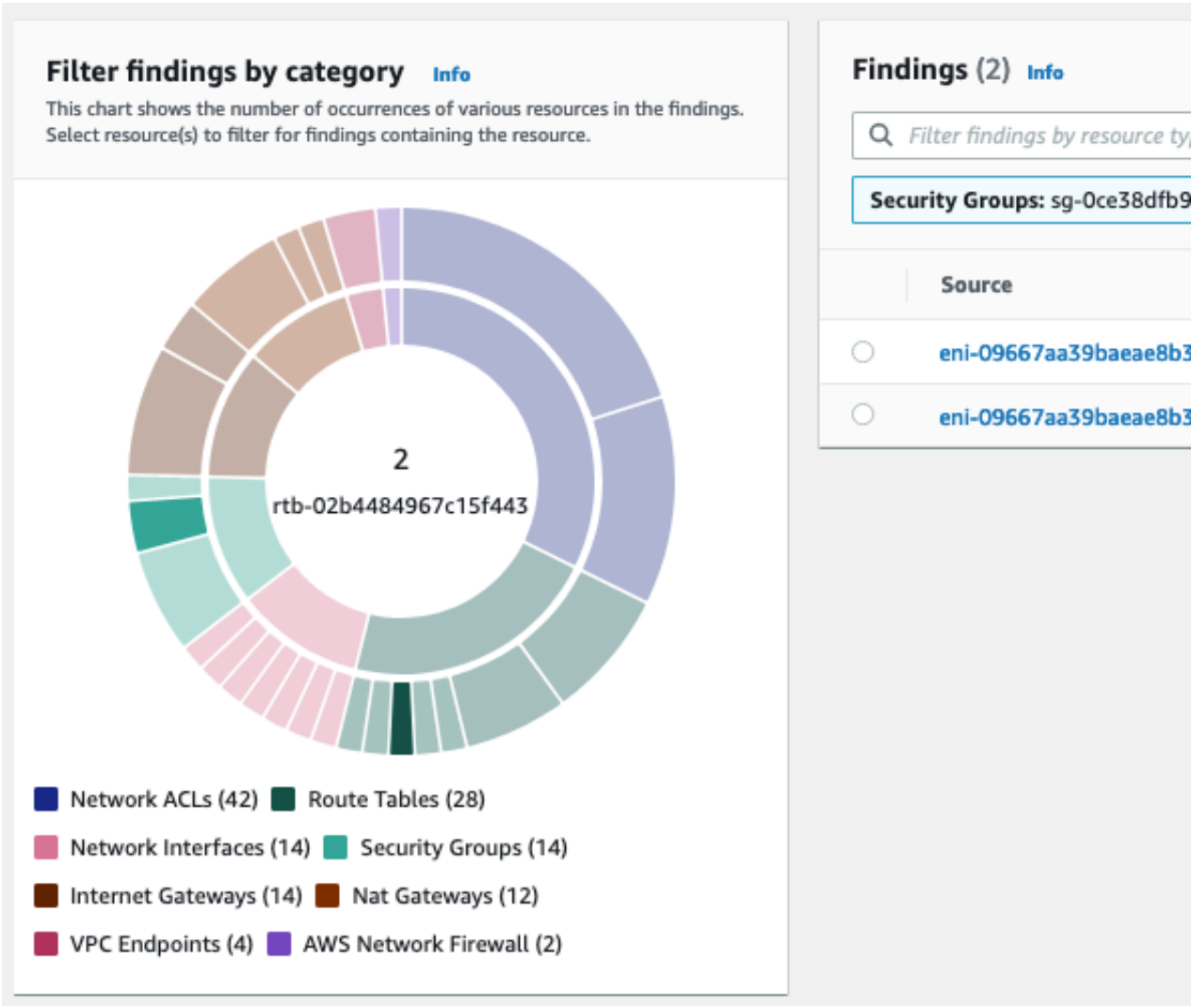
3. You can expand the pane from the bottom to view the details of the elements in the path. The information provided helps you understand the network configurations that produced the path. For example, as shown in the following figure, you can select a route table that appears on the path and see the route table entry that corresponds to traffic that is destined to 0.0.0.0/0.


Filter findings by category Info
This chart shows the number of occurrences of various resources in the findings.
Select resource(s) to filter for findings containing the resource.
Findings (1/14) Info
Filter findings by resource ty
Source
eni-0096ad03d76e70c7
eni-00bd04f33c8f7e6ab
eni-00e6fe2e80e5db62
eni-00bd04f33c8f7e6ab (FinanceApp) - igw-031ea2e3086f02275 (NA2-FinApp-fin
Source
eni-00bd04f33c8f7e6ab
SG
sg-0ce38dfb92d92532a
acl-0b88e5ae3cf3da254
rtb-04a14af2cabdbaaad
RouteTable information
Resource ID
rtb-04a14af2cabdbaaad
Route Table destinat
CIDR
0.0.0.0/0
Route Table Nat Gateway ID
nat-0d776ecf85e968105
Route Table origin
createroute
acl-0b88e5ae3cf3da254
eni-06461f5f069f310d8
nat-0d776ecf85e968105
eni-06461f5f069f310d8
acl-0b88e5ae3cf3da254
rtb-0ec584a6d1c53a199
Destination
igw-031ea2e3086f02275

4. You can filter for paths that have a particular resource type by selecting the inner ring of the chart. For example, you can filter for all paths that go through NAT gateways, as shown in the following figure. You can filter based on multiple resources or resource types to help understand your findings.



5. You can also filter based on any of the individual resource elements in your findings by selecting a resource from the outer ring of the chart. For example, you can filter for paths that contain a particular security group and a route table, as shown in the following figure.

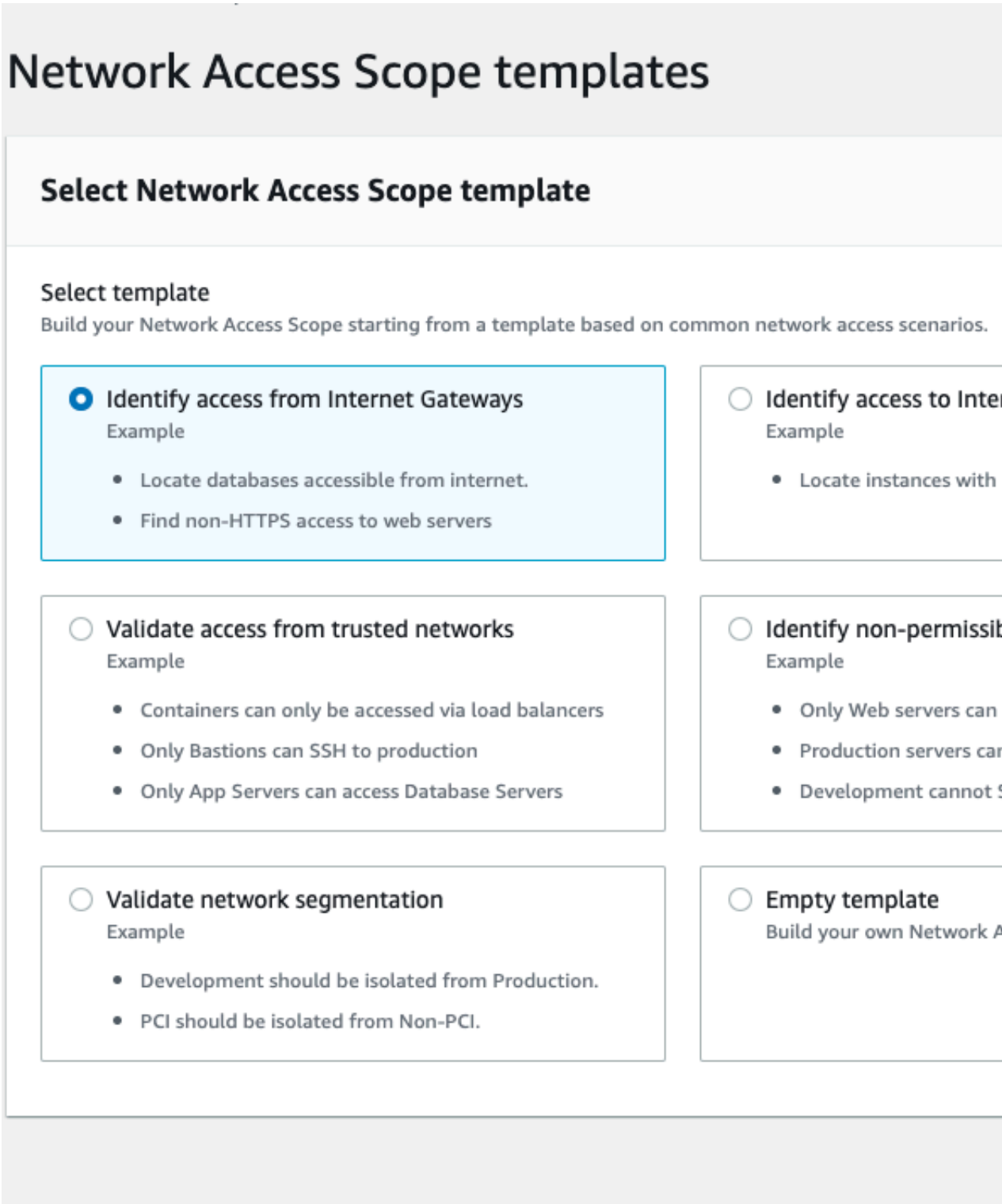


Create your own Network Access Scopes

You can also get started with Network Access Analyzer by using built-in templates based on common network access scenarios. For example, you can identify access from internet gateways or identify non-permissible traffic types. You can launch analyses that identify access paths and potential compliance issues for your AWS resources. You can also create your own scopes by choosing an empty template.

To create a Network Access Scope

1. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/.
2. In the navigation pane, choose **Network Access Analyzer**.
3. Choose **Create Network Access Scope**.



4. Add **Name** and **Description**. Optionally, enter **Match** and **Exclusion** conditions and **Tags** into your template. To specify the source resource, choose the resource type from **Source**, and then choose the specific **Resource selection** and **Resource types** options. You can choose multiple match conditions.

To specify the destination resource, choose the resource type from **Destination**, and then choose the specific resource from **Resources** and **Traffic type**. You can choose multiple exclusion conditions.

5. (Optional) For **Destination port**, enter the port number. By default, Network Access Analyzer considers all ports.
6. (Optional) To add a tag, choose **Add new tag** and then enter the tag key and tag value.
7. Choose **Next** and then choose **Create Network Access Scope**.

Duplicate a Network Access Scope

You can create a new Network Access Scope by starting from an existing one. You copy over all content from the previous Network Access Scope, and then change only the parameters that you want to alter.

To duplicate a Network Access Scope

1. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/.
2. Choose the Network Access Scope. Choose **Actions**, **Duplicate and modify**.
3. Change any of the details, match conditions, exclusion conditions, or tags as needed.
4. Choose **Duplicate and analyze Network Access Scope**.

Delete a Network Access Scope

If you no longer need a Network Access Scope, you can delete it. This action can't be undone.

To delete a Network Access Scope

1. On the Network Access Scopes page, select the check box next to the Network Access Scope that you want to remove.

   Note
   You can select only one Network Access Scope at a time.
2. Choose the **Actions** button and then choose **Delete Network Access Scope**.
3. When prompted for confirmation, enter **`Delete`**.

Getting started with Network Access Analyzer using the AWS CLI

The following procedure describes how to get started with Network Access Analyzer using the AWS CLI.

Tasks

- Step 1: Create a Network Access Scope (p. 14)
- Step 2: Analyze a Network Access Scope (p. 15)
- Step 3: Get the results of a Network Access Scope analysis (p. 16)

Step 1: Create a Network Access Scope

Use the following create-network-insights-access-scope command to create a Network Access Scope.

```
aws ec2 --region us-east-1 create-network-insights-access-scope
# optional/example input
--match-paths 'Source={ResourceStatement={Resources=vpc-abcd12e3}}'
# optional/example input
--exclude-paths 'Source={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}'
```

The following is example output.

```
{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-0e123eecc45c67d8",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-east-1:1234567891011:network-
insights-access-scope/nis-0e123eecc45c67d8",
        "CreatedDate": "2021-11-08T19:01:38.297000+00:00",
        "UpdatedDate": "2021-11-08T19:01:38.298000+00:00"
    },
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-0e123eecc45c67d8"
    }
}
```

You can also create a scope using the CLI JSON input option, as shown in the following example.

```
aws ec2 create-network-insights-access-scope --cli-input-json file://path-to-access-scope-
file.json
```

The following is an example input file.

```
{
     "MatchPaths": [
          {
               "Source": {
                    "ResourceStatement": {
                         "Resources": [
                              "vpc-abcd12e3"
```

```
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
}
```

See Generating an AWS CLI skeleton and input file for more details about using the CLI with JSON input.

Use the following describe-network-insights-access-scopes command to describe a Network Access Scope.

```
aws ec2 --region us-east-1 describe-network-insights-access-scopes
```

Use the following get-network-insights-access-scope-content command to get a Network Access Scope.

```
aws ec2 --region us-east-1 get-network-insights-access-scope-content --network-insights-
access-scope-id nis-0e123eecc45c67d8
```

Use the following delete-network-insights-access-scope command to delete a Network Access Scope.

```
aws ec2 --region us-east-1 delete-network-insights-access-scope --network-insights-access-
scope-id nis-0e123eecc45c67d8
```

Step 2: Analyze a Network Access Scope

Use the following start-network-insights-access-scope-analysis command to analyze a Network Access Scope. The analysis can take a few minutes to complete.

```
aws ec2 --region us-east-1 start-network-insights-access-scope-analysis --network-insights-
access-scope-id nis-0e123eecc45c67d8
```

The following is example output.

```
{
    "NetworkInsightsAccessScopeAnalysis": {
        "NetworkInsightsAccessScopeAnalysisId": "nisa-0e123eecc45c67d89",
        "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:936459623503:network-insights-access-scope-analysis/nisa-0e123eecc45c67d89",
        "NetworkInsightsAccessScopeId": "nis-0e123eecc45c67d8",
        "Status": "running",
        "StartDate": "2021-11-08T19:29:30.179000+00:00"
    }
}
```

Step 3: Get the results of a Network Access Scope analysis

After the analysis completes, you can view the results using the describe-network-insights-analyses command.

```
aws ec2 --region us-east-1 describe-network-insights-access-scope-analyses
```

Example 1: Success

The following is example output for a successful analysis.

```
{
    "NetworkInsightsAccessScopeAnalyses": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
            "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:936459623503:network-insights-access-scope-analysis/nisa-09aeb24f525f2d9f7",
            "NetworkInsightsAccessScopeId": "nis-0af1fcfd38e5cad4e",
            "Status": "succeeded",
            "StartDate": "2021-11-08T19:29:30.179000+00:00",
            "FindingsFound": "true",
            "Tags": []
        }
    ]
}
```

Example 2: No findings

The following is example output when no network paths are found in the analysis.

```
aws ec2 --region us-east-1 get-network-insights-access-scope-analysis-findings --network-
insights-access-scope-analysis-id nisa-07bcaad8bd8160e63
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
    "AnalysisFindings": []
}
```

Example 3: Findings reported

The following is example output where findings were reported in the analysis.

```
aws ec2 --region us-east-1 describe-network-insights-access-scope-analyses --network-
insights-access-scope-analysis-id nisa-0c0d3ec68a9bb2f22
{
    "NetworkInsightsAccessScopeAnalyses": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
            "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:1234567891011:network-insights-access-scope-analysis/nisa-0c0d3ec68a9bb2f22",
            "NetworkInsightsAccessScopeId": "nis-096f763940bb6bcf2",
            "Status": "succeeded",
            "StartDate": "2021-10-06T20:23:53.604000+00:00",
            "FindingsFound": "true",
            "Tags": []
        }
    ]
```

```
}

aws ec2 --region us-east-1 get-network-insights-access-scope-analysis-findings --network-
insights-access-scope-analysis-id nisa-0c0d3ec68a9bb2f22 --max-results 1
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
    "AnalysisFindings": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
            "NetworkInsightsAccessScopeId": "nis-096f763940bb6bcf2",
            "FindingComponents": [
                {
                    "SequenceNumber": 1,
                    "Component": {
                        "Id": "igw-1a23b4cd",
                        "Arn": "arn:aws:ec2:us-east-1:1234567891011:internet-gateway/
igw-1a23b4cd"
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": [
                            "172.31.22.225/32"
                        ]
                    },
                    "InboundHeader": {
                        "DestinationAddresses": [
                            "52.2.112.57/32"
                        ],
                        "DestinationPortRanges": [
                            {
                                "From": 80,
                                "To": 80
                            }
                        ],
                        "Protocol": "6",
                        "SourceAddresses": [
                            "0.0.0.0/5",
                            "11.0.0.0/8",
                            "12.0.0.0/6",
                            "128.0.0.0/3",
                            "16.0.0.0/4",
                            "160.0.0.0/5",
                            "168.0.0.0/6",
                            "172.0.0.0/12",
                            "172.128.0.0/9",
                            "172.32.0.0/11",
                            "172.64.0.0/10",
                            "173.0.0.0/8",
                            "174.0.0.0/7",
                            "176.0.0.0/4",
                            "192.0.0.0/9",
                            "192.128.0.0/11",
                            "192.160.0.0/13",
                            "192.169.0.0/16",
                            "192.170.0.0/15",
                            "192.172.0.0/14",
                            "192.176.0.0/12",
                            "192.192.0.0/10",
                            "193.0.0.0/8",
                            "194.0.0.0/7",
                            "196.0.0.0/6",
                            "200.0.0.0/5",
                            "208.0.0.0/4",
                            "224.0.0.0/3",
                            "32.0.0.0/3",
                            "64.0.0.0/2",
                            "8.0.0.0/7"
```

```
                        ],
                        "SourcePortRanges": [
                            {
                                "From": 0,
                                "To": 65535
                            }
                        ]
                    }
                },
                {
                    "SequenceNumber": 2,
                    "AclRule": {
                        "Cidr": "0.0.0.0/0",
                        "Egress": false,
                        "Protocol": "all",
                        "RuleAction": "allow",
                        "RuleNumber": 100
                    },
                    "Component": {
                        "Id": "acl-579af131",
                        "Arn": "arn:aws:ec2:us-east-1:1234567891011:network-acl/
acl-579af131"
                    }
                },
                {
                    "SequenceNumber": 3,
                    "Component": {
                        "Id": "sg-0cab31773e042794f",
                        "Arn": "arn:aws:ec2:us-east-1:1234567891011:security-group/
sg-0cab31773e042794f"
                    },
                    "SecurityGroupRule": {
                        "Cidr": "0.0.0.0/0",
                        "Direction": "ingress",
                        "PortRange": {
                            "From": 80,
                            "To": 80
                        },
                        "Protocol": "tcp"
                    }
                },
                {
                    "SequenceNumber": 4,
                    "Component": {
                        "Id": "eni-0680af09e502660e7",
                        "Arn": "arn:aws:ec2:us-east-1:1234567891011:network-interface/
eni-0680af09e502660e7"
                    },
                    "Subnet": {
                        "Id": "subnet-8061f9db",
                        "Arn": "arn:aws:ec2:us-east-1:1234567891011:subnet/subnet-8061f9db"
                    },
                    "Vpc": {
                        "Id": "vpc-abcd12e3",
                        "Arn": "arn:aws:ec2:us-east-1:1234567891011:vpc/vpc-abcd12e3"
                    }
                }
            ]
        }
    ],
    "NextToken":
 "AYADeDdyvQENR4bFEGARVczOdwQAhwACABFFbmNyeXB0aW9uQ29udGV4dAATVG9rZW5FbmNyeXB0aW9uVXRpbAAVYXdzLWNyeXB0
+v6C/JyLKmZzcGXs3NAp676D8RwoAdF/
sSfYUnAA7JwYLPlYSfBZ5fHHPjJ8Y6AVkJEzpGGza1CuzHFG9dqvkyuLoYxkpqGgbv0e0T2Q0rLfJID
+vNWEqSb03/6JXltR5ipYGD7yAnOb6vCBmheU9dDdbPE1SnidTc6XLpR8ihzdqSaJZnslAxYXNcsjrSEWmERdBhOIBaUUhRjvxaEABV
+lzU9BgN/NrgBnMGUCMQDSA4E1zrjcR
```

```
+iFS4RNJincDtRKZz3T2AmoI23+Xh44OHSrTR2XgBdewZZzvKX1tdkCMHDGRfeLrJMXLvVo/
sHL6ZqGR1FYWs3UWhMpkMGDdXZcQL+is60dXqAY1LOJLaDpaQ=="
}
```

Note
The list of **`SourceAddresses`** in the previous example includes everything in the 0.0.0.0/0 address range except the RFC1918 range.

Working with Network Access Scopes

With Network Access Analyzer, you can specify your network access requirements by using Network Access Scopes. A Network Access Scope defines outbound (egress) and inbound (ingress) traffic patterns, including sources, destinations, paths, and traffic types. Each Network Access Scope consists of one or more MatchPath entries, and zero or more ExcludePath entries, that together define a set of network traffic patterns.

When you start an analysis on a Network Access Scope, Network Access Analyzer produces findings. It identifies network paths in the Network Access Scope that match at least one of the MatchPath entries, and none of the ExcludePath entries. By combining MatchPaths and ExcludePaths, you can refine the findings produced by Network Access Analyzer to identify unexpected connectivity in your network.

MatchPath and ExcludePath entries have a similar structure, consisting of *ResourceStatements* and *PacketHeaderStatements* that define network traffic to match or exclude. In the following sections, we first describe ResourceStatements and PacketHeaderStatements, before describing MatchPath and ExcludePath entries.

ResourceStatements

A *ResourceStatement* defines the network components that the match or exclude statement applies to. Each ResourceStatement consists of either a list of resource IDs (or ARNs), or a list of resource types. A single ResourceStatement can list either resource IDs or resource types, but cannot list both.

The following elements can be specified as resource IDs:

- EC2 instances (source and destination fields only)
- Network interfaces (source and destination fields only)
- Security groups (source and destination fields only)
- Subnets (source and destination fields only)
- VPCs (source and destination fields only)
- Internet gateways (source and destination fields only)
- Virtual private gateways (source and destination fields only)
- Transit gateway attachments
- VPC peering connections
- VPC endpoints
- VPC endpoint services
- NAT gateways (through fields only)
- Network firewalls (through fields only)
- Classic, Application, Network and Gateway Load Balancers (through fields only)
- Resource groups (must contain resources of the preceding types)

Except for load balancers, all of the preceding elements can be specified as resource IDs (for example, vpc-XXXXXX), or as ARNs. Load balancers must be specified as ARNs.

The following elements can be specified as resource types:

- `AWS::EC2::InternetGateway` (only in source and destination fields)

- `AWS::EC2::VPNGateway` (only in source and destination fields)
- `AWS::EC2::TransitGatewayAttachment`
- `AWS::EC2::VPCPeeringConnection`
- `AWS::EC2::VPCEndpoint` (not supported in source fields)
- `AWS::EC2::VPCEndpointService`
- `AWS::EC2::NatGateway` (only in through fields)
- `AWS::ElasticLoadBalancing::LoadBalancer` (only in through fields)
- `AWS::ElasticLoadBalancingV2::LoadBalancer` (only in through fields)
- `AWS::NetworkFirewall::NetworkFirewall` (only in through fields)

PacketHeaderStatements

A *PacketHeaderStatement* defines the traffic type that a match or exclude statement applies to. If not specified, all traffic types are matched by default. All fields in a PacketHeaderStatement are optional, but if a PacketHeaderStatement is defined, at least one of its fields must be defined.

The following fields can be supplied in a PacketHeaderStatement:

- `Protocols` — a list of protocol strings to match. Supported values are `tcp` or udp. Either or both can be supplied. If this field is omitted, packets with either `tcp` or udp protocol are admitted.
- `SourceAddress` — a list of IP addresses or CIDR ranges. This option is mutually exclusive with SourcePrefixLists. If specified, only packets with matching source addresses are admitted. If neither SourcePrefixLists or SourceAddresses are specified, packets with any source address are admitted.
- `SourcePrefixLists` — a list of prefix lists in ID or ARN form. This option is mutually exclusive with SourceAddresses. If specified, only packets with matching source addresses are admitted. If neither SourcePrefixLists or SourceAddresses are specified, packets with any source address are admitted.
- `DestinationAddress` — a list of IP addresses or CIDR ranges. This option is mutually exclusive with DestinationPrefixLists. If specified, only packets with matching destination addresses are admitted. If neither DestinationPrefixLists or DestinationAddress are specified, packets with any destination address are admitted.
- `DestinationPrefixLists` — a list of prefix lists in ID or ARN form. This option is mutually exclusive with DestinationAddress. If specified, only packets with matching destination addresses are admitted. If neither DestinationPrefixLists or DestinationAddress are specified, packets with any destination address are admitted.
- `SourcePorts` — a list of ports or port ranges ("80","1024-65535"). If specified, only packets with source ports that match one of the ports or ranges are admitted. If omitted, packets with any source port are admitted.
- `DestinationPorts` — a list of ports or port ranges (for example,"80","1024-65535"). If specified, only packets with destination ports that match one of the ports or ranges are admitted. If omitted, packets with any destination port are admitted.

MatchPaths entries

A *MatchPath* entry consists of a source entry or a destination entry, or both, which define the types of network paths that should be produced as findings for a Network Access Scope. Source and destination entries have the same structure; each may define either a ResourceStatement or a PacketHeaderStatement, or both.

A Network Access Scope containing a MatchPath that specifies a source entry but no destination entry produces findings for network paths:

- that end at any supported resource,
- that start at a resource that matches the ResourceStatement condition of the source entry (if defined), and
- with a packet header that matches the PacketHeaderStatement of the source entry (if defined).

Similarly, if the MatchPath specifies a destination entry (but not a source entry), the Network Access Scope produces findings for network paths:

- that start at any supported resource, so long as the path ends at a resource that matches the ResourceStatement of the destination entry (if defined), and
- with a packet header that matches the PacketHeaderStatement of the destination entry (if defined).

A MatchPath can optionally specify both a source and destination entry, in which case both the start and end of the path must match the source and destination entry, respectively.

A Network Access Scope must specify at least one MatchPath entry, and can optionally specify multiple MatchPaths entries. If more than one MatchPath entry is specified, the Network Access Scope produces findings for any path that satisfies at least one of the MatchPath conditions.

ExcludePaths entries

A Network Access Scope can optionally define one or more ExcludePath entries. If a Network Access Scope defines any ExcludePaths, the Network Access Scope produces findings only for paths that match at least one MatchPath entry, and that do not match any ExcludePath entries.

An ExcludePath entry consists of at least one of the following, or a combination: a source entry, a destination entry, and a ThroughResource entry. Each field is optional, but at least one must be defined. Source and destination entries have the same structure; each can define either a ResourceStatement or a PacketHeaderStatement, or both. A ThroughResource entry, if defined, must be a list containing exactly one element that contains a ResourceStatement.

If an ExcludePath statement specifies a source entry, only paths that start at a component that matches the ResourceStatement (if defined) and that start with a packet header that matches the PacketHeaderStatement (if defined) are excluded from the findings. If the ExcludePath statement specifies a destination entry, only paths that end at a component and with a packet header that matches the ResourceStatement and PacketHeaderStatement of the destination entry are excluded.

Unlike MatchPaths entries, an ExcludePath can optionally define a ThroughResource entry. A ThroughResource entry excludes paths that match a given ResourceStatement anywhere along the path (not just at the beginning or end of the path). An ExcludePath with a ThroughResource entry can be used to exclude paths that pass through a NAT gateway, load balancer, or other intermediate path, regardless of where the path starts or ends, or can be used in combination with a source entry, destination entry, or both.

Example Network Access Scopes

In this section, we show some examples of Network Access Scopes. These examples are provided in JSON format.

Example: Identify all traffic between two subnets

To identify traffic between two subnets (subnet-814424dd and subnet-d75133f9) that are intended to be isolated from each other, create an access scope with two MatchPath entries, as shown in the following example.

```
{
      "MatchPaths": [
          {
              "Source": {
                  "ResourceStatement": {
                      "Resources": [
                          "subnet-814424dd"
                      ]
                  }
              },
              "Destination": {
                  "ResourceStatement": {
                      "Resources": [
                          "subnet-d75133f9"
                      ]
                  }
              }
          }, {
              "Source": {
                  "ResourceStatement": {
                      "Resources": [
                          "subnet-d75133f9"
                      ]
                  }
              },
              "Destination": {
                  "ResourceStatement": {
                      "Resources": [
                          "subnet-814424dd"
                      ]
                  }
              }
          }
    ]
}
```

The first MatchPath entry identifies any paths from ENIs in subnet-814424dd to ENIs in subnet-d75133f9; the second identifies any paths from ENIs in subnet-d75133f9 to ENIs in subnet-814424dd. Together, this network access scope identifies any traffic in either direction between the two subnets.

Instead of specifying the source and destinations as subnets, you can specify security groups (to identify any paths between the ENIs in those security groups), VPCs, or specific Amazon EC2 instances. You can also mix and match these (for example, to identify paths from a VPC to a security group).

Example: Use resource groups with Network Access Scopes

To identify paths to or from resources with specific resource tags, you can use AWS resource groups. With resource groups, you define a set of resources by their tags and their types. For example, to identify inbound (ingress) traffic to Amazon EC2 instances with the `Bastion` tag, create a resource group `bastions` as shown in the following example.

```
aws resource-groups create-group --name "bastions" --resource-query
 '{"Type":"TAG_FILTERS_1_0","Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],
\"TagFilters\":[{\"Key\":\"Bastion\"}]}"}'
```

```
{
      "MatchPaths": [
```

```
          {
              "Destination": {
                  "ResourceStatement": {
                      "Resources": [
                          "arn:aws:resource-groups:us-east-1:123456789012:group/bastions"
                      ]
                  }
              }
          }
      ]
}
```

The preceding example identifies any inbound (ingress) paths to any Amazon EC2 instances with the bastion tag. To identify any inbound (ingress) paths to bastion hosts using a port other than port 22 (ssh), combine the MatchPath with an ExcludePath that specifies destination port 22, as shown in the following example.

```
{
      "MatchPaths": [
          {
              "Destination": {
                  "ResourceStatement": {
                      "Resources": [
                          "arn:aws:resource-groups:us-east-1:123456789012:group/bastions"
                      ]
                  }
              }
          }
      ],
      "ExcludePaths": [
          {
              "Destination": {
                  "PacketHeaderStatement": {
                      "DestinationPorts": [
                          "22"
                      ]
                  }
              }
          }
      ]
}
```

Example: Identify all inbound (ingress) traffic from the public internet, except for a trusted CIDR range

To identify traffic from the internet while excluding paths that start in a trusted address range, create a MatchPath entry with a source type of **`AWS::EC2::InternetGateway`**, and an ExcludePath entry that lists the trusted address ranges in SourceAddresses, as shown in the following example.

```
{
      "MatchPaths": [
          {
              "Source": {
                  "ResourceStatement": {
                      "ResourceTypes": [
                          "AWS::EC2::InternetGateway"
                      ]
                  }
              }
          }
      ],
      "ExcludePaths": [
```

```
          {
              "Source": {
                  "PacketHeaderStatement": {
                      "SourceAddresses": [
                          "55.3.0.0/16"
                      ]
                  }
              }
          }
      ]
}
```

In the preceding example, the MatchPath entry restricts findings to traffic starting at an internet gateway, while the ExcludePaths entry suppresses any findings where the packet starts with a source address in the range 55.3.0.0/16.

Example: Exclude traffic originating at the addresses in a prefix list rather than a CIDR range

To exclude traffic that originates at the addresses in a prefix list rather than a CIDR range, use the SourcePrefixLists field instead of the SourceAddresses field, as shown in the following example.

```
{
      "MatchPaths": [
          {
              "Source": {
                  "ResourceStatement": {
                      "ResourceTypes": [
                          "AWS::EC2::InternetGateway"
                      ]
                  }
              }
          }
      ],
      "ExcludePaths": [
          {
              "Source": {
                  "PacketHeaderStatement": {
                      "SourcePrefixLists": [
                          "pl-02cd2c6b"
                      ]
                  }
              }
          }
      ]
}
```

Example: Identify all outbound (egress) traffic to the public internet, excluding a trusted CIDR range

To create an access scope that identifies traffic to the public internet while excluding a trusted address range, specify destination fields in the MatchPath and ExcludePath entries, rather than source fields, as shown in the following example.

```
{
      "MatchPaths": [
          {
              "Destination": {
                  "ResourceStatement": {
                      "ResourceTypes": [
                          "AWS::EC2::InternetGateway"
                      ]
                  }
              }
```

```
            }
        ],
        "ExcludePaths": [
            {
                "Destination": {
                    "PacketHeaderStatement": {
                        "DestinationAddresses": [
                            "55.3.0.0/16"
                        ]
                    }
                }
            }
        ]
}
```

Example: Identify inbound (ingress) traffic that bypasses a network firewall

To identify any inbound (ingress) traffic into the ENIs in a subnet that bypasses a middlebox such as a network firewall or load balancer, use an ExcludePath entry with a **ThroughResource** argument. In the following example, we assume that the subnet in question is subnet-814424dd.

```
{
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                },
                 "Destination": {
                    "ResourceStatement": {
                        "Resources": [
                            "subnet-814424dd"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "ThroughResources": [
                    {
                        "ResourceStatement": {
                            "ResourceTypes": [
                                "AWS::NetworkFirewall::Firewall"
                            ]
                        }
                    }
                ]
            }
        ]
}
```

Example: Identify traffic to ENIs that are members of a particular security group

To identify any traffic to ENIs that are members of a particular security group rather than a subnet, you can specify a security group as the destination resource, as shown in the following example.

```
{
        "MatchPaths": [
            {
```

```
            "Source": {
                "ResourceStatement": {
                    "ResourceTypes": [
                        "AWS::EC2::InternetGateway"
                    ]
                }
            },
             "Destination": {
                "ResourceStatement": {
                    "Resources": [
                        "sg-f15d59b3"
                    ]
                }
            }
        }
    ],
    "ExcludePaths": [
        {
            "ThroughResources": [
                {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::NetworkFirewall::Firewall"
                        ]
                    }
                }
            ]
        }
    ]
}
```

Identity and access management for Network Access Analyzer

To use Network Access Analyzer, you need an AWS account and AWS credentials. To increase the security of your AWS account, we recommend that you use an *IAM user* to provide access credentials instead of using your AWS account credentials. For more information, see AWS account root user credentials vs. IAM user credentials in the *Amazon Web Services General Reference*, and IAM best practices in the *IAM User Guide*.

For an overview of IAM users and why they are important for the security of your account, see AWS security credentials in the *Amazon Web Services General Reference*. For more information about working with IAM, see the *IAM User Guide*.

The following sections provide details on how an IAM administrator can use IAM to help secure your AWS resources, by controlling who can perform Network Access Analyzer actions.

Contents

- How Network Access Analyzer works with IAM (p. 28)
- Allow IAM users or groups to access Network Access Analyzer (p. 30)
- Required API permissions for Network Access Analyzer (p. 32)

How Network Access Analyzer works with IAM

Before you use IAM to manage access to Network Access Analyzer, you should understand what IAM features are available to use with Network Access Analyzer. To get a high-level view of how Network Access Analyzer and other AWS services work with IAM, see AWS services that work with IAM in the *IAM User Guide*.

Contents

- Network Access Analyzer identity-based policies (p. 28)
- Authorization based on Network Access Analyzer tags (p. 30)
- Network Access Analyzer IAM roles (p. 30)

Network Access Analyzer identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Network Access Analyzer supports specific actions and resources. There are no Network Access Analyzer service-specific condition keys that can be used in the `Condition` element of policy statements. To learn about all of the elements that you use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some

exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Network Access Analyzer shares its API namespace with Amazon EC2. Policy actions in Network Access Analyzer use the following prefix before the action: `ec2:`. For example, to grant someone permission to create a Network Access Scope with the `CreateNetworkInsightsAccessScope` API operation, you include the `ec2:CreateNetworkInsightsAccessScope` action in their policy. Policy statements must include either an `Action` or `NotAction` element.

To specify multiple actions in a single statement, separate them with commas as shown in the following example.

```
"Action": [
      "ec2:action1",
      "ec2:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action.

```
"Action": "ec2:Describe*"
```

The following actions are supported by Network Access Analyzer:

- `CreateNetworkInsightsAccessScope`
- `DeleteNetworkInsightsAccessScope`
- `DeleteNetworkInsightsAccessScopeAnalysis`
- `DescribeNetworkInsightsAccessScopeAnalyses`
- `DescribeNetworkInsightsAccessScopes`
- `GetNetworkInsightsAccessScopeAnalysisFindings`
- `GetNetworkInsightsAccessScopeContent`
- `StartNetworkInsightsAccessScopeAnalysis`

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

The following Network Access Analyzer API actions do not support resource-level permissions:

- `DescribeNetworkInsightsAccessScopeAnalyses`

- `DescribeNetworkInsightsAccessScopes`

Condition keys

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. For example, you might want a policy to be applied only after a specific date. To express conditions, use predefined condition keys.

Network Access Analyzer does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see Example: Restricting Access to a Specific Region.

The `Condition` element is optional.

Authorization based on Network Access Analyzer tags

You can attach tags to Network Access Analyzer resources or pass tags in a request. To control access based on tags, you provide tag information in the condition element of a policy using the `ec2:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, or `aws:TagKeys` condition keys. For more information, see Granting permission to tag resources during creation, Controlling access to specific tags, and Controlling access to EC2 resources using resource tags in the *Amazon EC2 User Guide*.

Network Access Analyzer IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with Network Access Analyzer

You can use temporary credentials to sign in with federation, to assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

Network Access Analyzer supports using temporary credentials.

Service-linked roles

Network Access Analyzer has no service-linked roles.

Service roles

Network Access Analyzer has no service roles.

Allow IAM users or groups to access Network Access Analyzer

Any IAM user that signs in to the AWS Management Console or AWS Command Line Interface (AWS CLI) must have permissions to access specific resources. You provide those permissions by using AWS Identity and Access Management (IAM), through policies.

The following procedure shows you how to attach an IAM policy to your IAM user or group that allows full access to Network Access Analyzer.

Note
We recommend creating a new IAM policy that grants only the permissions necessary to use Network Access Analyzer.

Create an IAM policy

Create an IAM policy that provides IAM users full access to Network Access Analyzer. Then attach the policy to your IAM user or group.

To create and attach an IAM policy (console)

1. Sign in to the IAM console at https://console.aws.amazon.com/iam/ with administrator credentials.
2. In the navigation pane, choose **Policies**.
3. In the content pane, choose **Create policy**.
4. Choose the **JSON** tab.
5. Paste the following JSON policy document in the text field.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:ListStackResources",
                "ec2:CreateNetworkInsightsAccessScope",
                "ec2:CreateTags",
                "ec2:DeleteNetworkInsightsAccessScopeAnalysis",
                "ec2:DeleteNetworkInsightsAccessScope",
                "ec2:DeleteTags",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInsightsAccessScopeAnalyses",
                "ec2:DescribeNetworkInsightsAccessScopes",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRegions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeTransitGatewayAttachments",
                "ec2:DescribeTransitGatewayConnects",
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayRouteTables",
                "ec2:DescribeTransitGatewayVpcAttachments",
                "ec2:DescribeTransitGateways",
                "ec2:DescribeVpcEndpointServiceConfigurations",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpnConnections",
                "ec2:DescribeVpnGateways",
                "ec2:GetManagedPrefixListEntries",
```

```
                "ec2:GetNetworkInsightsAccessScopeAnalysisFindings",
                "ec2:GetNetworkInsightsAccessScopeContent",
                "ec2:GetTransitGatewayRouteTablePropagations",
                "ec2:SearchTransitGatewayRoutes",
                "ec2:StartNetworkInsightsAccessScopeAnalysis",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancerAttributes",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeTags",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeTargetHealth",
                "network-firewall:DescribeFirewall",
                "network-firewall:DescribeFirewallPolicy",
                "network-firewall:DescribeResourcePolicy",
                "network-firewall:DescribeRuleGroup",
                "network-firewall:ListFirewallPolicies",
                "network-firewall:ListFirewalls",
                "network-firewall:ListRuleGroups",
                "resource-groups:ListGroupResources",
                "tag:GetResources",
                "tiros:CreateQuery",
                "tiros:GetQueryAnswer"
            ],
            "Resource": "*"
        }
    ]
}
```

6. When you are finished, choose **Review policy**.
7. On the **Review** page, enter a name for the policy, for example, `NetworkAccessAnalyzerPolicy`. Optionally, enter a description.
8. In **Summary**, review the policy to see the permissions that it grants, and then choose **Create policy**.
9. Attach the new policy to your IAM user or group.

   For information on attaching a policy to a user, see Changing permissions for an IAM user in the *IAM User Guide*. For information on attaching a policy to a group, see Attaching a policy to an IAM Group in the *IAM User Guide*.

Required API permissions for Network Access Analyzer

Network Access Analyzer relies on data from other AWS services. It uses the following permissions:

- `cloudformation:DescribeStacks`
- `cloudformation:ListStackResources`
- `ec2:CreateNetworkInsightsAccessScope`
- `ec2:CreateTags`
- `ec2:DeleteNetworkInsightsAccessScope`
- `ec2:DeleteNetworkInsightsAccessScopeAnalysis`
- `ec2:DeleteTags`
- `ec2:DescribeAvailabilityZones`
- `ec2:DescribeCustomerGateways`
- `ec2:DescribeInstances`
- `ec2:DescribeInternetGateways`

- ec2:DescribeManagedPrefixLists
- ec2:DescribeNatGateways
- ec2:DescribeNetworkAcls
- ec2:DescribeNetworkInsightsAccessScopeAnalyses
- ec2:DescribeNetworkInsightsAccessScopes
- ec2:DescribeNetworkInterfaces
- ec2:DescribePrefixLists
- ec2:DescribeRegions
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeTransitGatewayAttachments
- ec2:DescribeTransitGatewayConnects
- ec2:DescribeTransitGatewayPeeringAttachments
- ec2:DescribeTransitGatewayRouteTables
- ec2:DescribeTransitGateways
- ec2:DescribeTransitGatewayVpcAttachments
- ec2:DescribeVpcEndpoints
- ec2:DescribeVpcEndpointServiceConfigurations
- ec2:DescribeVpcPeeringConnections
- ec2:DescribeVpcs
- ec2:DescribeVpnConnections
- ec2:DescribeVpnGateways
- ec2:GetManagedPrefixListEntries
- ec2:GetNetworkInsightsAccessScopeAnalysisFindings
- ec2:GetNetworkInsightsAccessScopeContent
- ec2:GetTransitGatewayRouteTablePropagations
- ec2:SearchTransitGatewayRoutes
- ec2:StartNetworkInsightsAccessScopeAnalysis
- elasticloadbalancing:DescribeListeners
- elasticloadbalancing:DescribeLoadBalancerAttributes
- elasticloadbalancing:DescribeLoadBalancers
- elasticloadbalancing:DescribeRules
- elasticloadbalancing:DescribeTags
- elasticloadbalancing:DescribeTargetGroups
- elasticloadbalancing:DescribeTargetHealth
- network-firewall:DescribeFirewall
- network-firewall:DescribeFirewallPolicy
- network-firewall:DescribeResourcePolicy
- network-firewall:DescribeRuleGroup
- network-firewall:ListFirewallPolicies
- network-firewall:ListFirewalls
- network-firewall:ListRuleGroups
- resource-groups:ListGroupResources
- tag:GetResources

- `tiros:CreateQuery`
- `tiros:GetQueryAnswer`

Network Access Analyzer API calls

The following permissions are required to call the Network Access Analyzer APIs. Users need these permissions to create and start analyzing Network Access Scopes, or to view and delete existing paths and analyses in your account. You must grant IAM users permission to call the Network Access Analyzer API actions that hey need.

- `ec2:CreateNetworkInsightsAccessScope`
- `ec2:DeleteNetworkInsightsAccessScope`
- `ec2:DeleteNetworkInsightsAccessScopeAnalysis`
- `ec2:DescribeNetworkInsightsAccessScopeAnalyses`
- `ec2:DescribeNetworkInsightsAccessScopes`
- `ec2:GetNetworkInsightsAccessScopeAnalysisFindings`
- `ec2:GetNetworkInsightsAccessScopeContent`
- `ec2:StartNetworkInsightsAccessScopeAnalysis`

Describe API calls for networking-related resources

Network Access Analyzer uses describe calls while gathering information about your resources from Amazon VPC, Amazon EC2, Elastic Load Balancing, and AWS Network Firewall (for example, subnets, network interfaces, and security groups). To access Network Access Analyzer, IAM users must also have these API permissions.

If you specify a resource group in a resource statement, Network Access Analyzer uses `resource-groups:ListResourceGroups` while gathering information about your network configuration. This action requires the following permissions: `cloudformation:DescribeStacks` `cloudformation:ListStackResources`, and `tag:GetResources`.

Tagging-related API calls

To tag or untag Network Access Analyzer resources, users need the following Amazon EC2 API permissions. To allow IAM users to work with tags, you must grant them permission to use the specific tagging actions that they need.

- `ec2:CreateTags`
- `ec2:DeleteTags`

Tiros API calls

If you monitor API calls, you might see calls to Tiros APIs. Tiros is a service that is only accessible by AWS services and that surfaces network findings to Network Access Analyzer. Calls to the Tiros endpoint are required for Network Access Analyzer to function. To access Network Access Analyzer, IAM users must also have the same API permissions.

Quotas and considerations for Network Access Analyzer

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. You can request increases for some quotas, but not for all quotas.

To view the quotas for Network Access Analyzer, open the Service Quotas console. In the navigation pane, choose **AWS services**, and then select **Network Insights**. To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Your AWS account has the following quotas related to Network Access Analyzer.

Name	Default	Adjustable
Access scopes	1,000	Yes
Access scope analyses	10,000	Yes
Concurrent access scope analyses	25	Yes

Analysis runtime

All network interfaces in the account and Region are included in every analysis. The running analysis times out after 1 hour and 30 minutes.

Document history for Network Access Analyzer

The following table describes the releases for Network Access Analyzer.

Change	Description	Date
Initial release (p. 36)	This release introduces Network Access Analyzer.	December 1, 2021