



# Cisco Attack Surface Management

Getting Started Guide



---

# Table of Contents

<b>Understanding Cisco Attack Surface Management</b> .....	<b>5</b>
Attack Surface Management: Integrations .....	5
Attack Surface Management: Apps .....	6
Assets .....	6
Policies .....	6
Alerts .....	7
Compliance .....	7
Graph Viewer .....	7
Insights .....	8
Query Library .....	9
Ask Anything Search Bar .....	10
JupiterOne Query Language .....	10
<b>Configuring Managed Integrations</b> .....	<b>11</b>
<b>Setting Up Searches</b> .....	<b>12</b>
Ask Questions .....	12
Full Text Search .....	13
JupiterOne Query Language (J1QL) .....	13
Combining Full Text Search with J1QL .....	14
<b>Navigating the Graph</b> .....	<b>15</b>
Zoom and Move .....	17
<b>Learning the JupiterOne Query Language</b> .....	<b>18</b>
Part 1 - Simple Root Query .....	18
Part 2 - Infrastructure Analysis .....	20
2a - SSH Key Usage Examples .....	20
2b - EBS Volume Examples .....	21
2c - Unencrypted Data .....	22
2d - Tagging Resources .....	23
2e - Network Resources and Configurations .....	24

---

2f - Serverless Functions .....	26
Part 3 - User and Access Analysis .....	28
3a - IdP Users and Access .....	28
3b - Cloud users and Access .....	29
3c - Combined Users and Access Across All Environments .....	30
Part 4 - Cross Account Analysis .....	30
Part 5 - Endpoint Compliance .....	31
<b>Using Filters in the Asset Inventory App .....</b>	<b>34</b>
Quick Filter for Critical Assets .....	34
Quick Filters by Class and/or Type .....	35
Granular Filters by Properties .....	37
<b>Configuring Alerts .....</b>	<b>38</b>
Import Alert Rules from Rule Pack .....	38
Creating Custom Alert Rules .....	39
Setting Up Additional Alert Options .....	39
Managing Alerts .....	40
Configuring Daily Notification Emails .....	41
<b>Using the Visual Query Builder .....</b>	<b>42</b>
Permissions .....	42
Prerequisites .....	42
Creating Queries Using VQB .....	42
Using Wildcards .....	44
Filtering .....	45
<b>Managing Security Findings .....</b>	<b>46</b>
View Findings .....	46
Create Alerts for Findings .....	47
Examples: .....	47
Generate Views of Findings with J1QL Query and Graph .....	48
<b>Managing Policies and Procedures for Attack Surface Management .....</b>	<b>50</b>
Generating Policies and Procedures from Templates .....	50

---

---

Variables .....	50
Versioning .....	51
Download/ Export Policy and Procedure Documents .....	51
Policy Builder CLI .....	51
Using Your Own Existing Policies .....	51
<b>Inviting Users to Your Attack Surface Management Account/Org .....</b>	<b>52</b>
<b>Using Attack Surface Management APIs .....</b>	<b>53</b>
Endpoints .....	53
Authentication .....	53
API Key .....	53
Account ID .....	54
<b>Contacting Support .....</b>	<b>55</b>
<b>Change History .....</b>	<b>56</b>


# Understanding Cisco Attack Surface Management

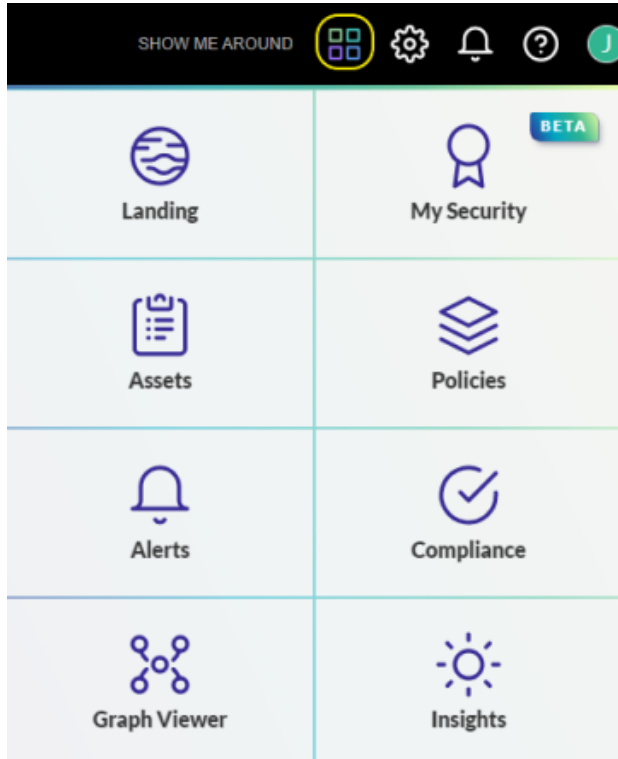
Cisco Attack Surface Management (formerly Cisco Secure Cloud Insights) is a cloud-native security platform that connects across your siloed security tools, empowering unobstructed visibility into security risks across your entire cyber asset universe. This extensible platform connects the dots between complex relationships and data, providing ultimate visibility to your environment, infrastructure, and operations.

## Attack Surface Management: Integrations

The first step in using Attack Surface Management is to bring your data into Attack Surface Management. There are numerous ready-made integrations that are easy to install and use to achieve end-to-end cyber asset visibility, context, and automation across every dimension of your digital universe. Attack Surface Management provides instructions on how to import the data to Attack Surface Management and understand the data model and mapping.

## Attack Surface Management: Apps

Attack Surface Management has separate apps that assist you in all the major components of security management. Click  to view the apps.



### Assets

After you import your data, you can analyze and visualize your complete infrastructure and security cyber asset inventory using the Assets app. In addition, the Assets app provides information about the types and classes of cyber assets you have, and the relationships between them.

### Policies

The Policies app enables you to articulate your organization policies and associate them to your compliance requirements.

Each policy and procedure document is written in its own individual Markdown file, and you can configure each policy file to link to other files. The templates are open-source that you can edit directly online using the Policies app.

To help you get started, Attack Surface Management provides 120+ [policy and procedure templates](#) to help your organization build your security program and operations. These

templates derive from the Attack Surface Management internal policies and procedures, and they have been through several rounds of compliance assessments.

## Alerts

Attack Surface Management enables you to configure alert rules in the Alerts app, using any query for continuous auditing and threat monitoring. You must have at least one active alert rule to trigger any alert. The easiest way to add some rules to an alert is to import rule packs that Attack Surface Management provides. You can also create custom rules.

## Compliance

Attack Surface Management provides a flexible platform for you to manage any compliance standard or framework as a set of controls or requirements. The platform enables you to:

- Import a compliance standard or security questionnaire
- Map policy procedures to each control or requirement
- Map data-driven compliance evidence by query questions
- Perform automated gap analysis based on query results
- Export compliance artifacts (summary or full evidence package)

## Graph Viewer

Attack Surface Management is built on a data-driven graph platform. JupiterOne Query Language (J1QL) is designed to traverse this graph and return a sub-graph or data from the entities and edges (such as relationships) of a sub-graph. You can view and interact with the sub-graph from any J1QL query result.

## Insights

The Insights app enables you to build reporting dashboards using J1QL queries.


You can configure each dashboard as either a team board that is shared with other account users or a personal board for the individual user. The layout of each board is individually saved per user, including the layout for team boards, so that each user can configure layouts according to their own preferences without impacting others.

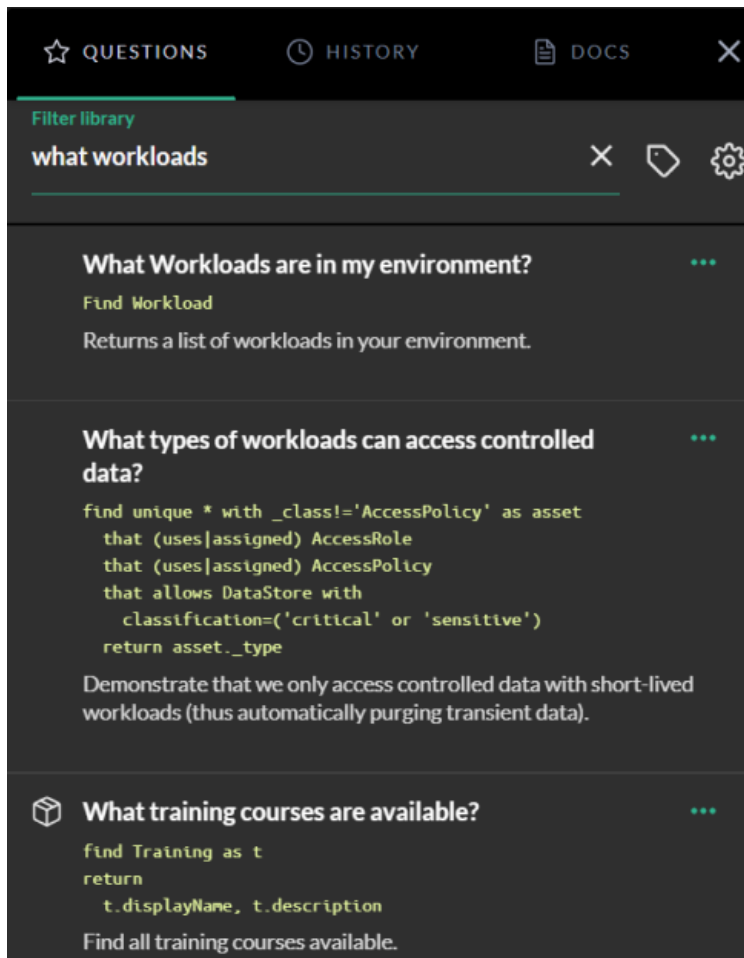
Administrators can save a team board layout as the default for other users.

You can build your own custom dashboards or utilize any of the existing boards that Attack Surface Management has already built.



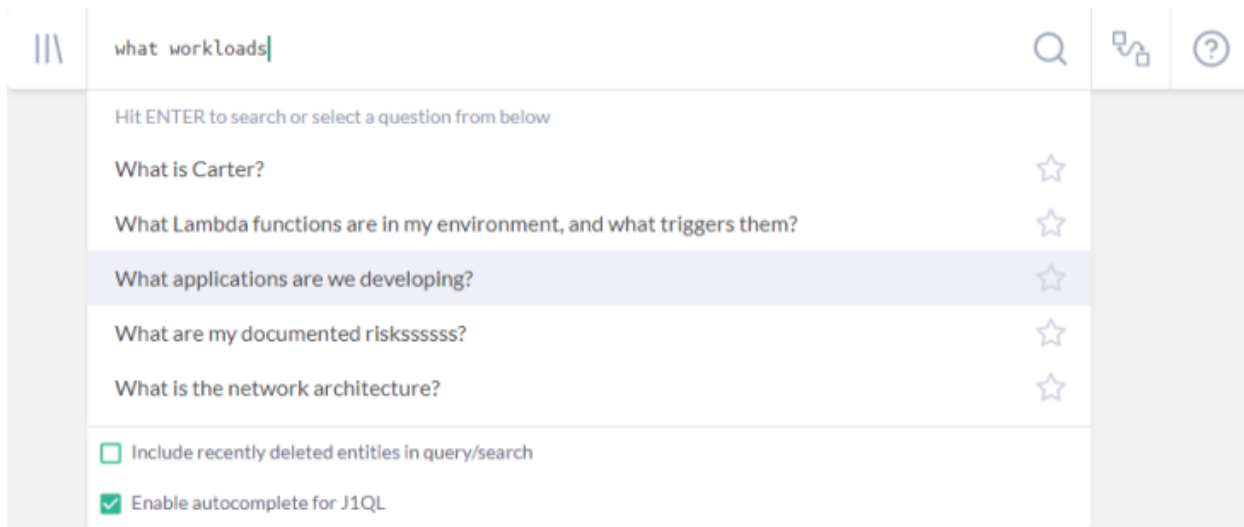
## Query Library

Attack Surface Management has hundreds of prebuilt and categorized queries for assessing the current state of your cyber assets. You can filter the queries on a specific topic, clone existing queries to create custom queries of your own, and save frequent searches for easy future reference. Click  on the landing page to access the query library.



## Ask Anything Search Bar

In addition to using the query library, from any Attack Surface Management page or app, you can enter questions in the search bar. By default, Attack Surface Management auto-completes the text and lists any related questions that you or someone else in your organization has asked.



## JupiterOne Query Language

The JupiterOne Query Language (J1QL) is a query language for finding the entities and relationships within your digital environment. J1QL blends together the capabilities of asking questions, performing full text search, and querying the complex entity-relationship graph.

J1QL is complex but a tutorial is available to help you learn. In addition, Attack Surface Management offers J1VQB, a visual query building app, a code-free tool for creating queries.

# Configuring Managed Integrations

You will need to have data in the Attack Surface Management platform to take advantage of its capabilities. The more data, the more powerful these capabilities become.

There are over a dozen managed integrations available out-of-the-box for turnkey configuration. More are added regularly. Each integration may have a slightly different mechanism for authentication and configuration, as required by the provider. For example, the AWS integration uses an IAM Role and Assume Role Trust policies for access. Other integrations may use an API key/token, OAuth, or Basic Auth.

Additionally, you can upload data outside of these managed integrations using the Attack Surface Management API Client or CLI. This allows you to centrally track, monitor, and visualize any of your data such as on-premises systems and security/compliance artifacts.

---

# Setting Up Searches

You can quickly search and get insight across your entire digital environment integrated with Attack Surface Management, right here from the Landing Zone. There are three modes of search:

1. **Ask questions** by typing in any keywords to search across all packaged/saved questions
2. **Full text search** across all entities based on their property values
3. **JupiterOne Query Language (J1QL)** for precise querying of entities and relationships

Results can be toggled in four different display modes: **Table**, **Graph**, **Raw JSON**, or **Pretty JSON**.



Note that for performance reasons, search results are limited to return up to 250 items. If you believe something is missing from a large result set, try tuning the query to generate more precise results

## Ask Questions

Just start typing any keyword (or combination of keywords) such as these (without quotes):

- compliance
- access
- traffic
- ssh
- data encrypted
- production

Or ask a question like:

- Who are my vendors?
- What lambda functions do I have in AWS?
- What is connected to the Internet?
- Who has access to ...?

## Full Text Search

Put your keywords in quotes ("keyword") to start a full text search. Or simply type in your keywords and press the Enter key. For example,

- "sg-123ab45c" will find an AWS EC2 Security Group with that group ID
- "Charlie" will find a Person and/or User with that first name, and potentially other resources related to that person/user

## JupiterOne Query Language (J1QL)

The JupiterOne Query Language (J1QL) is used here for searching for anything across all of your entities and relationships.

Here's the basic query structure:

- Start with an entity:

```
FIND {class or type of an Entity}
```

- Optionally add some property filters:

```
WITH {property}={value} AND|OR {property}={value}
```

- Get its relationships:

```
THAT {relationship_verb}|RELATES TO {class/type of another Entity}
```

For example:

```
FIND * WITH tag.Production='true'
```

(note the wildcard \* above to include everything)

```
FIND User THAT IS Person
```

If you don't know the exact relationship, you can just use the keyword `RELATES TO` to cover any/all relationship:

```
FIND User THAT RELATES TO Person
```

You can name an entity or relationship with an alias with the `AS {something}`. The alias can then be used in `WHERE` for additional filtering or comparison, or in `RETURN` for returning specific properties.

For example:

```
FIND Firewall AS fw
  THAT ALLOWS AS rule (Network|Host) AS n
WHERE
  rule.ingress=true and rule.fromPort=22
RETURN
  fw._type, fw.displayName, fw.tag.AccountName,
  n._type, n.displayName, n.tag.AccountName
```

The query language is case insensitive except for the following:

- **TitleCase Entity keyword after Find and the {relationship verb} will search for entities of that **Class**.** (e.g. CodeRepo)
- **lowercase Entity keyword after Find and the {relationship verb} will search for entities of that **Type**.** An entity type with more than one word is generally in `snake_case`. (e.g. github\_repo)
- Entity property names and values, and alias names defined as part of the query, are case sensitive.

## Combining Full Text Search with J1QL

You can also start with a full text search and then use J1QL to further filter the results from the initial search. For example:

```
Find "Administrator" with _class='AccessPolicy' that ASSIGNED (User|AccessRole)
```

```
Find 'security officer' with _type='employee'
```

```
Find 'roles responsibilities' with _class=('Policy' or 'Procedure')
```

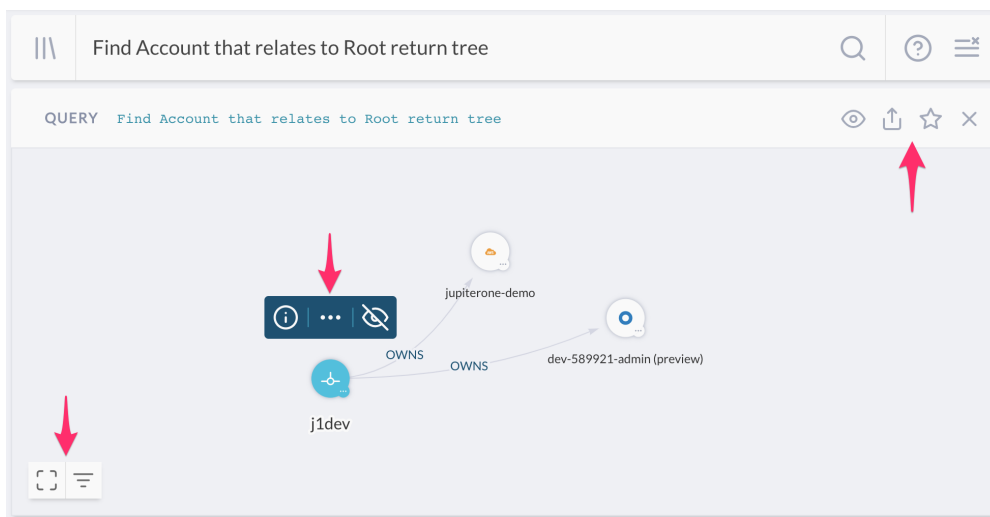
Note that either single quotes (') or double quotes (") will work for both full text search keywords and property string values.

# Navigating the Graph

Attack Surface Management is built on a data-driven graph platform. For the story that inspired us to build it, check out this [blog](#).

JupiterOne Query Language (J1QL) is designed to traverse this graph and return a sub-graph -- or data from the nodes (entities, for example) and edges (relationships, for example) of a sub-graph. You can view and interact with the sub-graph from any query result.






This screenshot below shows an example result graph from a query in the Ask Anything search bar:







The first set of controls on the upper right corner does the following:

Control	Function
	<b>Switch views</b> between Table, Graph, Raw JSON, and Pretty JSON.
	<b>Share the query</b> – shows a modal popup with the weblink to copy and share.
	<b>Save the query</b> – shows a modal popup where you can provide a title, description, and optionally some tags to save it to your own query library.
	<b>Remove the result</b> for this particular query/question from the page view.

Selecting any node (an entity, for example) on the graph will bring up a set of controls right on top of it that allows you to interact with the node. They serve the following functions:

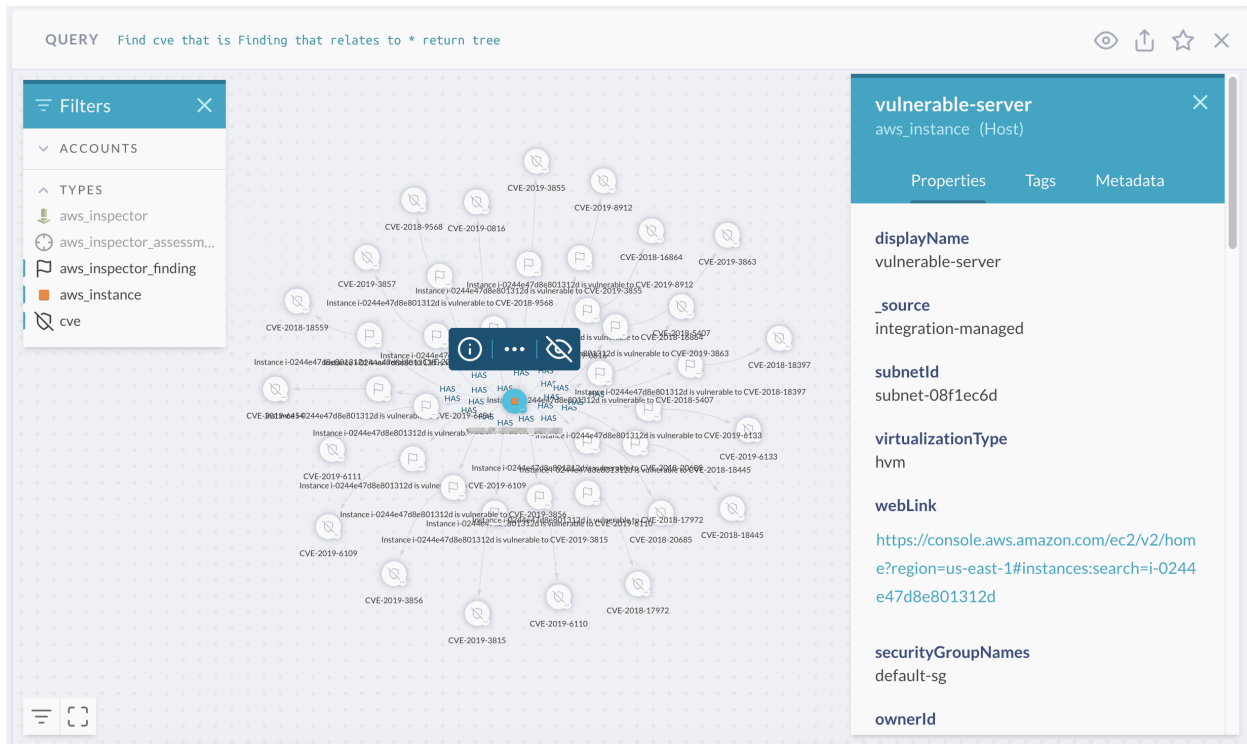
Control	Function
	<b>Open side panel</b> to show the detailed properties, tags, and metadata of the selected entity. Note that you can select an edge and see its properties in the side panel as well.
	<b>Load neighbors</b> – shows a modal popup with the weblink to copy and share.
	<b>Hide selected node</b> from graph to reduce clutter. You can unhide all hidden nodes from the bottom left control.
	<b>Expand grouped nodes</b> of the same type that have the same parent nodes. This option may not be always available depending on the data in the graph.
	<b>Collapse nodes</b> of the same type that have the same parent nodes into a group. This option may not be always available depending on the data in the graph.

The last set of controls are at the bottom left corner of the graph, and they do the following:

Control	Function
	<b>Maximize</b> graph in full screen mode.
	<b>Restore</b> graph in query result component.
	<b>Open filter panel</b> to let you filter (show/hide) nodes on the graph by <b>Account</b> and/or <b>Type</b> .
	<b>Unhide hidden nodes</b> – This control icon will only show up when there are hidden nodes on the graph.



Here's a screenshot of a graph with the **property panel** and **filter panel** open:



## Zoom and Move

Control	Function
 	<p><b>Scroll</b> using your mouse/touchpad to zoom in/out on the graph</p>
	<p><b>Click and Drag</b> on a blank spot on the graph using your mouse/touchpad to move the graph. Click and Drag on a selected node to move that particular node.</p>

The stand-alone Galaxy/Graph Viewer app uses the same sets of controls.

# Learning the JupiterOne Query Language

Querying can be the most challenging, yet the most fun and rewarding part of the Attack Surface Management experience. Once you become familiar with the query language, we are certain that you will find yourself uncovering all sorts of previously undiscovered insight from your data.

The JupiterOne Query Language (J1QL) is a query language for finding the entities and relationships within your digital environment. J1QL blends together the capabilities of asking questions, performing full text search, or querying the complex entity-relationship graph.

There are plenty of pre-packaged queries you can easily use in the Ask Anything search bar or browse in the Query Library. This tutorial focuses instead on helping you construct custom queries yourself.

This tutorial builds on the full J1QL documentation using some common use cases.



These example queries can be modified based on the specific structure of your data sources.

## Part 1 - Simple Root Query

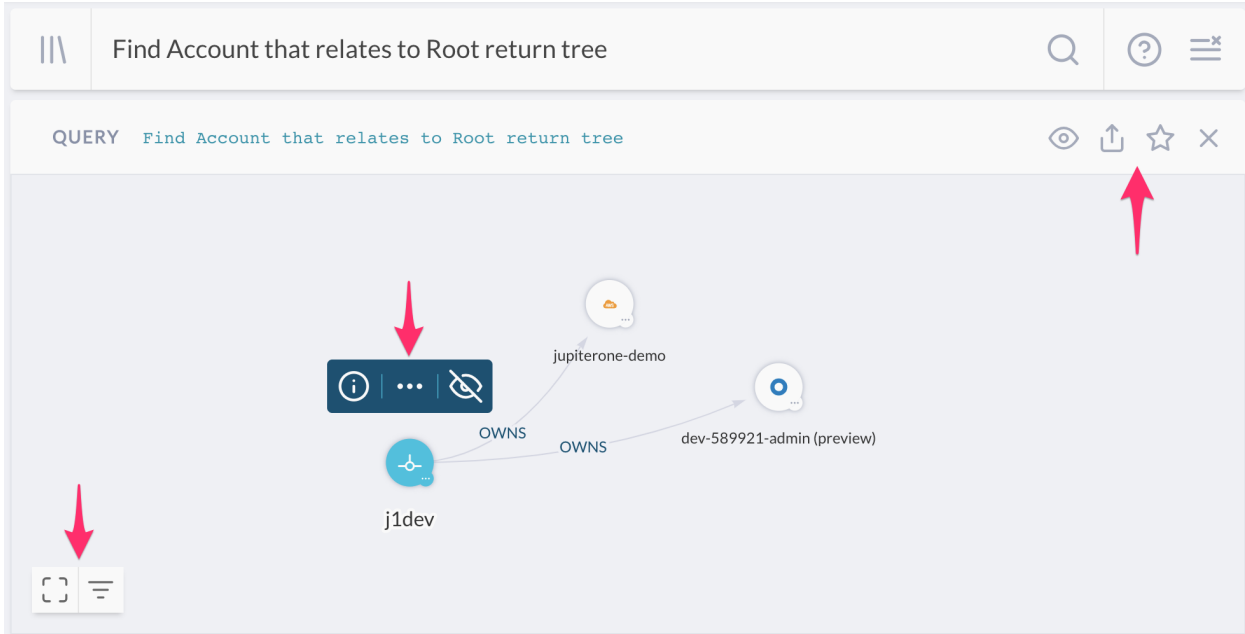
First, let's try this query:

```
Find Account that relates to Root return tree
```

Please note the noun that immediately follows the verb is case sensitive:

- A **TitleCase** word tells the query to search for entities of that **class** (e.g. Account, Firewall, Gateway, Host, User, Root, Internet, etc.);
- A **snake\_case** word tells the query to search for entities of that **type** (e.g. aws\_account, aws\_security\_group, aws\_internet\_gateway, aws\_instance, aws\_iam\_user, okta\_user, user\_endpoint, etc.)

You should get a result that looks like this (the `return tree` part of the query tells it to show the graph view by default):



The selected node in the above example is the special `Root` node, which represents your organization. Depending on the number of integration configurations you have, you'll see different number of accounts connected, showing that the `Root` entity `OWNS` these `Account` entities.

See the three sets of controls in the result panel. Starting from top right to bottom left --  
The first set of controls (next to the query) allows you to:

- Switch views between **Table**, **Graph**, **Raw JSON**, and **Pretty JSON**.
- Share the query -- shows a popup box with the weblink to copy and share.
- Save the query -- give it a title, description, and optionally some tags to save it to your own query library.
- Close / remove this result panel from the page.

The second set of controls (above the selected entity node) allows you to:


- Show the detailed properties, tags, and metadata of the selected entity.
- Expand the entity to see more of its connected neighbors - this will bring in additional data that may not have been returned by the original query, allowing you to further the search and analysis.
- Hide the selected entity node from the graph view - once you've hidden an entity, an unhide button will show up in the third set of controls at the bottom left of the graph, allowing you to unhide all currently hidden entities.

The last set of controls (at the bottom left corner) allows you to:

- Toggle the full screen mode.
- Opens up the filter panel to show/hide entities in the graph by account or entity type.
- Unhide all currently hidden entities (not shown in the above screenshot -- it only shows up when there is at least one hidden entity).

For more details about graph controls, see the [Navigating the Graph](#).

## Part 2 - Infrastructure Analysis

 Examples in this section require at least one AWS integration configuration.

If you've configured an AWS integration, you are now ready to try something a lot more interesting. Type in, or copy/paste the following query:

### 2a - SSH Key Usage Examples

```
Find AccessKey with usage='ssh'
```

This should find a set of `aws_access_key` entities used for SSH access into your EC2 instances, assuming you have some of those and they are configured to allow SSH access.

You can also query by the entity type instead of its class. The following query will get you the same result - unless you also have SSH Keys you've added from other integrations (non-AWS) or from the UI / API.

```
Find aws_key_pair
```

Now expand the search a little bit with the following:

```
Find Host as h
  that uses AccessKey with usage='ssh' as k
  return
    h.tag.AccountName,
    h._type,
    h.displayName,
    h.instanceId,
    h.region,
    h.availabilityZone,
```

```

h.publicIpAddress,
h.privateIpAddress,
h.platform,
h.instanceType,
h.state,
k._type,
k.displayName

```

This finds the `Host` entities that `USES` each `AccessKey` and returns a set of specific properties. You can add or remove properties returned as desired.



Note the keyword `that` is what tells the query to traverse the graph to find connections/relationships between entities, followed by a *verb* that represents :) the relationship class.

Also keep in mind you can switch to the **Graph** view to get a more visual result, and continue to drill down interactively.

Again, you can query using the more specific entity types. For example:

```
Find aws_instance that uses aws_key_pair
```

Or mix and match them:

```
Find Host that uses aws_key_pair
```



Note that the relationship keyword/verb is *not* case sensitive.

## 2b - EBS Volume Examples

First, let's see if there are any unencrypted EBS volumes:

```
Find aws_ebs_volume with encrypted != true
```



Note in the above query, the `with` keyword binds to the entity noun immediately to its left, and allows you to filter results on that entity's property values.

If the above query finds some unencrypted EBS volumes, it'll be interesting to see what's using them:

```
Find Host that uses aws_ebs_volume with encrypted != true
```

You can view the `aws_ebs_volume` entities and their relationships in the **Graph** mode, and further inspect the properties on each entity node or relationship edge. You can also expand to see more connected entities and relationships.

Are these actively in use? And in production?

```
Find Host with active = true and tag.Production = true
that uses aws_ebs_volume with encrypted != true
```

What subnets are these instances in? Let's also just return a few key properties from the type of entities related in this search:

```
Find Network as n
  that has Host as h
  that uses aws_ebs_volume with encrypted != true and tag.Production = true as e return
  n.displayName, h._type, h.displayName, e.displayName, e.encrypted
```

OK. How about any EBS Volumes *not* actively in use? Perhaps some of them can be removed...

```
Find aws_ebs_volume that !uses Host
```

You may notice the above query feels backwards. That's okay. The query will work the same way regardless of the direction of relationship. Because the query by default returns all properties from the initial set of entities, it is sometimes easier to reverse the query direction so that you get the data you're looking for more easily.

Technically, `Find Host that !uses aws_ebs_volume as v return v.*` may feel more correct, but it is definitely a bit more to type out.

## 2c - Unencrypted Data

There are many types of data stores you may have in AWS. For example, **EBS Volumes**, **S3 Buckets**, **RDS Clusters and Instances**, **DynamoDB Tables**, **Redshift Clusters**, to name a few. You likely want them to be encrypted if they store confidential data.

How do you find out if that's the case?

```
Find (aws_s3_bucket|aws_rds_cluster|aws_db_instance|aws_dynamodb_table|aws_redshift_
cluster) with encrypted!=true
```

The above query will certainly do the job, but it's quite complicated. This is where the abstract class labeling automatically assigned by Attack Surface Management serves its purpose. Querying by class makes it a whole lot simpler:

```
Find DataStore with encrypted != true
```

Now, you can start adding a few property filters to make the results much more focused, to help cut down the noise or to prioritize remediation. For example:

```
Find DataStore with
  encrypted != true and
  tag.Production = true and
  (classification = 'confidential' or classification = 'restricted')
```

## 2d - Tagging Resources

As you can see from some of the earlier examples, tagging resources can be very useful operationally. That's why we highly recommend tagging your resources at the source. These tags will be ingested by Attack Surface Management and you can use them in your custom queries.

By default, the packaged queries provided by Attack Surface Management, as seen in the **Query Library** and used in the **Compliance** app, rely on the following tags:

- Classification
- Owner
- PII or PHI or PCI (boolean tags to indicate data type)
- AccountName
- Production

All custom tags ingested by Attack Surface Management integrations are prefixed with `tag.<TagName>`. They need to be used as such in the query.

The `Classification` and `Owner` tags are automatically captured as properties so they can be used directly in the query without the `tag.` prefix - in all lower case:

```
classification = '...' or owner= '...'
```

The `tag.AccountName (string)` and `tag.Production (boolean)` tags can be added by Attack Surface Management as part of the Advanced Options in each integration configuration.

## 2e - Network Resources and Configurations

You may have a number of questions to ask or confirm about your network resources and their configurations. Here are a few examples.

Let's start with finding a few network resources and their connections:

```
Find (Gateway|Firewall) with category='network'
  that relates to *
  return tree
```

Keep in mind you can toggle the result back to **Table** view if you'd like.

How about networks and subnets?

```
Find Network that contains Network return tree
```

Or resources in a VPC:

```
Find Network that has (Host|Cluster|Database) return tree
```

The result looks like this (you may have a lot more going on than what's shown here from the demo environment):

QUERY Find Network that HAS (Host|Cluster|Database) return tree

subnet-ddcb48f1 (172.31.80.0/20)

aws\_subnet (Network)

Properties Tags Metadata

subnetId  
subnet-ddcb48f1

internal  
true

webLink  
<https://console.aws.amazon.com/vpc/home?reg>

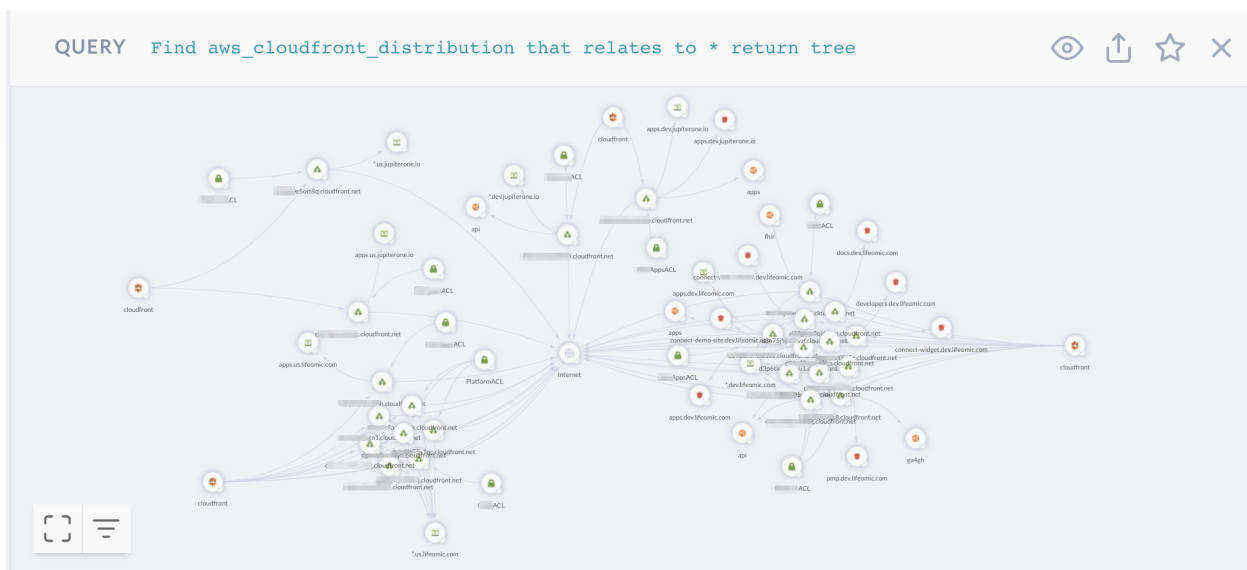


**i** Note that the properties panel for the selected `aws_subnet` has a `webLink` that will allow you to quickly get to the source directly in the AWS web console.

In AWS, you most likely have set up **CloudFront distributions** to distribute traffic to your API Gateways or static websites hosted in S3. What does that look like?

```
Find aws_cloudfront_distribution that relates to * return tree
```

Here, the result looks a little busier, from an account with multiple AWS integration configurations and quite a few `aws_cloudfront_distribution` entities and relationships.



This graph shows you the **origins** connected to the **distributions**: both **S3 buckets** (for static website/contents) and **API Gateways**. Additionally, the graph shows you the **ACM Certificate** being used by them and the **WAF ACL**, if any, configured to protect them.

Keep in mind you can select any entity node in the graph to inspect its detailed properties, or find a web link to quick get to the source in AWS web console.

If you use **AWS Transfer for SFTP**, you can find the **Transfer Servers, Users**, which **IAM Roles** are assigned to them, and which **S3 Buckets** the users have access to.

```
Find aws_account
that HAS aws_transfer
that HAS Host
that HAS User
that RELATES TO *
```

```
return tree
```

You'll get a visual that looks like this:



## 2f - Serverless Functions

Are you using serverless (lambda functions)? If you are, here are a few things that may help you see how they are set up.

Let's start with a listing of your lambda functions:

```
Find aws_lambda_function
```

Simple. Now, what triggers each function?

```
find aws_lambda_function as function
  that TRIGGERS * as trigger
  return
    trigger._type, trigger.displayName, trigger.arn, trigger.webLink,
    function.functionName, function.arn, function.webLink
```

Are there lambda functions with access to resources in a VPC?

```
Find aws_lambda_function that has aws_vpc return tree
```

The above query will give you a visual graph of the lambda functions and the VPC they are configured to run inside.

It is actually a best practice to **not** run lambda functions without access to a VPC unless they need direct access to resources within one -- for example, EC2 instances, RDS databases, or ElasticSearch/ElastiCache.

### Is inbound SSH allowed directly from an external host or network?

```
Find Firewall as fw
  that ALLOWS as rule (Host|Network)
    with internal=false or internal=undefined as src
  where rule.ingress=true and (rule.fromPort<=22 and rule.toPort>=22)
  return
    fw._type,
    fw.displayName,
    rule.fromPort,
    rule.toPort,
    src.displayName,
    src.ipAddress,
    src.CIDR
```



Notice the above query uses `where` to filter the property values of the relationship. You can use both `with` and `where` to filter property values of entities. See the full J1QL documentation for more details. Also keep in mind you can toggle to **Graph** View to see the above results more visually and interactively.

### What production resources are directly connected/exposed to the Internet/everyone?

```
Find (Internet|Everyone)
  that relates to *
  with tag.Production=true and _class!='Firewall' and _class!='Gateway'
  return tree
```

### What are my network layer resources?

```
Find (Firewall | Gateway) with category='network'
```

### What about Security Group protection?

```
Find aws_security_group that PROTECTS aws_instance return tree
```

**TIP** Selecting an edge in the graph to see the security group rule details (the properties on the edge, for example)

## Part 3 – User and Access Analysis

Once you have an Okta or OneLogin integration configured, try some of these example queries yourself.

### 3a – IdP Users and Access

**TIP** Examples in this section require an identity provider integration (Okta or OneLogin).

#### Are there system accounts that do not belong to an individual employee/user?

```
Find User that !is Person
```

User entities in Attack Surface Management are automatically mapped to a corresponding Person (`_type: 'employee'`) entity, when there is at least one Identity Provider (IdP) integration configuration – such as Okta or OneLogin.

**TIP** Set the `userType` property of the user profile in your IdP account to `'system'` or `'generic'` or `'bot'` will prevent Attack Surface Management from creating a Person entity for that user.

**TIP** Set the `username` of your `aws_iam_user` or other non-IdP users to be the email address of a Person / employee will allow Attack Surface Management to automatically map that User to its corresponding Person. Alternatively, you can add an `email` tag to your `aws_iam_user` for the mapping to work.

#### Which active user accounts do not have multi-factor authentication enabled?

```
Find User with active = true and mfaEnabled != true
that !(ASSIGNED|USES|HAS) mfa_device
```

Depending on the specific IdP integration, a `User` entity may have a relationship mapping to an `mfa_device` instead of the `mfaEnabled` flag directly as a property.

Therefore, the above query finds all `User` entities with the `active` flag but not the `mfaEnabled` flag set to true on its properties, and additionally, checks for the existence of an relationship between that `User` and any `mfa_device` assigned or in use.

### Are there users accessing my 'AWS' application without using MFA?

```
Find User with active = true and mfaEnabled != true
  that ASSIGNED Application with displayName = 'Amazon Web Services'
```

Replace the string value of the `displayName` to check for another application.

You can also use `shortName = 'aws'`, which will check for all AWS application instances, if you have more than one AWS SAML app configured with your IdP.

### Find all contractors and external users in the environment.

```
Find User that IS Person that !EMPLOYS Root
```

The above query finds user accounts belong to any individual not directly employed by your organization (`Root` entity).

```
Find User as u that IS Person as p
  where u.userType='contractor' or p.employeeType='contractor'
```

The above query finds contractor users.

## 3b - Cloud users and Access



Examples in this section require at least one AWS integration configuration.

### Who has been assigned full Administrator access in AWS?

```
find (aws_iam_role|aws_iam_user|aws_iam_group)
  that ASSIGNED AccessPolicy with policyName='AdministratorAccess'
```

### Which IAM roles are assigned which IAM policies?

```
find aws_iam_role as role
```

```

that ASSIGNED AccessPolicy as policy
return
  role._type as RoleType,
  role.roleName as RoleName,
  policy._type as PolicyType,
  policy.policyName as PolicyName

```

### 3c – Combined Users and Access Across All Environments



Examples in this section work best when there are both IdP and AWS integration configurations enabled in Attack Surface Management.

#### Who has access to what systems/resources?

```

Find (User|Person) as u
  that (ASSIGNED|TRUSTS|HAS|OWNS)
    (Application|AccessPolicy|AccessRole|Account|Device|Host) as a
return
  u.displayName, u._type, u.username, u.email,
  a._type, a.displayName, a.tag.AccountName
order by u.displayName

```

## Part 4 – Cross Account Analysis



Many examples in this section require both Okta and AWS integration configurations in Attack Surface Management, as well as an AWS SAML app configured in your Okta account. Some queries work best when you have multiple AWS configurations.

#### Who has access to my AWS accounts via single sign on (SSO)?


```

Find User as U
  that ASSIGNED Application as App
  that CONNECTS aws_account as AWS
return
  U.displayName as User,
  App.tag.AccountName as IdP,
  App.displayName as ssoApplication,
  App.signOnMode as signOnMode,
  AWS.name as awsAccount

```


#### Are there assume role trusts from one AWS account to other external entities?

```
Find aws_account
  that HAS aws_iam
  that HAS aws_iam_role
  that TRUSTS (Account|AccessRole|User|UserGroup) with _source='system-mapper'
return tree
```

Note from the above query, `_source='system-mapper'` is an indicator that the trusted entity is not one ingested by an integration configuration, rather,  mapped and created by Attack Surface Management during the analysis of Assume Role policies of the IAM roles in your account(s). Therefore, these entities are most likely external.

For example, you will most definitely see the Attack Surface Management integration IAM role with a `TRUSTS` relationship to the Attack Surface Management AWS account.

## Part 5 - Endpoint Compliance

 Examples in this section require the activation of at least one Endpoint Compliance Agent - powered by Stethoscope app.

### Do I have local firewall enabled on end-user devices?

```
Find HostAgent as agent
  that MONITORS user_endpoint as device
return
  device.displayName,
  device.platform,
  device.osVersion,
  device.hardwareModel,
  device.owner,
  agent.firewall,
  agent.compliant,
  agent._type,
  agent.displayName
```

### Whose endpoints are non-compliant?

```
Find Person as person
  that OWNS (Host|Device) as device
  that MONITORS HostAgent with compliant!=true as agent
return
  person.displayName,
  person.email,
  device.displayName,
  device.platform,
```

```
device.osVersion,  
device.hardwareModel,  
device.owner,  
agent.compliant,  
agent._type,  
agent.displayName
```



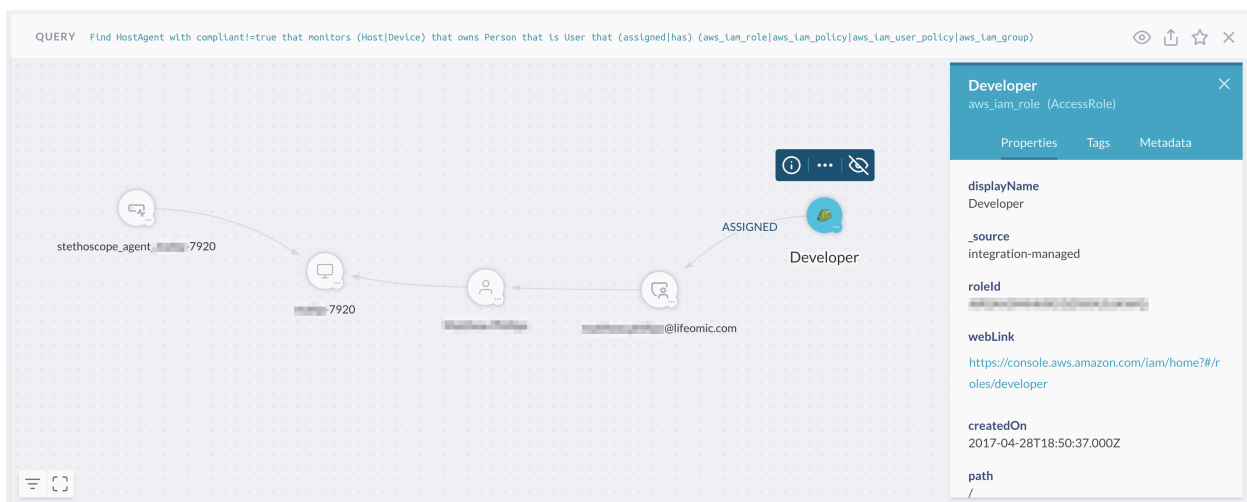
## What applications do those users have access to?

```
Find HostAgent with compliant!=true
  that MONITORS (Host|Device)
  that OWNS Person
  that IS User
  that Assigned Application
return tree
```

## Out of those above, any of them have access to AWS?

```
Find HostAgent with compliant!=true
  that MONITORS (Host|Device)
  that OWNS Person
  that IS User
  that (ASSIGNED|HAS) (aws_iam_role|aws_iam_policy|aws_iam_user_policy|aws_iam_group)
return tree
```

The resulting graph may look like this:



# Using Filters in the Asset Inventory App

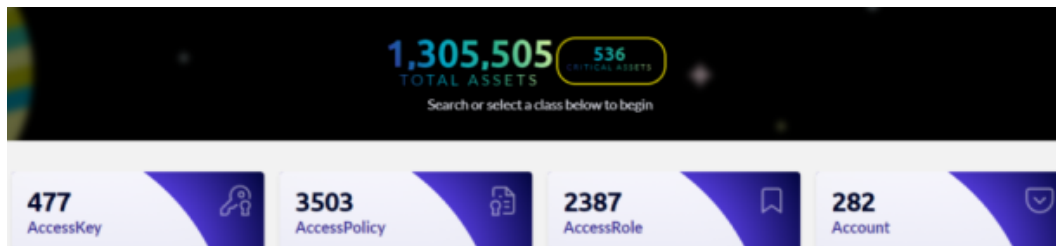
To see all the digital assets (entities) you have, in Attack Surface Management, go to **Apps > Assets**.

There are several ways to filter the large list of entities displayed in the Assets app:


- Quick filter the critical assets
- Additional filters by class and/or type
- Granular filters by properties

## Quick Filter for Critical Assets

Click **Critical Assets** in the top banner to go directly to the most important of your entities.



The Critical Assets feature allows you to create queries and alerts to quickly access the most crucial data. By default, Attack Surface Management determines which criteria defines an asset as the most important and, therefore, the most at risk but an administrator can edit this definition.

Click  to edit the critical asset definition default values. You can use classes, properties, and values to define what is critical.

### Define your critical assets ✕

Use classes, properties, and values to define which assets are critical to your organization.

Assets labeled with any of the classes and matching the property key/value pairs listed below are considered "critical assets". This should be a small subset of all assets that are the most important (of the highest risk) in your organization.

Which asset classes are critical?

AccessKey ✕    AccessRole ✕

Application ✕    CodeRepo ✕

DataStore ✕    Database ✕

Function ✕    Host ✕

Add more classes ✕ +

Which properties and values are critical?

Match  ANY  ALL of the following properties

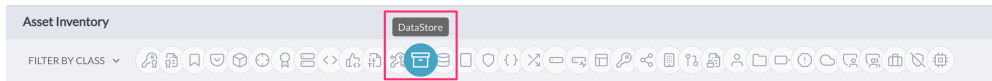
Key*	Value*	
tag.Production	true	✕
		+

CANCEL    UPDATE DEFINITION

Add asset classes and properties that your organization comprises a critical asset, and click **UPDATE DEFINITION**.

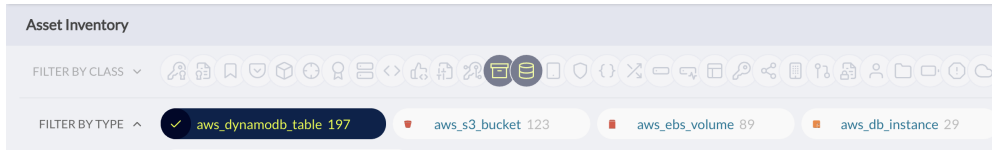
## Quick Filters by Class and/or Type

You can quickly filter your assets by **Class**, by selecting one or more icons that represent each class. The tooltip displays the class label when you move over it.



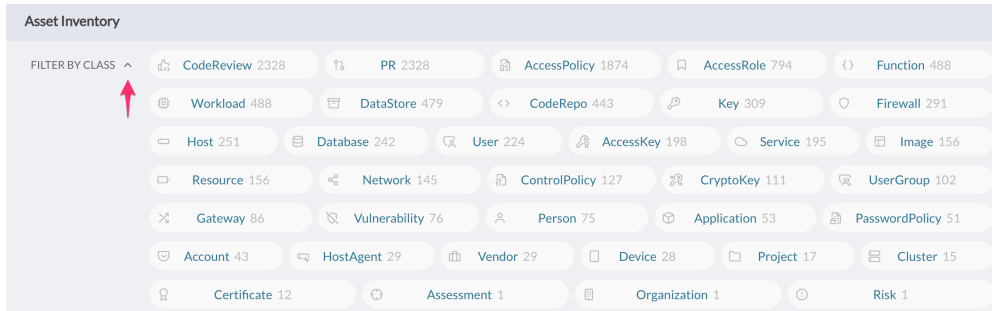
The class of an entity is an abstract label that defines what the entity is within the concept of security operations. For more details, see the Data Model documentation.

After you select one or more classes, you can further filter the entities/assets by **Type**:



**The **Type** of an entity represents the specific type of entity as defined by its source. For more details, see the Data Model documentation.**

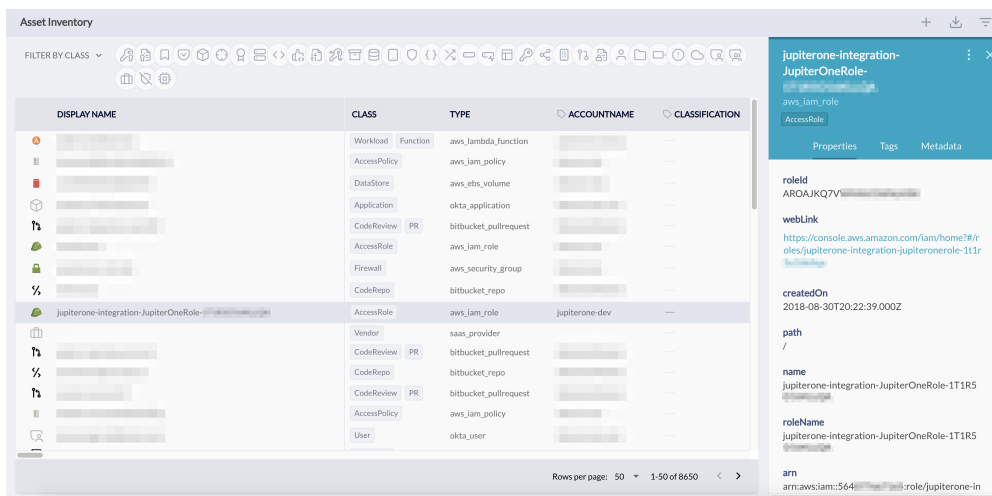
You can also expand the Class filter to get a more detailed, dashboard-like view of the entities/assets with a count for each class.



The data will respond correspondingly to the selection in the table below the quick filters. Note the pagination control at the bottom of the table:



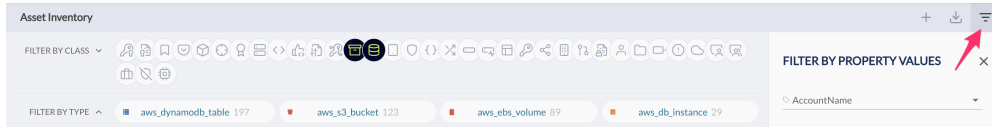
Selecting an entity in the table will bring up its detailed properties in a side panel on the right.



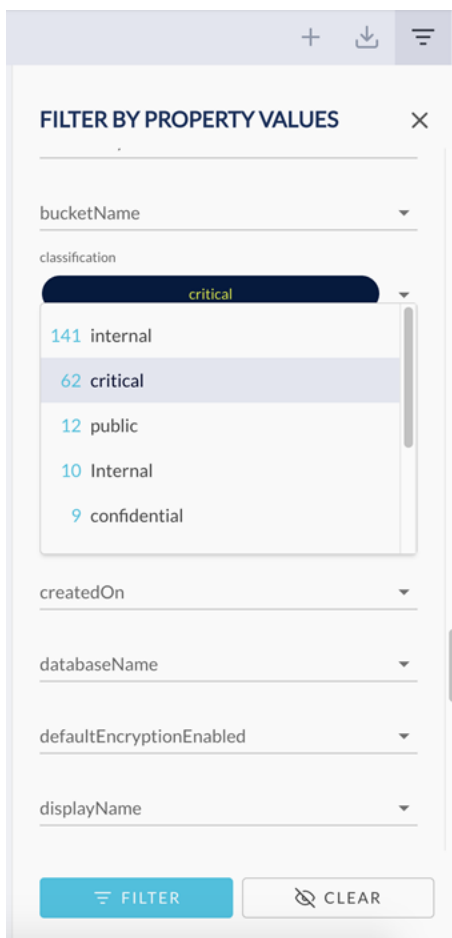
## Granular Filters by Properties

You can apply granular filters by selecting specific property values.

Open up the **Filter Panel** using the control icon near the top right corner:



Look for the property or properties you'd like to filter on to select one or multiple values to apply the filter. Clicking on a previously selected value from the property dropdown box will unselect it.



**Tips:** We recommend selecting Class/Type using the quick filter first, before apply more granular property filters. This will reduce the number of properties/values and make it a lot easier for selection.

# Configuring Alerts

Attack Surface Management allows you to configure alert rules using any J1QL query for continuous auditing and threat monitoring. This is done in the **Alerts** app.

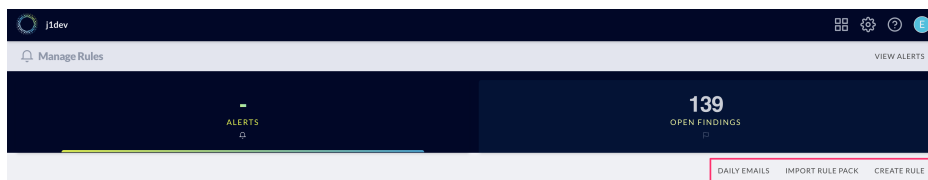
## Import Alert Rules from Rule Pack

You will need to have at least one active alert rule to trigger any alert. The easiest way to add some rules is to import rule packs, following these steps:

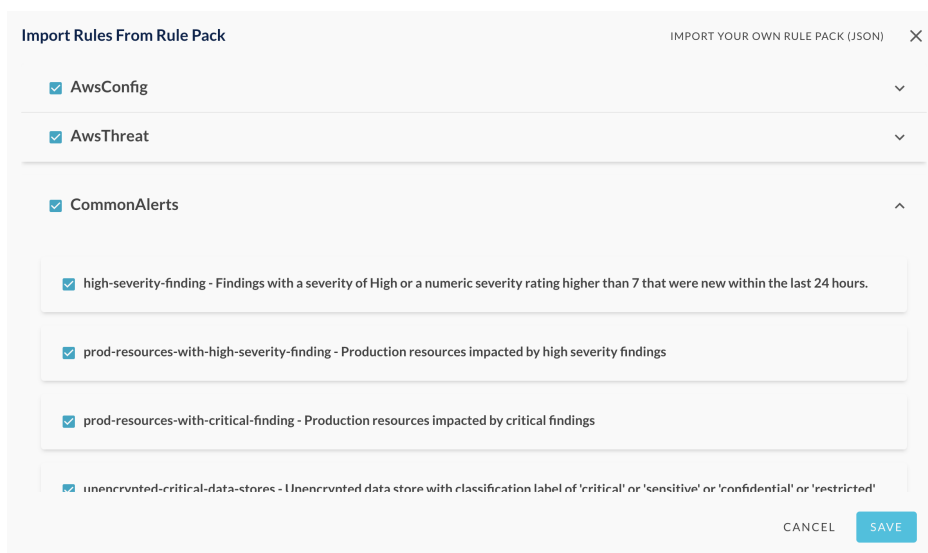
1. Go to **Manage Rules** from the Alerts app.



2. Click **Import Rule Pack** action button.



3. This will bring up the **Import Rules from Rule Pack** modal window, where you can select the rule packs or individual rules within a rule pack. Click **Save** to import the selected rules.



## Creating Custom Alert Rules

To create your own custom alert rule, do the following:

1. Go to **Manage Rules** from the Alerts app
2. Click **Create Rule** action button to bring up the modal window
3. Enter the following details for the custom rule and hit **SAVE**:
  - **Name**
  - **Description**
  - **Severity** (select from drop down list)
  - **Query** (any J1QL query)

The screenshot shows a 'Create Rule' modal window. At the top right, there is a 'SHOW ADVANCED' link and a close button 'X'. The form includes a 'Name' field, a 'Severity' dropdown menu currently set to 'CRITICAL', a 'Description' field, and a 'Query' field. The query text is: 'Find DataStore with classification='critical' and unencrypted=true as d return d.tag,AccountName as Account, d.displayName as UnencryptedDataStores, d.\_type as Type, d.encrypted as Encrypted'. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

The custom rule will be added and be evaluated daily, hourly, or with streaming evaluation for Enterprise customers. If the query you have specified in the rule returns at least one match, it will trigger an alert.

## Setting Up Additional Alert Options

We provide the ability to trigger workflows from alerts. Check the check box on the option you would like to utilize and fill in the required information via the drop downs and fields.

Create Rule
SHOW ADVANCED X

SEND\_EMAIL  
  
 SEND\_SLACK\_MESSAGE  
  
 CREATE\_JIRA\_TICKET

Recipient Email Addresses

New Recipient

Email Body

Affected Items: <br><br> \* [[queries.query0.data|mapProperty('displayName')|join('<br>')] ]

---

Integration Instance

Channels

Slack Message Body

Affected Items: <br><br> \* [[queries.query0.data|mapProperty('displayName')|join('<br>')] ]

---

Summary

Description

{{alertWebLink}}

\*\*Affected Items:\*\*

\* [[queries.query0.data|mapProperty('displayName')|join('

---

Project

---

Issue Type

CANCEL SAVE

### Some alert options require additional integrations/permissions:

1. Slack: You must configure the Slack integration for Attack Surface Management by [following these instructions](#). Be sure to include specify the channel in the format #channel.
2. JIRA: You must configure the JIRA integration for Attack Surface Management by [following these instructions](#).
3. SNS: The AWS Account you wish to send to must be configured as an AWS Integration, and the Attack Surface Management IAM Role for the AWS Account you want to publish to must have the `SNS:Publish` permission
4. SQS: The AWS Account you wish to send to must be configured as an AWS Integration, and the Attack Surface Management IAM Role for the AWS Account you want to publish to must have the `SQS:SendMessage` permission

## Managing Alerts

The alert rules are evaluated *daily* by default, or at the custom interval -- *hourly* or *every 30 minutes* -- you have specified for a specific rule.

Active alerts that matched the evaluation criteria of the alert rules will show up in the **Alerts** app in a data grid that looks like this:

TYPE	SEVERITY	ALERT TITLE / MESSAGE	COUNT	LAST ALERTED ON	
Alert	HIGH	s3-bucket-replication-enabled S3 buckets should enable cross-region replication.		05/3/19 7:01 am	DISMISS
Alert	HIGH	config-rule-noncompliant AWS Config rule evaluation found non-compliant resource configurations.		05/3/19 7:01 am	DISMISS
Alert	INFO	acm-certificate-expiration-check ACM certificate set to expire within 30 days.		05/3/19 6:58 am	DISMISS
Alert	HIGH	s3-bucket-versioning-enabled S3 buckets should enable versioning.		05/3/19 6:55 am	DISMISS
Alert	MEDIUM	unclassified-data-stores Data stores without a classification tag assigned		05/3/19 6:48 am	DISMISS
Alert	HIGH	s3-bucket-replication-enabled S3 buckets should enable cross-region replication.		05/2/19 7:01 am	DISMISS



- Click on an individual alert row will expand it to show the alert details.
- Click on the **DISMISS** button to dismiss an alert.

If an alert is not dismissed, you will not receive a follow up alert notification unless there are changes to the query result.

## Configuring Daily Notification Emails

To receive daily notification of new/active alerts, select:

- **Manage Rules**
- **Daily Emails**
- Enter the email addresses of the users or teams in the **Recipients** field

---

# Using the Visual Query Builder

Attack Surface Management Visual Query Builder (VQB) provides a no-code, drag-and-drop, visual interface for building Attack Surface Management queries without needing to learn JupiterOne Query Language (J1QL) syntax.

## Permissions

You must have at least the **accessLanding** and **readGraph** roles to use VQB.

## Prerequisites


Your organization must have already used an integration to import your data into Attack Surface Management.

## Creating Queries Using VQB

To access VQB, from the Ask Anything search bar, click  .

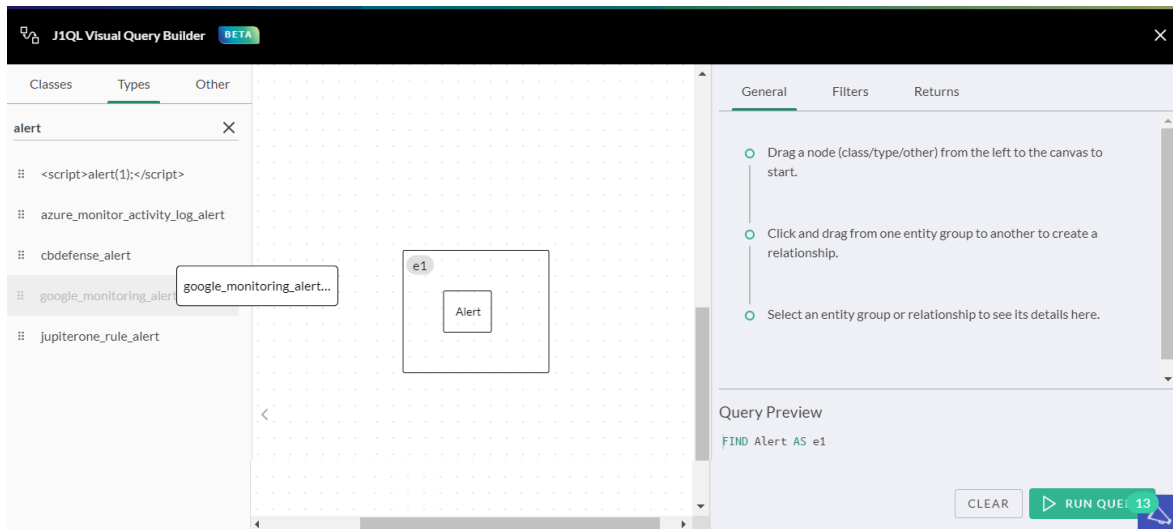
The VQB workspace consists of a:

- **Left pane** containing:
  - Default asset classes
  - J1QL-specific, data model asset types
  - Other entities, including text search and \* wildcard

 You cannot use a text search asset and a wildcard asset in the same query.

- **Center canvas pane**
- **Right information pane** that contains contextual details based on what is on the center canvas, including a query preview window

1. Drag an asset from the left pane to the center canvas to start.



As you place items in the center canvas, a query starts to build in the Query Preview window of the right pane.

If you want your query to do a full-text search of your entire graph on a custom text string, click **Other** in the left pane and drag over the Text asset. You are prompted to enter your custom text. You can only do this step at the very beginning of your query build because the custom text string must be the root asset.

2. Drag over all the Class and Type assets between which you want to build relationships.
  - If you want to group assets, drag assets on top of each other to form a group.
  - Each asset or asset group has an alias identifier in the top-left corner.
  - Click the asset group to see its details in the right information pane.
  - The first asset you drag over or asset group you create becomes the root asset or asset group, unless it is a text asset from the Other menu.
3. Click and drag from one asset or asset group box to another asset or asset group box to create a relationship.
  - Each relationship has an identifier.
  - Click the relationship identifier to see its details in the right information pane.
  - Relationship classes default to all available verbs. J1QL shows all the verbs that you can apply to that relationship. You can toggle between relationship classes to determine how you want the query to search.



The verb RELATES TO covers any and all relationship verbs. However, it is highly recommended to use specific relationship verbs for faster query performance.

- Continue to drag over assets and create groups and the relationships between them.

## Using Wildcards

There is a wildcard asset that you can use to represent a relationship with anything. Click **\*(any entity)** and drag it to the canvas. Then, in the right pane deselect the relationships that you do not want in your query. If you use the wildcard asset, it overrides all child assets in an asset group of which the wildcard asset is a member.

For example, in the following example, the text asset Security is the root asset. The query does a full text search on all alerts and all Attack Surface Management alerts assets with the word security in any of the properties. The wildcard asset lists all possible relationships, from which you can toggle on or off the options listed in the right pane according to the information you are seeking.

Classes Types Other

Use the entity below to search by a text value. Drag it onto the canvas to be prompted to type in your search text.

Text Node

A text node can only be added to the root node group, and there may only be one node present in that group.

Use the entity below to search for any entity.

\* (any entity)

If this entity is added to a node group, it will override all other child nodes in that group, and create a search with the wildcard character (\*).

General Filters Returns

HAS Select one or more specific relationship verbs from this list

ALLOWS

DENIES

RELATES TO Alternatively, RELATES TO covers any and all relationship verbs. It is highly

Query Preview

```

FIND "Security" AS e1
  THAT RELATES TO AS r1 (Alert | jupiterone_rule_alert)
  AS e2

```

**Note:** When connecting relationships between assets or asset groups, you must start with the root asset (such as e1) and drag it to the related asset (such as e4), and then connect the other assets as required.

## Filtering

You can create filters that are based on all the properties you have for the selected asset on the canvas. You can also apply AND OR conditions to the filters.

General Filters Returns

Filters for the e1 group of entities

Property name Operator

Attribute value OR undefined

ADD

e1.level = warning X

OR e1\_createdOn <= undefined X

AND

Query Preview

```

FIND Alert
  WITH ( level = "warning"
        OR _createdOn <= undefined ) AS e1

```

# Managing Security Findings

Attack Surface Management provides a centralized repository and dashboard to let you easily manage security findings from different sources, including:

- AWS Inspector findings
- AWS GuardDuty findings
- Veracode static and dynamic analysis findings
- WhiteHat application security findings
- Tenable Cloud scanning findings
- HackerOne report findings
- CVEs and other vulnerability findings
- Manual penetration testing findings (imported via API)



More vulnerability scanner integrations are being added. Current roadmap includes: Rapid7, Qualys, Bugcrowd, White Source, Source Clear, and Snyc.

## View Findings

Consolidated findings can be accessed in the **Alerts** app, under the **Findings** tab. The header tab shows a total count of currently open findings. Selecting it will bring you to the detailed findings view:

Type	Severity	Finding Title / Description	Created On
aws_inspector_finding	CRITICAL	Instance i-0244e47d8e801312d is not compliant with rule 4.1... Description Monitor changes to file permissions, attributes, ownership and group. The parameters in this section track changes for system calls that affect file permissions and attributes. The chmod , fchmod and fchmodat system calls affect the permissions associated with a file. The chown , fchown , fchownat and lchown system calls affect owner and group attributes on a file. The setxattr , lsetxattr , fsetxattr (set extended file attributes) and removexattr , lremovexattr , fremovexattr (remove extended file attributes) control extended file attributes. In all cases, an audit record will only be written for non-system user ids (audit >= 1000) and will ignore Daemon events (audit = 4294967295). All audit records will be tagged with the identifier "perm_mod." Rationale Monitoring for changes in file attributes could alert a system administrator to activity that could indicate intruder activity or policy violation.	04/11/19 7:53 pm
aws_inspector_finding	CRITICAL	Instance i-0244e47d8e801312d is not compliant with rule 4.2... Description Log files stored in /var/log/ contain logged information from many services on the system, or on log hosts others as well. Rationale It is important to ensure that log files have the correct permissions to ensure that sensitive data is archived and protected.	04/11/19 7:53 pm
aws_guardduty_finding	MEDIUM	Outbound portscan from EC2 instance i-031f61bc80ca33ceb. EC2 instance i-031f61bc80ca33ceb is performing outbound port scans against remote host 10.51.17.149.	04/20/19 2:25 am
aws_guardduty_finding	MEDIUM	Outbound portscan from EC2 instance i-09a81dbcd8d99ebb2. EC2 instance i-09a81dbcd8d99ebb2 is performing outbound port scans against remote host 10.51.12.142.	04/1/19 12:28 pm



Attack Surface Management will automatically map resources impacted by or related to each finding based on the available attributes from the finding source.

Selecting a finding from the list will show you a graph of those relationships. This allows you to visualize the context to further analyze the finding's impact and to determine a course of action for remediation.

TYPE	SEVERITY	FINDING TITLE / DESCRIPTION	CREATED ON
aws_inspector_finding	CRITICAL	Instance i-0244e47d8e801312d is not compliant with rule S.2... Description The PermitUserEnvironment option allows users to present environment options to the ssh daemon. Rationale Permitting users the ability to set environment variables through the SSH daemon could potentially allow users to bypass security controls (e.g. setting an execution path that has ssh executing trojan/d programs)	04/11/19 9:54 pm
<p>QUERY Find Finding with _key="208ksu00(LuLbSF6R55HT0B2V2h4zsqvR8sv8uQ1E+" that relates to * return tree</p>			
aws_inspector_finding	INFO	On instance i-0244e47d8e801312d, TCP port 9200 which is associat... On this instance, TCP port 9200, which is associated with Elasticsearch, is reachable from the internet with no process listening. The instance i-0244e47d8e801312d is located in VPC vpc-436e713a and has an attached ENI eni-012758c020f651db which uses network ACL acl-9a3a9967. The port is reachable from the internet through Security Group sg-008200348cb9961b2 and IGW igw-baf680dc	04/11/19 9:54 pm

## Create Alerts for Findings

You can create custom alert rules to notify you on certain findings, using J1QL to filter and correlate.

### Examples:

The following three rules are included in the **Common Alerts** Rule Pack:

- high-severity-finding**

Alerts on Findings with a severity of High or a numeric severity rating higher than 7 that were new within the last 24 hours.

```
Find Finding with
(severity='High' or severity='high' or numericSeverity>7) and
_createdOn > date.now-24hours
```

- prod-resources-with-high-severity-finding**

Alerts when Production resources are impacted by high severity findings.

```
Find (Host|DataStore|Application|CodeRepo|Account|Service|Network)
```

```
with tag.Production=true
that has Finding with severity=('High' or 'high') or numericSeverity=(7 or 8)
```

- **prod-resources-with-critical-finding**

Alerts when Production resources are impacted by critical findings.

```
Find (Host|DataStore|Application|CodeRepo|Account|Service|Network)
  with tag.Production=true
  that has Finding with severity=('Critical' or 'critical') or numericSeverity=(9
or 10)
```

The following rule is included in the **AWS Threat** Rule Pack:

- **aws-guardduty-inspector-finding-instance-correlation**

Identifies vulnerable EC2 instances (with medium or higher rated open Inspector finding) that are also targets of suspicious activities (such as medium or higher rated open GuardDuty finding).

```
Find aws_guardduty_finding with numericSeverity>5 and open=true as guardduty
  that relates to aws_instance as i
  that has aws_inspector_finding with numericSeverity>5 and open=true as inspector
return i.*, guardduty.*, inspector.*
```

## Generate Views of Findings with J1QL Query and Graph

You can execute J1QL queries to generate graph visualizations that help you analyze the relationships among findings, the agents/scanners/services that identified them, and the resources they impact.

Here's an example:

```
Find cve that relates to (Host|HostAgent) with active=true return
tree
```

This will give you a visual like this (you may need to move the nodes around to adjust their positioning):





# Managing Policies and Procedures for Attack Surface Management

Attack Surface Management provides a **Policies** app that allows users to generate and manage corporate security policies and procedures. It has the following capabilities:

- Generating policies and procedures from templates
- Managing policies and procedures online via the webapp
- Mapping controls/procedures to compliance requirements
- Policy Builder CLI

## Generating Policies and Procedures from Templates

The Policies app provides a set of over 120 policy and procedure templates to help your organization build your security program and operations from scratch. These templates are derived from our own internal policies and procedures, and have been through several rounds of compliance assessments.

To get started, simply navigate to the **Policies** app, fill in the following three sections of information in the web form:

- Company information
- Key personnel information (such as your Security and Privacy Officer)
- Security and DevOps tooling information

It may take a few minutes for the policy and procedure documents to be generated for the first time.

## Variables

Note that the Markdown text contains both global and local variables -- in this format: `{{variableName}}`. It is best not to edit the variables in the templates since they would be auto-replaced by the relevant text.

A **Procedure** document may contain an optional local `{{provider}}` variable. This allows you to configure the control provider that implements or has been designated the responsibility to fulfill that procedure. For example, the provider for "Single Sign On" could be "Okta", "OneLogin", "JumpCloud", "Google", etc. This `provider` value can be entered near the top of the document editor when it is open, right below the Document Title.

The procedure editor also presents you a short summary guidance description. Additionally, you may toggle the "Adopted" flag on or off depending on your readiness to adopt a particular procedure.

## Versioning

Edits to policies and procedure documents are automatically versioned upon save. The `{{defaultRevision}}` variable will be populated with the date the document was last edited.

Currently the web app does not have a UI to view previous versions of documents.

## Download/ Export Policy and Procedure Documents

The "Export / Download Zip" button at the upper right corner of the screen will generate a zip file containing the following three sets of files:

- templates in Markdown format
- final policies and procedures in Markdown format
- final policies and procedures in HTML format

## Policy Builder CLI

Attack Surface Management provides an offline CLI that allows you to manage your policies and procedures offline (for example, as code in a git repo), and publish to your Attack Surface Management account as needed.





## Using Your Own Existing Policies

The Policies app is an optional component of the platform. It is not a prerequisite for the rest of the platform. The Compliance app is the only app that depends on it for proper mapping to compliance framework requirements and controls.

You are not required to use Attack Surface Management provided policy/procedure templates. If your organization already has written documents for security policies/procedures and you would like to take advantage of Compliance app and its mapping capabilities, you can transform your existing policies and publish them to Attack Surface Management.

# Inviting Users to Your Attack Surface Management Account/Org

Adding others to your Attack Surface Management account/organization is done via a simple invite process. To send an invite, follow these steps:

1. Go to  **Settings**, select  **Users & Access**.
2. A modal screen should pop up showing the current User Groups. You can add a new group by click on the **Add Group** button and give the new group a `name` and optionally a `description`.
3. Click on the  **Add User** button, enter the user's email address, and click  **Send Invitation**.
4. The user should receive an invitation email to join your account/organization and be prompted to accept the invite upon login. New users will be prompted to create their Attack Surface Management user account.

- You need to be a member of the **Administrators** group to be able to invite other users.

- If you are an **Enterprise** customer and use SAML SSO, see the instructions [\[here\]\(./configure-sso-integration.md\)](#).

---

# Using Attack Surface Management APIs

The following information provides some of the basics required to get started using Attack Surface Management APIs.

## Endpoints

The Attack Surface Management platform exposes a number of public GraphQL endpoints.

The base url is `https://api.us.securecloudinsights.cisco.com`, with the following endpoints:


- **query and graph operations:** `/graphql`
- **alert and rules operations:** `:/rules/graphql`

## Authentication

The Attack Surface Management APIs use a Bearer Token to authenticate. Make sure you include the API Key in the header as a Bearer Token, and make sure to include the JupiterOne-Account as a header parameter.

## API Key

To generate the API Key:

1. Go to **Settings > Account Management**.
2. In the left panel, click the  **(Key)** icon.
3. In the User API Keys page, click **Add**.
4. In the API Keys modal, type the name of the key and the number of days before it expires, then click **Create**.
5. Copy or save the API Key, since it is only available once, then click **DONE**.

## Account ID

You can find the Account ID (the JupiterOne-Account value) for your account by running the following J1QL query:

```
FIND jupiterone_account as a return a._accountId
```

Here's an example of a cURL command with authentication:

```
curl --location --request POST
'https://api.us.securecloudinsights.cisco.com/graphql' \
--header 'JupiterOne-Account: accountId' \
--header 'Authorization: Bearer 123456abcdef' \
--header 'Content-Type: application/json' \
--data-raw '{"query":...}'
```

# Contacting Support

If you need technical support, please do one of the following:

- Contact your local Cisco Partner
- Contact Cisco Support
- To open a case by web: <http://www.cisco.com/c/en/us/support/index.html>
- To open a case by email: [tac@cisco.com](mailto:tac@cisco.com)
- For phone support: 1-800-553-2447 (U.S.)
- For worldwide support numbers:  
<https://www.cisco.com/c/en/us/support/web/tsd-cisco-worldwide-contacts.html>
- For Secure Cloud Analytics Free Trial customers, open a case by email: [swatchc-support@cisco.com](mailto:swatchc-support@cisco.com)

---

## Change History

<b>Document Version</b>	<b>Published Date</b>	<b>Description</b>
1_0	November 9, 2021	Initial Version.
1_1	September 20, 2022	Added Using Secure Cloud Insights APIs section.
1_2	May 24, 2023	Re-brand Cisco Secure Cloud Insights to Cisco Attack Surface Management.



---

# Copyright Information

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

