Vectra AI

SaaS API Guide

REST API v3.2

VECTRA®

Revision History

DATE	COMMENT
September 2023	• Add ordering query parameter for the field “event_timestamp” for events endpoints • Adds the patch query parameter “membership_action” for Groups
March 2023	• Refresh token is now in a valid Bearer token format. Adds enhanced documentatiosn and instruction regarding refresh tokens. • Rate limit is now 4 requests per second, 10 request burst per tenant • Enhanced error handling around `/oauth2/token`
Nov 2022	Initial Release • Adds ability to view account groups via `/api/v3.2/groups` endpoint • Adds ability to view a single account group via `/api/v3.2/accounts/<group_id>` endpoint

API Version 3.2 Changelog

Groups

- Adds ability to view account groups via `/api/v3.2/groups` endpoint
- Adds ability to view a single account group via `/api/v3.2/groups/<group_id>` endpoint

Overview

A REST API is available for administrators and developers to integrate Vectra's breach detection data into their applications. Vectra RESTful API provides access to security event data, platform configuration, and health information via URI paths.

Vectra REST API is based on open standards. You can use any web development language to access and retrieve information via the API. A common use-case would be to retrieve security event information generated by the Vectra platform and supply this information to a security operations dashboard or incident response and ticketing systems.

The REST API can be accessed via HTTPS connection to the interface IP address of the Vectra brain. The data in the response to the API query is in JSON format.

Examples of security event data that can be integrated into your application:

- Security event type detected
- Account information associated with the security event
- Severity of the Account activities

The Vectra REST API is accessible using OAuth2 authentication.

Security Detection Data

The "Detections" and "Accounts" elements retrieve security events that can be inserted into external applications. The REST API provides filtering options to extract data. Advanced parsing of the data can be performed after data has been retrieved and saved into your target application. Order of the response data returned is latest first.

Accessing REST API v3.2

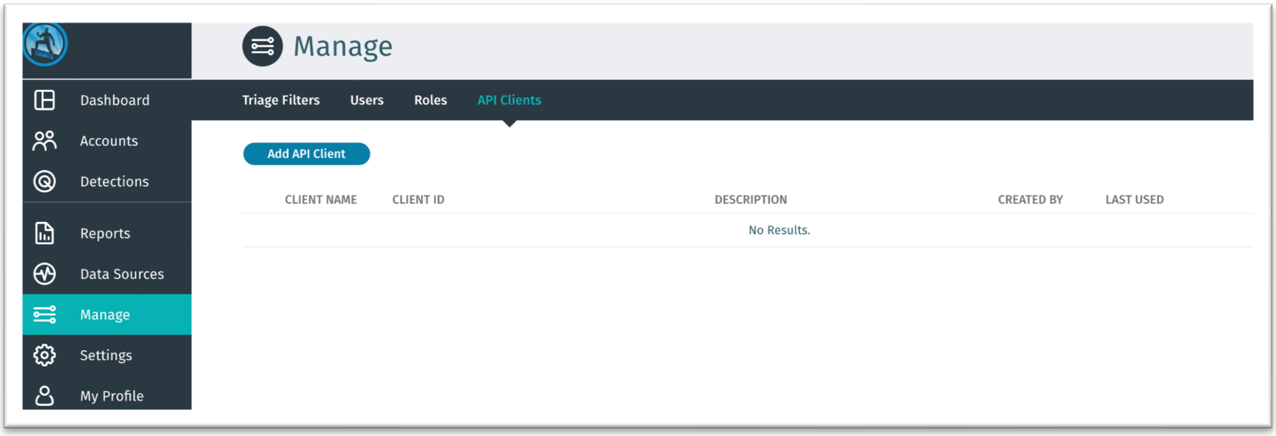
The REST API v 3.2 is accessed via the URL `https://<vectra_portal_url>/api/v3.2/<path>.`

The `<path>` options for REST API queries are listed in the table below.

URL	METHOD	DATA TO QUERY
/detections	GET, PATCH	All detection events
/detections/<detection_id>	GET	Single detection event details
/detections/<detection_id>/notes	GET, POST, PATCH, DELETE	Detection notes
/accounts	GET	All accounts
/accounts/<account_id>	GET	Single account details
/accounts/<account_id>/notes	GET, POST, PATCH, DELETE	Account notes
/rules	GET, POST, PUT, DELETE	Triage rule configuration
/tagging/detection/<detection_id>	GET, PATCH	Detection tags
/tagging/account/<account_id>	GET, PATCH	Account tags
/assignments	GET, POST, PUT, DELETE	Account assignments
/assignment_outcomes	GET, POST, PUT, DELETE	Assignment outcomes
/events/account_scoring	GET	Account scoring events
/events/account_detection	GET	Account detection events
/events/audits	GET	Audit log events
/entities	GET	All entities
/entities/<entity_id>	GET	Single entity details
/events/entity_scoring	GET	Entity scoring events
/groups	GET, POST	Account groups
/groups/<group_id>	GET, PATCH, DELETE	Account groups

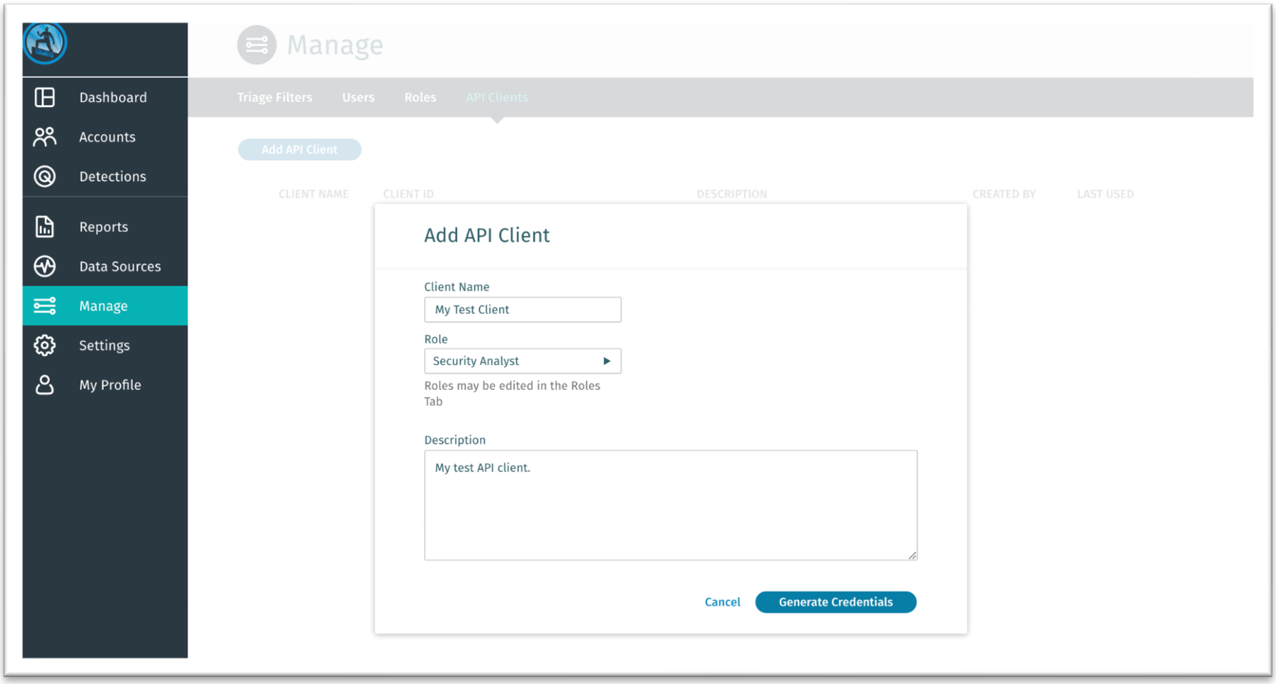
API Clients

Getting access to the SaaS API is done through the creation of an API Client. Creation of an API Client will provide a set of OAuth 2.0 credentials that will be used to gain authorization to the SaaS API. Please note that management of API Clients is restricted to Detect users with the role of “Super Admin”. To create an API client, log into your Detect portal and navigate to *Manage > API Clients*.



Creating a new API Client

From the API Clients page, select ‘Add API Client’ to create a new client.



Client Name: My Test Client

Role: Security Analyst

Roles may be edited in the Roles Tab

Description: My test API client.

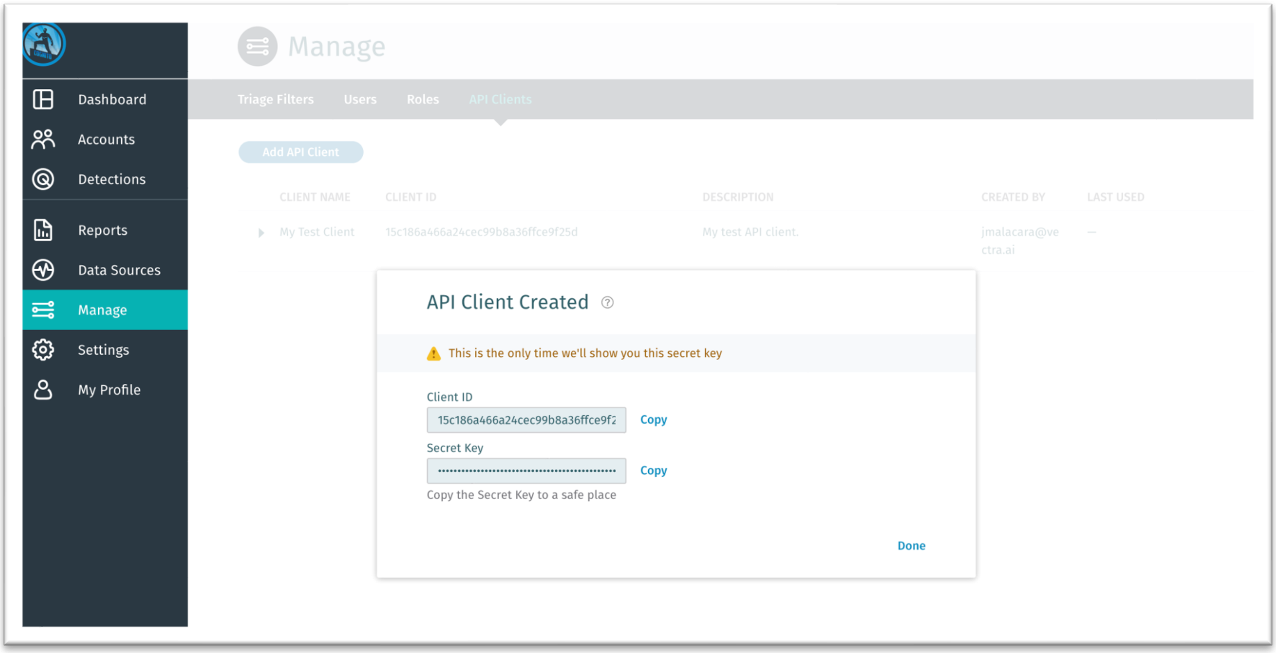
Creating a new API Client has one required parameter:

- **Role** – the role maps the API Client to a set of permissions, similar to the way a Detect UI user would be assigned a role. The role must be one of the following:
  - Read-Only
  - Restricted Admin
  - Security Analyst
  - Settings Admin
  - Auditor

Creating a new API Client has two optional parameters:

- **Name –** a user-friendly name to identify the client (up to 256 characters)
- **Description –** a brief description to aid in identifying the client (up to 2048 characters)

Once you have entered the API Client information, select 'Generate Credentials' to obtain your client credentials.



CLIENT NAME	CLIENT ID	DESCRIPTION	CREATED BY	LAST USED
My Test Client	15c186a466a24cec99b8a36ffce9f25d	My test API client.	jmalacara@vectra.ai	—

API Client Created

⚠ This is the only time we'll show you this secret key

Client ID

15c186a466a24cec99b8a36ffce9f2

Secret Key

••

Copy the Secret Key to a safe place

Be sure to record your **Client ID** and **Secret Key** for safekeeping. You will need these two pieces of information to obtain an access token from the SaaS API. An access token is required to make requests to all of the SaaS API endpoints.

Authenticating as an API Client

Many REST API tools and libraries can programmatically manage access tokens for you once provided the OAuth 2.0 parameters described below, or you can manually make requests to obtain them as needed.

First you must authenticate using the **Client ID** and **Secret Key** you obtained when creating your API Client.

Example Request:

```
POST https://<vectra_portal_url>/oauth2/token
Authorization: Basic <HTTPBasic(<client_id>:<secret_key>)>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
```

Example Response:

```
{
    "access_token": "Z0FBQUFB...",
    "expires_in": 21600,
    "refresh_expires_in": 86400,
    "refresh_token": "eyJzdWIi...",
    "token_type": "Bearer"
}
```

Notice that the response includes an **access_token** and **refresh_token** which will be used to make subsequent API requests as a client, outlined in the next sections.

Also, be sure to note the differences in the **expires_in** and **refresh_expires_in** timers, which are expressed in seconds. An access_token will expire in 6 hours, while a refresh_token will take 24 hours to expire. Once your access_token has expired, you will need to reauthenticate your API client with your client_id and secret_key in order to receive a new access_token. If your refresh_token has not expired, you may use it to obtain a new access token using the previous authorization grant.

Manually Refreshing Token via API Client (e.g. Postman or Insomnia)

A refresh token will have the following syntax when returned as part of the `/oauth2/token` response:

```
“refresh_token”: “eyJzdWIi..."
```

As of March 2023, if you manually attempt to refresh your token via an API Client, copying and pasting this value as is **will** be valid.

Refreshing Token via Script

There should be no formatting changes required for using the refresh token via script. For example, in Python, accessing and saving the refresh token should look like this:

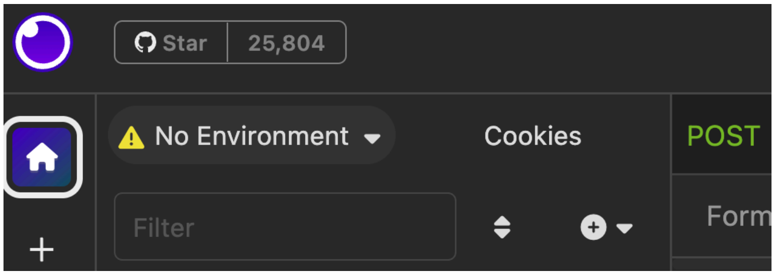
```
refresh_token = response.get(‘refresh_token’)
```

where response is the dictionary received from authenticating via /oauth2/token.

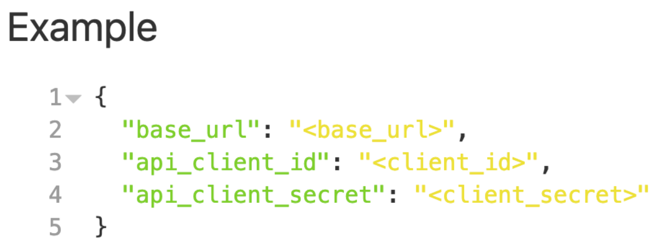
Auto-Refresh Token via Insomnia

If you are using Insomnia for API requests, we recommend the following setup for basic Authentication and refresh token requests:

1. Create a new environment with your API client credentials. Select the “No Environment” dropdown found on the top left corner of Insomnia, choose “Manage Environments”.



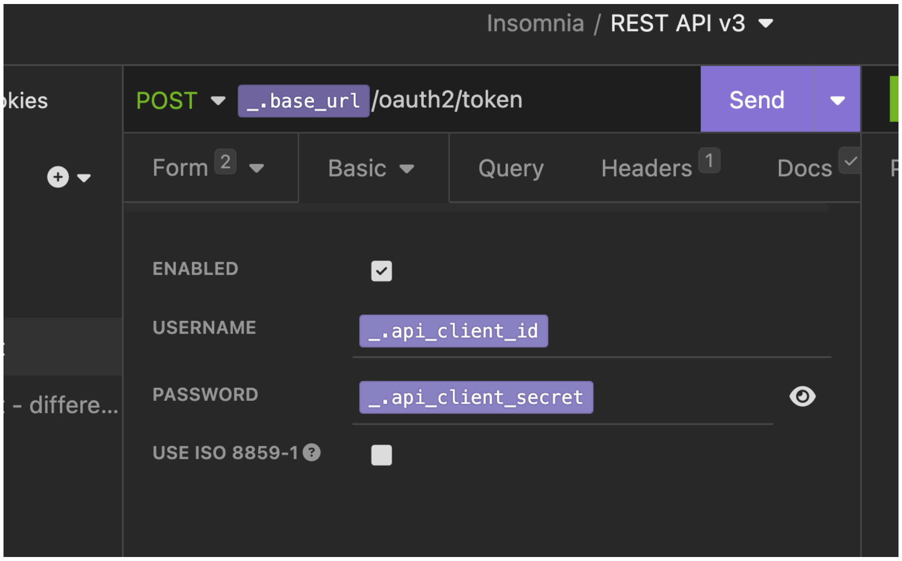
2. Use the example data below to create a new sub environment. Give the environment a unique name.



Example

```
{
  "base_url": "<base_url>",
  "api_client_id": "<client_id>",
  "api_client_secret": "<client_secret>"
}
```

3. Repeat for any other desired tenants.
4. Select your desired environment to start making requests.
5. Create a new request for /oauth2/token & set up basic authentication. Reference the client_id and secret from your environment file via Insomnia variables in the Basic auth tab (example below).



POST _.base_url/oauth2/token

ENABLED ☑

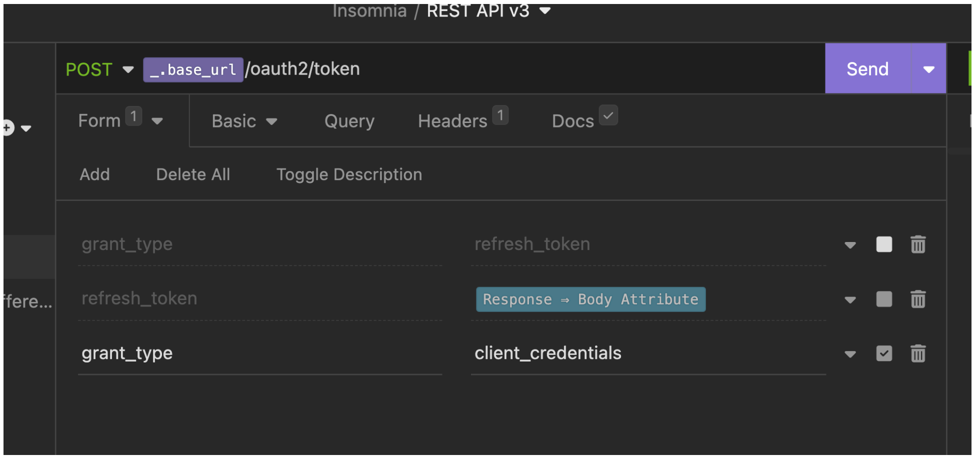
USERNAME _.api_client_id

PASSWORD _.api_client_secret

USE ISO 8859-1 □

Next, we’ll need to populate Form data for both types of requests: initial auth and refresh.

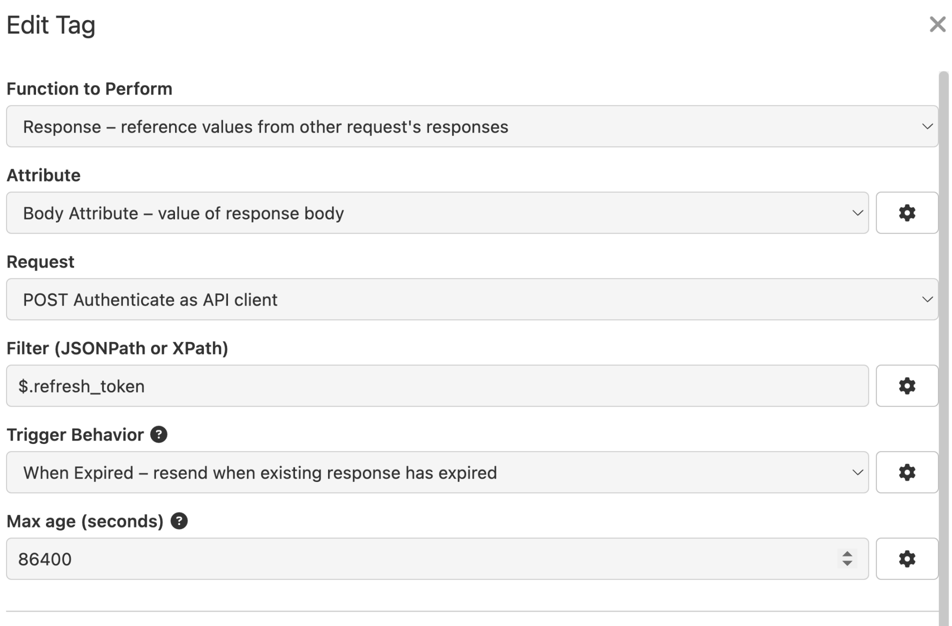
When authenticating initially for basic auth, form data should just include `grant_type: client_credentials` (example below):



POST _.base_url/oauth2/token

grant_type	refresh_token
refresh_token	Response ⇒ Body Attribute
grant_type	client_credentials

When refreshing your token, use `grant_type` and `refresh_token` form data (example below). refresh_token can be populated automatically via an Insomnia tag. To create a tag, enter “Response => Body Attribute” in the refresh_token form data field and Insomnia will auto-populate a configurable interface. Select the interface and input the following (Request should be the /oauth2/token request):



Edit Tag

Function to Perform: Response – reference values from other request's responses

Attribute: Body Attribute – value of response body

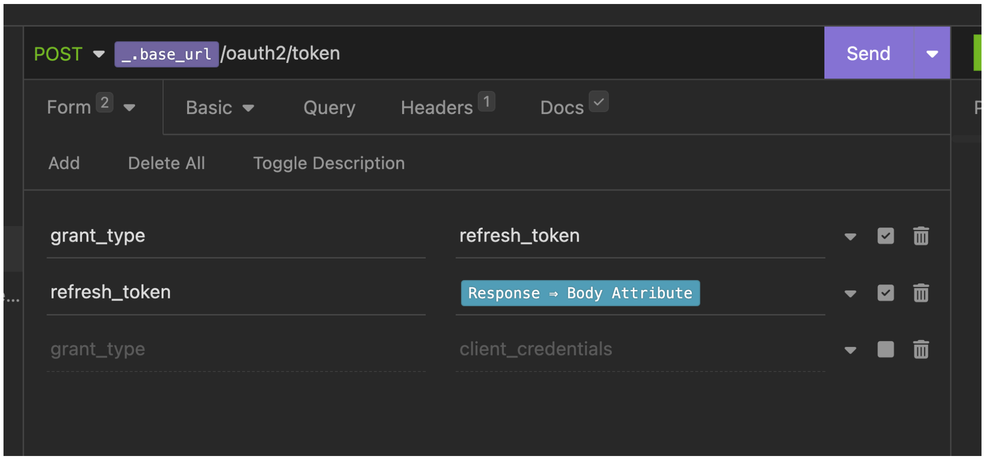
Request: POST Authenticate as API client

Filter (JSONPath or XPath): $.refresh_token

Trigger Behavior: When Expired – resend when existing response has expired

Max age (seconds): 86400

Press “Done”, and your refresh form data should appear as the following:



POST _.base_url/oauth2/token

grant_type refresh_token

refresh_token Response ⇒ Body Attribute

grant_type client_credentials

Insomnia should now be configured to automatically grab and use the refresh_token field returned.

Requests via Postman

If you prefer Postman, we have a public API collection you can import to your environment and use.

Example Request:

```
POST https://<vectra_portal_url>/oauth2/token
Authorization: Basic <HTTPBasic(client_id:secret_key)>
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
refresh_token=<refresh_token>
```

Example Response:

```
{
    "access_token": "Z0FBQUFB...",
    "expires_in": 21600,
    "token_type": "Bearer"
}
```

Make API request as a client

Now that you have obtained an **access_token**, you will use that token to make API requests as an API Client to all the SaaS API endpoints. Here are a couple quick examples.

Example Request to accounts endpoint:

```
GET /api/v3.2/accounts
Authorization: Bearer <access_token>
```

Example Response:

```
{
    "count": 16,
    "next": "http://<vectra_portal_url>/api/v3.2/accounts?page=1",
    "previous": null,
    "results": [
    ...
}
```

Example Request to detections endpoint:

```
GET /api/v3.2/detections
Authorization: Bearer <access_token>
```

Example Response:

```
{
    "count": 25,
    "next": "http://<vectra_portal_url>/api/v3.2/detections?page=2",
    "previous": null,
    "results": [
    ...
}
```

Full sample output for accounts and detections endpoints can be found in Appendix A.

API Rate Limits

Rate limits define the frequency of requests that can be made to the v3.2 API. API rate limits apply on a per-tenant basis.

The following rate limits are enforced across all v3.2 API endpoints and methods:

- Steady-state: 4 requests per second
- Burst: 10 requests per second

When request submissions exceed the steady-state request rate and burst limits, the v3.2 API will throttle requests and return a `429 Too Many Requests` error response.

Detections

Detection objects contain all the information related to security events detected in the environment. The URL to retrieve all detections is `https://<vectra_portal_url>/api/v3.2/detections` and uses OAuth2 authentication.

Detections are grouped into one of the following categories:

DETECTION CATEGORY	URL
Command & Control	/detections?category=command
Botnet	/detections?category=botnet
Reconnaissance	/detections?category=reconnaissance
Lateral Movement	/detections?category=lateral
Exfiltration	/detections?category=exfiltration
Info	/detections?category=info

The API query performs partial word match. For example, you can use `/detections?category=recon` to query all Reconnaissance category detections.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Detections

ELEMENT	DESCRIPTION	TYPE	NOTES
count	The number of object IDs retrieved in the output	integer	
next	URL to the next page of output	string	Useful when using a web-based REST API browser.
previous	URL link to the previous page of output	string	Useful when using a web-based REST API browser.
results	The list of Detections returned in the output	List of Objects	

The following table lists the fields and descriptions present in a Detection. These Detections will be contained inside the ‘results’ field, which is a top-level field described in the table above.

ELEMENT	DESCRIPTION	TYPE	NOTES
id	Object ID	integer	

ELEMENT	DESCRIPTION	TYPE	NOTES
detection	The name of the threat detected.	string	See Appendix B for the list of Detection names
detection_type	The name of the threat detected.	string	See Appendix B for the list of Detection names
category	The category of the vname attack detected.	string	See Appendix B for the list of categories. Will be deprecated in future release. Replaced by detection_category.
detection_category	The category of the vname attack detected.	string	See Appendix B for the list of categories.
src_ip	The source IP address of the host attributed to the security event.	string	
state	The state of the detection.	string	If the detection has aged out the state will transition to "active" from "inactive". If marked as fixed in the UI, or via the V3.2 API, state will be "fixed".
t_score	The threat score attributed to the detection.	integer	Will be deprecated in future release. Replaced by threat.
threat	The threat score attributed to the detection.	integer	
c_score	The certainty score attributed to the detection.	integer	Will be deprecated in future release. Replaced by certainty.
certainty	The certainty score attributed to the detection.	Integer	
first_timestamp	The timestamp when the event was first detected	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
last_timestamp	The timestamp when the event was last detected	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
description	System generated description of the event.	string	

ELEMENT	DESCRIPTION	TYPE	NOTES
proto	Protocol used in the communications.	string	
total_bytes_sent	Total bytes sent by the client.	integer	
total_bytes_rcvd	Total bytes received by the client.	integer	
url	The URL that links directly to this record.	string	
sensor_name	The name of sensor where this flow was detected from.	string	
src_host	A dictionary with fields that describe the Host the detection is from	JSON object	
url	The URL that links directly to the detection record	string	
summary	The summary information for the detection	JSON object	See Appendix A for examples
grouped_details	The detection details for the detection	JSON object	See Appendix A for examples
tags	User defined tags added to the detection	List of strings	
targets_key_asset	Indicates whether the detection targets a key asset	Boolean	Will be deprecated in future release. Use is_targeting_key_asset.
campaign_summaries	The summaries of campaigns of which this detection is part.	JSON object.	The summaries of campaigns this detection belongs to
is_targeting_key_asset	Indicates whether the detection targets a key asset	Boolean	

ELEMENT	DESCRIPTION	TYPE	NOTES
note	User defined note for this detection.	string	
note_modified_by	Username who last modified note	string	
note_modifed_timestamp	The timestamp when note was last modified	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
notes	An array of all notes on the host.	array of objects	Includes id, date_created, date_modified, created_by, modified_by, and note contents
assigned_to	User named assigned to this detection	string	
assigned_date	The timestamp when user was assigned this detection	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
src_account	A dictionary with fields that describe the Account the detection is from	JSON object	

You can also apply filters to the API response to query for specific elements.

The available options and filters for detection set is listed below.

QUERY PARAMETER	DESCRIPTION
fields	Filters results by fields provided. The available fields are listed in the table above.
page	Page number. Possible values are a positive integer or last
page_size	Page size. Possible values are a positive integer, up to 5000.
ordering	Orders records by last timestamp, threat score and certainty score. The default sorts threat and certainty score in ascending order. Scores can be sorted in descending order by prepending the query with “minus” symbol.
min_id	>= the id provided
max_id	<= the id provided
state	Filter by state: active, inactive, ignored, ignored for all

QUERY PARAMETER	DESCRIPTION
category	Filter by the detection category
detection_type	Filter by the name of the threat detected.
detection_category	Filter by the detection category
src_ip	Filter by source (ip address)
t_score	Filter by threat score
t_score_gte	Filter by threat score >= the score provided
threat_score	Filter by threat score
threat_gte	Filter by threat score >= the score provided
c_score	Filter by certainty score
c_score_gte	Filter by certainty score >= the score provided
certainty	Filter by certainty score
certainty_gte	Filter by certainty score >= the score provided
last_timestamp	Filter by last timestamp
host_id	Filter by id of the host object a detection is attributed to
tags	Filter by a tag or a comma-separated list of tags
destination	Filter by destination in the detection detail set
proto	Filter by the protocol in the detection detail set
is_targeting_key_asset	Filter by is_targeting_key_asset: True or False
note_modified_timestamp_gte	Filter by note_modified_timestamp >= the timestamp provided: '2019-08-27T20:55:29Z'
src_account	Filter by source account ID
id	Filter by detection IDs (comma separated list of IDs)

Examples of detection queries:

QUERY	COMMENT
/api/v3.2/detections/?ordering=t_score	Retrieves all detections with threat score sorted low to high (ascending).
/api/v3.2/detections/?ordering=-t_score	Retrieves all detections with threat score sorted high to low (descending).
/api/v3.2/detections/?c_score_gte=60	Retrieves all detections with certainty score greater than or equal to 60.
/api/v3.2/detections/?page=2	Retrieves page 2 of detections.
/api/v3.2/detections/?t_score_gte=90&c_score_gte=90	Retrieves all detections with threat &certainty score greater than or equal to 90

QUERY	COMMENT
/api/v3.2/detections/?tags=RAT	Retrieves all detections with tag value=RAT
/api/v3.2/detections/?tags=RAT,TOR	Retrieves all detections with tag value=RAT, and TOR

Mark/Unmark Detections as Fixed

Marking a detection as Fixed indicates that some remedial action was taken based upon the detection. Threat and Certainty scores will both be 0 once a detection has been marked as fixed.

Marking/unmarking a detection as fixed requires the following elements be present in the request body:

- detectionIdList
- mark_as_fixed

URL to mark/unmark a detection as fixed is `https://<vectra_portal_url>/api/v3.2/detections.`

An example of using the PATCH method to mark or unmark a detection as fixed can be found in Appendix A.

Accounts

Account objects contain information related to accounts observed in the environment. The URL to retrieve all accounts is `https://<vectra_portal_url>/api/v3.2/accounts` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Accounts

ELEMENT	DESCRIPTION	TYPE	NOTES
count	The number of object IDs retrieved in the output	integer	
next	URL to the next page of output	string	Useful when using a web-based REST API browser.
previous	URL to the previous page of output	string	Useful when using a web-based REST API browser.
results	A list of all Accounts (described in the table below) being returned by the query	list	

The following table lists the fields and descriptions present in an Account. These Accounts will be contained inside the ‘results’ field, which is a top-level field described in the table above.

ELEMENT	DESCRIPTION	TYPE	NOTES
id	The ID of the Account	integer	
url	A v3.2 URL to the Account	string	Useful when using a web-based REST API browser.
name	The name associated with the Account	string	
state	The state of the Account	string	
threat	The threat score attributed to the account	integer	
certainty	The certainty score attributed to the account	Integer	
severity	The severity of this Account	string	Either ‘Low’, ‘Medium’, ‘High’, or ‘Critical’. This is calculated based off of the threat and certainty of the Account
account_type	The method through which this account was discovered	List of strings	Can be one or both of “kerberos” or “o365”
tags	User defined tags added to the account	List of strings	
sensors	Sensors associated with the accounts detection set	List of strings	
note	User defined note added to the account	string	
note_modified_by	Username who last modified note	string	
note_modified_timestamp	The timestamp when note was last modified	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
notes	An array of all notes on the host.	array of objects	Includes id, date_created, date_modified, created_by, modified_by, and note contents
privilege_level	A number 1-10 to represent how privileged this account is	int	
privilege_category	A string to represent how privileged this account it	string	Either ‘Low’, ‘Medium’, or ‘High’. Privilege levels of 1 and 2 map to ‘Low’. Privilege levels of 3-7 map to ‘Medium’. Privilege levels of 8-10 map to ‘High’

ELEMENT	DESCRIPTION	TYPE	NOTES
last_detection_timestamp	Last detection activity from this Account	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
detection_set	List of Detections for Account	string	
detection_summaries	The summaries of detections attached to this Account.	List of Objects	
ldap	Information about the LDAP server this Account came from (if applicable)	JSON object	

The available options and filters for account set is listed below.

QUERY PARAMETER	DESCRIPTION
fields	Filters results by fields provided. The available fields are listed in the table above.
page	Page number. Possible values are a positive integer or last
page_size	Page size. Possible values are a positive integer up to 5000.
ordering	Orders records by last timestamp, threat score and certainty score. The default out sorts threat and certainty score in ascending order. Scores can be sorted in descending order by prepending the query with “minus” symbol.
name	filter by name
state	filter by state: active or inactive
t_score	filter by threat score
t_score_gte	filter by threat score >= the score provided
c_score	filter by certainty score
c_score_gte	filter by certainty score >= the score provided
tags	filter by a tag or a comma-separated list of tags (returns hosts that contain any of the tags specified), e.g.tags=baz | tags=foo,bar"
all	No filter, return all host objects. Only available in version 2.0 API
min_id	Filter hosts have id greater than or equal to min_id
max_id	Filter hosts have id less than or equal to max_id
note_modified_timestamp_gte	filter by note_modified_timestamp >= the timestamp provided: ‘2019-08-27T20:55:29Z’
privilege_level	filter by exact privilege level of hosts. 1-10
privilege_level_gte	filter hosts that have a privilege level greater than or equal to the supplied number. 1-10

QUERY PARAMETER	DESCRIPTION
privilege_category	filter hosts by privilege category. Options are ‘low’, ‘medium’ and ‘high’
id	Filter by account IDs (comma separated list of IDs)

Tagging

The tagging API can be used to manage tags for accounts and detections.

To manage tags for an account, use the following URL:

`https://<vectra_portal_url>/api/v3.2/tagging/account/<account_id>`

To manage tags for a detection, use the following URL:

`https://<vectra_portal_url>/api/v3.2/tagging/detection/<detection_id>`

The following table lists the fields and a description of the various elements for tagging operations. Detailed examples of using GET and PATCH methods on tags is described in Appendix A.

ELEMENT	DESCRIPTION	TYPE	NOTES
status	Status of tagging request.	string	Valid values: ‘success’ or ‘failure’
tag_id	The id of the tag	integer	Corresponds to the account or detection id of the object being tagged.
tags	List of tags for the host or detection object.	list of strings	When doing a PATCH operation, the object will be updated to match the list of tags provided. To remove tags for a host or detection set tags to an empty list.
message	Error message if operation was unsuccessful.	string	Only present when status is ‘failure’ of operation is a failure.
invalid_tags	List of tags which are invalid for the account or detection object.	list of strings	Only present when status is ‘failure’. This will only be returned in the response for a PATCH operation.

Triage

Version 3 of triage API supports GET, POST, PUT, and DELETE. The API endpoint for accessing triage rules is `https://<vectra_portal_url>/api/v3.2/rules`

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3 API for Triage rules.

ELEMENT	DESCRIPTION	TYPE	NOTES
count	The number of object IDs retrieved in the output	integer	
next	URL to the next page of output	string	Useful when using a web-based REST API browser.
previous	URL link to the previous page of output	string	Useful when using a web-based REST API browser.
results	A list of all Triage rules (described in the table below) being returned by the query	list	

The following table lists the fields and descriptions present in a Triage rule. These Triage rules will be contained inside the ‘results’ field, which is a top-level field described in the table above.

ELEMENT	DESCRIPTION	TYPE	NOTES
id	The ID of the Triage rule	integer	
url	A v3 URL to the Triage rule	string	Useful when using a web-based REST API browser.
enabled	Describes if the Triage rule is enabled	Boolean	
created_timestamp	Describes the time the Triage rule was created	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
last_timestamp	The timestamp when this Triage filter was triggered	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
is_whitelist	This whitelists all detections for this activity	Boolean	True or False
priority	Used in ordering execution of Triage filters	integer	
active_detections	The total number of active detections this Triage rule applies to	integer	
total_detections	The total number of detections (active or inactive) this Triage rule applies to	integer	
template	Specifies if this Triage rule was created based off of a template	Boolean	True of False
detection_category	Original detection category	string	
triage_category	Custom Triage label used to categorize specified detections	string	
detection	Original detection type	string	
source_conditions	Specifies the entity this Triage rule applies to	JSON	source_conditions represents the conditions that can be applied to

			the source of a detection, including host, account, IP, and sensor. For more information on the format of source_conditions, see below this table
additional_conditions	Specifies additional matching criteria for this Triage rule	JSON	additional_conditions are other conditions that are different on a per-detection-type basis. For more information on the format of additional_conditions, see below this table

Formatting source_conditions and additional_conditions

The format of version 2.2 Triage rules is based on AND/OR logic, making them more flexible and customizable to your specific situation. Both ‘source_conditions’ and ‘additional_conditions’ are now saved as JSON blobs which represents a tree-like structure. Each tree node is represented in the following format:

```
{operator: operand}
```

Where ”operator” is any of the following:

For non-leaf nodes: `‘AND’ or ‘OR’`
For leaf nodes: `‘ANY_OF’ or ‘NONE_OF’`

And “operand” is any of the following:

For non-leaf nodes: A list of children tree nodes of the form `[{operator: operand}, {operator: operand}, …]`

For leaf nodes: A dictionary of the form:

```
{
        ‘field’: <FIELD NAME>,
        ‘values’: [
                {‘value’: [<VALUE>]}
        ],
        ‘group’: [
                {‘value’: [<GROUP ID>]}
        ]
}
```

Stipulations on the format of the top-level tree structure as of 2.2 is that the top-level operator must be an ‘OR’ node, with a single ‘AND’ node as the only child. The ‘AND’ node may have an arbitrary number of leaf node children. All valid 2.2 Triage rules will look as follows:

```
{
‘OR’: [
{ ‘AND’: [
<LEAF NODE 1>,
<LEAF NODE 2>,
…
<LEAF NODE N>,
            ]}
      ]
}
```

source_conditions and additional_conditions have different fields that can be used in their leaf nodes. For source_conditions, the following fields are valid: ip, host, account, sensor. The fields ‘ip’ and ‘host’ also support triaging on groups, meaning that an ip or a host leaf node can be given an IP Group or Host Group ID, and the Triage rule will apply to every IP/host in the specified group.

For additional_conditions, the following fields are valid: remote1_ip, remote1_proto, remote1_port, remote1_dns, remote2_ip, remote2_proto, remote2_port, remote2_dns, account, named_pipe, uuid, identity, share, extensions, rdp client name, rdp client token, keyboard name. The fields ‘remote1_ip’, ‘remote2_ip’, ‘remote1_dns’, ‘remote2_dns’ support triaging on groups.

Either source_conditions or additional_conditions may be null. Setting source_conditions to null implies that this Triage rule with apply to All Hosts.

To see examples of complete valid source_conditions and additional_conditions, see Appendix A.

Assignments

Assignments are used to assign account objects to analysts for investigation. This information includes but is not limited to:

- Assigned to user
- Assigned by user
- Date assigned
- Date resolved
- Events
- Outcome
- Account ID
- Triaged detections

URL to retrieve assignment information is
`https://<vectra_portal_url>/api/v3.2/assignments`

An example of using curl to retrieve all assignments using token authentication:

```
curl 'https://<vectra_portal_url>/api/v3.2/assignments/' \
-X 'GET' \
-H "Content-Type: application/json" \
--header "Authorization: Bearer <access_token>” \
--compressed \
--insecure
```

Assignment Resolution

Once completed, assignments can be resolved. Assignment resolutions provide a way to label and track the outcomes of assignments. Outcomes get recorded in the assignment’s history.

Outcome choices may include one of the following built-in assignment outcomes:

- Benign True Positive
- Malicious True Positive
- False Positive

Alternatively, users can choose to define their own Custom outcomes for reporting purposes.

URL to resolve an assignment is
`https://<vectra_portal_url>/api/v3.2/assignments/<id>/resolve`

An example of using curl to retrieve all assignments using token authentication:

```
curl 'https://<vectra_portal_url>/api/v3.2/assignments/9/resolve' \
-X 'GET' \
-H "Content-Type: application/json" \
--header "Authorization: Bearer <access_token> " \
--data '{"outcome": 2, "note": "some note", "triage_as": "my triage rule",
"detection_ids": [23, 73, 85, 88]}' \
--compressed \
--insecure
```

The following table lists fields and description of the various elements for assignments.

ELEMENT	DESCRIPTION	TYPE	NOTES
count	Number of object IDs retrieved in the output	integer	
next	URL to the next page of output	string	
previous	URL link to the previous page of output	string	

results	List of users returned in the output	list of objects

The following table lists the fields and descriptions present in an assignment. These elements will be contained inside the 'results' field, which is a top-level field described in the table above.

ELEMENT	DESCRIPTION	TYPE	NOTES
id	Object ID	integer	
assigned_by	The user that made the assignment	list of objects	
date_assigned	Timestamp from when the assignment was assigned	datetime	
date_resolved	Timestamp from when the assignment was resolved	datetime	
events	Actions take on the assignment	list of objects	
outcome	Outcome of the assignment	list of objects	
resolved_by	The user that resolved the assignment	list of objects	
triaged_detections	The number of detections that were triaged as a part of the resolution of this assignment	integer	
account_id	The ID of the account associated with this assignment	integer	
assigned_to	The user this assignment has been assigned to	list of objects	

The available options and filters for assignments are listed below.

QUERY PARAMETER	DESCRIPTION
accounts	Filter by accounts
assignees	Filter by assignees
resolution	Filter by resolution (outcome)
resolved	Filters by resolved status. True or False.
created_after	Filter by created after timestamp.

Examples of assignments queries

QUERY	COMMENT
`https://<vectra_portal_url>/api/v3.2/assignments?accounts=1,2,3`	Retrieves all assignments on accounts 1, 2 and 3
`https://<vectra_portal_url>/api/v3.2/assignments?assignees=1,2,3`	Retrieves all assignments for users 1, 2 and 3
`https://<vectra_portal_url>/api/v3.2/assignments?resolution=1,2,3`	Retrieves all assignments with resolution 1, 2 or 3
`https://<vectra_portal_url>/api/v3.2/assignments?resolved=true`	Retrieves all assignments that have been resolved

`https://<vectra_portal_url>/api/v3.2/assignments?created_after=2021-01-01T00:00:00Z`	Retrieves all assignments that have been created since 2021-01-01T00:00:00Z

Event-based endpoints

Event-based endpoints provide a mechanism for API clients to poll Detect for system events. These endpoints provide a SaaS analog to syslog on the Detect appliance, where data is aggregated at an external collector for storage and analysis, such as a SIEM.

Event-based endpoints include the following event types:

- Account Scoring Events
- Account Detection Events
- Audit Log Events
- Entity Scoring Events

These endpoints make use of a checkpoint value, which is used in subsequent requests to get new events beginning from the last checkpoint.

Account Scoring Events

Account Scoring Events are generated when an account score is changed, which occurs upon initial threat detection, discovery of additional detections, and updates to any discovered detections. The account score is reduced over time if the underlying detection behavior subsides, either because of user intervention or because the account has left the environment.

The URL to retrieve accounts scoring events is `https://<vectra_portal_url>/api/v3.2/events/account_scoring` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Account Scoring Events.

ELEMENT	DESCRIPTION	TYPE	NOTES
events	A list of account scoring events	list	Event fields are described in the table below
remaining_count	The number of remaining events	integer	
next_checkpoint	The next checkpoint value to use to retrieve any remaining events	integer	Example: "next_checkpoint": 101 Use with the ‘from’ query parameter described below

The following table lists the fields and descriptions present in an Account Scoring Event.

Element	Description	Type	Notes
id	The ID of the Account Scoring Event	integer	
version	The version	string	
account_id	The account ID	integer	
account_uid	The account UID	string	
threat	The threat score attributed to the account	integer	
certainty	The certainty score attributed to the account	integer	
severity	The severity attributed to the account	integer	
score_decrease	Indicates whether both Threat and Certainty scores are decreasing	boolean	
href	URL link to see the account in the UI	string	
category	Event category	string	
last_detection_href	The URL of the last detection	string	
last_detection_type	The type of the last detection	string	
active_detection_types	A list of active detection types	list	
event_timestamp	Timestamp when the Account Scoring Event occurred	string	

The available query parameters to filter for Account Scoring Events are listed below.

QUERY PARAMETER	DESCRIPTION
from	Used to provide a checkpoint value for filtering Account Scoring Events
limit	Used to specify the batch size returned by the response from the /events/account_scoring endpoint. By default 500 events will be returned if a limit is not specified. A maximum of 1,000 events will be returned per request.
ordering	Orders events by “event_timestamp”. The default sorts event timestamps by ascending order. Events can be sorted in descending order by prepending the query with “minus” symbol.

The URL to retrieve accounts scoring events from the last checkpoint is `https://<vectra_portal_url>/api/v3.2/events/account_scoring?from=101`

Account Detection Events

Account Detection Events are generated upon initial detection and for each update of the detection.

The URL to retrieve accounts scoring events is `https://<vectra_portal_url>/api/v3.2/events/account_detection` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Account Detection Events.

ELEMENT	DESCRIPTION	TYPE	NOTES
events	A list of account scoring events	list	Event fields are described in the table below
remaining_count	The number of remaining events	integer	
next_checkpoint	The next checkpoint value to use to retrieve any remaining events	integer	Example: "next_checkpoint": 101 Use with the ‘from’ query parameter described below

The following table lists the fields and descriptions present in an Account Detection Event.

Element	Description	Type	Notes
id	The ID of the Account Detection Event	integer	
detection_id	The ID of the detection	integer	
category	The detection category	string	
threat	The threat score attributed to the detection	integer	
certainty	The certainty score attributed to the detection	integer	
d_type_vname	The detection name	string	
triaged	Indicates whether the detection has been triaged	boolean	

Element	Description	Type	Notes
href	URL link to see the detection in the UI	string	
account_uid	The account UID	string	
certainty	The certainty score attributed to the account	integer	
event_timestamp	Timestamp when the Account Detection Event occurred	string	
src_account	The source account	string	
ip	The IP address	string	
detail	The detection detail	string	Detail fields are specific to detection type
severity	The severity of the detection	integer	

The available query parameters to filter for Account Detection Events are listed below.

QUERY PARAMETER	DESCRIPTION
from	Used to provide a checkpoint value for filtering Account Detection Events
limit	Used to specify the batch size returned by the response from the /events/account_detection endpoint. By default 500 events will be returned if a limit is not specified. A maximum of 1,000 events will be returned per request.
ordering	Orders events by “event_timestamp”. The default sorts event timestamps by ascending order. Events can be sorted in descending order by prepending the query with “minus” symbol.

The URL to retrieve accounts detection events from the last checkpoint is `https://<vectra_portal_url>/api/v3.2/events/account_detection?from=101`

Audit Log Events

Audit Log Events are generated when a user action is performed on the system. Audit Log Event are sequential and can be used to create an audit trail of user activity on your system.

The URL to retrieve audit log events is `https://<vectra_portal_url>/api/v3.2/events/audits` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Audit Log Events.

ELEMENT	DESCRIPTION	TYPE	NOTES
events	A list of audit log events	list	Event fields are described in the table below
remaining_count	The number of remaining events	integer	
next_checkpoint	The next checkpoint value to use to retrieve any remaining events	integer	Example: "next_checkpoint": 101 Use with the ‘from’ query parameter described below

The following table lists the fields and descriptions present in an Audit Log Event.

Element	Description	Type	Notes
id	Autoincrementing ID	bigint	used for checkpoints
user_id	User ID of the user account associated with the event.	integer	
username	Username of the account associated with the event, at the time of the event	string	
user_type	User type, e.g. “LOCAL”, “SAML”, “API_CLIENT”	string / enum	
api_client_id	API client ID, if an event was caused by an API client interaction	string	
role	Role the user/API client had at the time of the event	string	
version	Vectra UI version at the time of the event	string	
source_ip	IP address of the user/API client	string	
event_timestamp	Event timestamp (UTC) in ISO-8601 format	timestamp	Example: 2022-03-01T13:00:00Z
message	Message describing the event.	string	
result_status	"success" or "failure"	string / enum	
event_data	JSON data specific to the event type	string / JSON	For example, which tags were added/removed for a tagging event
event_object	The object type the audited action is being performed on	string / enum	For example user, filter, or account

Element	Description	Type	Notes
event_action	What type of action is being audited	string / enum	For example, creation or deletion

The available query parameters to filter for Audit Log Events are listed below.

QUERY PARAMETER	DESCRIPTION
event_timestamp_gte	Start date/time for audit, inclusive, formatted in ISO-8601
event_timestamp_lte	End date/time for audit, inclusive, formatted in ISO-8601
from	Used to provide a checkpoint value for filtering Account Scoring Events
user_id	Audit events associated with a particular user_id
event_object	Audit events on a particular object (e.g. account, user, etc.)
event_action	Audit events regarding a particular action (e.g. delete, create, etc.)
limit	Used to specify the batch size returned by the response from the /events/audits endpoint. By default 500 events will be returned if a limit is not specified. A maximum of 1,000 events will be returned per request.
ordering	Orders events by “event_timestamp”. The default sorts event timestamps by ascending order. Events can be sorted in descending order by prepending the query with “minus” symbol.

The URL to retrieve accounts scoring events from the last checkpoint is `https://<vectra_portal_url>/api/v3.2/events/audits?from=101`

Entities

Entities is an overarching term to cover various types of models, including Accounts and Hosts. Entities are returned with an entity type, indicating what type of entity it is, along with urgency, importance, and fields from the original model.

The URL to retrieve entities is `https://<vectra_portal_url>/api/v3.2/entities` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Entities.

ELEMENT	DESCRIPTION	TYPE	NOTES
results	A list of entity objects	list	Will return only Accounts
next	URL to the next page of entity results	string	
previous	URL to the previous page of entity results	string	
count	Total number of entity objects	integer	

The following table lists the fields and descriptions present in an Entities object.

Element	Description	Type	Notes
id	Autoincrementing ID.	integer	Entity ID
breadth_contrib	Breadth contribution of the entity.	integer	
entity_importance	Importance score of the entity.	integer	
entity_type	Entity type, e.g. “Account”, “Host”.	string	
is_prioritized	Whether or not the priority of this entity is above the configured priority threshold.	boolean	
severity	Entity severity, e.g. “Low”, “Medium”, “High”.	string	
urgency_reason	Reason behind the urgency_score.	string	
urgency_score	Priority or urgency of the entity.	integer	
velocity_contrib	Velocity contribution of the entity.	integer	
detection_set	List of detections for this entity.	list	
last_detection_timestamp	Last detection activity from this Entity.	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
name	The name associated with the account, or the learned hostname.	string	
notes	A list of all notes on the entity.	list	
privilege_level	An integer from 1-10 to represent how privileges this account is.	string	
privilege_category	A string to represent how privileged this account is, related to privilege level. Either Low (1-2), Medium (3-7), or High (8-10).	string	
sensors	Sensors related to the entity.	list	

Element	Description	Type	Notes
state	State of the entity, e.g. “active” or “inactive”.	list	
tags	User defined tags added to the entity.	list	
url	The URL link directly to this entity.	string	

The available query parameters to filter for Entities are listed below.

QUERY PARAMETER	DESCRIPTION
is_prioritized	If it is set (present), only entities whose priority score is above the configured priority threshold will be included in the response.
entity_type	Allows the caller to request only entities of the given type. Valid values are "account" or "host". Multiple values can be provided using commas to separate values (e.g. entity_type=account,host). Required when querying single entity.
ordering	Option to order the records by last timestamp and/or priority score. By default the results are sorted by urgency score in descending order. Multi-ordering is supported by sending a comma-separated list of fields (e.g. ordering=urgency_score,-name)
last_detection_timestamp_gte	Return only the entities which have a last detection timestamp equal to, inclusive, formatted in ISO-8601
name	Filter by entity name.
note_modified_timestamp_gte	Filter by note_modified_timestamp, inclusive, formatted in ISO-8601
page	Page number. Possible values are a positive integer or last
page_size	Page size. Possible values are a positive integer up to 5000.
state	Filter on entity activation state. Valid values are "active" or "inactive" .
tags	Filter by a tag or a comma-separated list of tags (returns entities that contain any of the tags specified), e.g. tags=baz or tags=foo,bar

Entity Scoring Events

Entity Scoring Events are generated when an entity score is changed, which occurs upon initial threat detection, discovery of additional detections, and updates to any discovered detections. The entity score is reduced over time if the underlying detection behavior subsides, either because of user intervention or because the account has left the environment.

The URL to retrieve entity scoring events is `https://<vectra_portal_url>/api/v3.2/events/entity_scoring` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Entity Scoring Events.

ELEMENT	DESCRIPTION	TYPE	NOTES
events	A list of entity scoring events	list	Event fields are described in the table below
remaining_count	The number of remaining events	integer	
next_checkpoint	The next checkpoint value to use to retrieve any remaining events	integer	Example: "next_checkpoint": 101 Use with the 'from' query parameter described below

The following table lists the fields and descriptions present in an Entity Scoring object.

Element	Description	Type	Notes
entity_id	Entity ID.	integer	
breadth_contrib	Breadth contribution of the entity.	integer	
entity_importance	Importance score of the entity.	integer	
entity_type	Entity type, e.g. "Account", "Host".	string	
is_prioritized	Whether or not the priority of this entity is above the configured priority threshold.	boolean	
severity	Entity severity, e.g. "Low", "Medium", "High".	string	
urgency_reason	Reason behind the urgency_score.	string	
urgency_score	Priority or urgency of the entity.	integer	
velocity_contrib	Velocity contribution of the entity.	integer	
last_detection_url	URL of the last detection reported on the entity.	string	
event_timestamp	Timestamp when the detection event occurred.	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT

Element	Description	Type	Notes
name	The name associated with the account, or the learned hostname.	string	
active_detection_types	A list of all active detection types on the entity.	list	
last_detection_type	The type of the last detection.	string	
category	The event category (e.g. "ACCOUNT_SCORING").	string	
last_detection_id	The ID of the last detection associated with this event.	integer	
url	The URL link directly to this entity.	string	

The available query parameters to filter for Entity Scoring events are listed below.

QUERY PARAMETER	DESCRIPTION
include_score_decreases	If it is set (present), events reflecting score decreases will be included in event list.
from	Used to provide a checkpoint value for filtering Entity Scoring Events
limit	Used to specify the batch size returned by the response from the /events/entity_scoring endpoint. By default 500 events will be returned if a limit is not specified. A maximum of 1,000 events will be returned per request.
event_timestamp_gte	Start date/time for scoring event, inclusive, formatted in ISO-8601
ordering	Orders events by “event_timestamp”. The default sorts event timestamps by ascending order. Events can be sorted in descending order by prepending the query with “minus” symbol.

Groups

The groups API can retrieve a listing of account groups that are defined on the system.

Version 3.2 of groups API supports GET, PATCH, POST, and DELETE to not only query the list of account groups, but also create, modify, or delete them.

The URL to retrieve account groups is `https://<vectra_portal_url>/api/v3.2/groups` and uses OAuth2 authentication.

The following table lists the top-level fields and descriptions present in the response for a bulk GET request to the v3.2 API for Groups.

ELEMENT	DESCRIPTION	TYPE	NOTES
results	A list of group objects.	list	
next	URL to the next page of groups.	string	
previous	URL to the previous page of groups.	string	
count	The total number of groups matching the query parameters.	integer	

The following table lists the fields and descriptions present in a Group object.

Element	Description	Type	Notes
id	Group ID.	integer	
name	The group name.	string	
type	The group type.	string	In this version of the API this will always return “account”.
description	User defined description of the group.	string	
importance	User defined group importance.	string	One of "high", "medium", "low", or "never_prioritize".
last_modified_timestamp	Filter for all groups modified on or after the given timestamp (GTE).	string	Timestamp format: YYYY-MM-DD HH-MM-SS GMT
last_modified_by	The last user that modified the group.	string	
members	List of member account names.	list of strings	
rules	List of triage rules that the group is attached to.	list of objects	The rules objects have the following key names “id”, “description”, “triage_category” which all have string values.

The available query parameters to filter for Groups are listed below.

QUERY PARAMETER	DESCRIPTION
account_ids	Only valid when the type parameter is set to “account”. Provide a comma-delimited (,) list of integers to filter groups associated with accounts having one of the given IDs.
account_names	Only valid when the type parameter is set to “account”. Provide a comma-delimited (,) list of strings to filter groups associated with accounts having one of the given names.
importance	One of "high", "medium", "low", or "never_prioritize". Will return an empty response otherwise.
description	Filter by group description (case insensitive match).
last_modified_timestamp	Filters for all groups modified on or after the given timestamp (GTE), formatted in ISO-8601.
last_modified_by	Filters groups by the user who made the most recent modification.
name	Filters by group name (case insensitive match).
type	Filter by group type. This version of the API will only accept the value “account”.
membership_action	Exclusive parameter for PATCH requests. This will accept the values “append”, “remove”, and “replace”. Members passed in the request body will be used to perform the corresponding membership update. Appendix A covers this in more detail.

An example of using the GET, POST,PATCH, and DELETE methods for Groups can be found in Appendix A.

Appendix A

This section will include examples of data retrieved from the REST API 3.2. Due to the amount of data that can be retrieved from a single query, the output examples below show only a snippet of the actual data that can be retrieved.

Note: The information in the following examples was generated in a lab environment. Any reference to IP addresses similar to those used in your environment is purely coincidental.

Detections

GET

URL: `https://<vectra_portal_url>/api/v3.2/detections/4`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
    "summary": {
        "user_type": "Regular",
        "azure_ad_privilege": {
            "privilege": 2,
            "privilegeCategory": "Low"
        },
        "num_events": 2,
        "operations": [
            "Consent to application.",
            "Add delegated permission grant."
        ],
        "src_ips": [],
        "target_entities": [
            "atlassian",
            "microsoft graph"
        ],
        "description": "This account was seen using an operation associated with a
high privilege admin activity that was anomalous for the user."
    },
    "threat": 0,
    "note_modified_by": null,
    "detection_category": "LATERAL MOVEMENT",
    "is_marked_custom": false,
    "detection_type": "Azure AD Privilege Operation Anomaly",
    "note_modified_timestamp": null,
    "assigned_to": null,
    "detection": "Azure AD Privilege Operation Anomaly",
    "note": null,
    "groups": [],
    "tags": [],
    "assigned_date": null,
    "src_ip": null,
    "certainty": 0,
    "targets_key_asset": false,
    "last_timestamp": "2021-12-11T19:52:31Z",
    "src_account": {
        "id": 1,
        "name": "O365:jmalacara@example.com",
        "url": "http://123456789.uw2.portal.vectra.ai/api/v3.2/accounts/1",
        "threat": 0,
        "certainty": 0,
        "privilege_level": null,
        "privilege_category": null
    },
    "category": "LATERAL MOVEMENT",
    "sensor": "tzlgmx99",
    "detection_url": "http://123456789.uw2.portal.vectra.ai/api/v3.2/detections/4",
    "first_timestamp": "2021-12-11T19:52:31Z",
    "custom_detection": null,
    "sensor_name": "Vectra X",
    "t_score": 0,
    "is_custom_model": false,
    "id": 4,
    "state": "inactive",
    "notes": [],
    "is_targeting_key_asset": false,
    "triage_rule_id": null,
    "url": "http://123456789.uw2.portal.vectra.ai/api/v3.2/detections/4",
    "description": null,
    "c_score": 0,
    "grouped_details": [
        {
            "operation": "Consent to application.",
            "target_entity": "atlassian",
            "src_ips": [],
            "user_type": "Regular",
            "num_events": 1,
            "azure_ad_privilege": {
                "privilege": 2,
                "privilegeCategory": "Low"
            },
            "operation_details": [
                {
                    "display_name": "ConsentContext.IsAdminConsent",
                    "new_value": "False",
                    "old_value": null
                },
                {
                    "display_name": "ConsentContext.IsAppOnly",
                    "new_value": "False",
                    "old_value": null
                },
                {
                    "display_name": "ConsentContext.OnBehalfOfAll",
                    "new_value": "False",
                    "old_value": null
                },
                {
                    "display_name": "ConsentContext.Tags",
                    "new_value": "WindowsAzureActiveDirectoryIntegratedApp",
                    "old_value": null
                },
                {
                    "display_name": "ConsentAction.Permissions",
                    "new_value": "[] => [[Id:
kD4Wx1xB5kavt9rlFq0oJvsgHpNuEWpCvthhgPPoPIRqrXjJgw92RIqby2TABWol, ClientId: c7163e90-
415c-46e6-afb7-dae516ad2826, PrincipalId: c978ad6a-0f83-4476-8a9b-cb64c0056a25,
ResourceId: 931e20fb-116e-426a-bed8-6180f3e83c84, ConsentType: Principal, Scope:
openid profile email]]; ",
                    "old_value": null
                },
                {
                    "display_name": "TargetId.ServicePrincipalNames",
                    "new_value": "46cdaa3f-9339-4b60-bf04-4dd98cd903d7",
                    "old_value": null
                }
            ],
            "normal_operations": [
                "UserLoggedIn"
            ],
            "normal_account_objects": [
                {
                    "id": 11,
                    "uid": "O365:rvalero@example.com"
                }
            ],
            "last_timestamp": "2021-12-11T19:52:31Z"
        },
        {
            "operation": "Add delegated permission grant.",
            "target_entity": "microsoft graph",
            "src_ips": [],
            "user_type": "Regular",
            "num_events": 1,
            "azure_ad_privilege": {
                "privilege": 2,
                "privilegeCategory": "Low"
            },
            "operation_details": [
                {
                    "display_name": "DelegatedPermissionGrant.Scope",
                    "new_value": " openid profile email",
                    "old_value": null
                },
                {
                    "display_name": "DelegatedPermissionGrant.ConsentType",
                    "new_value": "Principal",
                    "old_value": null
                },
                {
                    "display_name": "ServicePrincipal.ObjectID",
                    "new_value": "c7163e90-415c-46e6-afb7-dae516ad2826",
                    "old_value": null
                },
                {
                    "display_name": "ServicePrincipal.DisplayName",
                    "new_value": null,
                    "old_value": null
                },
                {
                    "display_name": "ServicePrincipal.AppId",
                    "new_value": null,
                    "old_value": null
                },
                {
                    "display_name": "ServicePrincipal.Name",
                    "new_value": null,
                    "old_value": null
                },
                {
                    "display_name": "TargetId.ServicePrincipalNames",
                    "new_value":
"https://canary.graph.microsoft.com/;https://graph.microsoft.us/;https://dod-
graph.microsoft.us/;https://dod-
graph.microsoft.us;https://graph.microsoft.com/;https://graph.microsoft.us;https://can
ary.graph.microsoft.com;https://graph.microsoft.com;https://ags.windows.net;00000003-
0000-0000-c000-000000000000/ags.windows.net;00000003-0000-0000-c000-000000000000",
                    "old_value": null
                }
            ],
            "normal_operations": [
                "UserLoggedIn"
            ],
            "normal_account_objects": [],
            "last_timestamp": "2021-12-11T19:52:30Z"
        }
    ],
    "campaign_summaries": [],
    "is_triaged": false
}
```

Detections (Notes)

GET

URL: `https://<vectra_portal_url>/api/v3.2/detections/<detection_id>/notes`

Headers:

`"Authorization": "Bearer <access_token>"`

Response:

```
[
  {
    "id": 1,
    "date_created": "2021-01-11T13:42:43Z",
    "date_modified": null,
    "created_by": "vadmin",
    "modified_by": null,
    "note": "this is a detection note"
  },
  {
    "id": 2,
    … … …
  }
]
```

POST

URL: `https://<vectra_portal_url>/api/v3.2/detections/<detection_id>/notes`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{"note":"this is a detection note"}
```

Response:

```
{
  "id": 2,
  "date_created": "2021-01-11T14:14:10.527603Z",
  "date_modified": null,
  "created_by": "vadmin",
  "modified_by": null,
  "note": "this is a detection note"
}
```

GET

URL: `https://<vectra_portal_url>/api/v3.2/detections/<detection_id>/notes/<note_id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
  "id": 1,
  "date_created": "2021-01-11T13:54:47.987918Z",
  "date_modified": null,
  "created_by": "vadmin",
  "modified_by": null,
  "note": " this is a detection note "
}
```

PATCH

URL: `https://<vectra_portal_url>/api/v3.2/detections/<detection_id>/notes/<note_id>`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{
  "note": “updated note”
}
```

Response:

```
{
  "id": 1,
  "date_created": "2021-01-11T13:47:42Z",
  "date_modified": "2021-01-11T13:57:11Z",
  "created_by": "vadmin",
  "modified_by": "vadmin",
  "note": "updated note"
}
```

DELETE

URL: `https://<vectra_portal_url>/api/v3.2/detections/<detection_id>/notes/<note_id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Mark as Fixed

PATCH to mark/unmark a detection as fixed.

URL: `https://<vectra_portal_url>/api/v3.2/detections`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{
    "detectionIdList": [128, 129],
    "mark_as_fixed": "True"
}
```

Response:

```
{
    "_meta": {
        "level": "Success",
        "message": "Successfully marked detections"
    }
}
```

Tagging

GET to list tags on an account or detection.

URL for accounts:

```
https://<vectra_portal_url>/api/v3.2/tagging/account/<account_id>
```

URL for detections:

```
https://<vectra_portal_url>/api/v3.2/tagging/detection/<detection_id>
```

Headers:

```
“Authorization”: “Bearer <access_token>”
```

Response:

```
{
  "status": "success",
  "tag_id": "1000",
  "tags": [
    "newticket",
    "under_investigation"
  ]
}
```

PATCH to add “new_tag” for an account.

URL for accounts:

```
https://<vectra_portal_url>/api/v3.2/tagging/account/<account_id>
```

URL for detections:

```
https://<vectra_portal_url>/api/v3.2/tagging/detection/<detection_id>
```

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json”
```

Body:

```
{
    "tags": [“newticket”, “under_investigation”]
}
```

Response:

```
{
  "status": "success",
  "tag_id": "1000",
  "tags": [
    "newticket",
    "under_investigation"
  ]
}
```

PATCH to clear tags for a detection

URL for accounts:

```
https://<vectra_portal_url>/api/v3.2/tagging/account/<account_id>
```

URL for detections:

```
https://<vectra_portal_url>/api/v3.2/tagging/detection/<detection_id>
```

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json”
```

Body:

```
{
    "tags": []
}
```

Patch operation will alter the tags for the account or detection to match the “tags” list provided in the PATCH.

Accounts

GET

URL: `https://<vectra_portal_url>/api/v3.2/accounts/1`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
    "id": 1,
    "url": "http://123456789.uw2.portal.vectra.ai/api/v3.2/accounts/1",
    "name": "O365:jmalacara@vectra.ai",
    "state": "inactive",
    "threat": 0,
    "certainty": 0,
    "severity": "Low",
    "account_type": [
        "o365"
    ],
    "tags": [],
    "note": null,
    "notes": [],
    "note_modified_by": null,
    "note_modified_timestamp": null,
    "privilege_level": null,
    "privilege_category": null,
    "last_detection_timestamp": "2021-12-20T18:11:16Z",
    "detection_set": [
        "http://123456789.uw2.portal.vectra.ai/api/v3.2/detections/1",
        "http://123456789.uw2.portal.vectra.ai/api/v3.2/detections/4"
    ],
    "probable_home": null,
    "assignment": null,
    "past_assignments": [],
    "sensors": [
        "tzlgmx99"
    ],
    "detection_summaries": [
        {
            "detection_id": 1,
            "detection_url":
"http://123456789.uw2.portal.vectra.ai/api/v3.2/detections/1",
            "detection_type": "O365 Internal Spearphishing",
            "detection_category": "LATERAL MOVEMENT",
            "is_targeting_key_asset": false,
            "state": "inactive",
            "threat": 0,
            "certainty": 0,
            "is_triaged": false,
            "tags": [],
            "summary": {
                "subject": [
                    "Sandbox: Congratulations, you just won!"
                ],
                "recipients": [
                    "tkawale@example.com",
                    "johnny@example.com",
                    "idolcetest@example.com",
                    "abacsmith@example.com",
                    "monty@example.com"
                ],
                "recipients_count": 24,
                "description": "This account sent emails containing malware,
suspicious links, or bad reputation."
            },
            "assigned_to": null,
            "assigned_date": null
        },
        {
            "detection_id": 4,
            "detection_url":
"http://123456789.uw2.portal.vectra.ai/api/v3.2/detections/4",
            "detection_type": "Azure AD Privilege Operation Anomaly",
            "detection_category": "LATERAL MOVEMENT",
            "is_targeting_key_asset": false,
            "state": "inactive",
            "threat": 0,
            "certainty": 0,
            "is_triaged": false,
            "tags": [],
            "summary": {
                "user_type": "Regular",
                "azure_ad_privilege": {
                    "privilege": 2,
                    "privilegeCategory": "Low"
                },
                "num_events": 2,
                "operations": [
                    "Consent to application.",
                    "Add delegated permission grant."
                ],
                "src_ips": [],
                "target_entities": [
                    "atlassian",
                    "microsoft graph"
                ],
                "description": "This account was seen using an operation associated
with a high privilege admin activity that was anomalous for the user."
            },
            "assigned_to": null,
            "assigned_date": null
        }
    ]
}
```

Account (Notes)

GET

URL: `https://<vectra_portal_url>/api/v3.2/accounts/<account_id>/notes`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
[
  {
    "id": 1,
    "date_created": "2021-01-11T13:42:43Z",
    "date_modified": null,
    "created_by": "vadmin",
    "modified_by": null,
    "note": "this is an account note"
  },
  {
    "id": 2,
    … … …
  }
]
```

POST

URL: `https://<vectra_portal_url>/api/v3.2/accounts/<account_id>/notes`

Headers:

```
"Authorization": "Bearer <access_token>",
“Content-Type”: “application/json”
```

Body:

```
{"note": "this is a note"}
```

Response:

```
{
  "id": 2,
  "date_created": "2021-01-11T14:14:10.527603Z",
  "date_modified": null,
  "created_by": "vadmin",
  "modified_by": null,
  "note": "this is a note"
}
```

GET

URL: `https://<vectra_portal_url>/api/v3.2/accounts/<account_id>/notes/<note_id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
  "id": 1,
  "date_created": "2021-01-11T13:54:47.987918Z",
  "date_modified": null,
  "created_by": "vadmin",
  "modified_by": null,
  "note": " this is an account note "
}
```

PATCH

URL: `https://<vectra_portal_url>/api/v3.2/accounts/<account_id>/notes/<note_id>`

Headers:

```
"Authorization": "Bearer <access_token>",
“Content-Type”: “application/json”
```

Body:

```
{"note": "updated note"}
```

Response:

```
{
  "id": 1,
  "date_created": "2021-01-11T13:47:42Z",
  "date_modified": "2021-01-11T13:57:11Z",
  "created_by": "vadmin",
  "modified_by": "vadmin",
  "note": "updated note"
}
```

DELETE

URL: `https://<vectra_portal_url>/api/v3.2/accounts/<account_id>/notes/<note_id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Triage Rules

GET

URL: `https://<vectra_portal_url>/api/v3.2/rules/<id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
    "id": 68,
    "url": "https://1.1.1.1/api/v3.2/rules/68",
    "description": "Expected behavior from these devices",
    "enabled": true,
    "created_timestamp": "2019-08-27T20:55:29Z",
    "last_timestamp": null,
    "is_whitelist": false,
    "priority": null,
    "active_detections": 2,
    "total_detections":3,
    "template": true,
    "additional_conditions": {
        "OR": [
            {
                "AND": [
                    {
                        "ANY_OF": {
                            "field": "remote1_port",
                            "values": [
                                {
                                    "url": null,
                                    "value": "135",
                                    "label": "135"
                                }
                            ],
                            "groups": [],
                            "label": "Port"
                        }
                    }
                ]
            }
        ]
    },
    "source_conditions": {
        "OR": [
            {
                "AND": [
                    {
                        "ANY_OF": {
                            "field": "host",
                            "values": [],
                            "groups": [
                                {
                                    "url": "https://192.168.51.36/api/v3.2/groups/8",
                                    "value": 8,
                                    "label": "Cognito - IPAM"
                                }
                            ],
                            "label": "Host"
                        }
                    }
                ]
            }
        ]
    },
    "detection_category": "RECONNAISSANCE",
    "triage_category": "Expected IPAM Behavior",
    "detection": "Internal Darknet Scan"
}
```

POST

URL: `https://<vectra_portal_url>/api/v3.2/rules`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{
      "description": "Peer to peer triage rule",
      "detection_category": "COMMAND & CONTROL",
      "triage_category": "Miscategorization",
      "detection": "Peer-to-Peer",
      "is_whitelist": false,
      "additional_conditions": {
        "OR": [
          {
            "AND": [
              {
                "ANY_OF": {
                  "field": "remote1_ip",
                  "values": [],
                  "groups": [
                    {
                      "value": 2
                    }
                  ]
                }
              },
              {
                "ANY_OF": {
                  "field": "remote1_dns",
                  "values": [
                    {
                      "value": "test.server.com"
                    }
                  ],
                  "groups": [
                    {
                      "value": 11
                    }
                  ]
                }
              }
            ]
          }
        ]
      },
      "source_conditions": {
        "OR": [
          {
            "AND": [
              {
                "ANY_OF": {
                  "field": "host",
                  "values": [
                    {
                      "value": 1
                    }
                  ],
                  "groups": [
                    {
                      "value": 8
                    }
```

```
                  ]
                }
              }
            ]
          }
        ]
      }
    }
```

The following fields are mandatory:

“detection_category”: Must be set to the category of the detection – LATERAL MOVEMENT, RECONNAISSANCE, COMMAND & CONTROL, EXFILTRATION, BOTNET, INFO

“detection”: The detection that must be triaged. Use the detection name as seen in the UI or the “Understanding Vectra Detections” guide.

“triage_category”: The name that will be used for the triaged detection. Only applies if “is_whitelist” is set to 0.

“description”: User defined description for the rule.

“is_whitelist”: A Boolean flag indicating whether to create a “whitelist” or a “track without scores” rule
source_conditions: conditions that can be applied to the source of a detection. Can be NULL.

additional_conditions: other conditions that are different on a per-detection-type basis. Can be NULL.

Response:

Success
The response contains the id of the triage rule if successful:

```
{
    "_meta": {
        "message": "Successfully created triage filter",
        "level": "success"
    },
    "id": 11
}
```

Failure
Failure will result in an error message along with an indication of what was incorrect

```
{
    "_meta": {
        "message": "Invalid field(s) found",
        "level": "error"
    },
    "host": [
        "Invalid host: 5 does not map to a real host"
    ]
}
```

PUT

URL: `https://<vectra_portal_url>/api/v3.2/rules/<id>`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{
    "detection_category": "LATERAL MOVEMENT",
    "triage_category": "Susp Rmt Exec - Test",
    "detection": "Suspicious Remote Execution",
    "is_whitelist": 0,
    "description": "put test",
        "additional_conditions": {
          "OR": [
            {
              "AND": [
                {
                  "ANY_OF": {
                    "field": "remote1_ip",
                    "values": [
                      {
                        "value": "1.1.1.1"
                      }
                    ],
                    "groups": [],
                  }
                }
              ]
            }
          ]
        },
        "source_conditions": {
          "OR": [
            {
              "AND": [
                {
                  "ANY_OF": {
                    "field": "ip",
                    "values": [
                      {"value": "1.1.1.1"},
                      {"value": "1.2.1.1"},
                      {"value": "1.1.3.1"}
                    ],
                    "groups": []
                  }
                }
              ]
            }
          ]
        }

}
```

DELETE

URL: `https://<vectra_portal_url>/api/v3.2/rules/<id>`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{
"detectionIdLIst": [<detection_id1>, <detection_id2>, …]
}
```

Assignments

GET used to retrieve assignments

URL: `https://<vectra_portal_url>/api/v3.2/assignments`

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json"
```

Response:

```
{
    "count": 3,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 4,
            "assigned_by": {
                "id": 1,
                "username": "josem"
            },
            "date_assigned": "2021-05-18T04:34:48Z",
            "date_resolved": null,
            "events": [
                {
                    "assignment_id": 4,
                    "actor": 1,
                    "event_type": "created",
                    "datetime": "2021-05-18T04:34:48Z",
                    "context": {
                        "to": 20
                    }
                }
            ],
            "outcome": null,
            "resolved_by": null,
            "triaged_detections": {},
            "host_id": null,
            "account_id": 1,
            "assigned_to": {
                "id": 20,
                "username": "admin"
            }
        },
        {
            "id": 2,
            "assigned_by": {
                "id": 1,
                "username": "josem"
            },
            "date_assigned": "2021-05-18T03:51:38Z",
            "date_resolved": "2021-05-18T03:51:53Z",
            "events": [
                {
                    "assignment_id": 2,
                    "actor": 1,
                    "event_type": "resolved",
                    "datetime": "2021-05-18T03:51:53Z",
                    "context": {
                        "triage_as": "Miscategorization",
                        "detection_ids": [
                            1
                        ],
                        "created_rule_ids": [
                            27
                        ]
                    }
                },
                {
                    "assignment_id": 2,
                    "actor": 1,
                    "event_type": "created",
                    "datetime": "2021-05-18T03:51:38Z",
                    "context": {
                        "to": 20
                    }
                }
            ],
            "outcome": {
                "id": 3,
                "builtin": true,
                "user_selectable": true,
                "title": "False Positive",
                "category": "false_positive"
            },
            "resolved_by": {
                "id": 1,
                "username": "josem"
            },
            "triaged_detections": [
                1
            ],
            "host_id": null,
            "account_id": 1,
            "assigned_to": {
                "id": 20,
                "username": "admin"
            }
        },
        {
            "id": 1,
            "assigned_by": {
                "id": 1,
                "username": "josem"
            },
            "date_assigned": "2021-05-17T23:21:52Z",
            "date_resolved": "2021-05-17T23:23:53Z",
            "events": [
                {
```

```
                    "assignment_id": 1,
                    "actor": 1,
                    "event_type": "resolved",
                    "datetime": "2021-05-17T23:23:53Z",
                    "context": {
                        "triage_as": null,
                        "detection_ids": [],
                        "created_rule_ids": null
                    }
                },
                {
                    "assignment_id": 1,
                    "actor": 1,
                    "event_type": "created",
                    "datetime": "2021-05-17T23:21:52Z",
                    "context": {
                        "to": 20
                    }
                }
            ],
            "outcome": {
                "id": 1,
                "builtin": true,
                "user_selectable": true,
                "title": "Benign True Positive",
                "category": "benign_true_positive"
            },
            "resolved_by": {
                "id": 1,
                "username": "josem"
            },
            "triaged_detections": {},
            "host_id": null,
            "account_id": 1,
            "assigned_to": {
                "id": 20,
                "username": "admin"
            }
        }
    ]
}
```

POST used to assign an entity to a user.

URL: `https://<vectra_portal_url>/api/v3.2/assignments`

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json"
```

Body:

```
{
    "assign_account_id": "27",
    "assign_to_user_id": "30"
}
```

Response:

```
{
    "assignment": {
        "id": 25,
        "assigned_by": {
            "id": 45,
            "username": "api_client_df405cc568734d4a8b7ebec1753ee647"
        },
        "date_assigned": "2022-03-31T19:07:24.841453Z",
        "date_resolved": null,
        "events": [
            {
                "assignment_id": 25,
                "actor": 45,
                "event_type": "created",
                "datetime": "2022-03-31T19:07:24Z",
                "context": {
                    "to": 30,
                    "entity_t_score": 68,
                    "entity_c_score": 48
                }
            }
        ],
        "outcome": null,
        "resolved_by": null,
        "triaged_detections": null,
        "host_id": null,
        "account_id": 27,
        "assigned_to": {
            "id": 30,
            "username": "jmalacara@vectra.ai"
        }
    }
}
```

PUT used to modify/reassign assignments.

URL: `https://<vectra_portal_url>/api/v3.2/assignments/<id>`

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json"
```

Body:

```
{
    "assign_account_id": "27",
    "assign_to_user_id": "23"
}
```

Response:

```
{
    "assignment": {
        "id": 25,
        "assigned_by": {
            "id": 45,
            "username": "api_client_df405cc568734d4a8b7ebec1753ee647"
        },
        "date_assigned": "2022-03-31T19:07:24Z",
        "date_resolved": null,
        "events": [
            {
                "assignment_id": 25,
                "actor": 45,
                "event_type": "reassigned",
                "datetime": "2022-03-31T19:13:02Z",
                "context": {
                    "from": 30,
                    "to": 23,
                    "entity_t_score": 68,
                    "entity_c_score": 48
                }
            },
            {
                "assignment_id": 25,
                "actor": 45,
                "event_type": "created",
                "datetime": "2022-03-31T19:07:24Z",
                "context": {
                    "to": 30,
                    "entity_t_score": 68,
                    "entity_c_score": 48
                }
            }
        ],
        "outcome": null,
        "resolved_by": null,
        "triaged_detections": {},
        "host_id": null,
        "account_id": 27,
        "assigned_to": {
            "id": 23,
            "username": "bart@vectra.ai"
        }
    }
}
```

DELETE used to delete assignments.

URL: `https://<vectra_portal_url>/api/v3.2/assignments/<id>`

Headers:

```
“Authorization”: “Bearer <access_token>”
```

PUT used to resolve assignments.

URL: `https://<vectra_portal_url>/api/v3.2/assignments/<id>/resolve`

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json"
```

Body:

```
{
    "outcome": "1",
    "note": "Resolved by JoseM",
    "triage_as": "My triage rule",
    "detection_ids": [128,120]
}
```

Response:

```
{
    "assignment": {
        "id": 26,
        "assigned_by": {
            "id": 45,
            "username": "api_client_df405cc568734d4a8b7ebec1753ee647"
        },
        "date_assigned": "2022-03-31T19:16:37Z",
        "date_resolved": "2022-03-31T19:20:06Z",
        "events": [
            {
                "assignment_id": 26,
                "actor": 45,
                "event_type": "resolved",
                "datetime": "2022-03-31T19:20:06Z",
                "context": {
                    "entity_t_score": 68,
                    "entity_c_score": 48,
                    "triage_as": "My triage rule",
                    "triaged_detection_ids": [
                        128,
                        120
                    ],
                    "fixed_detection_ids": null,
                    "created_rule_ids": [
                        14,
                        15
                    ]
                }
            },
            {
                "assignment_id": 26,
                "actor": 45,
                "event_type": "created",
                "datetime": "2022-03-31T19:16:37Z",
                "context": {
                    "to": 30,
                    "entity_t_score": 68,
                    "entity_c_score": 48
                }
            }
        ],
        "outcome": {
            "id": 1,
            "builtin": true,
            "user_selectable": true,
            "title": "Benign True Positive",
            "category": "benign_true_positive"
        },
        "resolved_by": {
            "id": 45,
            "username": "api_client_df405cc568734d4a8b7ebec1753ee647"
        },
        "triaged_detections": [
            128,
            120
        ],
        "host_id": null,
        "account_id": 27,
        "assigned_to": {
            "id": 30,
            "username": "jmalacara@vectra.ai"
        }
    }
}
```

Assignment Outcomes

GET used to list assignment outcomes.

URL: `https://<vectra_portal_url>/api/v3.2/assignment_outcomes`

Headers:
`“Authorization”: “Bearer <access_token>”`

Response:

```
{
    "count": 3,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 1,
            "builtin": true,
            "user_selectable": true,
            "title": "Benign True Positive",
            "category": "benign_true_positive"
        },
        {
            "id": 2,
            "builtin": true,
            "user_selectable": true,
            "title": "Malicious True Positive",
            "category": "malicious_true_positive"
        },
        {
            "id": 3,
            "builtin": true,
            "user_selectable": true,
            "title": "False Positive",
            "category": "false_positive"
        }
    ]
}
```

POST used to create assignment outcomes.

URL: `https://<vectra_portal_url>/api/v3.2/assignment_outcomes`

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json"
```

Body:

```
{
    "title": "Custom outcome",
    "category": "benign_true_positive"
}
```

Response:

```
{
    "id": 6,
    "builtin": false,
    "user_selectable": true,
    "title": "Custom outcome",
    "category": "benign_true_positive"
}
```

PUT used to modify an assignment outcome. The title can always be modified, but category can only be modified if the assignment outcome has not been used as an outcome for an assignment.

URL: `https://<vectra_portal_url>/api/v3.2/assignment_outcomes/<id>`

Headers:

```
“Authorization”: “Bearer <access_token>”,
“Content-Type”: “application/json"
```

Body:

```
{
    "title": "My New Outcome Title",
    "category": "false_positive"
}
```

Response:

```
{
    "id": 6,
    "builtin": false,
    "user_selectable": true,
    "title": "My New Outcome Title",
    "category": "false_positive"
}
```

DELETE used to delete an assignment outcome.

URL: `https://<vectra_portal_url>/api/v3.2/assignment_outcomes/<id>`

Headers:

`“Authorization”: “Bearer <access_token>”`

Account Scoring Events

GET

URL: `https://<vectra_portal_url>/api/v3.2/events/account_scoring?from=12588`

Headers:

`"Authorization": "Bearer <access_token>"`

Response:

```
{
    "events": [
        {
            "id": 12588,
            "version": "2022.0.0",
            "account_id": 299,
            "account_uid": "O365:homer@vectra.ai",
            "threat": 67,
            "certainty": 39,
            "severity": 6,
            "score_decrease": false,
            "account_href": "https://517731205673.uw2.portal.vectra.ai/accounts/299",
            "category": "ACCOUNT SCORING",
            "last_detection_href":
"https://517731205673.uw2.portal.vectra.ai/detections/514",
            "last_detection_type": "Azure AD MFA-Failed Suspicious Sign-On",
            "active_detection_types": [
                "Azure AD MFA-Failed Suspicious Sign-On"
            ],
            "event_timestamp": "2022-04-12T06:08:30Z"
        },
        {
            "id": 12589,
            "version": "2022.0.0",
            "account_id": 307,
            "account_uid": "O365:lisa@vectra.ai",
            "threat": 67,
            "certainty": 41,
            "severity": 6,
            "score_decrease": false,
            "href": "https://517731205673.uw2.portal.vectra.ai/accounts/307",
            "category": "ACCOUNT SCORING",
            "last_detection_href":
"https://517731205673.uw2.portal.vectra.ai/detections/526",
            "last_detection_type": "Azure AD MFA-Failed Suspicious Sign-On",
            "active_detection_types": [
                "Azure AD MFA-Failed Suspicious Sign-On"
            ],
            "event_timestamp": "2022-04-12T06:08:30Z"
        },
        {
            "id": 12590,
            "version": "2022.0.0",
            "account_id": 304,
            "account_uid": "O365:bart@vectra.ai",
            "threat": 21,
            "certainty": 89,
            "severity": 2,
            "score_decrease": false,
            "href": "https://517731205673.uw2.portal.vectra.ai/accounts/304",
            "category": "ACCOUNT SCORING",
            "last_detection_href":
"https://517731205673.uw2.portal.vectra.ai/detections/523",
            "last_detection_type": "Azure AD Brute-Force Attempt",
            "active_detection_types": [
                "Azure AD Brute-Force Attempt",
                "Azure AD Successful Brute-Force"
            ],
            "event_timestamp": "2022-04-12T06:36:47Z"
        }
    ],
    "remaining_count": 0,
    "next_checkpoint": 12591
}
```

Account Detection Events

GET

URL: `https://<vectra_portal_url>/api/v3.2/events/account_detection?from=222`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
    "events": [
        {
            "id": 222,
            "category": "lateral_movement",
            "threat": 50,
            "certainty": 50,
            "triaged": false,
            "detection_type ": "O365 Internal Spearphishing",
            "d_type_vname": "O365 Internal Spearphishing",
            "detection_id": 426,
            "detection_href":
"http://517731205673.uw2.portal.vectra.ai/detections/426?detail_id=2287",
            “account_id”: 196
            "account_href": "http://517731205673.uw2.portal.vectra.ai/accounts/196",
            "href": "http://517731205673.uw2.portal.vectra.ai/accounts/196",
            "account_uid ": “O365:do-not-reply@vectra.ai”,
            "src_account": “O365:do-not-reply@vectra.ai”,
            "event_timestamp": "2022-04-11T08:21:40Z",
            "ip": null,
            "detail": {
                "recipients": [
                    "homer@vectra.ai",
                    "bart@vectra.ai",
                    "lisa@vectra.ai"
                ],
                "hashes": [],
                "subject": "Sandbox: Support Case: New Relic, has a new case 00050552
with a priority Low at 4/10/2022 3:15 PM",
                "last_timestamp": "2022-04-10T22:17:24Z"
            },
            "severity": 5
        },
        {
            "id": 223,
            "category": "lateral_movement",
            "threat": 50,
            "certainty": 50,
            "triaged": false,
            "detection_type": "Azure AD Successful Brute-Force",
            "d_type_vname": "Azure AD Successful Brute-Force",
            "detection_id": 535,
            "detection_href":
"http://517731205673.uw2.portal.vectra.ai/detections/535?detail_id=2288",
            "account_id": 2219,
            "account_href": "http://517731205673.uw2.portal.vectra.ai/accounts/2219",
            "account_uid ": "O365:marge@vectra.ai",
            "src_account": "O365:marge@vectra.ai",
            "event_timestamp": "2022-04-12T06:32:09Z",
            "ip": "143.244.99.206",
            "detail": {
                "operation": "UserLoggedIn",
                "num_attempts": 26,
                "first_timestamp": "2022-04-11T16:38:06Z",
                "last_timestamp": "2022-04-11T16:38:06Z"
            },
            "severity": 5
        }
    ],
    "next_checkpoint": 224,
    "remaining_count": 0
}
```

Audit Log Events

GET

URL: `https://<vectra_portal_url>/api/v3.2/events/audits?from=16&limit=2`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
    "events": [
        {
            "id": 16,
            "user_id": 24,
            "username": "josem@vectra.ai",
            "user_type": "JWT",
            "api_client_id": null,
            "user_role": "Admin",
            "version": "2022.0.0",
            "source_ip": "10.100.219.11",
            "event_timestamp": "2022-05-21T06:24:52Z",
            "message": "Tag josetag100 has been added to linked_accounts with ids
[45]",
            "result_status": "success",
            "event_data": {
                "tag": {
                    "tag_name": "josetag100",
                    "entity": "linked_account",
                    "entity_ids": [
                        45
                    ]
                }
            },
            "event_object": "account_tag",
            "event_action": "created"
        },
        {
            "id": 17,
            "user_id": 24,
            "username": "josem@vectra.ai",
            "user_type": "JWT",
            "api_client_id": null,
            "user_role": "Admin",
            "version": "2022.0.0",
            "source_ip": "10.100.114.24",
            "event_timestamp": "2022-05-21T07:15:58Z",
            "message": "note for detection@172 created This has b...",
            "result_status": "success",
            "event_data": {
                "detection_note": {
                    "text": "This has been fixed downstream and is being monitored for
additional activity.",
                    "note_id": 16,
                    "detection_id": "172"
                }
            },
            "event_object": "detection_note",
            "event_action": "created"
        }
    ],
    "next_checkpoint": 18,
    "remaining_count": 0
}
```

Entities

GET

URL: `https://<vectra_portal_url>/api/v3.2/entities/<entity_id>?entity_type=account`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
        "id": 1,
        "breadth_contrib": 0,
        "entity_importance": 0,
        "is_prioritized": false,
        "severity": "High",
        "urgency_reason": “Ransomware: This entity was prioritized because it was
implicated in an active ransomware detection”,
        "urgency_score": 0,
        "velocity_contrib": 0,
        "detection_set": [
                "http://202079170575.uw2.devportal.vectra.ai/api/v3/detections/1",
                "http://202079170575.uw2.devportal.vectra.ai/api/v3/detections/408",
                "http://202079170575.uw2.devportal.vectra.ai/api/v3/detections/549",
                "http://202079170575.uw2.devportal.vectra.ai/api/v3/detections/699"
        ],
        "last_detection_timestamp": "2022-10-06T17:34:23Z",
        "name": "SAML:testaccount@vectra.ai",
        "notes": [
                {
                        "id": 3,
                        "date_created": "2022-05-24T14:55:01Z",
                        "date_modified": null,
                        "created_by": "api_client_b58f8e28a46f40b59a77bf04d068af61",
                        "modified_by": null,
                        "note": "# IMPORTANT - PLEASE READ!! Account under **active
investigation**."
                }
        ],
        "privilege_level": 2,
        "privilege_category": “Low”,
        "sensors": [
                "abx1pcad"
        ],
        "state": "active",
        "tags": [
                "high_priority"
        ],
        "url": "https://202079170575.uw2.devportal.vectra.ai/accounts/1",
        "account_type": [
                "aws"
        ],
        "entity_type": "account"
}
```

Entity Scoring Events

GET

URL: `https://<vectra_portal_url>/api/v3.2/events/entity_scoring?from=100&limit=2`

Headers:

`"Authorization": "Bearer <access_token>"`

Response:

```
{
    "events": [
        {
            "entity_id": 100,
            "name": "AWS:None/us-1-east/config:AWSConfig-Describe",
            "entity_importance": 0,
            "entity_type": "account",
            "is_prioritized": false,
            "severity": "Low",
            "urgency_reason": “Ransomware: This entity was prioritized because it was
implicated in an active ransomware detection”,
            "urgency_score": 0,
            "url": "https://200888808432.uw2.devportal.vectra.ai/accounts/8",
            "category": "ACCOUNT SCORING",
            "last_detection_url":
"https://200888808432.uw2.devportal.vectra.ai/detections/103",
            "last_detection_type": "AWS S3 Enumeration",
            "last_detection_id": 103,
            "active_detection_types": [
                "AWS S3 Enumeration"
            ],
            "event_timestamp": "2022-08-07T00:14:31Z",
            "breadth_contrib": 0,
            "velocity_contrib": 0
        },
        {
            "entity_id": 101,
            "name": "AWS:884414556547/solus-cgid-domryder-ucw",
            "entity_importance": 0,
            "entity_type": "account",
            "is_prioritized": false,
            "severity": "Low",
            "urgency_reason": “Ransomware: This entity was prioritized because it was
implicated in an active ransomware detection”,,
            "urgency_score": 0,
            "url": "https://200888808432.uw2.devportal.vectra.ai/accounts/2",
            "category": "ACCOUNT SCORING",
            "last_detection_url":
"https://200888808432.uw2.devportal.vectra.ai/detections/80",
            "last_detection_type": "AWS Attack Tools",
            "last_detection_id": 80,
            "active_detection_types": [
                "AWS Attack Tools"
            ],
            "event_timestamp": "2022-08-07T06:15:04Z",
            "breadth_contrib": 0,
            "velocity_contrib": 0
        }
    ],
    "next_checkpoint": 102,
    "remaining_count": 659
}
```

Groups

GET

URL: `https://<vectra_portal_url>/api/v3.2/groups/<group_id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Response:

```
{
    "id": 5,
    "name": "test_group",
    "description": "a test group",
    "last_modified": "2022-11-18T18:21:34Z",
    "last_modified_by": "API Client c628cdee",
    "type": "account",
    "members": [
        {
            "uid": “name_1”
        },
        {
            "uid": “name_2”
        }
    ],
    "rules": [],
    "importance": "high"
}
```

POST

URL: `https://<vectra_portal_url>/api/v3.2/groups`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{

    "name": "test_group",

    "type": "account",

    "members":[

        "name_1","name_2"

    ],

    "description": "a test group",

    "importance": "high"

}
```

Response:

```
{
    "group": {
        "id": 5
    }
}
```

PATCH

PATCH method can be used to modify existing groups. A PATCH is a partial override, so the format can include one or more fields from the POST.

When using the default PATCH functionality to modify the members, a complete new list must be included. For example, if a group already contains accounts “acct1@test.com”, and “acct2@test.com”, and the group should be modified to include account “acct5@test.com”, then “acct5@test.com” should be added to the list, the members field of the PATCH request should be “acct1@test.com”,“acct2@test.com”,“acct1@test.com”.

Alternatively, the “membership_action” query parameter can be passed in with a value of “replace”, “append”, or “remove”. The “append” and “remove” values will only affect members listed in the request body; if a user wants to add or remove account “acct1@test.com” for a group, then the user can pass the members field of “acct1@test.com” in the request body with the appropriate value of “membership_action”. The “replace” value will perform the default PATCH behavior.

URL: `https://<vectra_portal_url>/api/v3.2/groups/<group_id>`

Headers:

```
"Authorization": "Bearer <access_token>",
"Content-Type": "application/json"
```

Body:

```
{
    "name": "test_group new name"
}
```

Response:

```
{
    "id": 5,
    "name": "test_group new name",
    "description": "a test group",
    "last_modified": "2022-11-18T18:28:11Z",
    "last_modified_by": "API Client c628cdee",
    "type": "account",
    "members": [
        {
            "uid": “name_1”
        },
        {
            "uid": “name_2”
        }
    ],
    "rules": [],
    "importance": "high"
}
```

DELETE

URL: `https://<vectra_portal_url>/api/v3.2/groups/<group_id>`

Headers:

```
"Authorization": "Bearer <access_token>"
```

Appendix B

Detections

Predefined categories and vnames

CATEGORY	VNAME
Command and Control	
	Azure AD Admin Account Creation
	Azure AD MFA-Failed Suspicious Sign-On

CATEGORY	VNAME
	O365 Power Automate HTTP Flow Creation
	Azure AD Redundant Access Creation
	Azure AD Suspicious OAuth Application
	O365 Suspicious Power Automate Flow Creation
	Azure AD Suspicious Sign-On
	Azure AD TOR Activity
	AWS Root Credential Usage
	AWS Suspicious Credential Usage
	AWS TOR Activity
Botnet	
	AWS Cryptomining
Reconnaissance	
	O365 Suspicious Compliance Search
	O365 Unusual eDiscovery Search
	O365 Suspect eDiscovery Usage
	AWS EC2 Enumeration
	AWS Organization Discovery
	AWS S3 Enumeration
	AWS Suspect Credential Access from EC2
	AWS Suspect Credential Access from ECS
	AWS Suspect Credential Access from SSM
	AWS Suspect Escalation Reconnaissance
	AWS User Permission Enumeration
Lateral Movement	
	Azure AD Successful Brute-Force
	O365 Suspicious Mailbox Manipulation
	O365 Attacker Tool: Ruler
	Azure AD Change to Trusted IP Configuration
	O365 Disabling of Security Tools
	O365 DLL Hijacking Activity
	O365 External Teams Access
	O365 Internal Spearphising
	O365 Log Disabling Attempt
	O365 Malware Stage: Upload

CATEGORY	VNAME
	Azure AD MFA Disabled
	Azure AD Newly Created Admin Account
	O365 Ransomware
	O365 Risky Exchange Operation
	Azure AD Privilege Operation Anomaly
	O365 Suspicious Sharepoint Operation
	O365 Suspicious Teams Application
	Azure AD Unusual Scripting Engine Usage
	AWS ECR Hijacking
	AWS Lambda Hijacking
	AWS Logging Disabled
	AWS Ransomware S3 Activity
	AWS Security Tools Disabled
	AWS Suspect Admin Privilege Granting
	AWS Suspect Console Pivot
	AWS Suspect Login Profile Manipulation
	AWS Suspect Privilege Escalation
	AWS User Hijacking
Exfiltration	
	O365 eDiscovery Exfil
	O365 Exfiltration Before Termination
	O365 Suspicious Download Activity
	O365 Suspicious Exchange Transport Rule
	O365 Suspicious Mail Forwarding
	O365 Suspect Power Automate Activity
	O365 Suspicious Sharing Activity
	AWS Suspect External Access Granting
	AWS Suspect Public EBS Change
	AWS Suspect Public EC2 Change
	AWS Suspect Public S3 Change