Identifying audit log events performed by an access token

In this article

You can identify the actions performed by a specific token in your enterprise.

About token data in the audit log for an enterprise

Your enterprise's audit log contains an event for each action that a user or integration performs. If the action occurred outside of GitHub's web UI, the event's data will show details about how the user or integration authenticated.

If you learn that a token was compromised, you can understand the actions taken by the compromised token by searching the audit log for all events associated with that token.

Token data appears in the audit log for the following authentication methods.

- Personal access token
- OAuth token
- GitHub Apps (authentication as an app installation or on behalf of a user)

Token data in audit log events

The following data about token use appears in the audit log to help you understand how the user or integration authenticated.

Information	Description
`hashed_token`	SHA-256 hash of the token used for authentication.
`programmatic_access_type`	Type of authentication used.
`token_scopes`	If applicable, the scopes for the token.

Identifying events associated with a token

To identify events associated with a specific token, you can use the UI or REST API. To identify any events, you will need to know the SHA-256 hash of the token first.

Generating a SHA-256 hash value for a token

If you only have a raw token value, you'll need to generate a SHA-256 hash before you can search for the token.

For MacOS and Linux, you can use `echo -n TOKEN | openssl dgst -sha256 -binary | base64`, replacing TOKEN with the token value.

For Powershell, you can use the following script to return a SHA-256 hash for a given string.

Shell

```
Param (
    [Parameter(Mandatory=$true)]
    [string]
    $ClearString
)

$hasher = [System.Security.Cryptography.HashAlgorithm]::Create('sha256')
$hash = $hasher.ComputeHash([System.Text.Encoding]::UTF8.GetBytes($ClearString))

$hashString = [System.BitConverter]::ToString($hash)
$hashString.Replace('-', '')
```

Searching on GitHub

While searching the audit log on GitHub, include `hashed_token:"VALUE"` in your search query, replacing `VALUE` with the SHA-256 hash of the token.

Note: Make sure to wrap the hashed token value in quotation marks.

Searching with the REST API

Before you can search for a token using the REST API, after you generate a SHA-256 hash, you also need to URI-escape the hash. Most major programming languages provide a utility for URI escaping. For example, encodeURIComponent() encodes a string for JavaScript.

Then, include `hashed_token:"VALUE"` in your search phrase, replacing VALUE with the URI-escaped hash.

For example, if the name of the enterprise account is `octo-corp`, the following curl command would search @octo-corp's audit log for all events that are associated with the token whose URI-encoded SHA-256 hash is `EH4L8o6PfCqipALbL%2BQT62lyqUtnI7ql0SPbkaQnjv8`.

```
curl --header "Accept: application/vnd.github+json" --header "Authorization:
Bearer YOUR-TOKEN" --header "X-GitHub-Api-Version:2022-11-28"
'https://api.github.com/enterprises/octo-corp/audit-log?
phrase=hashed_token:"EH4L8o6PfCqipALbL%2BQT62lyqUtnI7ql0SPbkaQnjv8"'
```

Further reading

- "Using the audit log API for your enterprise"