ThreatQuotient

GitLab Action User Guide

Version 1.0.0

October 02, 2023

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	>= 5.12.1
ThreatQ TQO License Required	Yes
Support Tier	ThreatQ Supported

Introduction

The GitLab Action for ThreatQ allows users to automatically create Issues in a GitLab Repository based on incidents in ThreatQ. Context, such as tags and the full HTML description, will be included in the GitLab Issue.

The integration provides the following action:

- **GitLab - Create Issues** - Creates GitLab Issues based on ThreatQ Incidents

The action is compatible with the Incident object type.

This action is intended for use with ThreatQ TDR Orchestrator (TQO). An active TQO license is required for this feature.

Prerequisites

- An active ThreatQ TDR Orchestrator (TQO) license.
- A data collection containing Incident objects.
- A GitLab Personal Access Token with the following permissions:
  - api
  - read_api
  - read_repository
  - write_repository

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 action zip file.
3. Navigate to the integrations management page on your ThreatQ instance.
4. Click on the **Add New Integration** button.
5. Upload the action zip file using one of the following methods:
    - Drag and drop the zip file into the dialog box
    - Select **Click to Browse** to locate the zip file on your local machine

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

You will still need to configure the action.

Configuration

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

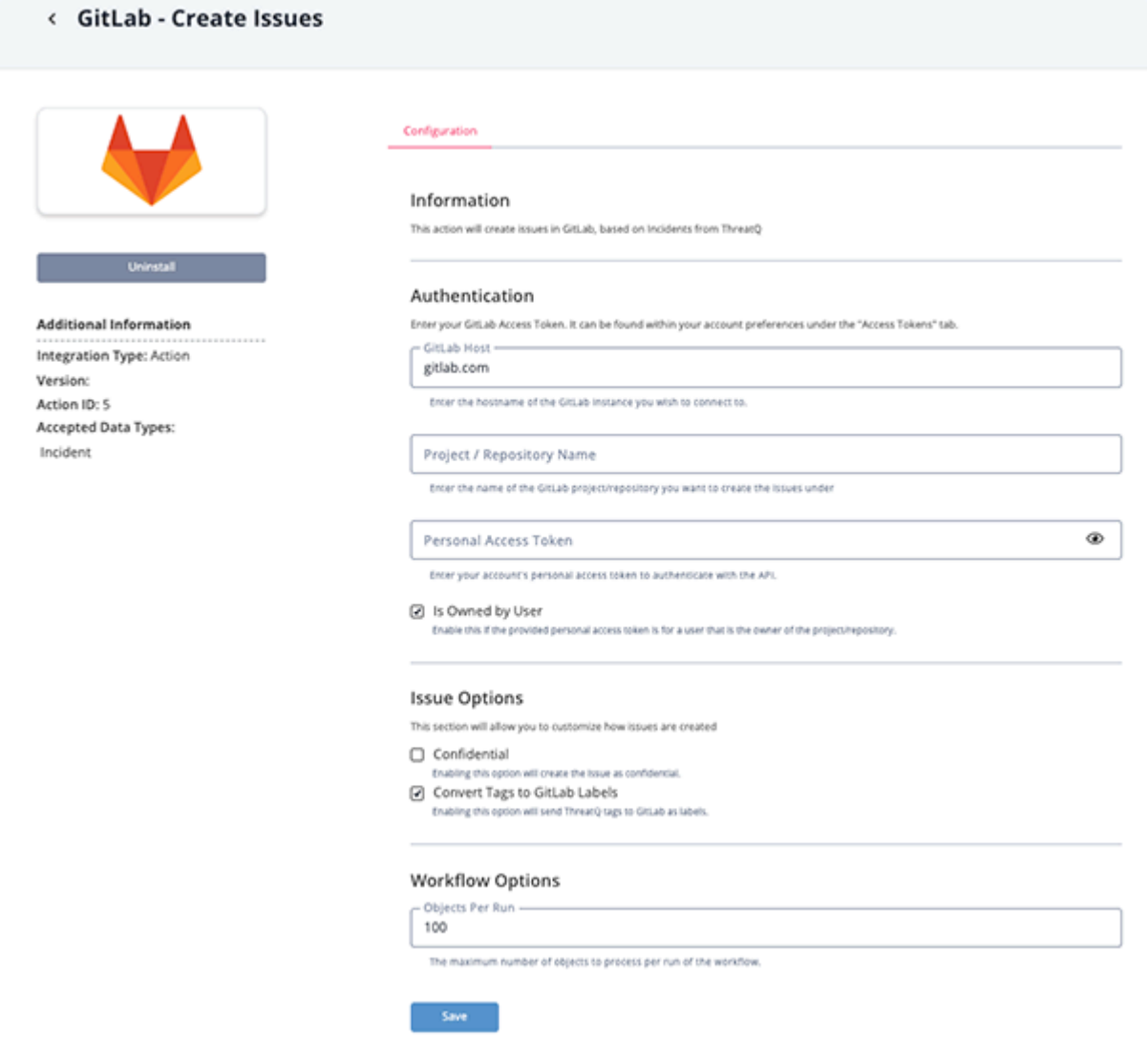
To configure the integration:

1. Navigate to your integrations management page in ThreatQ.
2. Select the **Actions** option from the *Category* dropdown (optional).
3. Click on the action entry to open its details page.
4. Enter the following parameters under the **Configuration** tab:

> The configurations set on this page will be used as the default settings when inserting this action into a new workflow. Updating the configurations on this page will not update any instances of this action that have already been deployed to a workflow. In that scenario, you must update the action's configurations within the workflow itself.

PARAMETER	DESCRIPTION
GitLab Host	Enter the hostname of the GitLab instance you wish to connect to. The default value is gitlab.com.
Project / Repository Name	Enter the name of the GitLab project/repository to create the issues. Issues created with this action will appear under this account.
Personal Access Token	Enter your account's personal access token to authenticate with the API.
Is Owned by User	Enable this if the provided personal access token is for a user that is the owner of the project/repository.
Confidential	Enabling this option will create the issue as confidential. This option is disabled by default.
Convert Tags to GitLab Labels	Enabling this option will send ThreatQ tags to GitLab as labels. This option is enabled by default.

PARAMETER	DESCRIPTION
Objects Per Run	The maximum number of objects to process per run of the workflow. The default value for this is 100.



5. Review any additional settings, make any changes if needed, and click on **Save**.

Actions

The integration provides the following action:

FUNCTION	DESCRIPTION	OBJECT TYPE	OBJECT SUBTYPE
GitLab - Create Issues	Creates GitLab Issues based on ThreatQ Incidents	Incident	N/A

GitLab - Create Issues

The Get Issues action will automatically send Incidents from ThreatQ to a Gitlab project's Issues. The action will create a new Issue for each Incident that is sent to it. The action will also include the Incident's tags as labels on the Issue (if enabled). The full HTML description for each Incident (up to 1 MB) will be included in the Issue.

Fetch/Validate ID

The first request that is made is to fetch the ID for the project, validating that it exists.

```
GET https://{{ host }}/api/v4/projects?search={{ project_name }}
```

Sample Response:


```
[
  {
    "id": 49138255,
    "description": null,
    "name": "Sample Repository",
    "name_with_namespace": "John Doe / Sample Repository",
    "path": "sample-repository",
    "path_with_namespace": "john.doe/sample-repository",
    "created_at": "2023-09-06T20:38:34.100Z",
    "default_branch": "main",
    "tag_list": [],
    "topics": [],
    "ssh_url_to_repo": "git@gitlab.com:john.doe/sample-repository.git",
    "http_url_to_repo": "https://gitlab.com/john.doe/sample-repository.git",
    "web_url": "https://gitlab.com/john.doe/sample-repository",
    "readme_url": "https://gitlab.com/john.doe/sample-repository/-/blob/main/
README.md",
    "forks_count": 0,
    "avatar_url": null,
    "star_count": 0,
    "last_activity_at": "2023-09-07T14:44:56.575Z",
    "namespace": {
      "id": 58452882,
      "name": "John Doe",
      "path": "john.doe",
      "kind": "user",
      "full_path": "john.doe",
      "parent_id": null,
      "avatar_url": "https://secure.gravatar.com/avatar/
eed9dc8b344dc6ad47f400d81885cbb9?s=80&d=identicon",
      "web_url": "https://gitlab.com/john.doe"
    },
    "container_registry_image_prefix": "registry.gitlab.com/john.doe/sample-
repository",
```

```
    "_links": {
      "self": "https://gitlab.com/api/v4/projects/49138255",
      "issues": "https://gitlab.com/api/v4/projects/49138255/issues",
      "merge_requests": "https://gitlab.com/api/v4/projects/49138255/
merge_requests",
      "repo_branches": "https://gitlab.com/api/v4/projects/49138255/repository/
branches",
      "labels": "https://gitlab.com/api/v4/projects/49138255/labels",
      "events": "https://gitlab.com/api/v4/projects/49138255/events",
      "members": "https://gitlab.com/api/v4/projects/49138255/members",
      "cluster_agents": "https://gitlab.com/api/v4/projects/49138255/
cluster_agents"
    },
    "packages_enabled": true,
    "empty_repo": false,
    "archived": false,
    "visibility": "private",
    "owner": {
      "id": 12657250,
      "username": "john.doe",
      "name": "John Doe",
      "state": "active",
      "avatar_url": "https://secure.gravatar.com/avatar/
eed9dc8b344dc6ad47f400d81885cbb9?s=80&d=identicon",
      "web_url": "https://gitlab.com/john.doe"
    },
    "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": "2023-09-07T20:38:34.152Z"
    },
    "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+john-doe-sample-
repository-49138255-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",
"analytics_access_level": "enabled",
"container_registry_access_level": "enabled",
"security_and_compliance_access_level": "private",
"releases_access_level": "enabled",
"environments_access_level": "enabled",
"feature_flags_access_level": "enabled",
"infrastructure_access_level": "enabled",
"monitor_access_level": "enabled",
"emails_disabled": false,
"emails_enabled": true,
"shared_runners_enabled": true,
"lfs_enabled": true,
"creator_id": 12657250,
"import_url": null,
"import_type": null,
"import_status": "none",
"open_issues_count": 2,
"description_html": "",
"updated_at": "2023-09-07T14:44:56.575Z",
"ci_default_git_depth": 20,
"ci_forward_deployment_enabled": true,
"ci_forward_deployment_rollback_allowed": true,
"ci_job_token_scope_enabled": false,
"ci_separated_caches": true,
"ci_allow_fork_pipelines_to_run_in_parent_project": true,
"build_git_strategy": "fetch",
"keep_latest_artifact": true,
"restrict_user_defined_variables": false,
"runners_token": "GR13489411NBhYYYs41RXAAKCVE_z",
"runner_token_expiration_interval": null,
"group_runners_enabled": true,
"auto_cancel_pending_pipelines": "enabled",
"build_timeout": 3600,
"auto_devops_enabled": false,
"auto_devops_deploy_strategy": "continuous",
"ci_config_path": "",
"public_jobs": true,
"shared_with_groups": [],
"only_allow_merge_if_pipeline_succeeds": false,
"allow_merge_on_skipped_pipeline": null,
"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,
    "issue_branch_template": null,
    "autoclose_referenced_issues": true,
    "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
    }
  },
  {
    "id": 39792303,
    "description": null,
    "name": "YARA Rules",
    "name_with_namespace": "John Doe / YARA Rules",
    "path": "yara-rules",
    "path_with_namespace": "john.doe/yara-rules",
    "created_at": "2022-09-28T14:16:08.174Z",
    "default_branch": "main",
    "tag_list": [],
    "topics": [],
    "ssh_url_to_repo": "git@gitlab.com:john.doe/yara-rules.git",
    "http_url_to_repo": "https://gitlab.com/john.doe/yara-rules.git",
    "web_url": "https://gitlab.com/john.doe/yara-rules",
    "readme_url": "https://gitlab.com/john.doe/yara-rules/-/blob/main/
README.md",
    "forks_count": 0,
    "avatar_url": null,
    "star_count": 0,
    "last_activity_at": "2022-10-06T15:01:37.200Z",
    "namespace": {
      "id": 58452882,
      "name": "John Doe",
      "path": "john.doe",
      "kind": "user",
      "full_path": "john.doe",
      "parent_id": null,
      "avatar_url": "https://secure.gravatar.com/avatar/
eed9dc8b344dc6ad47f400d81885cbb9?s=80&d=identicon",
      "web_url": "https://gitlab.com/john.doe"
    },
```

```
    "container_registry_image_prefix": "registry.gitlab.com/john.doe/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": "john.doe",
      "name": "John Doe",
      "state": "active",
      "avatar_url": "https://secure.gravatar.com/avatar/
eed9dc8b344dc6ad47f400d81885cbb9?s=80&d=identicon",
      "web_url": "https://gitlab.com/john.doe"
    },
    "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+john-doe-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",
"analytics_access_level": "enabled",
"container_registry_access_level": "enabled",
"security_and_compliance_access_level": "private",
"releases_access_level": "enabled",
"environments_access_level": "enabled",
"feature_flags_access_level": "enabled",
"infrastructure_access_level": "enabled",
"monitor_access_level": "enabled",
"emails_disabled": false,
"emails_enabled": true,
"shared_runners_enabled": true,
"lfs_enabled": true,
"creator_id": 12657250,
"import_url": null,
"import_type": null,
"import_status": "none",
"open_issues_count": 0,
"description_html": "",
"updated_at": "2022-10-06T15:01:37.200Z",
"ci_default_git_depth": 20,
"ci_forward_deployment_enabled": true,
"ci_forward_deployment_rollback_allowed": true,
"ci_job_token_scope_enabled": false,
"ci_separated_caches": true,
"ci_allow_fork_pipelines_to_run_in_parent_project": true,
"build_git_strategy": "fetch",
"keep_latest_artifact": true,
"restrict_user_defined_variables": false,
"runners_token": "GR1348941jHJWuJ-eZ6CWC241MZD7",
"runner_token_expiration_interval": null,
"group_runners_enabled": true,
"auto_cancel_pending_pipelines": "enabled",
"build_timeout": 3600,
"auto_devops_enabled": false,
"auto_devops_deploy_strategy": "continuous",
"ci_config_path": "",
"public_jobs": true,
"shared_with_groups": [],
"only_allow_merge_if_pipeline_succeeds": false,
"allow_merge_on_skipped_pipeline": null,
"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,
    "issue_branch_template": null,
    "autoclose_referenced_issues": true,
    "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
    }
  }
]
```


Create Incident

The following request is made to create each Incident as an Issue:

```
POST https://{{ host }}/api/v4/projects/{{ project_id }}/issues
```

Sample Response:


```
{
  "id": 134850195,
  "iid": 3,
  "project_id": 49138255,
  "title": "Test Issue",
  "description": "<p>Test Description</p>",
  "state": "opened",
  "created_at": "2023-09-19T14:19:56.338Z",
  "updated_at": "2023-09-19T14:19:56.338Z",
  "closed_at": null,
  "closed_by": null,
  "labels": ["demo", "test"],
  "milestone": null,
  "assignees": [],
  "author": {
    "id": 12657250,
    "username": "john.doe",
    "name": "John Doe",
    "state": "active",
    "avatar_url": "https://secure.gravatar.com/avatar/
```

```
eed9dc8b344dc6ad47f400d81885cbb9?s=80&d=identicon",
    "web_url": "https://gitlab.com/john.doe"
  },
  "type": "ISSUE",
  "assignee": null,
  "user_notes_count": 0,
  "merge_requests_count": 0,
  "upvotes": 0,
  "downvotes": 0,
  "due_date": null,
  "confidential": false,
  "discussion_locked": null,
  "issue_type": "issue",
  "web_url": "https://gitlab.com/john.doe/sample-repository/-/issues/3",
  "time_stats": {
    "time_estimate": 0,
    "total_time_spent": 0,
    "human_time_estimate": null,
    "human_total_time_spent": null
  },
  "task_completion_status": {
    "count": 0,
    "completed_count": 0
  },
  "blocking_issues_count": 0,
  "has_tasks": true,
  "task_status": "0 of 0 checklist items completed",
  "_links": {
    "self": "https://gitlab.com/api/v4/projects/49138255/issues/3",
    "notes": "https://gitlab.com/api/v4/projects/49138255/issues/3/notes",
    "award_emoji": "https://gitlab.com/api/v4/projects/49138255/issues/3/
award_emoji",
    "project": "https://gitlab.com/api/v4/projects/49138255",
    "closed_as_duplicate_of": null
  },
  "references": {
    "short": "#3",
    "relative": "#3",
    "full": "john.doe/sample-repository#3"
  },
  "severity": "UNKNOWN",
  "subscribed": true,
  "moved_to_id": null,
  "service_desk_reply_to": null
}
```

Use Case Example

Our organization tracks incidents in GitLab using the Issue Tracker. I want to be able to have Issues created in GitLab based on Incidents that analysts create and curate in ThreatQ regarding possible threats to our organization.

Example:

- Analysts create incidents in ThreatQ with rich context
- The action takes these incidents and creates Issues in GitLab for them
- Incident response team using GitLab can triage and respond to the Issues

Change Log

- **Version 1.0.0**
  - Initial release