

ASM GITHUB INTEGRATION

Mandiant Advantage Attack Surface Management (MA-ASM) provides two GitHub workflows which result in different types of Entities and Issues. Our token-based GitHub integration works with both GitHub organizations and accounts. Depending on which method you choose, the MA-ASM workflow varies.

- **Integrate with a GitHub Organization:** MA-ASM locates all of the users belonging to the integrated organization and creates a `GithubAccount` Entity for each of them.
- **Integrate with a GitHub Account:** MA-ASM creates a `GithubAccount` Entity of the user the token belongs to and `GithubRepository` Entities for all the repositories the token has access to (including private repositories owned by other users).

This integration eliminates the need for manual entry of the same data as Seeds.

Integration with GitHub requires two steps:

1. [Create a GitHub Access Token](#)
2. [Use GitHub Access Token for MA-ASM Integration](#)

Create a GitHub Access Token

There are two types of access tokens you can create for the GitHub integration: Fine-grained or Classic. To create either type of token, sign in to GitHub and navigate to <https://github.com/settings/tokens>. Then follow the steps for the type of token you want to generate:

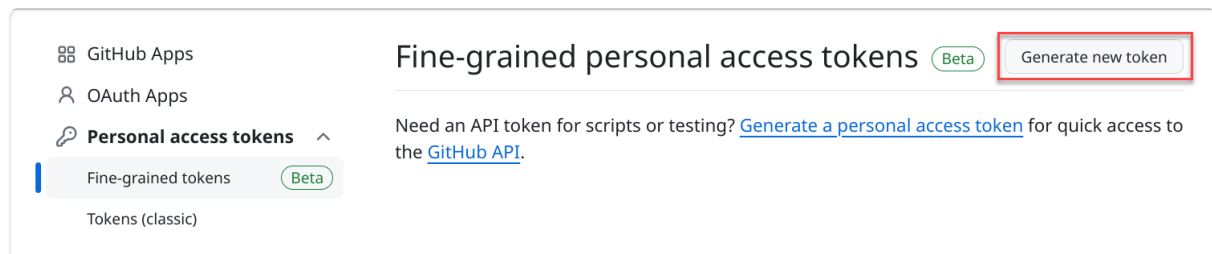
- **Fine-grained token (Beta):** Recommended by GitHub
- **Classic token:** Less secure option



- See [Managing your personal access tokens \(https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens\)](https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens) to determine which token best suits your needs.
- See <https://github.com/settings/tokens> for more information on how to create a token to access the **GitHub API** (<https://docs.github.com/en/rest>).

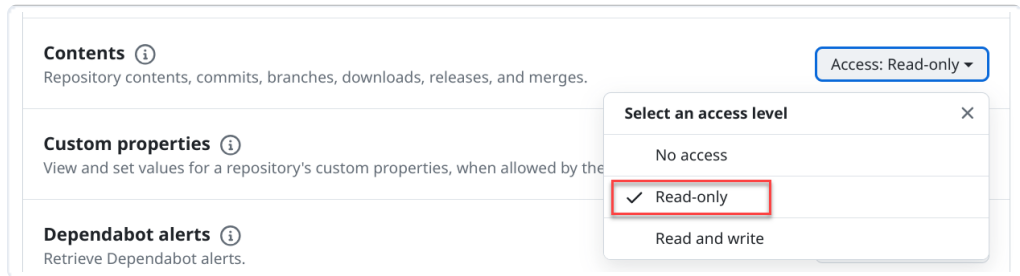
Create a fine-grained token (Beta)

1. Navigate to **Personal Access tokens > Fine-grained tokens**.
2. Click **Generate new token**.



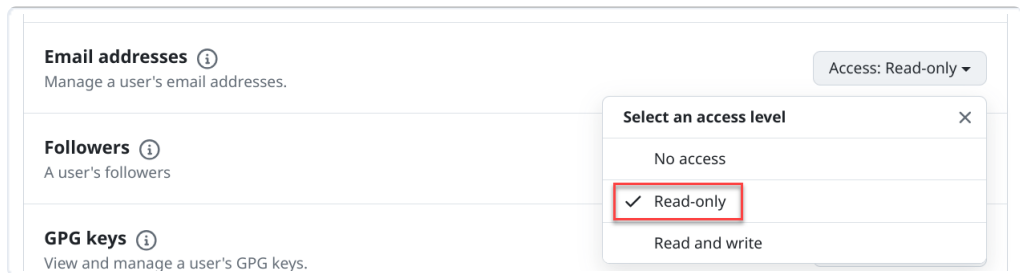
3. Fill in the fields as follows:
 - **Token name:** Enter a name such as *Mandiant-ASM* that's easy to remember.
 - **Expiration:** Set this based on your organizations's requirements and swap out the token when it expires.
 - **Description**
 - **Resource owner**

- **Repository access:** Select the choice which best fits your organization's requirements.
 - **Public Repositories (read-only)**
 - **All repositories:** This includes all current and future repositories you own.
 - **Only select repositories:** You must select at least one repository and no more than 50.
- **Permissions**
 - **Repository Permissions:** This subsection only appears if **All repositories** or **Only select repositories** is selected in the **Repository access** section.
 - Navigate to the **Contents** option and update this **Access** to **Read-only**.



The screenshot shows the 'Contents' settings for a repository. The 'Access' dropdown is set to 'Read-only'. A modal window titled 'Select an access level' is open, showing three options: 'No access', 'Read-only' (which is selected and highlighted with a red box), and 'Read and write'.

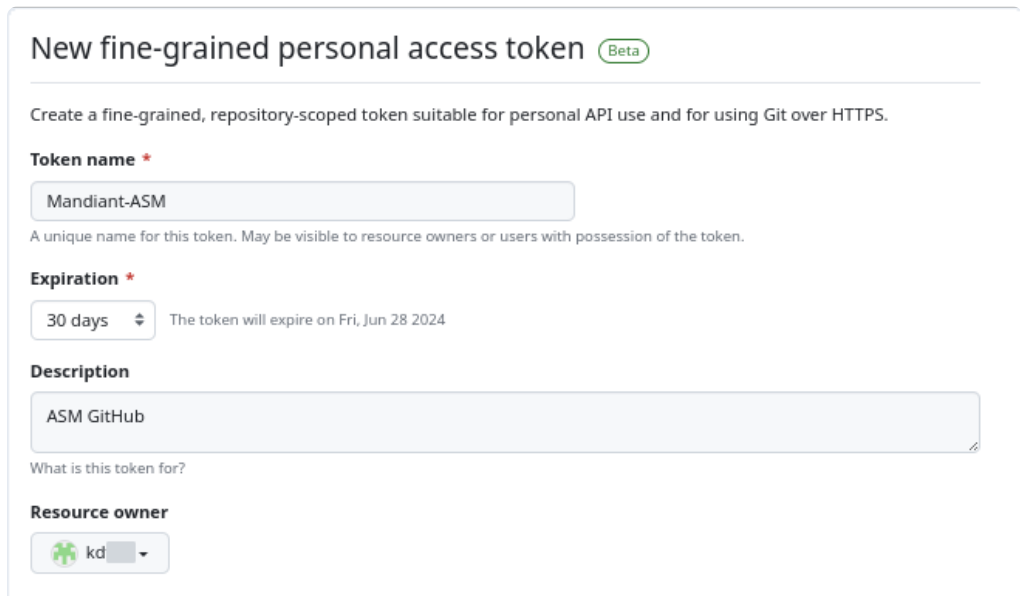
- **Account Permissions:**
 - Optional: Navigate to the **Email addresses** option and update this **Access** to **Read-only**.



The screenshot shows the 'Email addresses' settings for a user's account. The 'Access' dropdown is set to 'Read-only'. A modal window titled 'Select an access level' is open, showing three options: 'No access', 'Read-only' (which is selected and highlighted with a red box), and 'Read and write'.



Setting the Email addresses option to Read-only is recommended as this allows MA-ASM to display the email with which the access token is associated. This makes it easier to differentiate between many GitHub integrations.



The screenshot shows the 'New fine-grained personal access token' page. The token name is 'Mandiant-ASM', the expiration is set to '30 days' (expiring on Fri, Jun 28 2024), and the description is 'ASM GitHub'. The resource owner is 'kd'.

Repository access

Public Repositories (read-only)

All repositories
This applies to all current *and* future repositories you own.
Also includes public repositories (read-only).

Only select repositories
Select at least one repository. Max 50 repositories.
Also includes public repositories (read-only).

Permissions

Read our [permissions documentation](#) for information about specific permissions.

Repository permissions 2 Selected >

Repository permissions permit access to repositories and related resources.

Account permissions 1 Selected >

User permissions permit access to resources under your personal GitHub account.

Overview

2 permissions for all of your repositories >


1 Account permission >

This token will expire **June 28, 2024**.

Generate token [Cancel](#)

This token will be ready for use immediately.

4. Click **Generate token**.
5. Finally, copy the token for use in MA-ASM when establishing the integration.

 This token is only available until you navigate away from this page.

Fine-grained personal access tokens Beta Generate new token

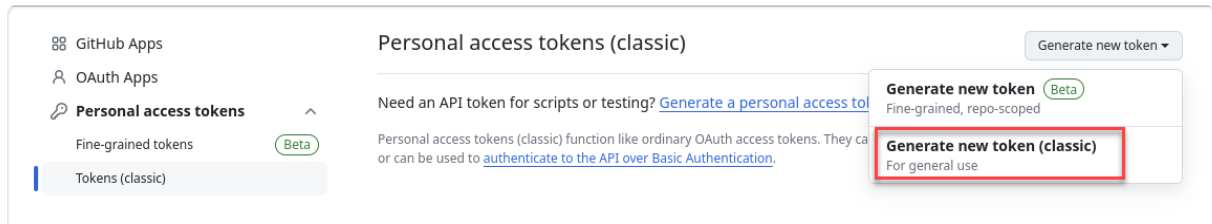
These are fine-grained, repository-scoped tokens suitable for personal [API](#) use and for using Git over HTTPS.

Make sure to copy your personal access token now as you will not be able to see this again. Never used Delete

github_pat...
Expires on Sun, Jun 16 2024.

Create a classic token

1. Navigate to **Personal Access tokens > Tokens (classic)**.
2. Click **Generate new token > Generate new token (classic)**.



The screenshot shows the GitHub interface for creating a classic personal access token. On the left, a sidebar lists 'Personal access tokens' with sub-options for 'Fine-grained tokens' (Beta) and 'Tokens (classic)'. The main content area is titled 'Personal access tokens (classic)' and includes a 'Generate new token' dropdown menu. A tooltip is visible over the dropdown, showing two options: 'Generate new token (Beta)' for fine-grained, repo-scoped tokens, and 'Generate new token (classic)' for general use, which is highlighted with a red box.

3. Fill out the mandatory fields as follows:
 - **Note:** Enter a name such as *Mandiant-ASM* that's easy to remember.
 - **Expiration:** Set this based on your organizations's requirements and swap out the token when it expires.
 - **Select scopes:** Select the following options:
 - **repo:** Full control of private repositories
 - Optional: **user > user:email:** Access user email addresses (read-only)



If enabled, the **user:email** scope allows MA-ASM to display the email with which the access token is associated. This makes it easier to differentiate between many GitHub integrations.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Mandiant-ASM

What's this token for?

Expiration


30 days The token will expire on Sat, Jun 29 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo:deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input checked="" type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:org	Read organization metadata
<input type="checkbox"/> read:public_key	Read public user keys
<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

- Finally, copy the token for use in MA-ASM when establishing the integration.

 This token is only available until you navigate away from this page.

Personal access tokens (classic)

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_kD2[REDACTED]hYF

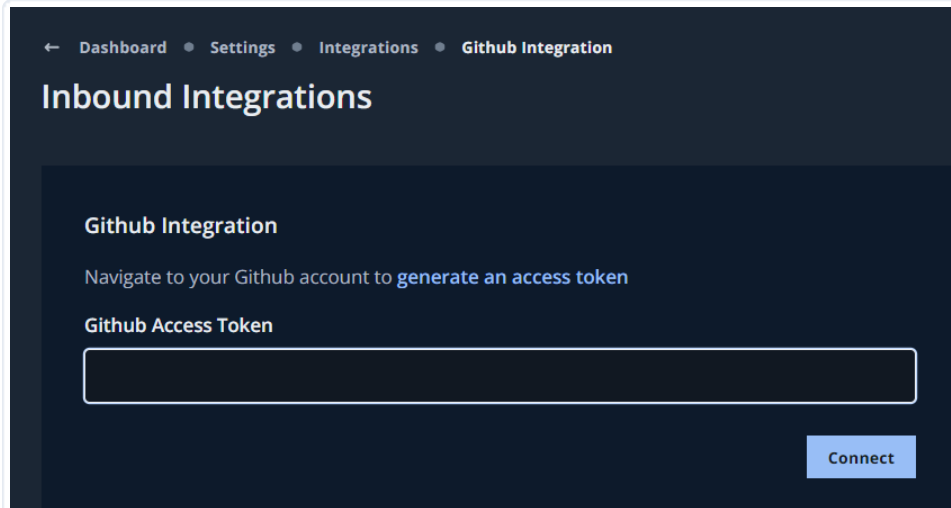
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).


Use GitHub Access Token for MA-ASM Integration

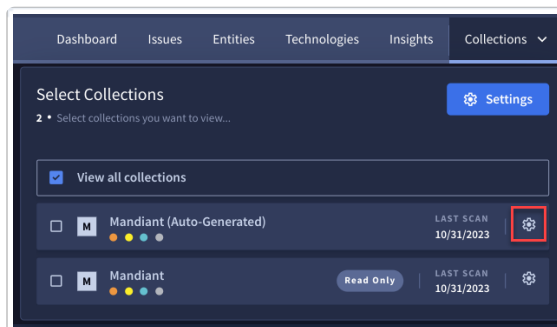
- From the **Projects and Settings** menu in MA-ASM, select the appropriate Project then click **Account Settings**.
- Click **Integrations**.
- From **Inbound Integrations**, click **Add New** for **GitHub**.

Github
Integration provides greater visibility into Github entities, and it is no longer necessary to manually enter the same data as a seed. + Add New

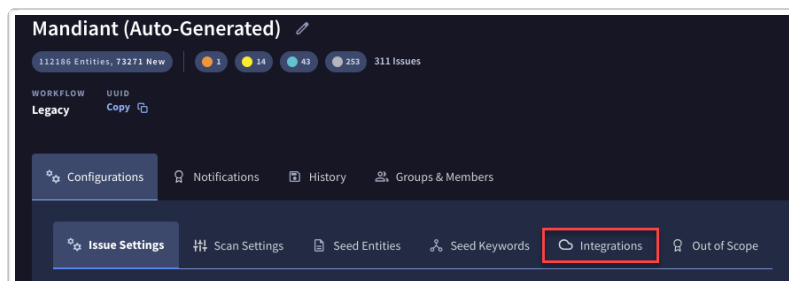
4. Paste the **Github Access Token** value into the **Github Access Token** field and click **Connect**.



5. Connect the integration to the appropriate Collection.
- Click **Collections** and click  **Collection Settings** for the Collection that you want to connect the integration to.



- Select the **Integrations** tab.




- Select **Connect Integration** and **Link** the integration.

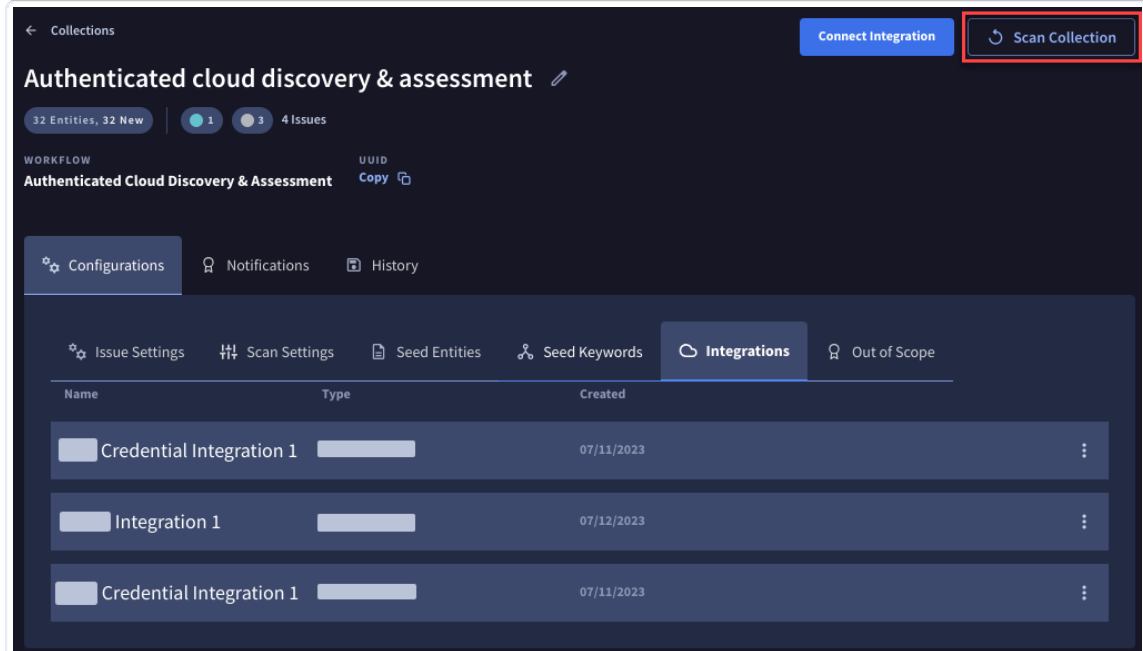


The integration is immediately added to the Collection.

Click  to remove the integration from this Collection.



- d. Click  to close the **Connect Integration** pane. Click **Scan Collection** to update your Collection with the current settings and integrations. Otherwise, your newly configured integration is incorporated at your regularly scheduled scan interval.



Keyword Seeds and the GitHub Integration

After integrating with GitHub, when a Seed is added (see [Creating & Seeding Collections](#)

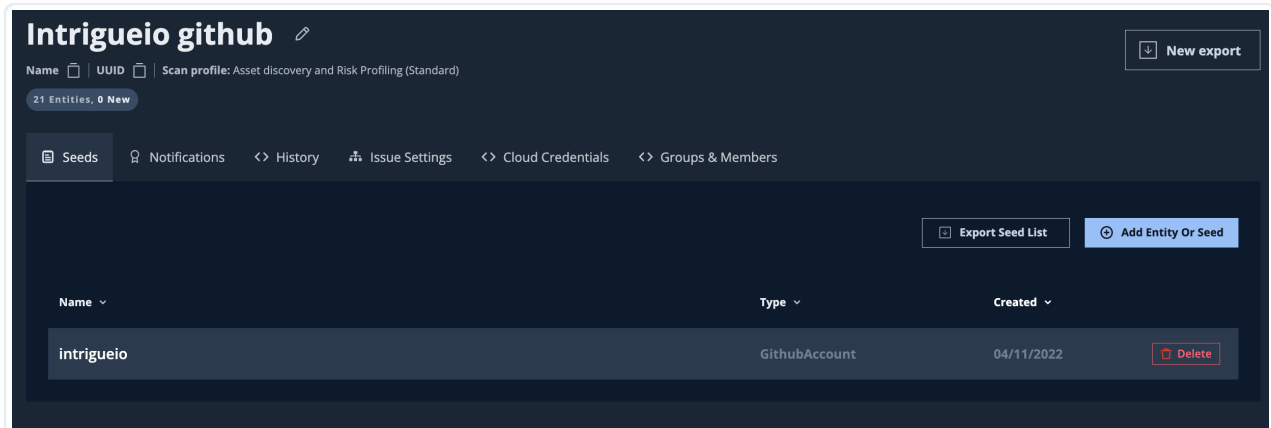
(<https://docs.mandiant.com/home/creating-seeding-collections>) and used, MA-ASM works in the following ways:

- **GitHub Organization:** MA-ASM locates the public users and creates `GithubAccount` Entities for them.
- **GitHub Account:** For each `GithubAccount` Entity, available repositories are created as `GithubRepository` Entities.
- **GitHub Repository:** MA-ASM uses [Gitleaks](https://github.com/zricethezav/gitleaks) (<https://github.com/zricethezav/gitleaks>) with a modified configuration file to detect items such as: common API keys, tokens, private keys, suspicious file names, or file extensions.

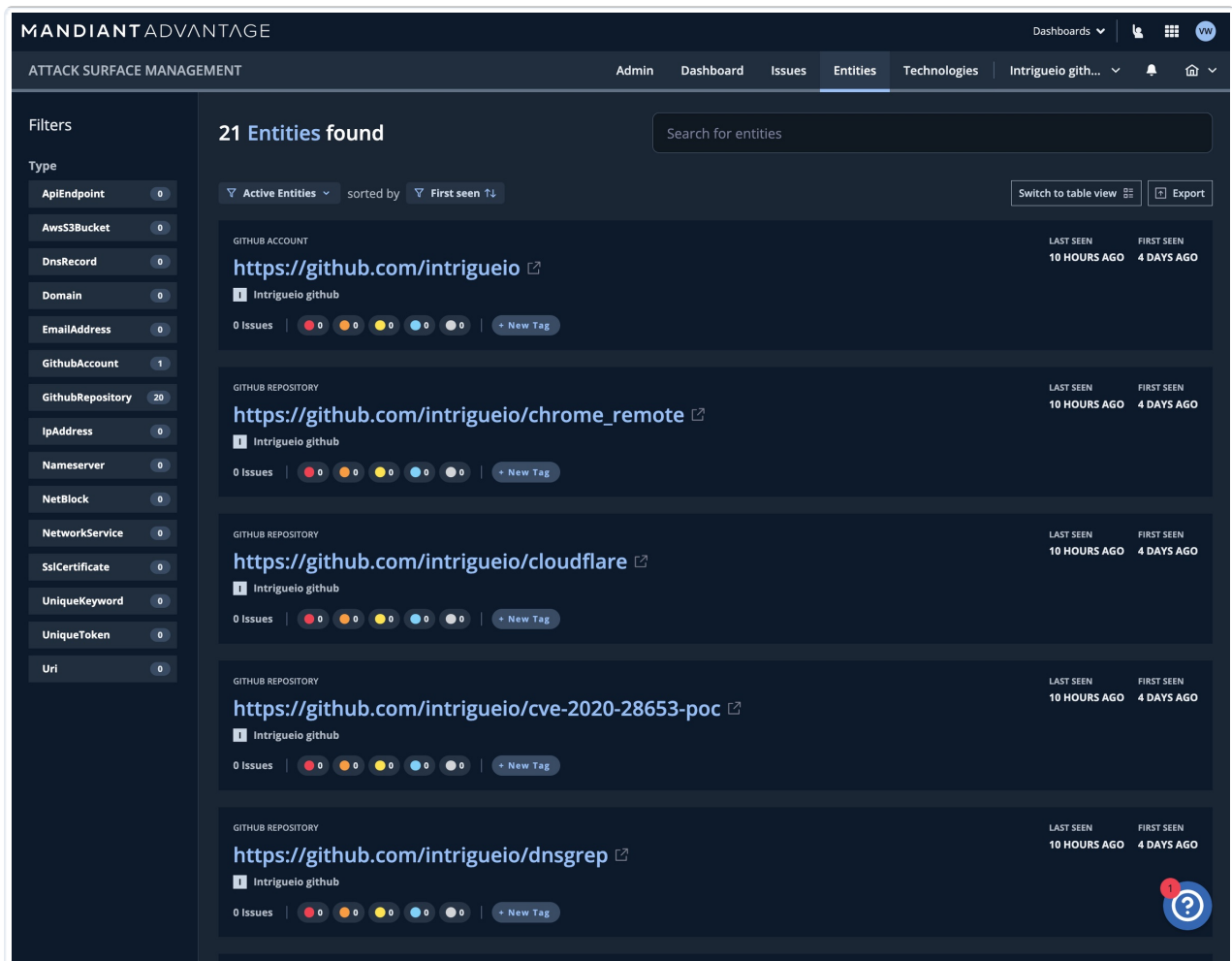
An Example Use Case

This example illustrates a sample MA-ASM GitHub tracking workflow.

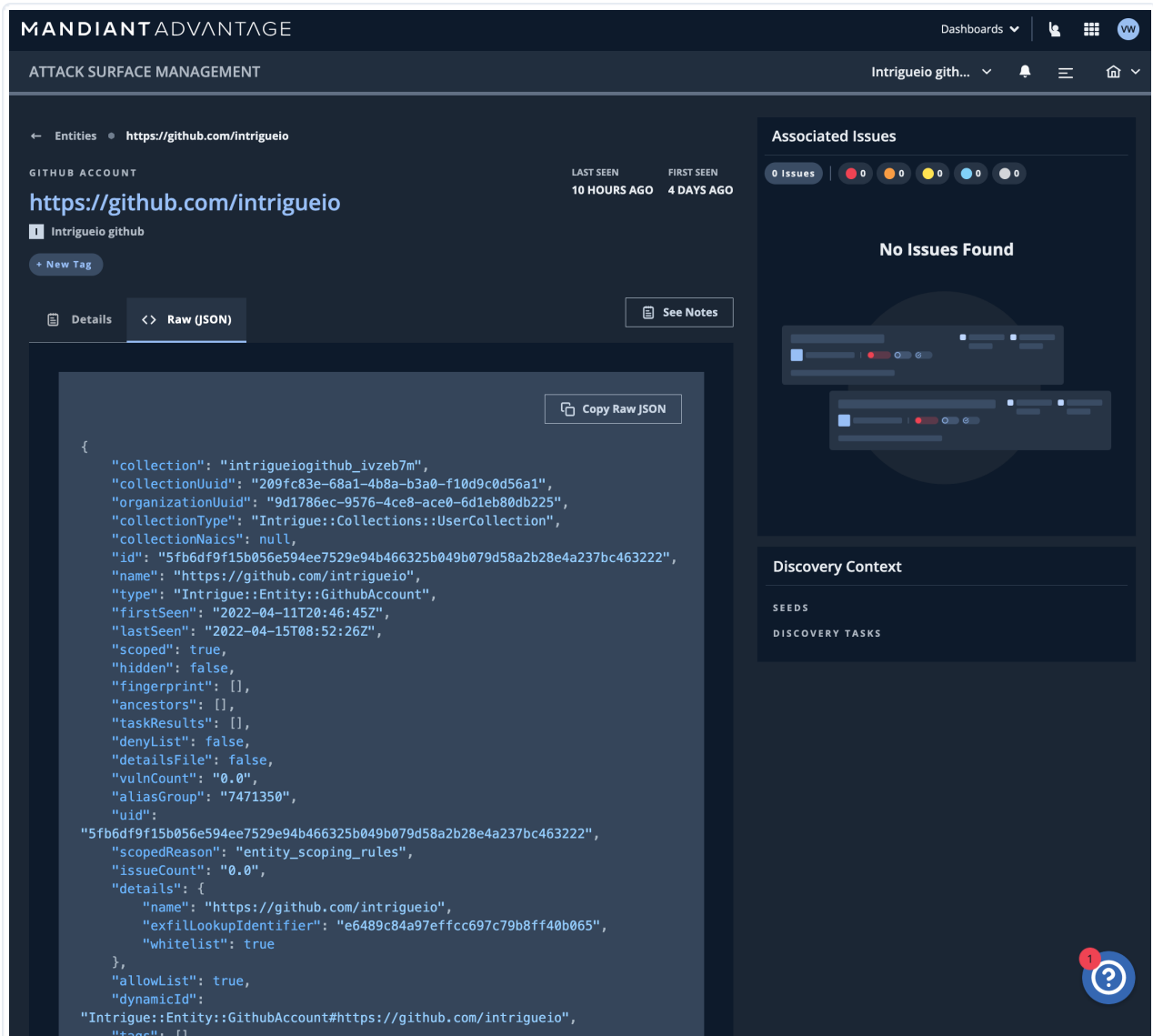
In this example, the user adds a Seed, a `GithubAccount` : "intrigueio."



Using that `GithubAccount`, MA-ASM finds 21 related Entities which include related repositories:



Selecting the first `GithubAccount` Entity from this list displays the following page of detailed information:



The screenshot displays the Mandiant Advantage interface for an entity named "https://github.com/intrigueio". The interface shows the following details:

- Entity Name:** https://github.com/intrigueio
- Last Seen:** 10 HOURS AGO
- First Seen:** 4 DAYS AGO
- Associated Issues:** 0 Issues (indicated by a row of colored circles: 0 red, 0 orange, 0 yellow, 0 blue, 0 grey).
- Discovery Context:** SEEDS, DISCOVERY TASKS
- Raw (JSON) View:** A large code block containing the following JSON data:

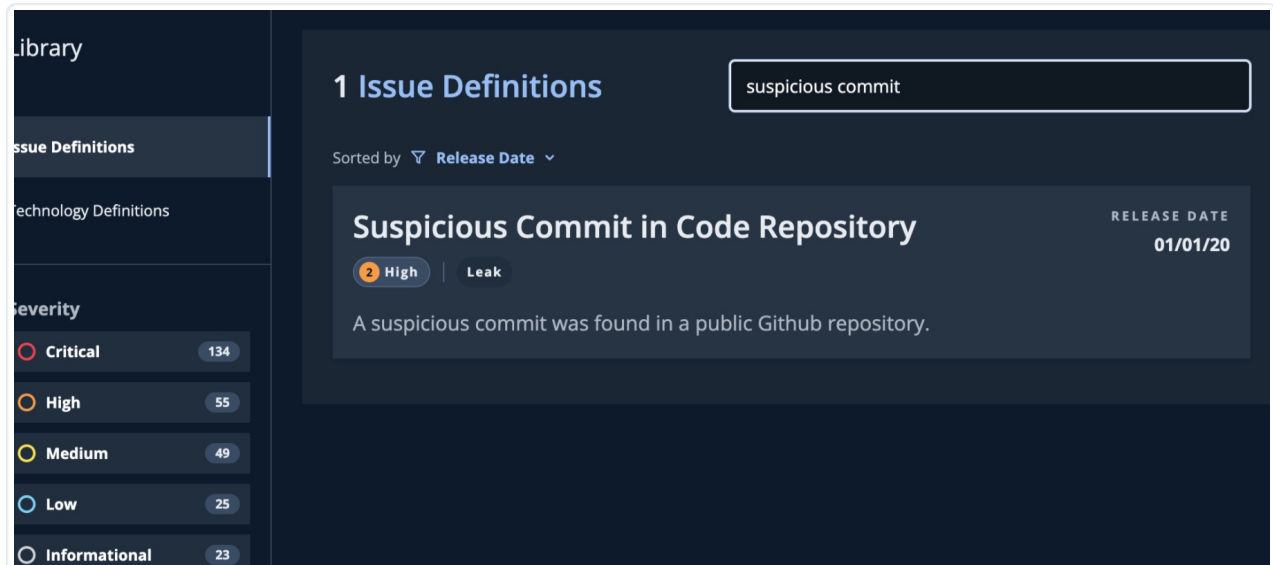

```
{
  "collection": "intrigueiogithub_ivzeb7m",
  "collectionUuid": "209fc83e-68a1-4b8a-b3a0-f10d9c0d56a1",
  "organizationUuid": "9d1786ec-9576-4ce8-ace0-6d1eb80db225",
  "collectionType": "Intrigue::Collections::UserCollection",
  "collectionNaics": null,
  "id": "5fb6df9f15b056e594ee7529e94b466325b049b079d58a2b28e4a237bc463222",
  "name": "https://github.com/intrigueio",
  "type": "Intrigue::Entity::GithubAccount",
  "firstSeen": "2022-04-11T20:46:45Z",
  "lastSeen": "2022-04-15T08:52:26Z",
  "scoped": true,
  "hidden": false,
  "fingerprint": [],
  "ancestors": [],
  "taskResults": [],
  "denyList": false,
  "detailsFile": false,
  "vulnCount": "0.0",
  "aliasGroup": "7471350",
  "uid": "5fb6df9f15b056e594ee7529e94b466325b049b079d58a2b28e4a237bc463222",
  "scopedReason": "entity_scoping_rules",
  "issueCount": "0.0",
  "details": {
    "name": "https://github.com/intrigueio",
    "exfilLookupIdentifier": "e6489c84a97efcc697c79b8ff40b065",
    "whitelist": true
  },
  "allowList": true,
  "dynamicId": "Intrigue::Entity::GithubAccount#https://github.com/intrigueio",
  "tags": []
}
```

When a keyword is used as a Seed, MA-ASM turns on the GitHub tracking workflow for this collection. Keywords can contain phrases and are not required to be one word. This keyword finds related GitHub Repositories and if any have leaked secrets, an Issue type of **"Suspicious commit"** is created.

Two limitations associated with this type of workflow include:

1. Choosing poor keywords or phrases can create an overload of irrelevant results.
2. The API rate limit can result in incomplete search results.

If any Issues are found relating to that keyword, they show up as a **Suspicious Commit in Code Repository** Issue:



library

Issue Definitions

Technology Definitions

Severity

- Critical 134
- High 55
- Medium 49
- Low 25
- Informational 23

1 Issue Definitions

suspicious commit

Sorted by Release Date

Suspicious Commit in Code Repository RELEASE DATE
01/01/20

2 High | Leak

A suspicious commit was found in a public Github repository.

In some instances, you could find 1000 `GithubRepository` Entities from a keyword, and they could all have no leaked secrets. In this case, no Issues would be created but you would still have the Entities created.