ThreatQuotient

GitLab YARA Rules Parser CDF

Version 1.0.0

December 17, 2024

ThreatQuotient
20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

ThreatQ Supported

Support
Email: support@threatq.com
Web: support.threatq.com
Phone: 703.574.9893

Contents

Warning and Disclaimer

ThreatQuotient, Inc. provides this document “as is”, without representation or warranty of any kind, express or implied, including without limitation any warranty concerning the accuracy, adequacy, or completeness of such information contained herein. ThreatQuotient, Inc. does not assume responsibility for the use or inability to use the software product as a result of providing this information.

Support

This integration is designated as **ThreatQ Supported**.

Support Email: support@threatq.com
Support Web: https://support.threatq.com
Support Phone: 703.574.9893

Integrations/apps/add-ons designated as **ThreatQ Supported** are fully supported by ThreatQuotient's Customer Support team.

ThreatQuotient strives to ensure all ThreatQ Supported integrations will work with the current version of ThreatQuotient software at the time of initial publishing. This applies for both Hosted instance and Non-Hosted instance customers.

> ThreatQuotient does not provide support or maintenance for integrations, apps, or add-ons published by any party other than ThreatQuotient, including third-party developers.

Integration Details

ThreatQuotient provides the following details for this integration:

Current Integration Version	1.0.0
Compatible with ThreatQ Versions	>= 6.5.0
Support Tier	ThreatQ Supported

Introduction

The GitLab YARA Rules Parser CDF integration ingests YARA rules from a repository/project located on an on-prem GitLab server.

The integration provides the following feed:

- **GitLab YARA Rules** - fetches YARA rules from a user-specified GitLab repository.

The integration ingests the following object types:

- Adversaries
- Attack Patterns
- Indicators
- Malware
- Signatures
- Vulnerabilities

Prerequisites

The following is required by the integration:

- A GitLab Account and Personal Access Token.
- A GitLab project containing YARA rules within sub-directories.

Installation

Perform the following steps to install the integration:

> The same steps can be used to upgrade the integration to a new version.

1. Log into https://marketplace.threatq.com/.
2. Locate and download the integration yaml file.
3. Navigate to the integrations management page on your ThreatQ instance.
4. Click on the **Add New Integration** button.
5. Upload the integration yaml file using one of the following methods:
   - Drag and drop the file into the dialog box
   - Select **Click to Browse** to locate the file on your local machine
6. Select the individual feeds to install, when prompted and click **Install**.

   > ThreatQ will inform you if the feed already exists on the platform and will require user confirmation before proceeding. ThreatQ will also inform you if the new version of the feed contains changes to the user configuration. The new user configurations will overwrite the existing ones for the feed and will require user confirmation before proceeding.

The feed(s) will be added to the integrations page. You will still need to configure and then enable the feed.

Configuration

> ThreatQuotient does not issue API keys for third-party vendors. Contact the specific vendor to obtain API keys and other integration-related credentials.

To configure the integration:

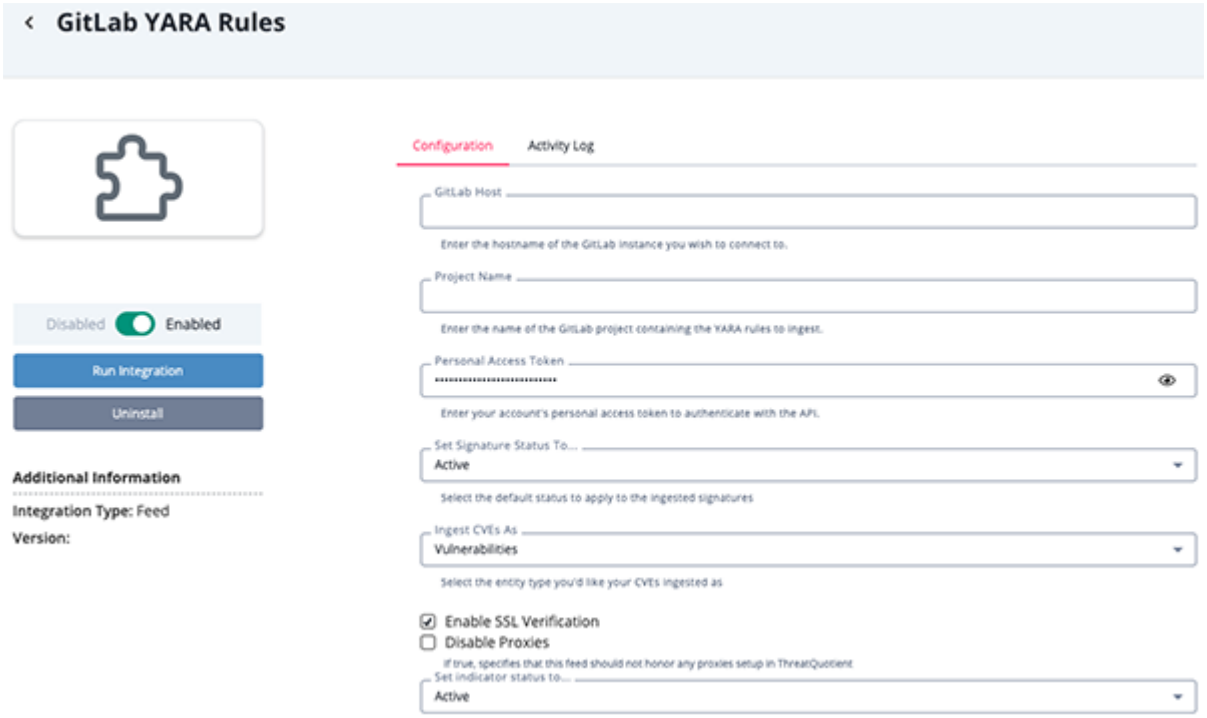
1. Navigate to your integrations management page in ThreatQ.
2. Select the **OSINT** option from the *Category* dropdown (optional).

   > If you are installing the integration for the first time, it will be located under the **Disabled** tab.

3. Click on the integration entry to open its details page.
4. Enter the following parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
GitLab Host	Enter the hostname of your GitLab instance.
Project Name	Enter the name of the GitLab project containing the YARA rules to ingest.
Personal Access Token	Enter your account's personal access token to authenticate with the API.
Set Signature Status to	Select the default status to apply to the ingested signatures. Options include: ◦ Active (default) ◦ Expired ◦ Inactive ◦ Non-Malicious ◦ Review ◦ Whitelisted
Ingest CVEs as	Select the entity type you'd like your CVEs ingested as in the ThreatQ platform. Options include: ◦ Indicators ◦ Vulnerabilities

PARAMETER	DESCRIPTION
Enable SSL Verification	Enable this parameter for the feed to validate the host-provided SSL certificate. This option is enabled by default.
Disable Proxies	Enable this option if the feed should not honor proxies set in the ThreatQ UI.



5. Review any additional settings, make any changes if needed, and click on **Save**.
6. Click on the toggle switch, located above the *Additional Information* section, to enable it.

ThreatQ Mapping

GitLab YARA Rules

The GitLab YARA Rules feed parses and ingests the YARA Signatures from a GitLab project.

`GET https://{{ host }}/api/v4/projects`

Sample Response:


```
[
  {
    "id": 39792303,
    "description": null,
    "name": "YARA Rules",
    "name_with_namespace": "Zach Shames / YARA Rules",
    "path": "yara-rules",
    "path_with_namespace": "zach.shames/yara-rules",
    "created_at": "2022-09-28T14:16:08.174Z",
    "default_branch": "main",
    "tag_list": [],
    "topics": [],
    "ssh_url_to_repo": "git@gitlab.com:zach.shames/yara-rules.git",
    "http_url_to_repo": "https://gitlab.com/zach.shames/yara-rules.git",
    "web_url": "https://gitlab.com/zach.shames/yara-rules",
    "readme_url": "https://gitlab.com/zach.shames/yara-rules/-/blob/main/
README.md",
    "avatar_url": null,
    "forks_count": 0,
    "star_count": 0,
    "last_activity_at": "2022-09-28T14:16:08.174Z",
    "namespace": {
      "id": 58452882,
      "name": "Zach Shames",
      "path": "zach.shames",
      "kind": "user",
      "full_path": "zach.shames",
      "parent_id": null,
      "avatar_url": "https://secure.gravatar.com/avatar/
eed9dc8b344dc6ad47f400d81885cbb9?s=80\u0026d=identicon",
      "web_url": "https://gitlab.com/zach.shames"
    },
    "container_registry_image_prefix": "registry.gitlab.com/zach.shames/yara-
rules",
    "_links": {
      "self": "https://gitlab.com/api/v4/projects/39792303",
      "issues": "https://gitlab.com/api/v4/projects/39792303/issues",
      "merge_requests": "https://gitlab.com/api/v4/projects/39792303/
merge_requests",
```

```
      "repo_branches": "https://gitlab.com/api/v4/projects/39792303/repository/
branches",
      "labels": "https://gitlab.com/api/v4/projects/39792303/labels",
      "events": "https://gitlab.com/api/v4/projects/39792303/events",
      "members": "https://gitlab.com/api/v4/projects/39792303/members",
      "cluster_agents": "https://gitlab.com/api/v4/projects/39792303/
cluster_agents"
    },
    "packages_enabled": true,
    "empty_repo": false,
    "archived": false,
    "visibility": "private",
    "owner": {
      "id": 12657250,
      "username": "zach.shames",
      "name": "Zach Shames",
      "state": "active",
      "avatar_url": "https://secure.gravatar.com/avatar/
eed9dc8b344dc6ad47f400d81885cbb9?s=80\u0026d=identicon",
      "web_url": "https://gitlab.com/zach.shames"
    },
    "resolve_outdated_diff_discussions": false,
    "container_expiration_policy": {
      "cadence": "1d",
      "enabled": false,
      "keep_n": 10,
      "older_than": "90d",
      "name_regex": ".*",
      "name_regex_keep": null,
      "next_run_at": "2022-09-29T14:16:08.193Z"
    },
    "issues_enabled": true,
    "merge_requests_enabled": true,
    "wiki_enabled": true,
    "jobs_enabled": true,
    "snippets_enabled": true,
    "container_registry_enabled": true,
    "service_desk_enabled": true,
    "service_desk_address": "contact-project+zach-shames-yara-rules-39792303-
issue-@incoming.gitlab.com",
    "can_create_merge_request_in": true,
    "issues_access_level": "enabled",
    "repository_access_level": "enabled",
    "merge_requests_access_level": "enabled",
    "forking_access_level": "enabled",
    "wiki_access_level": "enabled",
    "builds_access_level": "enabled",
    "snippets_access_level": "enabled",
    "pages_access_level": "private",
    "operations_access_level": "enabled",
```

```
  "analytics_access_level": "enabled",
  "container_registry_access_level": "enabled",
  "security_and_compliance_access_level": "private",
  "emails_disabled": null,
  "shared_runners_enabled": true,
  "lfs_enabled": true,
  "creator_id": 12657250,
  "import_url": null,
  "import_type": null,
  "import_status": "none",
  "open_issues_count": 0,
  "ci_default_git_depth": 20,
  "ci_forward_deployment_enabled": true,
  "ci_job_token_scope_enabled": false,
  "ci_separated_caches": true,
  "ci_opt_in_jwt": false,
  "ci_allow_fork_pipelines_to_run_in_parent_project": true,
  "public_jobs": true,
  "build_timeout": 3600,
  "auto_cancel_pending_pipelines": "enabled",
  "ci_config_path": "",
  "shared_with_groups": [],
  "only_allow_merge_if_pipeline_succeeds": false,
  "allow_merge_on_skipped_pipeline": null,
  "restrict_user_defined_variables": false,
  "request_access_enabled": true,
  "only_allow_merge_if_all_discussions_are_resolved": false,
  "remove_source_branch_after_merge": true,
  "printing_merge_request_link_enabled": true,
  "merge_method": "merge",
  "squash_option": "default_off",
  "enforce_auth_checks_on_uploads": true,
  "suggestion_commit_message": null,
  "merge_commit_template": null,
  "squash_commit_template": null,
  "auto_devops_enabled": false,
  "auto_devops_deploy_strategy": "continuous",
  "autoclose_referenced_issues": true,
  "keep_latest_artifact": true,
  "runner_token_expiration_interval": null,
  "external_authorization_classification_label": "",
  "requirements_enabled": false,
  "requirements_access_level": "enabled",
  "security_and_compliance_enabled": true,
  "compliance_frameworks": [],
  "permissions": {
    "project_access": {
      "access_level": 50,
      "notification_level": 3
    },
```

```
      "group_access": null
    }
  }
]
```

ThreatQuotient provides the following default mapping for this feed:

> The mapping for this feed is based on the results of the parse-yara CDF filter, after the YARA file content is downloaded using the supplemental feeds.

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES	
`data.metadata[].[key]`	Indicator Value	MD5, SHA-1, SHA-256, SHA-512	N/A	N/A	N/A	Parsed when metadata key references a hash type
`data.metadata[]`	Attack Pattern Value	N/A	N/A	N/A	N/A	N/A
`data.metadata[]`	Adversary Value	N/A	N/A	N/A	N/A	N/A
`data.metadata[]`	Malware Value	N/A	N/A	N/A	N/A	N/A
`data.metadata[]`	Indicator.Value	CVE	N/A	N/A	N/A	User-configurable
`data.metadata[]`	Vulnerability.Value	CVE	N/A	N/A	N/A	User-configurable
`data.metadata[].[key]`	Attribute	[key]	Title-cased	N/A	N/A	Hash metadata is filtered out
`data.tags`	Tag	N/A	N/A	N/A	webshell	N/A
`data.rule_name`	Signature Name	YARA	N/A	N/A	N/A	N/A
`data.raw_signature`	Signature Value	YARA	N/A	N/A	N/A	N/A
`N/A`	Signature Attribute	Category	N/A	N/A	N/A	N/A

GitLab - Get Project Tree Supplemental

The GitLab - Get Project Tree Supplemental feed fetches all the files from a given GitLab project (recursively).

`GET https://{{ host }}/api/v4/projects/{{ project_id }}/repository/tree`

Sample Response:


```
[
  {
    "id": "8763cced4b5455a1155e7c4cb7dd5fa741a8b285",
    "name": "EDR_TDR",
    "type": "tree",
    "path": "EDR_TDR",
    "mode": "040000"
  },
  {
    "id": "235af0dc181b17067da4d4847169c8cc12a12a2d",
    "name": "Malware_YARA",
    "type": "tree",
    "path": "Malware_YARA",
    "mode": "040000"
  },
  {
    "id": "9ffb60db56816592f93c41255b0b0f94d7af4707",
    "name": "Test_Files",
    "type": "tree",
    "path": "Test_Files",
    "mode": "040000"
  },
  {
    "id": "e69de29bb2d1d6434b8b29ae775ad8c2e48c5391",
    "name": ".gitkeep",
    "type": "blob",
    "path": "EDR_TDR/.gitkeep",
    "mode": "100644"
  },
  {
    "id": "df586c79d14e9f2575ab7d3bd8f4972995a1a995",
    "name": "CVE-2010-0805.yar",
    "type": "blob",
    "path": "EDR_TDR/CVE-2010-0805.yar",
    "mode": "100644"
  },
  {
    "id": "07f1e599e693f610d41c2f8a5d9d8a46e4dac924",
    "name": "CVE-2010-0887.yar",
    "type": "blob",
    "path": "EDR_TDR/CVE-2010-0887.yar",
    "mode": "100644"
```

```
  },
  {
    "id": "e69de29bb2d1d6434b8b29ae775ad8c2e48c5391",
    "name": ".gitkeep",
    "type": "blob",
    "path": "Malware_YARA/.gitkeep",
    "mode": "100644"
  },
  {
    "id": "9b3e37e7e236e14661bbaa9ea7f458315bdcd259",
    "name": "APT_APT15.yar",
    "type": "blob",
    "path": "Malware_YARA/APT_APT15.yar",
    "mode": "100644"
  },
  {
    "id": "1f47a1fbfe30f38575f9d1e34b0080aa79c7ba10",
    "name": "capabilities.yara",
    "type": "blob",
    "path": "Malware_YARA/capabilities.yara",
    "mode": "100644"
  },
  {
    "id": "102198284c0480f0cb826e6ef83c76e6e1c5a2c3",
    "name": "README.md",
    "type": "blob",
    "path": "README.md",
    "mode": "100644"
  },
  {
    "id": "e69de29bb2d1d6434b8b29ae775ad8c2e48c5391",
    "name": ".gitkeep",
    "type": "blob",
    "path": "Test_Files/.gitkeep",
    "mode": "100644"
  },
  {
    "id": "ee41e90c12a874888abea72acc10ffc2eef1e0ce",
    "name": "crypto.yar",
    "type": "blob",
    "path": "Test_Files/crypto.yar",
    "mode": "100644"
  }
]
```

GitLab - Get YARA Content Supplemental

The GitLab - Get YARA Content supplemental feed fetches all the files from a given GitLab project (recursively).

```
GET https://{{ host }}/api/v4/projects/{{ project_id }}/repository/blobs/{{ file_id }}/raw
```

Response Body: Raw YARA Rule Content

Average Feed Run

Object counts and Feed runtime are supplied as generalities only - objects returned by a provider can differ based on credential configurations and Feed runtime may vary based on system resources and load.

METRIC	RESULT
Run Time	1 minute
Indicators	16
Signatures	185
Signature Attributes	749

Known Issues / Limitations

- Each time the feed runs, it will only fetch projects that have been updated/modified after the feed's last run date. For the first run, it will look back 24 hours. If you are not seeing any data being brought back in the activity log, perform a manual run of the integration, setting the date back to before the project's last activity date.
- The MITRE filter uses cache memory to load all MITRE ATT&CK data, with the cache being refreshed every 24 hours.

Change Log

- **Version 1.0.0**
  - Initial release