Edge Firewall ATP - URL Filtering/ IP Filtering (Management Plane) Functional Spec

VeloCloud Engineering

Exported on 11/20/2024

Table of Contents

Author(s)	@Ben Shapero @Nandini Rangaswamy @Qing Li @Thiaga Sankaran @Aditya Agarwal @William Zapata
Approver(s)	☑ @Unknown User (qxinzhou) ☑ @Tom Speeter ☑ @Gurudutt Maiya Belur
Reviewer(s)	<QE Lead, QE Manager, Tech Ops Lead, Security, Product Mgr, Support Engineering, Tech Pubs, UX, API> *Please note that API team review is **mandatory** if the feature impacts the public API (e.g. device configuration schema).
Status	APPROVED
Approval Date	05 Sep 2023
Target Release	Yamazaki
PRD	Edge Firewall ATP services (IDS/IPS, URLF, Reputation)[1]
UX Design Brief (if applicable)	https://www.figma.com/file/15iMXtz6z2ZWIrFGa4Hfs2/FWaaS-E2E-Story?node-id=393%3A26230&t=h45zwBXrEEstEvLj-1
Jira Epic	VLENG-112975[2] - MP for Edge Firewall ATP services - URL Filtering & IP Reputation IN PROGRESS
Project Page	
References	Webroot Cache Redesign[3] NSX Threat Intelligence Cloud - Architecture[4] NSX Threat Intel Cloud: SDWAN Edge support for URL/IP Filtering[5]

1 https://confluence.eng.vmware.com/pages/viewpage.action?pageId=1481319163
2 https://vmw-jira.broadcom.net/browse/VLENG-112975?src=confmacro
3 https://vmw-confluence.broadcom.net/display/NSBU/Webroot+Cache+Redesign
4 https://vmw-confluence.broadcom.net/display/NSBU/NSX+Threat+Intelligence+Cloud+-+Architecture
5 https://vmw-confluence.broadcom.net/pages/viewpage.action?pageId=1774069929

1 1. Change log

Version	Date	Changes
0.1	4/13/23	First Version

2 2. Introduction

Next Generation Firewall (NGF) products include Advanced Threat Detection/Prevention (ATD/ATP). The Enhanced Firewall Services (EFS) functionality will be powered by NSX technology. The EFS service will support protecting VCE traffic from intrusions across branch 2 branch, branch 2 hub or branch to internet traffic patterns. Customers configure and manage the EFS services via Firewall functionality in the VCO UI.

The management plane must support all configuration operations, including creating/changing firewall rules, updating reference databases and collecting/exposing logs for dashboards and reports. To support the increased volume of firewall logs and provide better monitoring performance,

- All firewall logs are saved in GSM Logging Service for auditing/monitoring
- EFS logs are saved in the ClickHouse database that currently exists on VCO. These logs are used for analytics and metrics for dashboards in the VCO UI

Woodford Release (5.2.0) Introduced EFS features to the Velocloud Firewall in the form of IDS/IPS Signature matching. After the upcoming Xante Release, URL Filtering and IP Reputation EFS features will be incorporated. URL Filtering consists of assigning one or more categories and a reputation score to URLs/ Domains. IP Reputation looks up the score of destination IPs, blocking if their scores indicate a threat. Customers will be able to configure Firewall Rules to block web traffic based on category and/or reputation of the URL or IP, in addition to all previously available filters/checks.

3 3. Dependencies

Edge firewall enhancements depends on the following services outside of VCO.

NTICS:

NSX team provides REST API endpoints to download URL category lists and reputation databases. Unavailability of these APIs will affect ability of VCE to provide URL Filtering and IP Reputation.

Access Log Infrastructure:

Firewall logs are sent to a service on GSM to store, view, search, and export logs. If GSM is not available or slows down then it will impact VCO log ingestion causing the log files take up more VCO disk space.

Search Service:

VCO UI uses this service to query the logs. Availability this service plays a crucial role in viewing/exporting the logs by the customer.

4. Risks

- Full URL filtering is only possible with non-SSL traffic. As penetration of TLS 1.3 increases across the Internet over the next 12 months, certain elements of SNI used for domain-name filtering may no longer be usable.

4 5. Functional overview

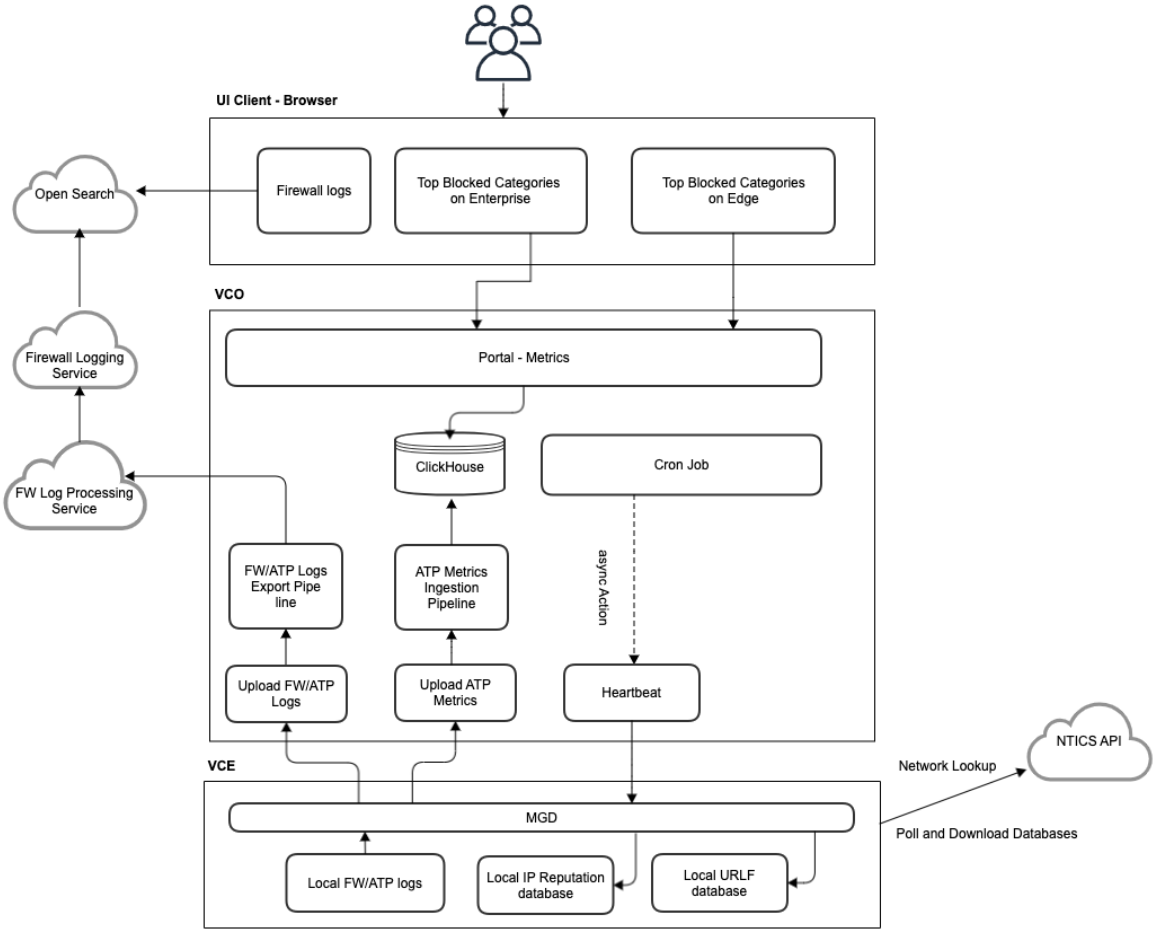
This section should minimally address the following

SDWAN VCO and Edges will integrate with NSX Threat Intelligent Cloud Service (NTICS) to deliver Enhanced Firewall Service capabilities like URL Filtering and IP Reputation. NTICS is a service hosted on AWS that provides data such as URL categories, URL and IP reputations, IDS/IPS signature bundles, etc. NTICS mimics the APIs exposed by Webroot's BrightCloud Threat Intelligence Service. NTICS acts as proxy/caching layer before Webroot's cloud service.

Webroot's BrightCloud Web Classification Service provides one of the broadest website intelligence and classification across 82 website categories. Webroot provides:

1. **URL Filtering and IP Reputation database**
2. **URL Filtering Categories**
3. **SDK** that can be integrated with on-prem deployment to provide URL/IP lookup
4. **Cloud Service** to download the latest Webroot's URL Filtering and IP Reputation database and perform lookups

The SDWAN VCO shall provide the required license to the edges that enables edges to authenticate with NTICS. Webroot SDK running on edges will be configured to talk to NTICS and download URL filtering and IP reputation database using Webroot SDK.



4.1 5.1. Configuration (PRD URL 1.5, URL 1.6, URL 1.7)

For configuration, we are adding the ability for customers to:

- Enable IP reputation and/or URL filtering features using the SD-Wan UI, within "Configure" > "Edge" OR "Profile"
- Enable IP reputation and/or URL filtering features within their firewall rules using the SD-Wan UI, within "Configure" > "Edge" OR "Profile" > "Firewall" > "Firewall Rules"

Disabling EFS

In order to disable EFS for a customer, it is not enough to toggle the EFS setting to off in the Global Settings app, this will only disable the EFS features in the UI. These features must first be disabled in the SD-WAN app within "Configure" > "Edge" OR "Profile" > "Firewall" (toggle EFS off). There is a current known bug where disabling EFS at the global level does not actually turn off EFS if it is still enabled at the Profile level. To counter this, the UI will only allow disabling EFS if it is disabled at all its dependent components (all rules on a profile, etc).

Disabling ATP

In order to disable EFS for a customer, it is not enough to toggle the EFS setting to off in the Global Settings app, this will only disable the EFS features in the UI. These features must first be disabled in the SD-WAN app within "Configure" > "Edge" OR "Profile" > "Firewall" (toggle EFS off)

4.1.1 5.2 Firewall Log Ingestion (PRD LR 2.5)

The addition of URL Filtering and IP Reputation requires adding several new fields to the Protobuf definition of a Firewall log. The underlying pipeline of ingesting logs will not change from Woodford other than adding support for these new fields.

4.2 5.3 Monitoring (PRD LR 1.4)

Aggregate firewall EFS metrics in various time interval, e.g. 5-minutes, will be powered by the ClickHouse tables. Visualizing the raw logs will be powered by Search Service querying OpenSearch. URL Filtering and IP Reputation metrics will each be powered by their own respective aggregate tables. These 5-minute aggregate tables are all derived from the shared VELOCLOUD_FIREWALL_STATS table. The schema of this table will be updated to include the new fields necessary for URL Filtering and IP Reputation, such as URL Category or IP Reputation score. These changes will require small updates to the Protobuf definitions as well as the backend job processing the uploaded logs. No new components or dependencies are added other than new ClickHouse aggregate tables for URL Filtering and IP Reputation.

4.3 5.4. URL Filtering Database Downloads by the Edge (PRD URL 1.7)

NTICS service provides REST APIs to poll the latest version of URL Filtering databases and to download them.

The Webroot SDKs on the Edges will pull updated versions of the databases directly from NSX via NTICS APIs, so there is no need for the GSM or VCO components to request or store these files.The polling frequency can be configured in SDK configuration.

5.5. IP Reputation Database Downloads by the Edge (PRD MP 1.6)

Downloading IP Reputation databases uses a flow that is mostly the same as for URL Reputations, but without the need for a separate NTICs call to download a Category list file. As with URL Reputation, the IP Reputation database is downloaded directly by the Webroot SDKs included on the Edges. Notably, the default for NSX is to not request RTUs of IP Reputation, relying only on the daily full database versions. Edges will notify VCO when a new version of the database is downloaded via events in the MGD heartbeat.

4.4 5.6 NTICS License Key

In order to download database of URL Reputation and the IP Reputation, the Edges will have to register to NTICS and A license key is a prerequisite.

Refer to the link[6] for the design idea of license key provision and validation.

6 https://confluence.eng.vmware.com/display/VELOENG/URL+Filtering+and+IP+Reputation+High+Level+Design#URLFilteringandIPReputationHighLevelDesign-URLFilteringandIPReputation(PostXante)

5 6. Backward compatibility

6.1. Configuration

The atp_enabled flag will be retained in the heartbeat response of firewall module for backwards compatibility. However, each feature will be controlled by a separate configuration parameter, one for IDPS, one for URL Filtering, one for IP Reputation.

6.2. Monitoring (PRD LR 1.4)

New fields to support URL Filtering and IP Reputation will be added to the ClickHouse VELOCLOUD_FIREWALL_STATS table and the derived tables.

6 7. Security impact

Edge Firewall EFS Url Filtering Threat Modeling (see page 78)

7 8. Platform & system dependencies

The MGD process on Edges will make calls into NSX's NTICS APIs. A service on GSM registers enterprises with NTICS and receives a license key. This license key is propagated to the Edges, which then use this license to authenticate with NTICS for URL Filtering and IP Reputation. This license is only generated when URL Filtering and/or IP Reputation are enabled.

8 9. API, Events and System properties

9.1. System properties

A new system properties denotes the NTICS public endpoint, which must be sent in the configuration heartbeat to all Edges.

```
INSERT IGNORE INTO VELOCLOUD_SYSTEM_PROPERTY
    (name, value, description, isReadOnly, isPassword, dataType)
VALUES
    ('ntics.public.endpoint.address', '', 'Public endpoint address of NSX Cloud
threat intelligence service', false, false, 'STRING')
```

A new backend job will be created to periodically poll NTICS for the list of Url Categories. The frequency of the invocation of this job will be controlled by a system property, following the style of pollIDPSSignatureFromGSM.js.

8.1 9.2. New APIs

8.1.1 9.2.1. GSM APIs for VCO

Sl. No.	PRD Item	Description	Endpoint URL	Notes
1	ATPD 1.3	VCO request from GSM to create/update License Key	POST https://{GSM}/firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses	Called when EFS is enabled/ disabled in global settings on VCO
2	ATPD 1.3	VCO notify GSM to update the status for License Key	PATCH https://{GSM}//firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses/{licenseLogicalId}	Called by VCO to mark license as inactive when EFS is disabled.
3	ATPD 1.3	NTICS call into GSM to validate the License key	POST https://{GSM}/firewallAtp/v1/licenses/validate	Called by NTICS to validate the license key

Sl. No.	PRD Item	Description	Endpoint URL	Notes
4	ATPD 1.3	VCO notify GSM to delete the License	Delete https://{GSM}//firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses/{licenseLogicalId}	Called by VCO when an Enterprise is deleted.
5	URL 1.6	GSM exposed API for VCO to download URL Filtering Category List	https://{gsm-endpoint}/firewallAtp/v1/urlFiltering/categories	A proxy for NTICS getcatlist API, called by a backend job.

8.1.2 9.2.2. VCO APIs for Edges

Sl.No.	PRD Item	Description	Endpoint URL	Notes
1		Send Firewall logs to VCO **(Pre Woodford Edges)**	/upload/firewallLogsUpload	For backwards compatibility
2	LR 1.1	Send Firewall + EFS logs to VCO **(Woodford /Xante Edges)**	/upload/firewallLogsUploadV2	Same endpoint as before, but with additional fields in the Protobuf
3	LR 1.4	Send EFS logs to VCO for Monitoring **(Woodford/Xante Edges)**	/upload/firewallStatsUpload	Same endpoint as before, but with additional fields in the Protobuf

8.1.3 9.2.3. Portal APIs

Sl. No.	PRD Item	Description	Endpoint URL	Notes
1	LR 1.4	Get Firewall Url Filtering Metrics at Enterprise level	/portal/metrics/getEnterpriseFirewallUrlCategoryMetrics /portal/metrics/getEnterpriseFirewallUrlReputationMetrics	Refer to the detail design Monitoring
2	LR 1.4	Get Firewall Url Filtering Metrics at Edge level	/portal/metrics/getEdgeFirewallUrlCategoryMetrics /portal/metrics/getEdgeFirewallUrlReputationMetrics	Refer to the detail design Monitoring
4.	MP 1.6	Get Firewall IP Reputation Metrics at Enterprise level	/portal/metrics/getEnterpriseFirewallMaliciousIpMetrics	Refer to the detail design Monitoring
5	MP 1.6	Get Firewall IP Reputation Metrics at Edge level	/portal/metrics/getEdgeFirewallMaliciousIpMetrics	Refer to the detail design Monitoring
6	LR 1.4	Get Summary Metrics across Edges	/portal/metrics/getEnterpriseFirewallEdgeSummaryMetrics	Refer to the detail design Monitoring
7	LR 1.4	Get Edge Counts	/portal/metrics/getEnterpriseFirewallEdgeCountMetrics	Refer to the detail design Monitoring
6	URL 1.6	Get URL Category List	/portal/firewall/getUrlFilteringCategories	Maps ids to strings for UI readability

8.2 See URL/IP Filtering Query APIs - VeloCloud Engineering - VMware Core Confluence[7] for Request/Response schemas.

8.3 9.3. Events

MGD_* events are generated by the Edges, and sent to the VCO as part of the heartbeats. VCO_* are generated in the VCO. It is the responsibility of the Edge to listen to the callbacks from the Webroot SDK to generate some events. Not all events will contain details (e.g. VCO_ENTERPRISE_NTICS_LICENSE_REQUEST_SUCCEEDED).

Edge Events:

MGD_EFS_NTICS_AUTHENTICATE_FAILED
MGD_EFS_NTICS_AUTHENTICATE_SUCCEEDED

MGD_EFS_URL_DB_VERSION_UPDATE
MGD_EFS_IP_DB_VERSION_UPDATE

VCO Events:
VCO_ENTERPRISE_NTICS_LICENSE_REQUEST_FAILED
VCO_ENTERPRISE_NTICS_LICENSE_REQUEST_SUCCEEDED
URL_CATEGORIES_STORE_SUCCESS
URL_CATEGORES_STORE_FAILURE

7 https://confluence.eng.vmware.com/pages/viewpage.action?pageId=1845286339

9 10. Upgrade & Migrations

10.1 License key (PRD URL 1.1)

The event of VCO requesting for license key is triggered by config setting change, i.e. EFS. For the VCOs EFS setting have been enabled in the previous release, one patch will be required and executed during VCO upgrading.

The new patch will do,

1. Find out the enterprises with EFS enable
2. Request license key for all enterprises found in #1. It will be desirable to make one API call.
3. Propagate the license keys to the Edges. It should be asynchronous manner, e.g. make necessary setup and the Edge will see the config change at the following Heartbeat message.

Woodford and Pre-woodford edges will ignore license key configuration. Post upgrade to version above woodford, the edges will read the license key and NITCS endpoint configuration from configuration file and obtain client id and secret.

10.2 IDPS object groups

This feature plans to introduce four new object groups for IDPS, URL Filtering and URL reputation and IP filtering which is different from the existing IDPS configuration . This requires the IDPS configuration to be translated to object groups during upgrades from prior releases.

- A **patch will be introduced to create four different IDPS object groups based on the configuration from the below table.**

IDS	IPS	Log
Enabled	Disabled	Disabled
Enabled	Disabled	Enabled
Enabled	Enabled	Disabled
Enabled	Enabled	Enabled

- **Four new security service groups will be created to which the new IDPS object groups are associated.**
- **These security service groups are associated to firewall rules which had ATP enabled prior to upgrade.**
- All the keys inside **atp_action** of firewall module will be derived from IDPS object groups and sent to all edges.
    - **ids_enabled is derived from idsEnabled.**

- **ips_enabled is derived from ipsEnabled.**
- **atp_logging_enabled is derived from logEnabled.**
- **atp_enabled should be set to true.**

10 11. Operations impact / Supportability

10.1 11.1 NTICS Authenticate with GSM

1. There will be a new VCE role certificate introduced in GSM for allowing NTICS to authenticate with GSM Firewall-Atp service.
2. The new role will be with limited privilege to a few API calls, e.g. license validation API.
3. Three pairs of certificates and keys will be created and provided for NTICS to access GSM service at test/preprod/prod envs respectively. The common name of the certificates is ntics_firewall_atp::VCE.
4. The certificate expiration is 1 year after issued, it has to be provided in manual way right now.

10.2 11.2 GSM Service Monitoring

TBD

11 12. Scale impact

This section should minimally address the following

License keys managed by GSM are stored in cloud storage. EFS logging infrastructure in VCO file_store will see a small increase in file sizes and memory on VCOs and Clickhouse will have 2 extra aggregate tables, between a 1-3x increase in Clickhouse storage requirements.

12 13. Detailed design & implementation

This section should minimally address the following

- [ ] **Have separate sections for orchestrator, Control Plane and Data Plane changes.**
- [ ] **Describe the design of this feature as it applies to the Velocloud ecosystem.**
- [ ] **Are there any caveats or limitations? What were the alternative approaches considered? What are the tradeoffs in the selected approach?**
- [ ] **Explain with sequence & interaction diagrams, if possible.**
- [ ] **Describe all components & modules that will be touched by this feature and explicitly call out the before and after behavior of those components in relation to this feature.**
- [ ] **Are there new modules being introduced? Explain the reasoning for this module.**

VCO

- [ ] **If this is not a UI specific project, are there any UI changes? If so, does that need its own spec or is it captured here?**
- [ ] **Are there any special considerations for the on-prem version of VCO?**
- [ ] **Are there any failure modes for the implementation like degraded operation etc.?**
- [ ] **Are there new services being introduced? If so, call them out and explain their design in detail.**
- [ ] **List out the dependencies on the Data/Control plane.**
- [ ] **Indicate the database schema design if it applies.**
- [ ] **Should the newly-added data model be considered for Enterprise Cloning feature and Customer Migration tool?**
- [ ] **Is there a scope of role customization based on requirements in PRD? If yes, call out the privileges, UI fields to hide and explain in detail about the standard roles to be granted/denied.**
  Define privilege labels and description in i18n file.

DP

- [ ] **Type your task here, using "@" to assign to a user and "//" to select a due date**
- [ ] **List out the dependencies on the Management Plane including UI/UX changes if needed.**

12.1 13.1. Config (PRD URL 1.6, URL 1.8, URL 1.9, URL 1.14)

For configuration, additional toggle buttons with be introduced for URL filtering and IP reputation at profile and edge level(override).The ability for customers to enable individual EFS features on their firewall rules will also be introduced. Each EFS feature can be toggled independently on firewall rules.Although EFS features can be updated separately from each other, and each Action and API concerns only a single EFS feature, multiple actions can be triggered in a single VCO heartbeat, if multiple updates are needed simultaneously.

13.1.1 Url Category List

Categories will be a list of the catids as defined by the Webroot getcatlist API. A single Url Group can select multiple categories. The categoryId 0 has a special meaning: that the URL is uncharacterized. When a Url lookup is unknown, the Webroot SDK returns a categoryId of 0. When downloading the Url Category List to the VCO, the "Unknown" category is prepended to the list of categories, and is included in the response to the Portal API. The SDK on the Edges does not need the "Unknown" category included with the list, but the Portal API will receive it to populate the UI.

Furthermore, some EFS features require the use of databases for performing lookups. URL Filtering and IP Reputation databases are downloaded daily in full by edges.

EFS may be disabled on a per-profile, per-edge, and per-rule basis. A customer-level configuration option for disabling EFS is currently non-operative, a known bug as of Xante release. Disabling EFS on a profile will override the settings of any edges/rules on that profile. Likewise, disabling EFS on an edge will apply for all rules on that edge, even if EFS is enabled on the rules. For consistency/clarity, the UI will only allow disabling EFS for a Profile when all rules have disabled EFS.

Firewall rules can also filter traffic by minimum Url Reputation score, regardless of the domain or category. Reputation scores range between 0 and 100, with 100 being most trustworthy. The rule will match against all addresses that have lower reputation than the configured minimum.

12.1.1 13.1.2 Config Table VELOCLOUD_FIREWALL_RULE

The firewall rule table in MySQL does not need to have an explicit schema change, but the data field of the rule will have new fields for supporting security service group.

Column	Type	Description
id	number	
created	timestamp	
deactivated	timestamp	
logicalId	uuid	
type	enum	
name	string	
data	JSON	**Configuration of firewall rule, includes references to new security service group.**
edgeLogicalId	uuid	
enterpriseId	number	
segmentObjectId	number	
segmentName	string	

12.1.2 13.1.2 HB Response with configurationUpdate action for ATPMetadata module when URL filtering/IP reputation is enabled at profile/Edge

This HB is sent in 2 scenarios:

- When IP Reputation / URL filtering is enabled at profile/edge level.
- When NTICS endpoint is updated.

As part of HB response the below fields are sent.

- **atpUpdateEnabled** flag was used in Woodford to indicate if IDS/IPS was enabled at profile level/ edge override.
- ntics license key information is sent to Edges

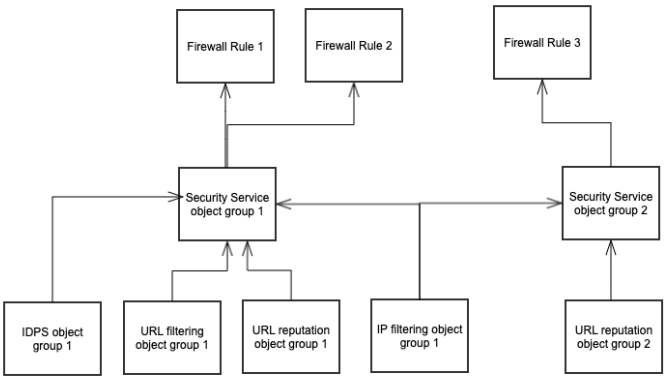
HeartBeat Response

```
    "actions": [{
        "action": "configurationUpdate",
        "data": {
            "module": "atpMetadata",
            "version": "1614693596464",
            "schemaVersion": "3.0.0",
            "use": {
                "atpUpdateEnabled" : true,
                "ntics": {=====================================================> New
key added to send license information to edge
                    "licenseLogicalId": "33f8a91b-324a-11ee-9dca-0e813ba16025",
                    "licenseKey": "ju8d228c-k22w-ur70-vw23-0242ac120002",
                    "endpoint" : "https://test-ntics.com",
                    "deviceType": "SDWAN-Edge",
                    "registerAPI": "/1.0/auth/register",
                    "authenticateAPI": "/1.0/auth/authenticate",
                    "version": "1614693596464",
                }
            }
        }
    }]
} }
```

12.1.3 13.1.3 Security Service Groups

- This feature will introduce new object group configurations for individual security engines namely IDPS, URL filtering , URL reputation and malicious IP detection(IP filtering).
- Enterprise objects will be created for individual engines and added to VELOCLOUD_ENTERPRISE_OBJECT table (each engine having its own type described below).
- Additionally, Security service groups which are also an object type, will be introduced, that can be used to group together the individual object groups and can be associated with a firewall rule.

- The data inside the security service object will hold references i.e. logical IDs for the individual security engines.
- The configuration of all the newly introduced object groups will be sent as part of firewall module.
- When security service groups config is changed, this will update firewall module version and updated object group config is sent to edge.
- When security service group is associated with firewall rule, then firewall module version would change and updated rule config is sent to edge.
- A single security service group can be associated with multiple firewall rules. A single security engine object group like IDPS can be associated with multiple security service groups.
- Not more than one security service group can be associated with a firewall rule.



12.1.3.1 13.1.3.1. Enterprise object to represent IDPS groups

- When security service groups are created on the UI, the UI will create enterprise objects which are stored in VELOCLOUD_ENTERPRISE_OBJECT table, similar to address groups and port groups.
- New enterprise object types called **idps, url_filtering, ip_reputation, malicious_ip_detection** will be introduced for the individual engines. The **security_groups** object type is for the complete selection of enterprise objects for the engines.

Column	Type	Description
id	number	1
created	timestamp	'2023-12-10T00:30:50.000Z'
operatorId	timestamp	NULL

Column	Type	Description
networkId	uuid	NULL
enterpriseId	enum	56
edgedId	string	NULL
gatewayId	JSON	NULL
parentGroupId	uuid	NULL
description	number	**"store IDPS policies "**
object	number	PROPERTY
name	string	**idps-1**
type	string	**idps**
logicalId	type	**'87e768fb-a0d4-4a34-a094-7b7a1179c80c'**
alertsEnabled	Boolean	1
operatorAlertsEnabled	Boolean	1
status	text	NULL
statusModified	timestamp	'0000-00-00 00:00:00'
previousData	mediumtext	NULL
previousCreated	timestamp	'0000-00-00 00:00:00'

Column	Type	Description
draftData	mediumtext	NULL
draftCreated	timestamp	'0000-00-00 00:00:00'
draftComment	string	NULL
data	mediumtext	`{ "idsEnabled": true/false, "ipsEnabled" : true/false, "logEnabled" : true/false }`
lastContact	timestamp	'0000-00-00 00:00:00'
version	string	'0'
modified	timestamp	'0000-00-00 00:00:00'

12.1.3.2 **13.1.3.2. Enterprise object to represent URL filtering**

Column	Type	Description
id	number	1

Column	Type	Description
created	timestamp	'2023-12-10T00:30:50.000Z'
operatorId	timestamp	NULL
networkId	uuid	NULL
enterpriseId	enum	56
edgedId	string	NULL
gatewayId	JSON	NULL
parentGroupId	uuid	NULL
description	number	**"Store URL filtering policies"**
object	number	PROPERTY
name	string	**url-filtering-1**
type	string	**urlCategoryFiltering**
logicalId	type	**'87e768fb-a0d4-4a34-a094-7b7a1179c80d'**
alertsEnabled	Boolean	1
operatorAlertsEnabled	Boolean	1
status	text	NULL
statusModified	timestamp	'0000-00-00 00:00:00'

Column	Type	Description
previousData	mediumtext	NULL
previousCreated	timestamp	'0000-00-00 00:00:00'
draftData	mediumtext	NULL
draftCreated	timestamp	'0000-00-00 00:00:00'
draftComment	string	NULL

data	mediumtext	{
		"monitorCategories" : [],
		"blockedCategories" : [],
		"unknownCategoryAction" : "allow/block"
		}

lastContact	timestamp	'0000-00-00 00:00:00'
version	string	'0'
modified	timestamp	'0000-00-00 00:00:00'

12.1.3.3 **13.1.3.3. Enterprise object to represent URL reputation**

Column	Type	Description
id	number	1
created	timestamp	'2023-12-10T00:30:50.000Z'
operatorId	timestamp	NULL
networkId	uuid	NULL
enterpriseId	enum	56
edgedId	string	NULL
gatewayId	JSON	NULL
parentGroupId	uuid	NULL
description	number	**"store URL reputation policies "**
object	number	PROPERTY
name	string	**url-reputation-1**
type	string	**urlReputationFiltering**
logicalId	type	**'87e768fb-a0d4-4a34-a094-7b7a1179c80e'**
alertsEnabled	Boolean	1

Column	Type	Description
operatorAlertsEnabled	Boolean	1
status	text	NULL
statusModified	timestamp	'0000-00-00 00:00:00'
previousData	mediumtext	NULL
previousCreated	timestamp	'0000-00-00 00:00:00'
draftData	mediumtext	NULL
draftCreated	timestamp	'0000-00-00 00:00:00'
draftComment	string	NULL

data	mediumtext	{
		"minReputationScore" : 0-100,
		"monitorReputations" : [0-4],
		"unknownCategoryAction" : "allow/block"
		}

lastContact	timestamp	'0000-00-00 00:00:00'
version	string	'0'

Column	Type	Description
modified	timestamp	'0000-00-00 00:00:00'

12.1.3.4 13.1.3.4. Enterprise object to represent Malicious IP filtering

Column	Type	Description
id	number	1
created	timestamp	'2023-12-10T00:30:50.000Z'
operatorId	timestamp	NULL
networkId	uuid	NULL
enterpriseId	enum	56
edgedId	string	NULL
gatewayId	JSON	NULL
parentGroupId	uuid	NULL
description	number	**"store malicious IP filtering policies "**
object	number	PROPERTY
name	string	**mal-ip-filtering-1**
type	string	**maliciousIpFiltering**

Column	Type	Description
logicalId	type	**'87e768fb-a0d4-4a34-a094-7b7a1179c80f'**
alertsEnabled	Boolean	1
operatorAlertsEnabled	Boolean	1
status	text	NULL
statusModified	timestamp	'0000-00-00 00:00:00'
previousData	mediumtext	NULL
previousCreated	timestamp	'0000-00-00 00:00:00'
draftData	mediumtext	NULL
draftCreated	timestamp	'0000-00-00 00:00:00'
draftComment	string	NULL
data	mediumtext	`{ "action" : "monitor/block" }`
lastContact	timestamp	'0000-00-00 00:00:00'

Column	Type	Description
version	string	'0'
modified	timestamp	'0000-00-00 00:00:00'

12.1.3.5 **13.1.3.5. Enterprise object to store security service groups**

Column	Type	Description
id	number	1
created	timestamp	'2023-12-10T00:30:50.000Z'
operatorId	timestamp	NULL
networkId	uuid	NULL
enterpriseId	enum	56
edgedId	string	NULL
gatewayId	JSON	NULL
parentGroupId	uuid	NULL
description	number	**"store security service groups"**
object	number	PROPERTY
name	string	**security-groups-1**

Column	Type	Description
type	string	**securityServiceGroup**
logicalId	type	**'87e768fb-a0d4-4a34-a094-7b7a1179c80g'**
alertsEnabled	Boolean	1
operatorAlertsEnabled	Boolean	1
status	text	NULL
statusModified	timestamp	'0000-00-00 00:00:00'
previousData	mediumtext	NULL
previousCreated	timestamp	'0000-00-00 00:00:00'
draftData	mediumtext	NULL
draftCreated	timestamp	'0000-00-00 00:00:00'
draftComment	string	NULL

Column	Type	Description

data	mediumtext	{
		"idps": {
		"logicalId" : "87e768fb-a0d4-4a34-a094-7b7a1179c80c",
		"id" : 1
		},
		"urlCategoryFiltering" : {
		"logicalId" : "87e768fb-a0d4-4a34-a094-7b7a1179c80d",
		"id" : 2
		},
		"urlReputationFiltering" : {
		"logicalId" : "87e768fb-a0d4-4a34-a094-7b7a1179c80e",
		"id" : 3
		},
		"maliciousIpFiltering" : {
		"logicalId" : "87e768fb-a0d4-4a34-a094-7b7a1179c80f",
		"id" : 4
		}
		}

lastContact	timestamp	'0000-00-00 00:00:00'
version	string	'0'
modified	timestamp	'0000-00-00 00:00:00'

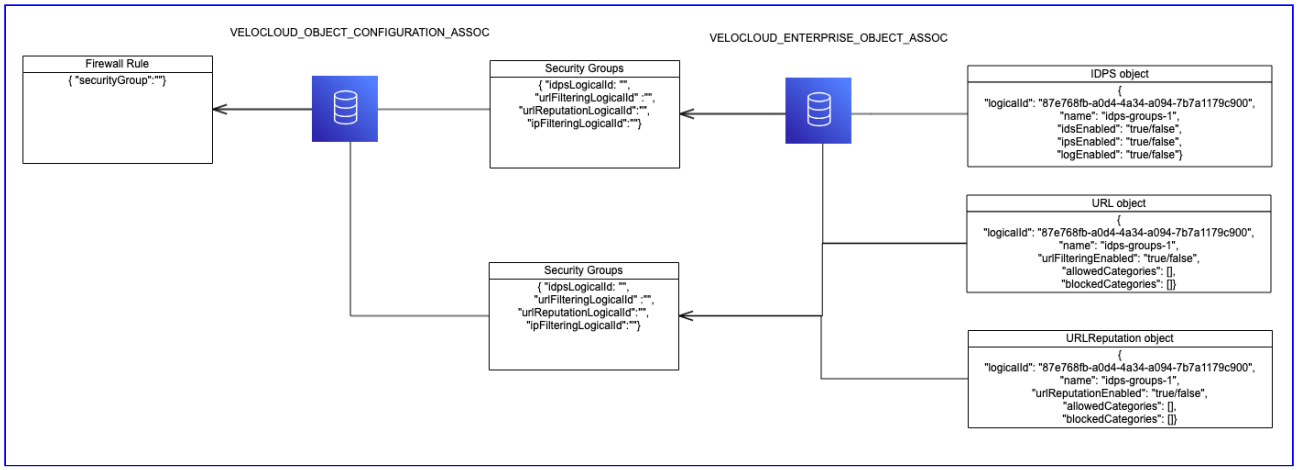
13.1.3.6 Storing associations between IDPS/URL/IP filtering object groups and security service groups

- When security service group is associated with a firewall rule in a profile, an association is created between the object and profile and stored in the existing VELOCLOUD_OBJECT_CONFIGURATION_ASSOC table.

- The associations between the individual engines's object groups (IDPS object group, URL filtering object group etc) and security object group is maintained through newly introduced ASSOC table VELOCLOUD_ENTERPRISE_OBJECT_ASSOC

```
CREATE TABLE VELOCLOUD_ENTERPRISE_OBJECT_ASSOC {
    id                      BIGINT NOT NULL AUTO_INCREMENT,
    enterpriseId   BIGINT NOT NULL,
    parentObjectId          BIGINT NOT NULL, // security service group object id
    childObjectId           BIGINT NOT NULL, // URL Filtering/IDPS/IP filtering
object id
    type                    ENUM ()
    ref                     VARCHAR(255), //
"objectGroup:ssg:idps/"objectGroup:ssg:urlCategoryFiltering"
    FOREIGN KEY             (parentObjectId) references
VELOCLOUD_ENTERPRISE_OBJECT(id) ON DELETE CASCADE,
    FOREIGN KEY  (enterpriseId) REFERENCES VELOCLOUD_ENTERPRISE(id) ON DELETE
CASCADE,
    PRIMARY KEY  (id)
 }
```

|



12.1.3.6 13.1.3.7 Storing security service groups in refs in firewall module and rendering it to UI

- When security groups are associated to a firewall rule the object's data is stored as a reference in firewall modules and sent to UI as part of getConfiguration API.
- The individual security engines's objects are also resolved as shown below.

```
{
```

```
    "modules": [
        {
            "id": 65,
            "created": "2023-06-22T23:11:49.000Z",
            "name": "firewall",
            "type": "ENTERPRISE",
            "description": null,
            "schemaVersion": "3.0.0",
            "version": "1689891242740",
            "configurationId": 12,
            "enterpriseLogicalId": "fd200ac2-80b0-4542-a26c-0d6cb83d2894",
            "data": {
                "module": "firewall",
                "schemaVersion": "3.0.0",
                "version": "1689863253297",
                "use": {
                    "firewall_enabled": true,
                    "segments": [
                        {
                            "segment": {
                                "segmentId": 0,
                                "name": "Global Segment",
                                "type": "REGULAR",
                                "segmentLogicalId": "b86a934c-941f-49c7-
a40f-52f32a8f8ece"
                            },
                            "firewall_logging_enabled": false,
                            "outbound": [
                                {
                                    "name": "Rule-1",
                                    "match": {
                                        "os_version": -1,
                                        "sInterface": "",
                                        "s_rule_type": "prefix",
                                        "sip": "any",
                                        "sipV6": "any",
                                        "ssm": "255.255.255.255",
                                        "smac": "any",
                                        "svlan": -1,
                                        "sport_high": -1,
                                        "sport_low": -1,
                                        "dvlan": -1,
                                        "dInterface": "",
                                        "dip": "any",
                                        "dipV6": "any",
                                        "dsm": "255.255.255.255",
                                        "hostname": "",
                                        "proto": -1,
                                        "dport_high": -1,
                                        "dport_low": -1,
                                        "d_rule_type": "prefix",
                                        "classid": -1,
                                        "dscp": -1,
```

```
                                        "appid": -1,
                                        "ipVersion": "IPv4v6"
                                    },
                                    "action": {
                                        "allow_or_deny": "allow"
                                    },
                                    "atp_action": {
                                        "atp_enabled": false,
                                        "ids_enabled": false,
                                        "ips_enabled": false,
                                        "atp_logging_enabled": false
                                    },
                                    "securityServiceGroup": "87e768fb-a0d4-4a34-
a094-7b7a1179c80g",-----------> Will store security service group logical id
                                    "ruleLogicalId": "RnhdWnn2TL+O1Dz0iIXOWA",
                                    "comments": null,
                                    "loggingEnabled": false
                                }
                            ]
                        }
                    ],
                    "atp_enabled": null,=====================> introduced in
woodford and will be used to represent if EFS is enabled at profile level
                    "securityFeatures" : {
                      "idpsEnabled": true/false/null,   /*Per feature knob*/
                      "urlFilteringEnabled" : true/false/null,
                      "maliciousIpFilteringEnabled" : true/false/null
                    }
                },
     "refs" : {
            "objectGroup:securityServiceGroup": [
                {
                    "id": 11551,
                    "enterpriseObjectId": 289,
                    "configurationId": 12,
                    "moduleId": 65,
                    "segmentObjectId": null,
                    "ref": "objectGroup:securityServiceGroup",
                    "data": [
                        {
                            "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80g",
                            "name": "security-groups-2",
                            "idps": {
                                "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80h",
                                "name": "idps-groups-1",
                                 "data": {
                                    "idsEnabled": true/false,
                                    "ipsEnabled": true/false,
                                    "logEnabled": true/false
                                }
                            },
                            "urlCategoryFiltering": {
                                "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80i",
```

```
                                "name": "urlf-groups-1",
                                "data": {
                                    "monitorCategories": [],
                                    "blockedCategories":[],
                                    "unknownCategoryAction" : "allow/block"
                                }
                            },
                            "urlReputationFiltering": {
                                "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80j",
                                "name": "urlr-groups-1",
                                 "data": {
                                    "minReputationScore": 0-100,
                                    "monitorReputations" : [0-4],
                                    "unknownCategoryAction" : "allow/block"
                                 },
                             },
                            "maliciousIpFiltering": {
                                "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80k",
                                "name": "ipf-groups-1",
                                "data": {
                                    "action": "monitor/block"
                                }
                            }
                        ],
                        "modified": "2023-07-20T14:35:52.000Z",
                        "version": "0",
                        "object": "PROPERTY",
                        "name": "test-group1",
                        "type": "security_group",
                        "logicalId": "e3bd6591-baa1-40e4-9189-ba77bfb572bc",
                        "parentGroupId": null,
                        "segmentLogicalId": null
                    }
                ]
            }
        }
    ]
}
```


12.1.3.7 13.1.3.8 HB Response with configurationUpdate action for firewall module when security service group is associated to a firewall rule/ Security service group config is changed

- When security service group's association to a firewall rule changes or when security group config is modified, firewall module version is updated and configurationUpdate action is sent for firewall module.
- All the object groups's configuration is flattened and sent in HB.

HeartBeat Response


```
{
    "actions": [{
        "action": "configurationUpdate",
        "data": {
            "module":"Firewall",
            "version":"1369786365000",
            "schemaVersion":"1.0.0",
            "use": {
                "firewall_enabled": true,
                "firewall_logging_enabled": true,
                "atp_enabled": true,
                "securityFeatures" : {
                   "idpsEnabled" : true,
                   "urlFilteringEnabled" : true,
                   "maliciousIpFilteringEnabled" : true
                },
                "inbound": [],
                "stateful_firewall_enabled": false,
                "syslog_forwarding": false,
                "segments": [{
                    "segment": {
                        "segmentId": 0,
                        "name": "Global Segment",
                        "type": "REGULAR",
                        "segmentLogicalId": "0f966250-942d-48cc-be87-d656734f6449"
                     },
                     "firewall_logging_enabled": false,
                     "atp_enabled": true,
                     "outbound": [{
                        "name": "Block Google DNS",
                        "match": {
                            "ipVersion":"IPv4v6",
                            "appid": -1,
                            "classid": -1,
                            "dscp": -1,
                            "sip": "any",
                            "smac": "any",
                            "sport_high": -1,
                            "sport_low": -1,
                            "ssm": "255.255.255.255",
                            "svlan": -1,
                            "os_version": -1,
                            "hostname": "",
                            "dip": "8.8.8.8",
                            "dport_low": 53,
                            "dport_high": 53,
                            "dsm": "255.255.255.255",
                            "dvlan": -1,
```

```
                            "proto": 6,
                            "s_rule_type": "prefix",
                            "d_rule_type": "exact",
                        },
                        "action": {
                            "allow_or_deny": "allow",
                        },
                       "atp_action": {
                            "atp_enabled": true,/*Retained for backward
compatibility*/
                            "ids_enabled": true,
                            "ips_enabled": true,
                            "atp_logging_enabled": true,
                         },
                         "urlCategoryFiltering" : {
                            "monitorCategories": [], //list of category IDs
                            "blockedCategories": [], //list of category IDs
                            "unknownCategoryAction" : "allow/block"
                         },
                         "urlReputationFiltering" : {
                             "minReputationScore" : 0-100,
                             "monitorReputations" : [0-4] ,
                             "unknownCategoryAction" : "allow/block"
                          },
                         "maliciousIpFiltering" : {
                             "action" : "monitor/Block"
                         }
                    }]
                }]
            }
        }
    }]
}
```

12.1.3.8 13.1.3.9 Portal APIs to configure security service groups

Operation	API endpoint	Parameters

Insert and Update	'enterprise/insertSecurityObject'
	'enterprise/updateSecurityObject'
		{
		"name": "test-1",
		"description": null,
		"enterpriseId": 1,
		"data": {
		"idsEnabled": true,
		"ipsEnabled": true,
		"logEnabled ": true
		},
		"type": "idps"
		}
		{
		"name": "test-3",
		"description": null,
		"enterpriseId": 1,
		"data": {
		"monitorCategories": [3,5],
		"blockedCategories": [40,50],
		"unknownCategoryAction": "allow/block"
		},
		"type": "urlCategoryFiltering"
		}
		{
		"name": "test-3",
		"description": null,
		"enterpriseId": 1,
		"data": {
		"action" : "monitor/block"

Operation	API endpoint	Parameters

		`}, "type": "maliciousIpFiltering" }`
		
		`{ "name": "test-4", "description": null, "enterpriseId": 1, "data": { "minReputationScore": 0-100, "monitorReputations": [0-4], "unknownCategoryAction" : "allow/block" }, "type": "urlReputationFiltering" }`
		
		`{ "name": "test-5", "description": null, "enterpriseId": 1, "data": { "idps" : { "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80c", "id" : 1 } "urlCategoryFiltering" : { "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80d", "id" : 2 } "urlReputationFiltering" : { "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80e", "id" : 3 } "maliciousIpFiltering" : { "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80f", "id" : 4 } }, "type": "securityServiceGroup" }`

Operation	API endpoint	Parameters
Delete	'enterprise/deleteSecurityObject'	`{ "id": 70, "enterpriseId": 1 }`
Get	'enterprise/getSecurityObjects'	`{ "enterpriseId": 1, "type": "idps" | "urlCategoryFiltering" | "urlReputationFiltering" | "maliciousIpFiltering" | "securityServiceGroup" "with": ["securityGroups"] | ["profiles"] }` securityGroups adds a field "securityGroups" to each element in the result that is an array of {id, name} objects of the securityServiceGroups each securityObject is a member of. profiles adds two fields to each object, "profileCount" and "edgeCount". Both are simple integers representing how many profiles or edges reference each securityServiceGroup object.

For example, if a customer has an edge that uses the security group "group-1" which includes the "idps-1" security object, then the request/response is:

request	response

{
enterpriseId: 1,
type: "idps"
}
	[{
	id: 2,
	enterpriseId: 1,
	name: idps-1,
	type: idps,
	data: {
	idsEnabled: true,
	ipsEnabled: true,
	logEnabled: true
	}
	}]

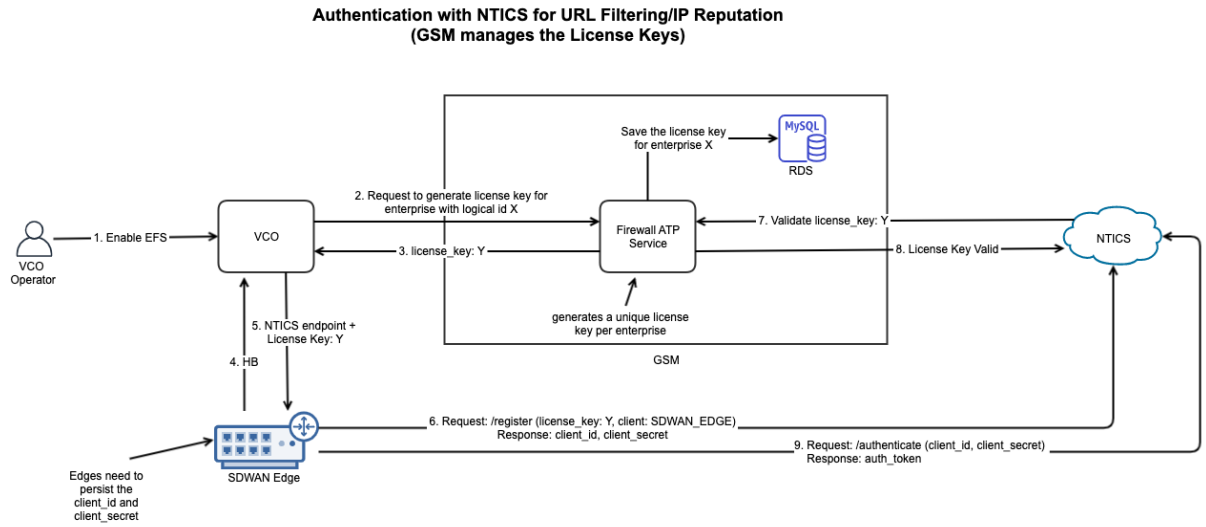
{
"enterpriseId": 1,
"type": "idps",
"with": ["securityGroups"]
}
	[{
	"id": 2,
	"enterpriseId": 1,
	"name": "idps-1",
	"type": "idps",
	"data": {
	"idsEnabled": true,
	"ipsEnabled": true,
	"logEnabled": true
	},
	"serviceGroups":
	}]

{
"enterpriseId": 1,
"type":
"securityServiceGroup",
}
	{
	"id": 1,
	"enterpriseId": 1,
	"name": "group-1",
	"data": {
	"idps": { "id": 2, "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80c" }
	}
	}

request	response

{
"enterpriseId: 1,
"type": "securityServiceGroup",
"with": ["profiles"]
}
	{
	"id": 1,
	"enterpriseId": 1,
	"name": "group-1",
	"data": {
	"idps": { "id": 2, "logicalId": "87e768fb-a0d4-4a34-a094-7b7a1179c80c" }
	},
	"profileCount": 1,
	"edgeCount": 1
	}

12.2 13.2. Generating NTICS Licenses Key



For URL Filtering/IP Reputation, we will be integrating Webroot SDK on the SDWAN Edges. The URL module on Webroot SDK will make network queries to NTICS for URLs that are not found in local DB and cache. As Webroot SDK on Edge needs to communicate with NTICS we will have to generate credentials for each SDWAN edge. As mentioned above, the authentication workflow is as follows:

1. When EFS is enabled on VCO, VCO sends a request to GSM Firewall ATP service to generate license key for an enterprise with particular logical ID.
2. Firewall ATP service on GSM generates a license key and stores it in its local DB and returns it to VCO
3. VCO notifies the the edges through HB response about NTICS endpoint and license key.

4. Edges sends registration request to client using license key.
5. NTICS will call GSM Firewall ATP service to validate the license key,
6. NTICS responds with a client_id and client_secret after validation. This is a one time activity.
7. Using client id and secret, edges send a request to NTICS to generate an Auth token (JWT).
8. NTICS respond with authentication token.
9. Edges start using the auth token as part of 'Authorization' header in requests
10. Repeat step 7 as and when auth token expires
11. When EFS is disabled on the VCO, there is no synchronous notification sent to GSM. Unused licenses will be cleaned up by GSM periodically, as specified below.

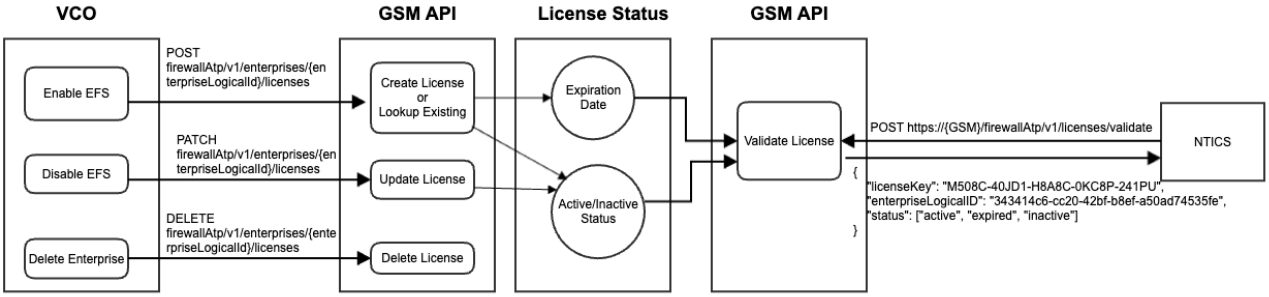
Auth credentials (client_id and client_secret) are stored in a globally synchronized database. This means that we can use a single instance of Firewall ATP Service deployed in us-west2 and connected to NTICS in US region to provision auth credentials for a VCO/Edge in EU region. The edge will still be able to authenticate with NTICS service in EU when it tries to generate an auth token.

Thus the main design goals are as follows:

1. License Key provision from GSM to the Edges.
2. License validation from NTICS to GSM, NTICS validates whether or not the license key is valid, expired, or active.

12.2.1 13.2.1 GSM

- An Enterprise should have only one valid license key in GSM database at any given time.
- The license key expires is 1 year by default. (The expires should associate with the Enterprise license when it come out)
- In license validation, there should have 72 hours license grace period for tolerate with possible delay from VCO renew certification action.
- Each license switches between active and inactive status until it is deleted. VCO will call the GSM API to change the license status according to EFS setting.
- There should be a mechanism to prevent unused licenses from persisting. This includes licenses for enterprises that disabled EFS. If EFS is re-enabled before the license becomes "stale", there is no need to generate a new NTICS license.



12.2.1.1 **13.2.1.1 Database to store License Key**

```
CREATE TABLE IF NOT EXISTS FIREWALL_ATP_LICENSE (
  `id`                        BIGINT NOT NULL AUTO_INCREMENT,
  `logicalId`                 VARCHAR(128) UNIQUE DEFAULT (uuid()),
  `created`                   DateTime DEFAULT now(),
  `modified`                  DateTime NOT NULL DEFAULT now() ON UPDATE now(),
  `licenseKey`                VARCHAR(128) UNIQUE,
  `enterpriseLogicalId`       VARCHAR(128),
  `expires`                   DateTime,
  `lastReferenceTime`         DateTime DEFAULT NOW(),
  `active`                    BOOLEAN NOT NULL DEFAULT 1,
  PRIMARY KEY (`id`)
) ENGINE = InnoDB DEFAULT CHARSET = UTF8;
```

12.2.1.2 13.2.1.2 License Provision

Request License at 1st time

1. VCO initiate and make the API call like, POST /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses.

2. In API request, VCO need provide EnterpriseLogicalId as parameter.

3. The firewall-atp API lookups and not find an valid license associated with the EnterpriseLogicalId, then it creates a new unique license and return it in API response. The expires of the new License is 1 year.

4. NTICS licenses can be generated by the VCO without a configured NTICS endpoint (VCO only directly communicates with GSM). The NTICS endpoint is only used by edges for License validation. The NTICS endpoint must still be configured for edges to support Malicious IP and URL Filtering.

Request License at 2nd time

The steps are very similar as above "Request 1st License", the only difference is in step-3, and there are two cases.

3.1 The firewall-atp API lookups and finds an existing one (not expired), GSM update the status of license as active and return the license in API response.

3.2 The firewall-atp API lookups and finds an existing one (expired or about to expired), GSM creates a new unique license and return the license in API response. The expired license will be taken care by the backend job which is response with license cleanup.

API	Request	Response Body
POST /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses	`POST /firewallAtp/v1/enterprises/343414c6-cc20-42bf-b8ef-a50ad74535fe/licenses`	`{ "licenseLogicalId": "33f8a91b-324a-11ee-9dca-0e813ba16025", "enterpriseLogicalId": "2f5e20d4-ba80-4a55-91b9-9bc599a3ea08", "licenseKey": "B4J8B-ZATHN-FNEG6-B8ABE-8HQJZ", "expires": "2024-08-02T22:07:58.000Z" }`

VCO notify GSM to update the License status

1. VCO initiate and make the API call like, PATCH /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses/{licenseLogicalId}.
2. In API request, VCO provides the inactive status when EFS is disabled at enterprise.
3. The firewall-atp API updates license status to the database.

API	Request	Response Body

PATCH /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses/{licenseLogicalId}	`PATCH /firewallAtp/v1/enterprises/2f5e20d4-ba80-4a55-91b9-9bc599a3ea08/licenses/33f8a91b-324a-11ee-9dca-0e813ba16025`
	
	`HTTP Body`
	`{`
	`  "active" : "false"`
	
	`}`
		license updated successfully

Summaries

VCO Events	VCO Actions	VCE Status	License key Status in GSM
EFS Enabled	• **Request License**.	Acquiring or holding the License Key	Active
EFS Disabled	• **Update License status to inactive**	Deleting the License key	Inactive
Enterprise Deleted	• **Delete License**	Deleting the License key	

12.2.1.3 13.2.1.3 License Validation

1. Ntics make the API call to firewall-atp service API for License validate when the Edges register/ Authenticate with NTICS.

2. In API request, NTICS need provide licenses info. Batch License validation request is supported.
3. The firewall-atp API return enterpriseLogicalID and License status (active, inactive, invalid, expired) in the API response.
4. Given the possible frequent requests for validation, the firewall-atp API should try to load license info from cache before query database.
   For example, 100K edges, each edge raise Authentication every 30 minutes, then QPS will be 100,000 / 30 / 60 ~= 55.

API Request	HTTP Status Code	HTTP Response

POST https://{GSM}/firewallAtp/v1/licenses/validate
 	
HTTP Body
{
"licenses": [
"M508C-40JD1-H8A8C-0KC8P-241PU",
"$(inactive-key)",
"$(invalid-key)",
"$(expired-key)"
]
}
	200	{
		"results": [
		{
		"licenseKey": "M508C-40JD1-H8A8C-0KC8P-241PU",
		"enterpriseName": "abc", #TBD
		"enterpriseLogicalID": "aaaaaaaa-cc20-42bf-b8ef-a50ad74535fe",
		"expirationTime": "2024-05-22T00:00:00Z",
		"status": "active",
		},
		{
		"licenseKey": "inactive-key",
		"enterpriseLogicalID": "bbbbbbbb-cc20-42bf-b8ef-a50ad74535fe",
		"status": "inactive"
		},
		{
		"licenseKey": "invalid-key",
		"enterpriseLogicalID": "",
		"status": "invalid"
		},
		{
		"licenseKey": "expired-key",
		"enterpriseLogicalID": "dddddddd-cc20-42bf-b8ef-a50ad74535fe",
		"status": "expired"
		}
		]
		}

12.2.1.4 13.2.1.4 License Eviction

The licenses are removed from the databases at the cases below,

1. VCO initiate to notify the GSM firewall-atp service when the Enterprise is deleted.

2. Auto-Delete after expired.
3. Auto-Delete after long idle. The last active timestamp per license is updated when NTICS call the License Validation API.

12.2.2 13.2.2. VCO

12.2.2.1 13.2.2.1 EFS Enabled

When EFS is enabled for the first time in the customer settings for an enterprise

- When EFS is enabled for the first time in an enterprise at customer settings, vco sends a POST request to GSM specifying enterprise logical ID and expiration date.
- Once VCO gets new license key, it creates an enterprise object and stores the license key information.

API	Request	Response Body
POST /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses	`POST /firewallAtp/v1/enterprises/343414c6-cc20-42bf-b8ef-a50ad74535fe/licenses`	`{ "licenseLogicalId": "33f8a91b-324a-11ee-9dca-0e813ba16025", "enterpriseLogicalId": "2f5e20d4-ba80-4a55-91b9-9bc599a3ea08", "licenseKey": "B4J8B-ZATHN-FNEG6-B8ABE-8HQJZ", "expires": "2024-08-02T22:07:58.000Z" }`

VCO - Enterprise object

- This object will be created per enterprise when a customer enables EFS for the first time and receives license key information from GSM. Disabling EFS will mark the object as inactive. Objects are deleted only when the enterprise is deleted.
- A new type will be introduced in **EnterpriseServiceType** called **ntics_license**

Column	Type	Description
id	number	1
created	timestamp	'2023-12-10T00:30:50.000Z'
operatorId	timestamp	NULL
networkId	uuid	NULL
enterpriseId	enum	56
edgedId	string	NULL
gatewayId	JSON	NULL
parentGroupId	uuid	NULL
description	number	"store license key "
object	number	PROPERTY
name	string	'licenseKey'
type	string	**ntics_license**
logicalId	type	'87e768fb-a0d4-4a34-a094-7b7a1179c80c'
alertsEnabled	Boolean	1
operatorAlertsEnabled	Boolean	1
status	text	NULL

Column	Type	Description
statusModified	timestamp	'0000-00-00 00:00:00'
previousData	mediumtext	NULL
previousCreated	timestamp	'0000-00-00 00:00:00'
draftData	mediumtext	NULL
draftCreated	timestamp	'0000-00-00 00:00:00'
draftComment	string	NULL

data	mediumtext	{
		"licenseKey": "M508C-40JD1-H8A8C-0KC8P-241PU",
		"licenseLogicalId": "33f8a91b-324a-11ee-9dca-0e813ba16025",
		"enterpriseLogicalId": "343414c6-cc20-42bf-b8ef-a50ad74535fe",
		"status": "active",
		"expires": "2024-05-22T00:00:00Z",
		"endpoint": "https://gsm.net",
		"version": "123456789",
		"registerAPI": "/2.0/auth/register",
		"authenticateAPI": /1.0/auth/authenticate"
		}

lastContact	timestamp	'0000-00-00 00:00:00'
version	string	'0'

Column	Type	Description
modified	timestamp	'0000-00-00 00:00:00'

VCO - Edges notification

- When URL Filtering /IP Reputation is enabled at profiles/edges, VCO adds "configurationUpdate" action in Heartbeat response for all those edges in enterprise to be updated with the license key.
- The configuration module **atpMetadata** will be updated with license key and NTICS endpoint and sent to edges.

HB Request from edges

HeartBeat Request


```
{
  "params": {
    "logicalId": "7e13ba94-2b92-40dd-8dba-24fa5c0fde50",
    "endpointPkiMode": "CERTIFICATE_DISABLED",
    "crlNumber": "0",
    "certDigest": "",
    "actionUpdates": [],
    "serviceUpSince": 1652800678628,
    "events": [],
    "buildNumber": "R450-20211007-GA-72423-2da3b08e35",
    "systemUpSince": 1652800655000,
    "edgeBfdNeighbors": {
      "totalEntries": 0,
      "startEntryIndex": 0,
      "bfdNeighborSummary": [],
      "dispEntries": 0
    },
    "haState": "UNKNOWN",
    "endpointTrustedIssuerVersion": "0",
    "deviceId": "00:50:56:82:b4:73",
    "edgeBgpNeighbors": {
      "startEntryIdx": 0,
      "bgpNeighborSummary": [],
      "totalEntries": 0,
      "dispEntries": 0
    },
    "isLive": false,
    "configuration": [
      {
        "version": "1643808529009",
        "module": "WAN"
```

```
      },
      {
        "version": "1652889871892",
        "module": "QOS"
      },
      {
        "version": "1655603384673",
        "module": "firewall"
      },
      {
        "version": "1666042368318",
        "module": "controlPlane"
      },
      {
        "version": "1652968702684",
        "module": "analyticsSettings"
      },
      {
        "version": "1643066611000",
        "module": "properties"
      },
      {
        "version": "1619507235682",
        "module": "managementPlane"
      },
      {
        "version": "0",
        "module": "metaData"
      },
      {
        "version": "1644529322115",
        "module": "imageUpdate"
      },
      {
        "version": "1666042368317",
        "module": "deviceSettings"
      },
      {
        "version": "1666042368222",     /***************** added for ATP Suricate
signatures and will be used for URL Filtering file updates as well**************/
        "module": "atpMetadata"
      }
    ],
    "softwareVersion": "4.5.0",
    "token": {
      "logicalId": "7e13ba94-2b92-40dd-8dba-24fa5c0fde50",
      "hmac": "3351c725e574901f400e66fa375f3dba82bfe6d4fe0e88005bb22c87745114e9"
    },
    "serialNumber": "VMware-4202c21592f4f3e3-23b05bfa1a1910bb"
  },
  "jsonrpc": "2.0",
  "method": "edge/edgeHeartbeat",
```

```
  "id": 1669829037552
}
```

HB response from VCO:

HeartBeat Response

```
{
    "actions": [{
        "action": "configurationUpdate",
        "data": {
            "module": "atpMetadata",
            "version": "1614693596464",
            "schemaVersion": "3.0.0",
            "use": {
                "atpUpdateEnabled" : true,
                "ntics": {
                    "licenseKey": "ju8d228c-k22w-ur70-vw23-0242ac120002",
                    "endpoint" : "https://test-ntics.com", /* Required for edges to
call register/authenticate. Creation of license does not require this */
                    "deviceType": "SDWAN-Edge",
                    "registerAPI": "/2.0/auth/register",
                    "authenticateAPI": "/1.0/auth/authenticate",
                    "version": "1614693596464",
                }
            }
        }
    }]
}
```

- Edges that receive this configurationUpdate action makes the calls to ntics register and authenticate endpoints and provide the license key in its request.
- NTICS validates license key and responds with client id and secret. Edges request authentication token using client id and secret to NTICS and NITCS provides auth token which will be used by edges for subsequent requests until token expires.

12.2.2.2 13.2.2.2 EFS Disabled

When EFS is disabled on the customer settings in VCO (which can be done only when EFS is disabled at all the profiles and/or edges belonging to that enterprise),

- VCO sends a PUT request to GSM to mark license status as inactive
- **VCO updates status of nticsLicense object to inactive**

API	Request	Response Body

PATCH /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses/{licenseLogicalId}	`PATCH /firewallAtp/v1/enterprises/343414c6-cc20-42bf-b8ef-a50ad74535fe/licenses/33f8a91b-324a-11ee-9dca-0e813ba16025`
	
	`HTTP Body`
	`{`
	`  "status" :  "inactive"`
	
	`}`
		`HTTP status code: 200`
		`license updated successfully`

> The license key status need not be sent to edges through HB response as the features would have been disabled at profile/edge level prior to that and that would have already been sent to edges through configurationUpdate.

12.2.2.3 13.2.2.3 NTICS/GSM endpoint updated

When the NTICS endpoint is updated. All active NTICS licenses will be regenerated, triggering the update of the related atpMetadata configs and heartbeat response as shown above.

12.2.2.4 13.2.2.4 Enterprise deleted

When enterprise is deleted the VCO should notify GSM to delete the license key. Enterprise object is deleted

API	Request	Response Body
DELETE /firewallAtp/v1/enterprises/{enterpriseLogicalId}/licenses/{licenseLogicalId}	`DELETE /firewallAtp/v1/enterprises/343414c6-cc20-42bf-b8ef-a50ad74535fe/licenses/33f8a91b-324a-11ee-9dca-0e813ba16025`	HTTP Status code: 200 license deleted successfully

12.2.2.5 13.2.2.5 License key expiry

- VCO should have a background job to check if license key is about to expire in one week and make a call to GSM to update the license.
- VCO stores the new license obtained from GSM as a new enterprise object.
- VCO sends the new license to the edges in HB response.
- VCO will always store a single license corresponding to an enterprise.

12.3 13.3. URL Category List

The Webroot SDK installed on the Edge handles URL lookups. If the URL is not found locally, then a network request is made to a configured NTICS API, which acts as a proxy to Webroot. The VCO and GSM are not involved

12.3.1 13.3.1 URL Category List (PRD URL 1.6)

Webroot assigns membership of all seen URLs to 80 categories that are further combined into 10 groups. The first 9 groups are recognized site types and the 10th is the "unknown" category for URLs that Webroot does not know. This list should rarely, if ever, change. A default version of URL category list will be included in the software package and copied into the proper filepath as part of the build process, similar to the default Application Map. A job in the backend service will run daily to poll the Category List from Webroot using the NTICS APIs, and will store the new version, if any, in this file location. This same Url Category list is also available to the UI for creating/editing Url Groups and monitoring. A backend job checks for (rare) new categories every 12 hours. The URL_CATEGORIES_STORE_SUCCESS event contains which categories were added. The customer must update security group configurations to include the new categories.

12.3.2 13.3.1.1 VCO

The default URL Category List will be added to the meta/enums package of the velocloud.src repository. As part of the production build, this file will be copied into the proper location on the server/container. This parallels how the Qosmos Application ID map is compiled into the product. This file will be stored locally in the VCOs as part of blob store; and will be used for mapping URL Category IDs to readable strings for UI and monitoring requests.

12.3.3 13.3.1.1 GSM

The GSM ATP service introduced in Woodford can be extended to provide the URL Category list to VCOs. It will serve as a simple proxy and send a request to NTICS to get the category list upon receiving a request from the VCO and relay the response back to VCO.

API	Request	Response Body

POST / firewallAtp/v1/urlFilter/urlCategories	**POST** / firewallAtp/v1/urlFilter/categories	{
		"categoryList": [
		{
		"catid": 1,
		"catname": "Real Estate",
		"catgroup": "Productivity"
		},
		{
		"catid": 2,
		"catname": "Computer and Internet Security",
		"catgroup": "Productivity"
		},
		.
		.
		.
		.
		]
		}

12.4 13.4. Monitoring

12.4.1 13.4.1. ClickHouse Table - VELOCLOUD_FIREWALL_STATS

New columns for supporting URL Filtering and IP reputation metrics will be added to this table. These columns are optional and will be blank if URL Filtering is not provided in the uploaded EFS logs, either because the Edge is pre-Xante or EFS URL Filtering is disabled.

Column	EFS Feature	Type	Description
engineType	All	ENUM	What kind of Security Service engine generated the log line, corresponding to N/A (legacy edges, treated as idps), `idps` , `urlCategoryFilteirng, urlReputationFiltering, maliciousIpFiltering`
domainName	URL Category Filtering, Url Reputation Filtering	String	Top-level domain. HTTPS can only see domain, not full url, and NSX only has domain-level reputations
urlCategories	URL Category Filtering	Array(number)	Array of integers, max length 5
urlRisk	URL Reputation Filtering	Enum	0 = HIGH, 1 = SUSPICIOUS, 2 = MEDIUM, 3 = LOW, 4 = TRUSTWORTHY
ipCategories	Malicious IP Filtering	Array(number)	Array of integers
ipReputation	Malicious IP Filtering	number	Reputation score between 1-100

During migration to Yamazaki, any rows still present in this table will automatically have the type "idps" as that engine was the only one in use.

12.4.2 13.4.2. Clickhouse Database Schema Changes

Rename table VELOCLOUD_FIREWALL_STATS_FIVE_MINUTES_AGGREGATE to VELOCLOUD_FIREWALL_STATS_IDPS_FIVE_MINUTES_AGGREGATE

New table VELOCLOUD_FIREWALL_STATS_URLF_CAT_FIVE_MINUTES_AGGREGATE

```
CREATE TABLE VELOCLOUD_FIREWALL_STATS_URLF_CAT_FIVE_MINUTES_AGGREGATE (
    `startTime`             DateTime DEFAULT now(),
    `endTime`               DateTime DEFAULT now(),
    `enterpriseLogicalId`   UUID,
    `edgeLogicalId`         UUID,
    `segmentLogicalId`      UUID,
    `ruleId`                String,
    `domainName`            String,
    `action`                Enum('ALLOW' = 0, 'MONITOR' = 1, 'DENY' = 2),
    `urlCategories`         Array(UInt16),
    `threatsCount`          AggregateFunction(count)
) ENGINE = AggregatingMergeTree
PARTITION BY toYYYYMMDD(startTime)
PRIMARY KEY (enterpriseLogicalId, startTime, edgeLogicalId)
ORDER BY (enterpriseLogicalId, startTime, edgeLogicalId, segmentLogicalId, ruleId,
domainName, action, urlCategories)
TTL addYears(toStartOfDay(startTime), 1)
SETTINGS index_granularity = 8192;
```

New table VELOCLOUD_FIREWALL_STATS_URLF_REP_FIVE_MINUTES_AGGREGATE

```
CREATE TABLE VELOCLOUD_FIREWALL_STATS_URLF_REP_FIVE_MINUTES_AGGREGATE (
    `startTime`             DateTime DEFAULT now(),
    `endTime`               DateTime DEFAULT now(),
    `enterpriseLogicalId`   UUID,
    `edgeLogicalId`         UUID,
    `segmentLogicalId`      UUID,
    `ruleId`                String,
    `sourceIp`              String,
    `domainName`            String,
    `action`                Enum('ALLOW' = 0, 'MONITOR' = 1, 'DENY' = 2),
    `urlRisk`               Enum('HIGH' = 0, 'SUSPICIOUS' = 1, 'MEDIUM' = 2, 'LOW' =
3, 'TRUSTWORTHY' = 4),
    `threatsCount`          AggregateFunction(count)
) ENGINE = AggregatingMergeTree
PARTITION BY toYYYYMMDD(startTime)
PRIMARY KEY (enterpriseLogicalId, startTime, edgeLogicalId)
ORDER BY (enterpriseLogicalId, startTime, edgeLogicalId, segmentLogicalId, ruleId,
sourceIp, domainName, action, urlRisk)
```

```
TTL addYears(toStartOfDay(startTime), 1)
SETTINGS index_granularity = 8192;
```

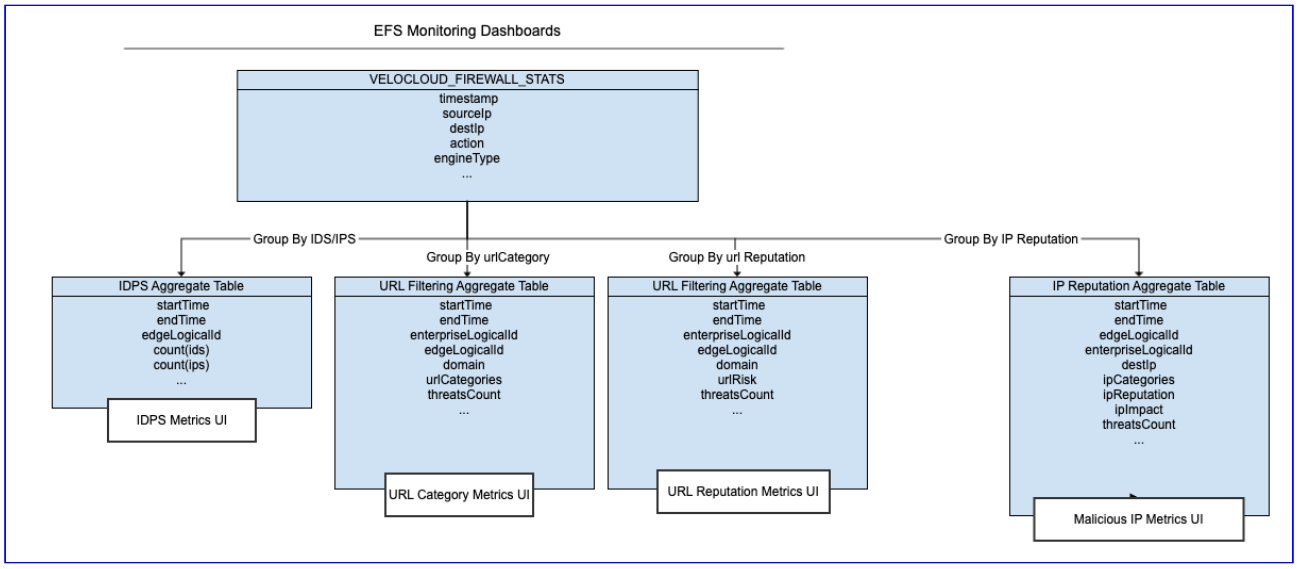
New table VELOCLOUD_FIREWALL_STATS_MALICIOUS_IP_FIVE_MINUTES_AGGREGATE

```
CREATE TABLE VELOCLOUD_FIREWALL_STATS_MALICIOUS_IP_FIVE_MINUTES_AGGREGATE (
    `startTime`             DateTime DEFAULT now(),
    `endTime`               DateTime DEFAULT now(),
    `enterpriseLogicalId`   UUID,
    `edgeLogicalId`         UUID,
    `segmentLogicalId`      UUID,
    `ruleId`                String,
    `sourceIp`              String,
    `destIp`                String,
    `action`                Enum('ALLOW' = 0, 'MONITOR' = 1, 'DENY' = 2),
    `ipCategories`          Array(UInt16),
    `threatSourceGeoCountry` String,
    `threatsCount`          AggregateFunction(count)
) ENGINE = AggregatingMergeTree
PARTITION BY toYYYYMMDD(startTime)
PRIMARY KEY (enterpriseLogicalId, startTime, edgeLogicalId)
ORDER BY (enterpriseLogicalId, startTime, edgeLogicalId, segmentLogicalId, ruleId,
sourceIp, destIp, action, ipCategories, threatSourceGeoCountry)
TTL addYears(toStartOfDay(startTime), 1)
SETTINGS index_granularity = 8192;
```

The rows for the derived tables will be filtered on type, so only "idps" metrics go into the VELOCLOUD_FIREWALL_STATS_IDPS_FIVE_MINUTES_AGGREGATE table, etc.

12.4.3 13.4.3. UI Dashboard

The monitoring dashboards pull their data from ClickHouse tables that are aggregated according to the respective dimensions. All Firewall EFS logs are uploaded to the VELOCLOUD_FIREWALL_STATS table, and there are separate aggregation queries for each EFS feature, aggregating metrics into separate derived tables grouped in 5 minute intervals.



The dashboards for URL Filtering will look very similar to the NSX dashboards, but filtered on Edges and Enterprises rather than Gateways, as in NSX.

For Enterprise-level dashboards, there will be additional graphs showing a breakdown of URL/IP filtering per-Edge, including such displays as "Top 5 Edges with the most blocked sessions", etc.

12.4.4 13.4.4. Metric Queries

Clickhouse supports Arrays as a column type, along with a function to "unroll" these arrays when performing queries, creating multiple result rows per single row with a single element of the array. Urls and IPs are categorized into one or more categories. The Webroot database store Url categories as an array of up to 5 integers. IP categories are stored as an integer corresponding to a bitmask. As part of processing the uploaded logs, IP categories will be reformatted into an array to match the format of Url categories. These arrays are inserted into Clickhouse, but are not unrolled in the tables.

Depending on the type of metric query, the categories may be unrolled to provide the correct counts. Top N urls/domains or IPs will not perform an Array Join, while Top N Url Categories or Top N IP categories will perform the Array Join. The topK(N)(column) function in Clickhouse returns an array of the N most frequent values in the given column. This array can itself be unrolled as a subquery to retrieve the counts. Using this, a single query will return the top N results, along with their respective counts. Critically, all final queries must use the countMerge(threatsCount) to properly aggregate data across all Clickhouse parts/rows.

Example:

A table has the following 2 rows:

domain	categories	count
A	[1, 2, 3]	3

domain	categories	count
B	[1, 5]	1

The topK category query will return:

category	count
1	4
2	3
3	3
5	1

12.5 13.5 Firewall Logging

We need to examine the scenarios that generate Firewall ATP Logs when a single Firewall rule has all secure engines configured. Here are the scenarios to consider:

For Firewall Logs:

1. In the 'Flow Allow' case, how many firewall logs should we expect if all secure engines, including the firewall rule, allow the traffic? (As discussed yesterday, it appears that IDPS cannot be aggregated with others.)
2. In the 'Flow Drop' case:
    a. How many firewall logs are expected if one secure engine drops the traffic while the others allow it?
    b. If there is only one drop log, will the fields related to the secure engine allowing the flow be included in the firewall logs?

For Firewall Stats for Monitoring, we should have the same set of questions as above. The current idea is *Count the metrics of all relevant secure engines when a flow is allowed. If the flow is not allowed, we should only count the secure engine responsible for dropping the flow*. For example, if a flow is allowed by URL filtering but subsequently dropped by the malicious IP filter, the monitoring dashboard's URL filtering 'allow' metrics should not count it because the flow is ultimately dropped. However, the malicious IP drop metrics should be counted.

Firewall ATP Logs	IDPS	URL Filtering	URL Reputation	Malicious IP	Total
Flow Allow	1 or 0 depends on if hit IDPS rule	One Log incorporate all 2 Engines related fields, including URL Category, URL Reputation Score, etc,.		NA	1 or 2 Allow Log • (Optional) One Log for IDPS • One combined Log for the 2 engines
Flow Drop	1 or 0 depends on if hit IDPS rule	One Log incorporate all 3 Engines related fields if they are present. As long as the engine process the flow, the related fields can be insert into the log.			1 Drop Log, either of below • (Optional) One Log for IDPS • (Optional) One combined Log for the 3 engines

13 14. Testing

13.1 14.1. General approach

This section should minimally address the following

13.2 14.2. Unit testing

This section should minimally address the following

13.3 14.3. System testing

This section should minimally address the following

- [ ] **Since QA does system testing it would be important to explain a general approach to how to exercise this functionality from a system test perspective.**
- [ ] **List the dependent features that may be used or have an impact on this feature.**
- [ ] **Will this impact existing QA Regression scripts ? If yes, explain what needs to be updated so that QA can modify script accordingly.**

13.4 14.4. Scale testing

This section should minimally address the following

1. NTICS will be receiving geturlinfo requests from 1000s of edges

13.5 14.5. Upgrade / interoperability testing

This section should minimally address the following

13.6 14.6. Documentation impact

This section should minimally address the following

New APIs and explanations of the EFS features and dashboards will have documentation for Operators and Support teams.

14 15. Future considerations

15 Edge Firewall ATP - URL & IP Filtering Monitoring (Phase 2) Roadmap

15.1 ~~Start Date: Oct 23 2023~~

15.2 Finish date: 01 31 2023

~~Time Available: 13.5 Engineer-Weeks [EW] (Veterans' Day + Thanksgiving leave ~4 weeks in November)~~

Number of engineers: 3 (Ben Shapero, Qing Li, Nandini Rangaswamy)

Monitoring Tasks:

ID	Task Description	Owner	Estimated Time	Jira Link	ETA
1	Add getIdpsSignature and getIpCategories APIs for Firewall Logs UI	Nandini Rangaswamy	1 EW	VLENG-130473[8] - [VCO] Add getIdpsSignature and getIpCategories APIs for Firewall Logs CLOSED	10/31
2	Process new firewall logs by engine type for insertion to Clickhouse	Ben Shapero	1 EW	VLENG-125627[9] - [VCO] Update firewall stats processing for new clickhouse monitoring CLOSED	11/2

8 https://vmw-jira.broadcom.net/browse/VLENG-130473?src=confmacro
9 https://vmw-jira.broadcom.net/browse/VLENG-125627?src=confmacro

ID	Task Description	Owner	Estimated Time	Jira Link	ETA
3	Write Clickhouse queries for URL Filtering	Ben Shapero	2.5 EW	VLENG-132498[10] - [VCO] Generate clickhouse query for urlf cat metrics CLOSED	1/7
5	Malicious IP - Portal metrics APIs & Clickhouse queries	Qing Li	2 EW	VLENG-122784[11] - [Firewall Atp] Implement Clickhouse Query for malicious IP Monitoring CLOSED	1/7
4	Portal metrics APIs - URL Filtering	Nandini Rangaswamy	2 EW	VLENG-134039[12] - Add new portal metrics API for URL category and URL reputation Filtering FIXED NOT VERIFIED	1/15
6	Enterprise Edges Count - Portal metrics APIs & Clickhouse queries	Ben Shapero	2 EW	VLENG-135890[13] - [VCO portal metrics] Enterprise security overview edgesCount CLOSED	01/19

10 https://vmw-jira.broadcom.net/browse/VLENG-132498?src=confmacro
11 https://vmw-jira.broadcom.net/browse/VLENG-122784?src=confmacro
12 https://vmw-jira.broadcom.net/browse/VLENG-134039?src=confmacro
13 https://vmw-jira.broadcom.net/browse/VLENG-135890?src=confmacro

ID	Task Description	Owner	Estimated Time	Jira Link	ETA
7	Enterprise Reporting Edges detail - Portal metrics APIs & Clickhouse queries	Qing Li	2 EW	VLENG-135891[14] - [VCO portal metrics] Enterprise security overview reporting Edges detail FIXED NOT VERIFIED	1/23
8	UI Integration Testing Code review for merge back to master	Qing Li Ben Shapero	2 EW		01/31
10	Scale Test - Agent Smith support firewall stats	Qing Li	2 EW		

~~Total: 11.5 EW~~

~~This leaves about a week for final approvals, rebasing, merging.~~

14 https://vmw-jira.broadcom.net/browse/VLENG-135891?src=confmacro

16 Edge Firewall ATP Url Filtering Threat Modeling

16.1 Feature Overview

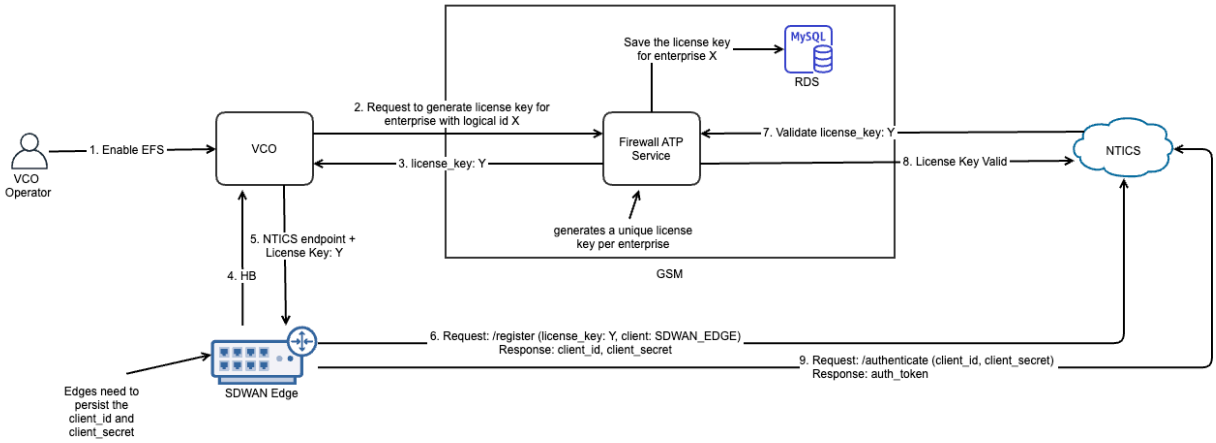
Next Generation Firewall (NGF) products include Advanced Threat Detection/Prevention (ATD/ATP). The ATP functionality will be powered by NSX technology. The ATP service will support protecting VCE traffic from intrusions across branch 2 branch, branch 2 hub or branch to internet traffic patterns. The NSX powered ATP functionality shall include IDS/IPS, Reputation & Threat Intelligence and URL Filtering security services. An end customer configures and manages the ATP services via Firewall functionality in VCO.

The management plane must support all configuration operations, including creating/changing firewall rules, updating ATP threat databases and Suricata signatures, and collecting/exposing logs for dashboards and reports.

Please refer to Functional Spec (see page 6) for more details on the feature.

16.2 Product Changes

16.2.1 VCE Authenticate with NTICS



For URL Filtering/IP Reputation, we will be integrating Webroot SDK on the SDWAN Edges. The URL module on Webroot SDK will make network queries to NTICS for URLs that are not found in local DB and cache. As Webroot SDK on Edge needs to communicate with NTICS we will have to generate credentials for each SDWAN edge. As mentioned above, the authentication workflow is as follows:

1. Register a client using license key → this generates a client_id and client_secret. This is a one time activity.
2. Generate an Auth token (JWT) using client_id and client_secret
3. Use the auth token as part of 'Authorization' header in requests

4. Repeat step 2 as and when auth token expires

License Keys will be generated by Firewall ATP Service on GSM. Whenever a SDWAN client registers with NTICS and presents a license key, NTICS will call into GSM to validate the key.

Auth credentials (client_id and client_secret) are stored in a globally synchronized database. This means that we can use a single instance of Firewall ATP Service deployed in us-west2 and connected to NTICS in US region to provision auth credentials for a VCO/Edge in EU region. The edge will still be able to authenticate with NTICS service in EU when it tries to generate an auth token.

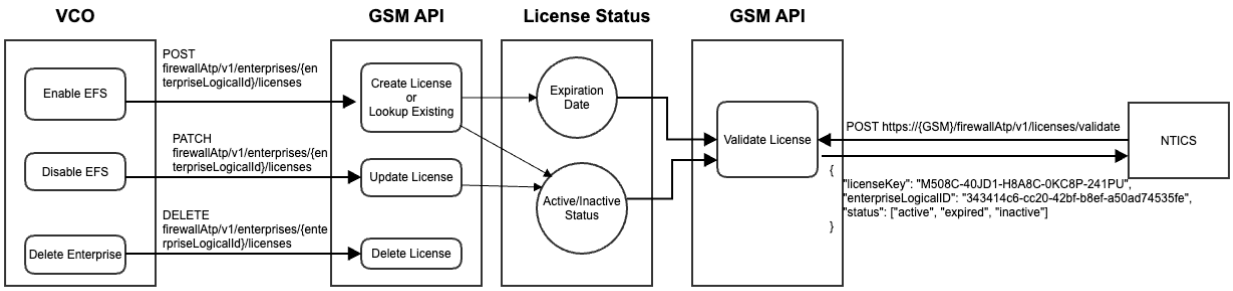
16.2.2 VCO

16.2.2.1 Customer Capability

1. New customer capability will be added to enable ATP for an enterprise
2. Operators will be able to configure this capability for an enterprise
3. Individual ATP modules (IDPS, URL Filtering, etc) can be disabled, even if ATP feature is enabled. Note: If all ATP features are disabled, then enabling ATP itself does nothing

16.2.2.2 NTICS Licenses

- An Enterprise should have only one valid license key at any given time.
- The license key expires is 1 year by default. (The expires should associate with the Enterprise license when it come out)
- In license validation, there should have 72 hours license grace period for tolerate with possible delay from VCO renew certification action.
- Each license switches between active and inactive status until it is deleted. VCO will call the GSM API to change the license status according to EFS setting.
- There should be a mechanism to prevent unused licenses from persisting. This includes licenses for enterprises that disabled EFS. If EFS is re-enabled before the license becomes "stale", there is no need to generate a new NTICS license.



16.2.2.3 **Security Groups**

This feature will introduce a new configuration called security groups that will act as a placeholder to configure/enable all the security engines that can be associated with a firewall rule.

The security groups consists of the ATP features as below:

- IDS for detection, IPS for prevention (Since Woodford)
- Malicious IP, with Drop and Monitor actions (New ATP feature in Yamazaki)
- URL Reputation based filter, with Drop and Monitor actions (New ATP feature in Yamazaki)
- URL categories based filter, with Allow, drop and Monitor actions. (New ATP feature)

16.2.2.4 Firewall Rule Configuration

1. Users will be able to CRUD Security Groups at Enterprise level.
2. Users will be able to attach a security group object for an Firewall rule configuration at both enterprise profile and edge.

16.2.2.5 ATP Metadata Configuration Module

1. Existing action field will be deprecated, but kept for backwards-compatibility with Woodford (to be removed after 3 versions)
2. Each EFS feature will be configured in its own subsection, one for each "Engine".
3. Each Engine will have its own action field, one for IDPS, one for URL Filtering, one for IP/Botnet, etc.

16.2.2.6 Logs

1. The new ATP features will generate Firewall alert logs while it is enabled at the firewall level and matches with a Security Group with DROP or MONITOR (allow but log) action.
2. Each engine runs in parallel, generating separate Log lines. A unique identifier per flow will be added to associate the log lines together, so they are all counted as one session
3. VCE upload the Firewall + ATP Logs to VCO in protobuf format through the existing endpoint to the upload service.
4. These logs will then be forwarded to Firewall Logging service on the GSM where they will decoded and finally stored in the Open Search Cluster as part of the logging infrastructure
5. VCO UI will query the firewall logs using the Search service that is being build as part of Log Management project.

16.2.2.7 Monitoring Dashboards

1. The new ATP features will generate Firewall stats while it is enabled and one or more engines have been added.
2. VCE upload the Firewall + ATP stats to VCO in protobuf format as metrics through the existing endpoint to the upload service.
3. These metrics will be stored on the VCO in ClickHouse database
4. New API endpoints will be added in Portal service to query these metrics and display them on the UI

16.2.3 GSM

16.2.3.1 Firewall ATP Service

1. Existing service for downloading IDPS Suricata signature bundles
2. Existing APIs for IDPS are unchanged
3. New endpoint for VCO to download URL Category list
4. New CRUD endpoints for VCO to call when managing NTICS license provisions for Enterprises
5. New MySQL table to store NTICS license keys/expirations
6. New endpoint on GSM for Edges to call to validate/authenticate NTICS licenses; authentication request is proxied to NTICS
7. New background job that deallocates stale/expired NTICS licenses

16.3 Authentication between Services

16.3.1 Edge and VCO

1. There's no change to how edges and VCO communicate and authenticate.
2. The heartbeat response from VCO to edges will contain the GSM endpoint and NTICS license key information needed for Edge→GSM validation/authentication

16.3.2 VCO and GSM

1. VCO will talk to two GSM services.
    a. ATP Service for querying/downloading IDPS bundles and URL Filtering databases
    b. Logs service to upload the Firewall logs.

2. All API requests will be secured via HTTPS and will use mutual auth (GSM certs) for authentication. This is the existing scheme used today to authenticate any API requests coming in from VCO to GSM services.
3. VCO registers with NTICS through GSM APIs.
    a. Creating a new enterprise with EFS enabled, or enabling EFS on an enterprise for the first time sends a CREATE request to GSM to create the NTICS license key.
    b. If a license key expires, VCO sends a new CREATE to GSM to generate a new NTICS license key.
    c. Disabling EFS on an enterprise sends an UPDATE to GSM to mark license as inactive.
    d. Re-enabling EFS on an enterprise makes an UPDATE call to GSM. If not expired, the license is marked active. If expired, a new license key is generated.
    e. Deleting an Enterprise sends a DELETE to GSM to remove the license information immediately.
    f. Expired/inactive license keys are deleted by the background cleanup job.
    g. License keys are encrypted before storing in the DB and decrypted after reading from the DB.

16.3.3 Edge and NTICS

1. Edge receives NTICS license key information from VCO as part of the heartbeat.
2. Edge uses license key to call NSX authenticate API, which return a client secret and token used by the Webroot SDK when making network queries to NTICS

16.3.4 GSM with NTICS

1. NTICS team will provide credentials of the special user to the SDWAN Edge Ops team.
2. These credentials will be then added to AWS Secrets Manager from where the service will read them and authenticate with NTICS.
3. More details on NTICS authentication API can be found here https://confluence.eng.vmware.com/x/dxHKEw

16.3.5 NTICS with GSM

There will be a dedicated role for NTICS privilege introduced in GSM for allowing Edges to authenticate with to NTICS via the GSM service. The new NTICS role will be on par with the existing roles VCO, SUPPORT, OPERATOR.

Three pairs of certificate-key will be created and provided for NITCS to access GSM service at test/preprod/prod envs respectively with common name as below,

- VeloCloudEdgeFwProd::NTICS
- VeloCloudEdgeFwPreprod::NTICS
- VeloCloudEdgeFwTest::NTICS

The certificate expiration is 1 year after issued, it has to be provided in manual way right now.

16.4 Customer/System Data Interaction

All other data flow diagrams have been captured in the Functional Spec (see page 6). Please refer to the Functional Spec section 15.5 for more details.

16.5 Attack Surface

Service	Action	Impact
Firewall ATP Service (GSM) Firewall Logging Service (GSM)	Malicious user intercepts the requests	No Impact since the connection is secured with HTTPS.
Firewall ATP Service (GSM)	GSM Certificate is compromised	Fake enterprise NTICS licenses can be generated, but can't be used and will be eventually cleaned up. Existing customers not impacted since enterpriseLogicalId is needed to update license keys. Risk of running out of space on GSM boxes with junk licenses.
VCO UI	Malicious user tries to view logs of another enterprises	No impact since the enterprise logical id is verified to make sure they are same in the request and authentication cookie before returning the logs
Edge	Malicious user intercepts heartbeat	Can authenticate with NTICS to obtain a client token/secret. No customer data exposed because it only allows access to NTICS' public APIs.

17 Effort estimation for security groups

Task	Sub tasks		Individual coding tasks involved	Time taken
UI	This involves UI 1. Writing models for security groups similar to address_group.js (William/ Praveen).Models should insert/ get/update/ delete enterprise objects (William/ Praveen)			2 weeks (Praveen to confirm on Aug 22nd).This can be done in parallel

Task	Sub tasks		Individual coding tasks involved	Time taken
Backend Changes for configuration on VCO	CRUD operation for IDPS/URL filtering/ URL reputation/Mal IP filtering object groups	INITIALIZATIONS	define a new type for all object groups in enums.js	3 weeks
		INSERT	**insertObjectGroup.js:** Add any object group type specific handling if required	
			insertOrUpdateObjectGroup.js Add semantic Validation for IDPS/URL filtering/ URL reputation/ IP filtering group	
		DELETE	**deleteObjectGroup.js :** 1. Update getObjectUsageCounts / add new API to check if object is in new ASSOC table `VELOCLOUD_ENTERPRISE_OBJECT_ASSOC` and if so do not allow deletion. Return error to front end that IDPS/URL object is in use by security group.	
		GET	**getObjectGroups.js:(support required with profile and edges)** Have to add additional parameter keys in data	
		UPDATE	**updateObjectGroup.js** Add any object group type specific handling if required Check if it being used as part of security object group and modify security object version	

Task	Sub tasks		Individual coding tasks involved	Time taken
			insertOrUpdateObjectGroup.js Add semantic Validation for IDPS/URL filtering/ URL reputation/ IP filtering group	
		TEST	Unit tests	
	CRUD operation for security service groups	INITIALIZATIONS	1. Define a new type for all object groups in enums.js 2. Define new table `VELOCLOUD_ENTERPRISE_OBJECT_ASSOC`	
		INSERT	**InsertObjectGroups.js** 1. Modify insertObjectGroup API to check if the type is security service group and add entries to `VELOCLOUD_ENTERPRISE_OBJECT_ASSOC(`associations between IDPS/URL group and security service group). Do the required validations and return error if incorrect config.	
			insertOrUpdateObjectGroup.js Add semantic Validation for security groups during insert	

Task	Sub tasks		Individual coding tasks involved	Time taken
		DELETE	**deleteObjectGroup.js :** 1. Verify if getObjectUsageCounts works as is to catch association between security service group and firewall rule(if not update the API) and return error to front-end that security service object is in use by a rule.	
		GET	**getObjectGroups.js :(support required with profile and edges)** 1. Have to send data for security service objects. 2. Need to show what profiles use this object	
		UPDATE	**updateObjectGroup.js** Update entries in `VELOCLOUD_ENTERPRISE_OBJECT_ASSOC` when type is type is security service group. Modify security object version	
			insertOrUpdateObjectGroup.js Add semantic Validation for security groups during update.	
		TEST	Unit test	

Task	Sub tasks		Individual coding tasks involved	Time taken
	Security group is associated with a rule	Creating refs	**processObjectGroupAssoc.js and policyutils.js** Add entries to VELOCLOUD_OBJECT_CONFIGURATION_ASSOC with refs "objectGroup:securityGroup"	2 week
		Resolving and showing refs on UI Validate object groups are returned as part of config fetch	**getConfiguration.js** Ensure refs show up on the UI and are resolved	
			resolveModuleAssocObjectData.js Resolve individual security engines' refs embedded in security service groups	
	Security group is dis-associated from a rule	Removing refs	**processObjectGroupAssoc.js and policyutils.js** Remove entries from VELOCLOUD_OBJECT_CONFIGURATION_ASSOC	
		Resolving and showing refs on UI	**getConfiguration.js** Ensure refs are removed	
	Define JSON schema			
	Special handling for **atp_action**		Where do we populate atp_action? 1. During object insert/ update or 2. During rule association	

Task	**Sub tasks**		**Individual coding tasks involved**	**Time taken**
	Enabling disabling EFS config at profile level/edge level	Ensure config is properly populated in firewall module when feature is enabled at profile level / edge level	**propagateConfigurationModuleChangeFirewall.js** 1. populate new flag config in firewall module for URL filtering /IDPS	
Propagate Changes to edges	Heartbeat	Version change to modules	**FW module version change when any of the object groups are updated**	2 weeks
			policyutils.js (properties module version) 1. Check timestamp of security group object and check if it is later than fw moduler version. if, so Update HB repsonse with refs (take care to remove data key in url fi) 2. Modify getPropertiesModule to skip objects of type idps/ urlf/ipr	
		Updating HB response	1. Verify securityServiceGroupLogicalId is sent as part of rules 2. Verify atp_action is populated when idps is present	

Task	Sub tasks		Individual coding tasks involved	Time taken
		Ensure getComposite configuration returns correct config when features are disabled at profile level but overridden at edge level.	getEdgeCompositeConfiguration.js 1. Copy the correct configuration 2. Update the correct version(profile vs edge version)	
Upgrade patches	1. Create default profiles and associate to rules 2. updating the rules with newly created sec service groups id			2 weeks

Storing	
Initializing	
Creating/Deleting refs	

18 URL/IP Filtering Query APIs

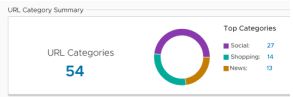
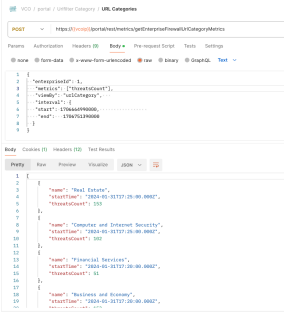
18.1 UX Mockup

Sorry, the widget is not supported in this export.
But you can reach it using the following URL:

https://www.figma.com/file/udG3bfEiaJYuUD5z0hZAVf/Edge-Firewall-ATP?type=design&node-id=5466-33815&mode=design&t=xOhbzZXX7fi6qtRF-0

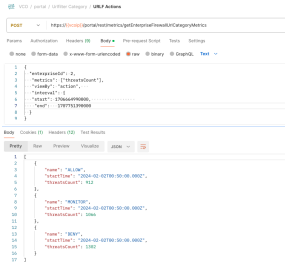
18.2 URL Filtering Cateogry

	Dashboard Widget	API	Request	Response

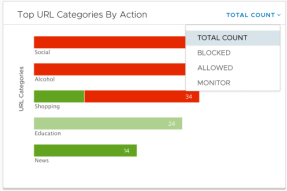
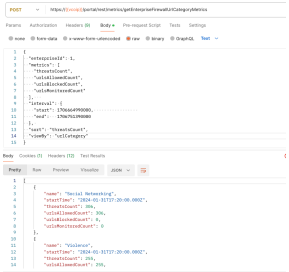
1		Edge:
		/portal/metric/getEdgeFirewallUrlCategoryMetrics
		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlCategoryMetrics
		 
			{
			"enterpriseId": 1,
			"metrics": ["threatsCount"],
			"viewBy": "urlCategory",
			"interval": {
			"start": 1706664990000,
			"end": 1706751390000
			}
			}
				{
				jsonrpc: "2.0",
				result: [
				{
				"name": "Real Estate",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 153
				},
				{
				"name": "Computer and Internet Security",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 102
				},
				{
				"name": "Financial Services",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 51
				},
				{

	Dashboard Widget	API	Request	Response
				"name": "Business and Economy", "startTime": "2024-01-31T17:20:00.000Z", "threatsCount": 153 }, { "name": "Computer and Internet Info", "startTime": "2024-01-31T17:25:00.000Z", "threatsCount": 102 }], id: 10, }

	Dashboard Widget	API	Request	Response

2		Edge:
		/portal/metric/getEdgeFirewallUrlCategoryMetrics
		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlCategoryMetrics
		 
			{
			"enterpriseId": 2,
			"metrics": ["threatsCount"],
			"viewBy": "action",
			"interval": {
			"start": 1706664990000 0,
			"end": 1707751390000 0
			}
			}
				{
				"jsonrpc": "2.0",
				"result": [
				{
				"name": "ALLOW",
				"startTime": "2024-02-02T00:50:00.000Z",
				"threatsCount": 912
				},
				{
				"name": "MONITOR",
				"startTime": "2024-02-02T00:50:00.000Z",
				"threatsCount": 1066
				},
				{
				"name": "DENY",
				"startTime": "2024-02-02T00:50:00.000Z",
				"threatsCount": 1302
				}
				]
				"id": 10
				}

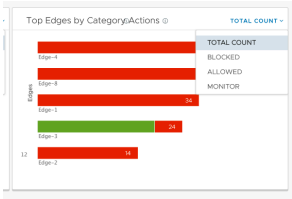
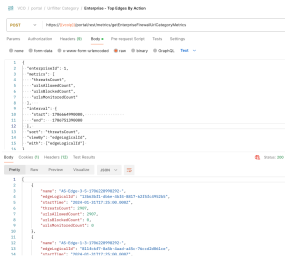
	Dashboard Widget	API	Request	Response

3		Edge:
		/portal/metric/getEdgeFirewallUrlCategoryMetrics
		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlCategoryMetrics
		 
			{
			enterpriseId: 1,
			metrics: [
			"threatsCount",
			"urlsAllowedCount",
			"urlsBlockedCount",
			"urlsMonitoredCount"
			],
			sort: "threatsCount",
			viewBy: "urlCategory",
			interval: { start: 100, end: 200 }
			}
				{
				jsonrpc: "2.0",
				result: [
				{
				"name": "Social Networking",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 306,
				"urlsAllowedCount": 306,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{
				"name": "Violence",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 255,
				"urlsAllowedCount": 255,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{

	Dashboard Widget	API	Request	Response
				"name": "Recreation and Hobbies", "startTime": "2024-01-31T17:25:00.000Z", "threatsCount": 255, "urlsAllowedCount": 255, "urlsBlockedCount": 0, "urlsMonitoredCount": 0 }, { "name": "Search Engines", "startTime": "2024-01-31T17:25:00.000Z", "threatsCount": 204, "urlsAllowedCount": 204, "urlsBlockedCount": 0, "urlsMonitoredCount": 0 }, { "name": "SPAM URLs", "startTime": "2024-01-31T17:20:00.000Z",

	Dashboard Widget	API	Request	Response
				"threatsCount": 204, "urlsAllowedCount": 204, "urlsBlockedCount": 0, "urlsMonitoredCount": 0 }], id: 10, }

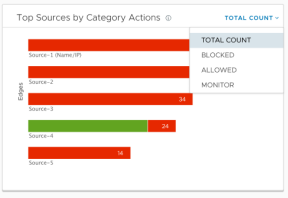
	Dashboard Widget	API	Request	Response

4		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlCategoryMetrics
		 
			{
			"enterpriseId": 1,
			"metrics": [
			"threatsCount",
			"urlsAllowedCount",
			"urlsBlockedCount",
			"urlsMonitoredCount"
			],
			"interval": {
			"start": 1706664990000,
			"end": 1706751390000
			},
			"sort": "threatsCount",
			"viewBy": "edgeLogicalId",
			"with": ["edgeLogicalId"]
			}
				{
				jsonrpc: "2.0",
				result: [
				{
				"name": "AS-Edge-3-5-1706228998292-",
				"edgeLogicalId": "13b63bf1-db6e-4bf4-8817-62f5fc4952b5",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 2907,
				"urlsAllowedCount": 2907,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{
				"name": "AS-Edge-1-3-1706228998292-",
				"edgeLogicalId": "811fc6f7-8a5b-4aad-a45c-76ccd2d061ce",
				"startTime": "2024-01-31T17:25:00.000Z",

	Dashboard Widget	API	Request	Response

				"threatsCount": 2754,
				
				"urlsAllowedCount": 2754,
				
				"urlsBlockedCount": 0,
				
				"urlsMonitoredCount": 0
				},
				{
				"name": "AS-Edge-2-4-1706228998292-",
				
				"edgeLogicalId": "06d27df9-a22d-432a-8d5f-a753cc3083a1",
				
				"startTime": "2024-01-31T17:25:00.000Z",
				
				"threatsCount": 2652,
				
				"urlsAllowedCount": 2652,
				
				"urlsBlockedCount": 0,
				
				"urlsMonitoredCount": 0
				}
				],
				id: 10,
				}

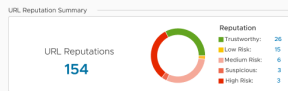
	Dashboard Widget	API	Request	Response

5		Edge:
		/portal/metric/getEdgeFirewallUrlCategoryMetrics
			{
			
			enterpriseId: 1,
			edgeId: 2,
			
			metrics: [
			
			"threatsCount",
			
			"urlsBlockedCount",
			
			"urlsBlockedCount",
			
			"urlsMonitoredCount"
			],
			sort: "threatsCount",
			viewBy: "sourceIp",
			interval: { start: 100, end: 200 }
			}
				{
				jsonrpc: "2.0",
				result: [
				{
				name: "10.12.1.16",
				
				threatsCount: 54,
				
				urlsAllowedCount: 14,
				urlsBlockedCount: 14,
				urlsMonitoredCount: 40,
				},{
				name: "10.12.1.17",
				
				threatsCount: 54,
				
				urlsAllowedCount: 14,
				urlsBlockedCount: 14,
				urlsMonitoredCount: 40,
				},{
				name: "10.12.1.18",
				
				threatsCount: 54,
				
				urlsAllowedCount: 14,
				urlsBlockedCount: 14,
				urlsMonitoredCount: 40,
				},{
				name: "10.12.1.19",

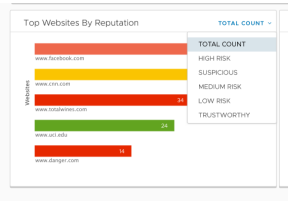
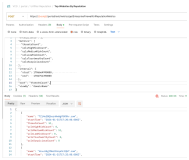
	Dashboard Widget	API	Request	Response

				threatsCount: 54,
				
				urlsAllowedCount
				: 14,
				urlsBlockedCount
				: 14,
				urlsMonitoredCou
				nt: 40,
				},{
				name:
				"10.12.1.20",
				
				threatsCount: 54,
				
				urlsAllowedCount
				: 14,
				urlsBlockedCount
				: 14,
				urlsMonitoredCou
				nt: 40,
				}
				]
				id: 10,
				}

18.3 URL Filtering Reputation

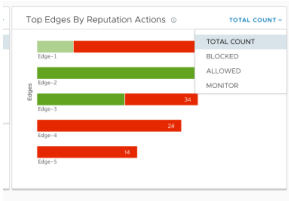
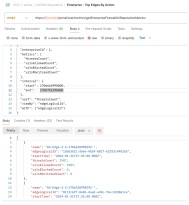
1		Edge: /portal/metric/getEdgeFirewallUrlReputationMetrics
		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlReputationMetrics
		 
			{
			enterpriseId: 1,
			edgeId: 2,
			metrics: ["threatsCount"],
			viewBy: ["urlRisk"],
			interval: { start: 100, end: 200 }
			}
				{
				"jsonrpc": "2.0",
				"result": [
				{
				"name": "HIGH",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 1530
				},
				{
				"name": "SUSPICIOUS",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 1785
				},
				{
				"name": "MEDIUM",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 1326
				},
				{
				"name": "LOW",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 1887
				},
				{
				"name": "TRUSTWORTHY",
				"startTime": "2024-01-31T17:25:00.000Z",

				"threatsCount": 1785 }] "id": 10 }
2		Edge: /portal/metric/getEdgeFirewallUrlReputationMetrics Enterprise: /portal/metric/getEnterpriseFirewallUrlReputationMetrics 	{ enterpriseId: 1, edgeId: 2, metrics: ["threatsCount"], viewBy: "action", interval: { start: 100, end: 200 } }	{ "jsonrpc": "2.0", "result": [{ "name": "ALLOW", "threatsCount": 14 }, { "name": "DENY", "threatsCount": 27 }, { "name": "MONITOR", "threatsCount": 13 },] "id": 10 }

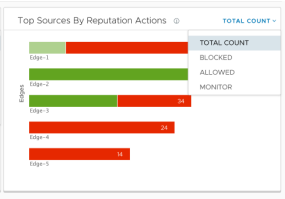
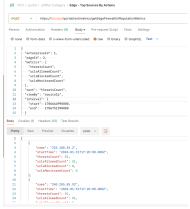
3		Edge:
		/portal/metric/getEdgeFirewallUrlReputationMetrics
		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlReputationMetrics
		 
			{
			"enterpriseId": 1,
			"metrics": [
			"threatsCount",
			"urlsHighRiskCount",
			"urlsMediumRiskCount",
			"urlsLowRiskCount",
			"urlsTrustworthyCount",
			"urlsSuspiciousCount"
			],
			"interval": {
			"start": 1706664990000,
			"end": 1706751390000
			},
			"sort": "threatsCount",
			"viewBy": "domainName"
			}
				{
				jsonrpc: "2.0",
				result: [
				{
				"name": "TZi4efBQbaznNxWgY8VOKv.com",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 51,
				"urlsHighRiskCount": 0,
				"urlsMediumRiskCount": 51,
				"urlsLowRiskCount": 0,
				"urlsTrustworthyCount": 0,
				"urlsSuspiciousCount": 0
				},
				{
				"name": "Usvu15gtMmLCKrqe9cJQb7.com",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 51,
				"urlsHighRiskCount": 51,
				"urlsMediumRiskCount": 0,
				"urlsLowRiskCount": 0,
				"urlsTrustworthyCount": 0,
				"urlsSuspiciousCount": 0
				},

```
    {
        "name":
"oaTIyd3A9jqzw8Dv1nuHQ7
.com",
        "startTime":
"2024-01-31T17:20:00.00
0Z",
        "threatsCount":
51,

"urlsHighRiskCount": 0,

"urlsMediumRiskCount":
51,

"urlsLowRiskCount": 0,

"urlsTrustworthyCount":
0,

"urlsSuspiciousCount":
0
    },
    {
        "name":
"iYNjWUoFZf3PyktcuRgrHV
.com",
        "startTime":
"2024-01-31T17:20:00.00
0Z",
        "threatsCount":
51,

"urlsHighRiskCount": 0,

"urlsMediumRiskCount":
0,

"urlsLowRiskCount": 51,

"urlsTrustworthyCount":
0,

"urlsSuspiciousCount":
0
    },
    {
        "name":
"IbWHMycw5KDjrlv7PdGigJ
.com",
```

				"startTime": "2024-01-31T17:20:00.000Z", "threatsCount": 51, "urlsHighRiskCount": 0, "urlsMediumRiskCount": 0, "urlsLowRiskCount": 0, "urlsTrustworthyCount": 51, "urlsSuspiciousCount": 0 }] id: 10, }

4		Enterprise:
		/portal/metric/getEnterpriseFirewallUrlReputationMetrics
		 
			{
			"enterpriseId": 1,
			"metrics": [
			"threatsCount",
			"urlsAllowedCount",
			"urlsBlockedCount",
			"urlsMonitoredCount"
			],
			"interval": {
			"start": 1706664990000,
			"end": 1706751390000
			},
			"sort": "threatsCount",
			"viewBy": "edgeLogicalId",
			"with": ["edgeLogicalId"]
			}
				{
				jsonrpc: "2.0",
				result: [
				{
				"name": "AS-Edge-3-5-1706228998292-",
				"edgeLogicalId": "13b63bf1-db6e-4bf4-8817-62f5fc4952b5",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 2907,
				"urlsAllowedCount": 2907,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{
				"name": "AS-Edge-1-3-1706228998292-",
				"edgeLogicalId": "811fc6f7-8a5b-4aad-a45c-76ccd2d061ce",
				"startTime": "2024-01-31T17:25:00.000Z",
				"threatsCount": 2754,
				"urlsAllowedCount": 2754,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{
				"name": "AS-Edge-2-4-1706228998292-",

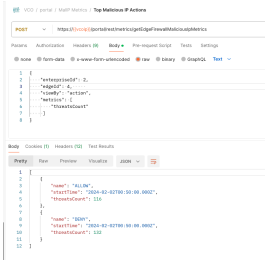
				"edgeLogicalId": "06d27df9-a22d-432a-8d5f-a753cc3083a1", "startTime": "2024-01-31T17:25:00.000Z", "threatsCount": 2652, "urlsAllowedCount": 2652, "urlsBlockedCount": 0, "urlsMonitoredCount": 0 }] id: 10, }

5	
	There is typo in the screenshot by the time it is take. Edge-X should be IP-X
		Edge:
		/portal/metric/getEdgeFirewallUrlReputationMetrics
		 
			{
			"enterpriseId": 1,
			"edgeId": 2,
			"metrics": [
			"threatsCount",
			"urlsAllowedCount",
			"urlsBlockedCount",
			"urlsMonitoredCount"
			],
			"sort": "threatsCount",
			"viewBy": "sourceIp",
			"interval": {
			"start": 1706664990000,
			"end": 1706751390000
			}
			}
				{
				jsonrpc: "2.0",
				result:
				[
				{
				"name": "225.180.45.2",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 51,
				"urlsAllowedCount": 51,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{
				"name": "240.205.85.92",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 51,
				"urlsAllowedCount": 51,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},
				{
				"name": "253.42.31.118",
				"startTime": "2024-01-31T17:20:00.000Z",
				"threatsCount": 51,
				"urlsAllowedCount": 51,
				"urlsBlockedCount": 0,
				"urlsMonitoredCount": 0
				},

				{ "name": "231.91.106.44", "startTime": "2024-01-31T17:20:00.000Z", "threatsCount": 51, "urlsAllowedCount": 51, "urlsBlockedCount": 0, "urlsMonitoredCount": 0 }, { "name": "233.45.125.200", "startTime": "2024-01-31T17:20:00.000Z", "threatsCount": 51, "urlsAllowedCount": 51, "urlsBlockedCount": 0, "urlsMonitoredCount": 0 }] id: 10, }

18.4 Malicious IP

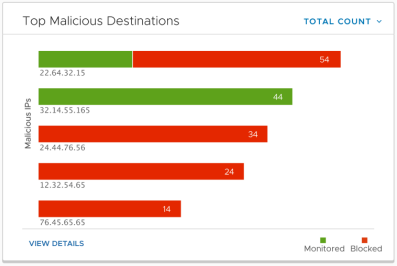
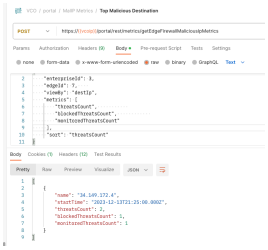
Metrics view Name	**Dashboard Widget**	**Metrics APIs**	**API Request**	**API Response**

Top Malicious IP Actions (**Both Edge and Enterprise**)		Edge:
		/portal/metric/getEdgeMaliciousIpMetrics
		Enterprise:
		/portal/metric/getEnterpriseMaliciousIpMetrics
		 
			{
			"edgeId": 1,
			"enterpriseId": 1,
			"limit": 5,
			"sort": "threatsCount",
			"metrics": [
			"threatsCount"
			],
			"viewBy": "action",
			"interval": {
				{
				"jsonrpc": "2.0",
				"result": [
				{
				"name": "ALLOW",
				"threatsCount": 150
				},
				{
				"name": "DENY",
				"threatsCount": 154
				},
				]
				"id": 10
				}

Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response

			`"start": 1664577627120,`
			`"end": 1677532677000`
			`}`
			`}`	

Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response

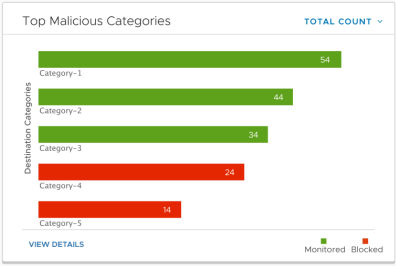
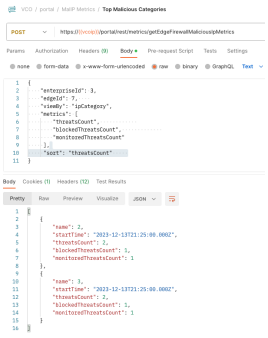
Top Malicious Destination (Both Edge and Enterprise)		Edge:
		/portal/metric/getEdgeMaliciousIpMetrics
		Enterprise:
		/portal/metric/getEnterpriseMaliciousIpMetrics
		 
			{
			"edgeId": 1,
			"enterpriseId": 1,
			"limit": 5,
			"sort": "threatsCount",
			"metrics": [
			"threatsCount",
			"blockedThreatsCount",
			"monitoredThreatsCount",
			],
			"viewBy": "destIp",
				{
				jsonrpc: "2.0",
				result: [
				{
				name: "22.64.32.15",
				threatsCount: 54,
				blockedThreatsCount: 30,
				monitoredThreatsCount: 24,
				},{
				name: "32.14.55.165",
				threatsCount: 44,
				blockedThreatsCount: 0,
				monitoredThreatsCount: 24,
				},{
				name: "24.44.76.56",
				threatsCount: 34,
				blockedThreatsCount: 34,
				monitoredThreatsCount: 0,

Metrics View Name	Dashboard Widget	Metrics APIs	API Request	API Response
			"interval": { "start": 1664577627120, "end": 1677532677000 } }	},{ name: "12.32.54.65", threatsCount: 24, blockedThreatsCount: 24, monitoredThreatsCount: 0, },{ name: "76.45.65.65", threatsCount: 14, blockedThreatsCount: 14, monitoredThreatsCount: 0, }] id: 10, }

Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response
	Top Malicious Destination - Blocked Count		{ "edgeId": 1, "enterpriseId": 1, "limit": 5, "sort": "blockedThreatsCount", "metrics": ["blockedThreatsCount"], "viewBy": "destIp", "interval": { "start": 1664577627120,	{ jsonrpc: "2.0", result: [{ name: "24.44.76.56", blockedThreatsCount: 34, },{ name: "22.64.32.15", blockedThreatsCount: 30, },{ name: "12.32.54.65", blockedThreatsCount: 24, },{ name: "76.45.65.65", blockedThreatsCount: 14, }, { name: "32.14.55.165", blockedThreatsCount: 0, }] id: 10, }

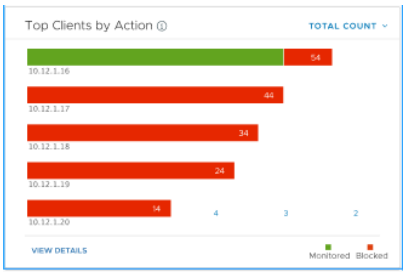
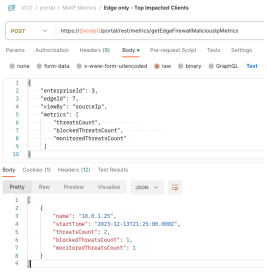
Metricsview Name	Dashboard Widget	Metrics APIs	API Request	API Response
			"end": 1677532677000 } }	

Metrics View Name	Dashboard Widget	Metrics APIs	API Request	API Response

Top Malicious Categories (Both Edge and Enterprise)		Edge:
		/portal/metric/getEdgeMaliciousIpMetrics
		Enterprise:
		/portal/metric/getEnterpriseMaliciousIpMetrics
		 
			{
			"edgeId": 1,
			"enterpriseId": 1,
			"limit": 5,
			"sort": "threatsCount",
			"metrics": [
			"threatsCount",
			"blockedThreatsCount",
			"monitoredThreatsCount",
			],
			"viewBy": "ipCategory",
				{
				jsonrpc: "2.0",
				result: [
				{
				name: "Category-1",
				threatsCount: 54,
				blockedThreatsCount: 0,
				monitoredThreatsCount: 54,
				},{
				name: "Category-2",
				threatsCount: 44,
				blockedThreatsCount: 0,
				monitoredThreatsCount: 44,
				},{
				name: "Category-3",
				threatsCount: 34,
				blockedThreatsCount: 0,
				monitoredThreatsCount: 34,
				},{
				name: "Category-4",

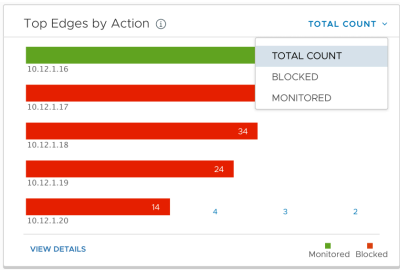
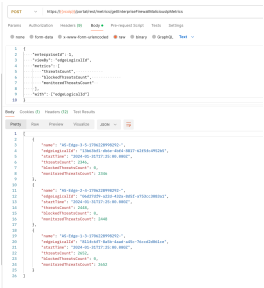
Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response
			"interval": { "start": 1664577627120, "end": 1677532677000 } }	threatsCount: 24, blockedThreatsCount: 24, monitoredThreatsCount: 0, },{ name: "Category-5", threatsCount: 14, blockedThreatsCount: 14, monitoredThreatsCount: 0, }] id: 10, }

Metrics view Name	**Dashboard Widget**	**Metrics APIs**	**API Request**	**API Response**

Top Impacted Clients (**Edge Only**)		Edge:
		/portal/metric/getEdgeMaliciousIpMetrics
		 
			{
			"edgeId": 1,
			"enterpriseId": 1,
			"limit": 5,
			"sort": "threatsCount",
			"metrics": [
			"threatsCount",
			"blockedThreatsCount",
			"monitoredThreatsCount"
			],
			"viewBy": "sourceIp",
				{
				jsonrpc: "2.0",
				result: [
				{
				name: "10.12.1.16",
				threatsCount: 54,
				blockedThreatsCount: 14,
				monitoredThreatsCount: 40,
				},{
				name: "10.12.1.17",
				threatsCount: 44,
				blockedThreatsCount: 44,
				monitoredThreatsCount: 0,
				},{
				name: "10.12.1.18",
				threatsCount: 34,
				blockedThreatsCount: 34,
				monitoredThreatsCount: 0,
				},{
				name: "10.12.1.19",

Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response
			"interval": { "start": 1664577627120, "end": 1677532677000 } }	threatsCount: 24, blockedThreatsCount: 24, monitoredThreatsCount: 0, },{ name: "10.12.1.20", threatsCount: 14, blockedThreatsCount: 14, monitoredThreatsCount: 0, }] id: 10, }

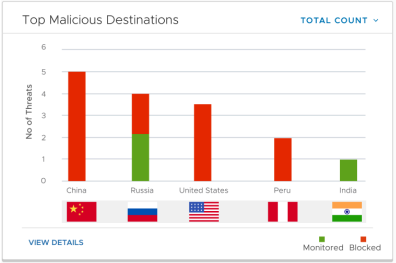
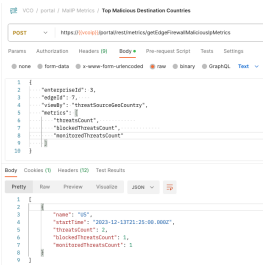
Metrics view Name	**Dashboard Widget**	**Metrics APIs**	**API Request**	**API Response**

Top Impacted Edges (**Enterprise Only**)		Enterprise:
		/portal/metric/getEnterpriseMaliciousIpMetrics
		 
			{
			"enterpriseId": 1,
			"limit": 5,
			"sort": "threatsCount",
			"metrics": [
			"threatsCount",
			"blockedThreatsCount",
			"monitoredThreatsCount",
			],
			"viewBy": "edgeLogicalId",
			"interval": {
				{
				jsonrpc: "2.0",
				result: [
				{
				name: "Edge-1",
				"edgeLogicalId": "3344b248-53d7-11ed-ac8d-0242ac160001",
				threatsCount: 54,
				blockedThreatsCount: 14,
				monitoredThreatsCount: 40,
				},{
				name: "Edge-2",
				"edgeLogicalId": "3344b248-53d7-11ed-ac8d-0242ac160002",
				threatsCount: 44,
				blockedThreatsCount: 44,
				monitoredThreatsCount: 0,

Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response
			"start": 1664577627120, "end": 1677532677000 }, "with": ["edgeLogicalId"] }	},{ name: "Edge-3", "edgeLogicalId": "3344b248-53d7-11ed-ac8d-0242ac160003", threatsCount: 34, blockedThreatsCount: 34, monitoredThreatsCount: 0, },{ name: "Edge-4", "edgeLogicalId": "3344b248-53d7-11ed-ac8d-0242ac160004", threatsCount: 24, blockedThreatsCount: 24, monitoredThreatsCount: 0, },{ name: "Edge-5",

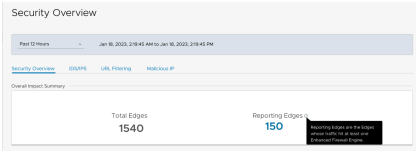
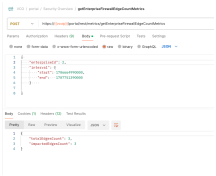
Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response
				"edgeLogicalId": "3344b248-53d7-11ed-ac8d-0242ac160005", threatsCount: 14, blockedThreatsCount: 14, monitoredThreatsCount: 0, }] id: 10, }

Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response

Top Malicious Destinations (Both Edge and Enterprise)		Edge:
		/portal/metric/getEdgeMaliciousIpMetrics
		Enterprise:
		/portal/metric/getEnterpriseMaliciousIpMetrics
		 
			{
			"edgeId": 1,
			"enterpriseId": 1,
			"limit": 5,
			"sort": "threatsCount",
			"metrics": [
			"threatsCount",
			"blockedThreatsCount",
			"monitoredThreatsCount",
			],
				{
				jsonrpc: "2.0",
				result: [
				{
				name: "CH",
				threatsCount: 5,
				blockedThreatsCount: 5,
				monitoredThreatsCount: 0,
				},{
				name: "RU,
				threatsCount: 4,
				blockedThreatsCount: 2,
				monitoredThreatsCount: 2,
				},{
				name: "US",
				threatsCount: 3,
				blockedThreatsCount: 3,
				monitoredThreatsCount: 0,
				},{
				name: "PE",

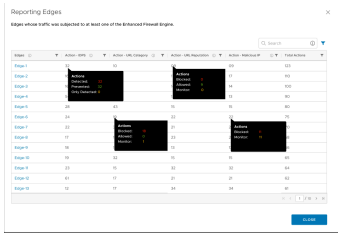
Metrics view Name	Dashboard Widget	Metrics APIs	API Request	API Response
			"viewBy": "threatSourceGeoCountry", "interval": { "start": 1664577627120, "end": 1677532677000 } }	threatsCount: 2, blockedThreatsCount: 2, monitoredThreatsCount: 0, },{ name: "IN", threatsCount: 1, blockedThreatsCount: 0, monitoredThreatsCount: 1, }] id: 10, }

18.5 Enterprise - Overall Impact Summary

		API Call	Request Example	Response Example
		A new API will be introduced /portal/metrics/getEnterpriseFirewallEdgeCountMetrics 	`{ "method": "metrics/getEnterpriseFirewallEdgeCountMetrics", "params": { "enterpriseId": 1, "interval": { "start": 1664577627120, "end": 1677532677000 } } }`	`{ "totalEdgesCount": 100, "impactedEdgesCount": 50 }`

18.6 Enterprise - Reporting Edges Summary

		API Call	Request Example	Response Example

		A new API will be introduced
		/portal/metrics/getEnterpriseFirewallEdgeSummaryMetrics
		1. search and filter, need confirm if both search and filter are necessary, it might make design and implementation complex.
			{
			"method": "metrics/getEnterpriseFirewallEdgeSummaryMetrics",
			"params": {
			"enterpriseId": 1,
			"metrics": [
			"idpsActions",
			"urlCatActions",
			"urlRepActions",
			"malIpActions"
			],
			"interval": {
			"start": 1664577627120,
			"end": 1677532677000
			}
			}
				[
				{
				"name": "AS-Edge-2-4-1706228998292-",
				"edgeLogicalId": "21fa42d6-b29c-4dd7-8961-387fe0a2e269",
				"totalActions": {
				"threatsCount": 884
				},
				"idpsActions": {
				"threatsCount": 200,
				"detectedThreatsCount": 136,
				"preventedThreatsCount": 64
				},
				"urlCatActions": {
				"threatsCount": 220,
				"urlsBlockedCount": 108,
				"urlsMonitoredCount": 0,
				"urlsAllowedCount": 112
				},
				"urlRepActions": {
				"threatsCount": 220,
				"urlsBlockedCount": 96,

		API Call	Request Example	Response Example
		2. pagination, the API need pull data from 4 separated CH table, each query might return different set of Edges. So we probably can't support pagination at backend API. It might be much easier to do so at UI side, give the number of edge is at most in thousands.		"urlsMonitoredCount": 0, "urlsAllowedCount": 124 }, "malIpActions": { "threatsCount": 244, "blockedThreatsCount": 124, "monitoredThreatsCount" : 120 }, "edgeId": 3 }, { "name": "AS- Edge-3-5-1706228998292- ", "edgeLogicalId": "08fcc3ef-7ba2-48ca-9b4 1-49e0cdd7269b", "totalActions": { "threatsCount": 1085 }, "idpsActions": { "threatsCount": 245, "detectedThreatsCount": 190, "preventedThreatsCount" : 55 }, "urlCatActions": { "threatsCount": 295, "urlsBlockedCount": 95,

		API Call	Request Example	Response Example
		3. Sort, it is TBD the result sorted by edge name or edge Id. The new API acts as proxy to redirect the request to four existing common lib API for IDPS, urlCat, urlRep and maliciousIp respetively. 		"urlsMonitoredCount": 0, "urlsAllowedCount": 200 }, "urlRepActions": { "threatsCount": 295, "urlsBlockedCount": 150, "urlsMonitoredCount": 0, "urlsAllowedCount": 145 }, "malIpActions": { "threatsCount": 250, "blockedThreatsCount": 140, "monitoredThreatsCount" : 110 }, "edgeId": 2 }, { "name": "AS- Edge-1-3-1706228998292- ", "edgeLogicalId": "3c591516-27fd-44c2- b4c1-7a4a3710616d", "totalActions": { "threatsCount": 1270 }, "idpsActions": { "threatsCount": 380, "detectedThreatsCount": 295,

		API Call	Request Example	Response Example
				"preventedThreatsCount" : 85 }, "urlCatActions": { "threatsCount": 285, "urlsBlockedCount": 145, "urlsMonitoredCount": 0, "urlsAllowedCount": 140 }, "urlRepActions": { "threatsCount": 285, "urlsBlockedCount": 150, "urlsMonitoredCount": 0, "urlsAllowedCount": 135 }, "malIpActions": { "threatsCount": 320, "blockedThreatsCount": 170, "monitoredThreatsCount" : 150 }, "edgeId": 1 }]

18.7 Enterprise and Edge Detail View (Draft) (POST Yamazaki)

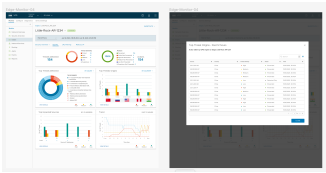
API Request Parameters

- The viewBy and metrics parameters should be absent. The query is supposed to fetch and filter DB records without grouping.
- The firewallStats metric API will support **filters** parameter as flowStats does.
- The firewallStats metric APIs support **sortBy** parameter as flowStats does, e.g. sortBy: [{attribute: "sourceIp", type: "DESC"}].
- The firewallStats metric APIs support **quickSearch** parameter, e.g. quickSearch: "keyword".
- Pagination, the API will return **prevPageLink** and **nextPageLink** in metaData field which could be used as parameters at next API request.
- TBD, if the firewallStats metric APIs support filterSpec?

API Backend

- Expanding getEdgeFirewallMetricsMultiple to support all new requirements.
- The subquery is used for aggregation cases to reduce the scope before grouping, but it might be unnecessary for the detail view scenario.
- Modify the existing portal metrics APIs to allow UI make the proper call.

	API Call	Request Example	Response Example

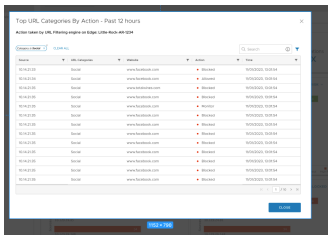
IDPS view details 	/portal/metrics/getEdgeFirewallIdpsMetrics	{
		"method": "metrics/getEdgeFirewallIdpsMetrics",
		"params": {
		"edgeId": 1,
		"enterpriseId": 1,
		"interval": {
		"start": 1664577627120,
		"end": 1665787227120
		},
		"limit": 50,
		"filters": {
		"and": [
		{
		"field": "threatImpact",
		"operator": "is",
		"value": "HIGH"
		}
			{
			"jsonrpc": "2.0",
			"result": {
			"metaData": {
			"limit": 50,
			"prevPageLink" 'securityutils.encryptToken'
			"nextPageLink": 'securityutils.encryptToken'
			},
			"data": [
			{
			"startTime": "2023-09-27 22:30:00",
			"endTime": "2023-09-27 22:35:00",
			"enterpriseLogicalId": "70d2aa07-7592-41a5-b2f8-b0fd16088d46",
			"edgeLogicalId": "acac92dd-99ad-48a8-b467-8cc87b58ef24",
			"segmentLogicalId": "956e5d9c-1ff0-4452-b6e7-56a7fa0178b2",
			"ruleId": "",
			"alert": "IPS",
			"protocol": 6,
			"sourceIp": "10.0.1.25",
			"destIp": "34.149.172.4",
			"destPort": 80,
			"signatureId": 1102994,

	API Call	Request Example	Response Example

		]
		},
		
		"quickSearch": "example"
		
		"sortBy": [
		{
		
		"attribute": "sourceIp",
		
		"type": "DESC"
		}
		]
		}
		}
			"signatureName": "NSX - QE Test signature",
			
			"signatureCategory": "A Network Trojan was Detected",
			
			"signatureSeverity": 2,
			
			"threatImpact": "HIGH",
			
			"threatSourceIp": "34.149.172.4",
			
			"threatSourceGeoCountry": "",
			
			"threatTargetIp": "10.0.1.25",
			
			"threatsCount": "\u0002"
			},
			{
			"startTime": "2023-12-12 23:05:00",
			"endTime": "2023-12-12 23:10:00",
			
			"enterpriseLogicalId": "2f5e20d4-ba80-4a55-91b9-9bc599a3ea08",
			
			"edgeLogicalId": "55fa202b-b195-45d7-8bfc-b35bd59d4260",
			
			"segmentLogicalId": "cddd7405-15ac-45f3-8c41-05f02d30ca71",
			"ruleId": "Z6p8OflmSMKZe9RU2UQ3Pw",
			"alert": "IDS",
			"protocol": 6,
			"sourceIp": "10.0.1.25",

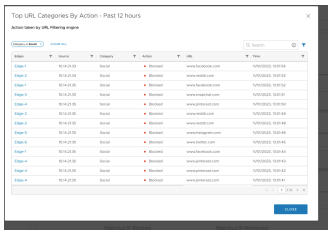
	API Call	Request Example	Response Example
			"destIp": "34.149.172.4", "destPort": 80, "signatureId": 0, "signatureName": "", "signatureCategory": "", "signatureSeverity": 0, "threatImpact": "HIGH", "threatSourceIp": "", "threatSourceGeoCountry": "US", "threatTargetIp": "", "threatsCount": "\u0001" }] }, "id": 115 }

	API Call	Request Example	Response Example

Edge - URL Categories view details
 
	/portal/metrics/getEdgeUrlCatMetrics	{
		"method": "metrics/getEdgeUrlCatMetrics",
		"params": {
		"edgeId": 1,
		"enterpriseId": 1,
		"interval": {
		"start": 1664577627120,
		"end": 1665787227120
		},
		"limit": 50,
		"filters": {
		"and": [
		{
		"field": "action",
		"operator": "is",
		"value": "DENY"
		}
		]
		},
			{
			"jsonrpc": "2.0",
			"result": {
			"metaData": {
			"limit": 50,
			"prevPageLink" 'securityutils.encryptToken'
			"nextPageLink": 'securityutils.encryptToken'
			},
			"data": [
			{
			"startTime": "2023-09-27 22:30:00",
			"endTime": "2023-09-27 22:35:00",
			"enterpriseLogicalId": "70d2aa07-7592-41a5-b2f8-b0fd16088d46",
			"edgeLogicalId": "acac92dd-99ad-48a8-b467-8cc87b58ef24",
			"segmentLogicalId": "956e5d9c-1ff0-4452-b6e7-56a7fa0178b2",
			"ruleId": "",
			"domainName": "www.vmware.com",
			"action": "DENY",
			"urlCategories": 1,
			"threatsCount": "\u0002"
			},
			{
			"startTime": "2023-09-27 22:30:00",

	API Call	Request Example	Response Example

		"quickSearch": "example"
		
		"sortBy": [
		{
		
		"attribute": "domainName",
		
		"type": "DESC"
		}
		]
		}
		}
			"endTime": "2023-09-27 22:35:00",
			
			"enterpriseLogicalId": "70d2aa07-7592-41a5-b2f8-b0fd16088d46",
			
			"edgeLogicalId": "acac92dd-99ad-48a8-b467-8cc87b58ef24",
			
			"segmentLogicalId": "956e5d9c-1ff0-4452-b6e7-56a7fa0178b2",
			"ruleId": "",
			
			"domainName": "www.broadcom.com",
			"action": "DENY",
			
			"urlCategories": 1,
			
			"threatsCount": "\u0002"
			}
			]
			},
			"id": 115
			}

	API Call	Request Example	Response Example
Enterprise - URL Categories view details 	/portal/metrics/getEnterpriseUrlCatMetrics	{ "method": "metrics/getEnterpriseUrlCatMetrics", "params": { "edgeId": 1, "enterpriseId": 1, "interval": { "start": 1664577627120, "end": 1665787227120 }, "limit": 50, "filters": { "and": [{ "field": "urlCategories", "operator": "is", "value": "11" }	Similar as Above, i.e. Edge - URL Categories view details

	API Call	Request Example	Response Example

		``
		]
		},
		
		"sortBy": [
		{
		
		"attribute"
		:
		"domainName
		",
		
		"type":
		"DESC"
		}
		]
		}
		}
		``	

19 URL Filtering - Monitoring APIs

~~Scope~~	~~Path~~	~~Description~~	~~Parameters~~	~~Response~~
~~Enterprise~~	~~metrics/getEnterpriseFirewallUrlMetrics~~	~~Top 5 Blocked URLs~~	`{ viewBy: "topUrlThreats", enterpriseId: 1, interval: { start: 0, end: 100 }, metrics: ["risksCount", "edgesCount"], with: ["category", "reputation"] }`	`... result: [ { "url": "megaworm.com", "category": "Gambling", "reputation": 0, "risksCount": 24, "edgesCount": 4 }, { "url": "cnn.com", "category": "News", "reputation": 85, "risksCount": 10, "edgesCount": 1 } ]`

~~Scope~~	~~Path~~	~~Description~~	~~Parameters~~	~~Response~~
		~~Top 5 Blocked Categories~~	`{ viewBy: "topCategoryThreats", enterpriseId: 1, interval: { start: 0, end: 100 }, metrics: ["risksCount"], with: [] }`	`... result: [{ "category": "Gambling", "risksCount": 24, "highRisksCount": 20, "suspiciousRisksCount: 4, "mediumRisksCount": 0, "lowRisksCount": 0, "noRisksCount": 0 }, { "category": "News", "risksCount": 10, "highRisksCount": 0, "suspiciousRisksCount": 6, "mediumRisksCount": 2, "lowRisksCount": 2, "noRisksCount: 0 }]`

~~Scope~~	~~Path~~	~~Description~~	~~Parameters~~	~~Response~~
		~~Edges with most blocked sessions~~	`{ viewBy: "edgesBlockedCount", enterpriseId: 1, interval: { start: 0, end: 100 }, metrics: ["risksCount"], with: ["category", "reputation"] }`	`... result: [ { "edgeLogicalId": "abcd", "risksCount": 10, "highRisksCount": 5, "suspiciousRisksCount": 5, "mediumRisksCount": 0, "lowRisksCount": 0, "noRisksCount": 0 } ]`

~~Scope~~	~~Path~~	~~Description~~	~~Parameters~~	~~Response~~
~~Edge~~	~~metrics/getEdgeFirewallUrlMetrics~~	~~Top 5 Blocked URLs~~	`{ viewBy: "topUrlThreats", enterpriseId: 1, edgeId: 1, interval: { start: 0, end: 100 }, metrics: ["risksCount"], with: ["category", "reputation"] }`	`... result: [ { "url": "megaworm.com", "category": "Gambling", "reputation": 0, "risksCount": 24 }, { "url": "cnn.com", "category": "News", "reputation": 85, "risksCount": 10 } ]`

~~Scope~~	~~Path~~	~~Description~~	~~Parameters~~	~~Response~~
		~~Top 5 Blocked Categories~~	`{ viewBy: "topCategoryThreats", enterpriseId: 1, edgeId: 1, interval: { start: 0, end: 100 }, metrics: ["risksCount"], with: [] }`	`... result: [{ "category": "Gambling", "risksCount": 24, "highRisksCount": 20, "suspiciousRisksCount: 4, "mediumRisksCount": 0, "lowRisksCount": 0, "noRisksCount": 0 }, { "category": "News", "risksCount": 10, "highRisksCount": 0, "suspiciousRisksCount": 6, "mediumRisksCount": 2, "lowRisksCount": 2, "noRisksCount: 0 }]`