ARGUS: A Framework for Staged Static Taint Analysis of GitHub Workflows and Actions

Siddharth Muralee, Igibek Koishybayev, Aleksandr Nahapetyan, Greg Tystahl, Brad Reaves, Antonio Bianchi, William Enck, Alexandros Kapravelos, Aravind Machiry

Purdue University North Carolina State University

Github Actions

- CI/CD (Continuous Integration/Continuous Deployment) platform developed by GitHub in 2018
- Features
  - Developers define **Workflows** which automate various steps of the software development lifecycle, such as building, testing, and deploying code.
  - Workflows can use **Actions** which are applications that perform commonly repeated tasks. These are developed by the community and can be found on the market place
  - Workflows are **Event Driven** and can be triggered by specific GitHub events such as a pushing a commit, creating a pull request, or opening an issue.

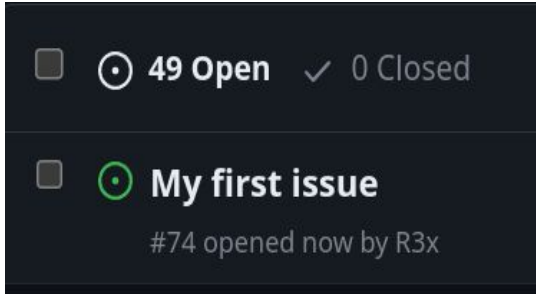
Our Goal : Identify Command Injection vulnerabilities in Github Actions

Motivating Example

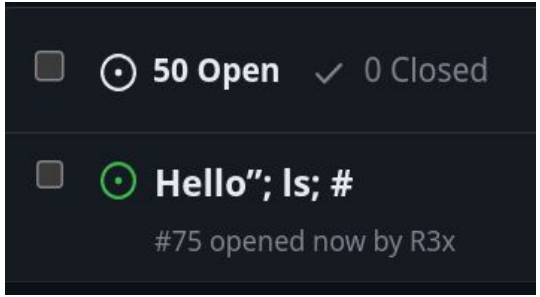
```yaml
name: Sample Workflow
on:
  issues:
    types: [opened]

jobs:
  notify:
    - name: Log title
      run: echo "${{ github.event.issue.title }}"
    ...
```

```
echo “${{ github.event.issue.title}}”
```



```
echo “My first issue”
```



```
echo “Hello”; ls; #”
```

Triggers
Determines the events on which the workflow is run

Jobs
Independent tasks that usually run in different Virtual Environments

Steps
Jobs have a number of steps that get executed sequentially

workflow.yml

```yaml
name: My custom workflow
on:
  pull_request:
    types: [opened]
jobs:
  build:
    outputs:
      build: ${{ steps.build.outputs.build }}
    steps:
    - uses: actions/checkout@v3
      with:
        ref: main
    - id: build
      run: |
       sudo ./build.sh
       if [ $? -eq 0 ]; then
         echo "build=true" >> $GITHUB_OUTPUT
       fi
  test:
    needs: build
    steps:
    - if: needs.build.outputs.build == 'true'
      run: |
        echo "Successfully built - ${{ github.event.pull_request.title }}"
```

Triggers
Determines the events on which the workflow is run

Jobs
Independent tasks that usually run in different Virtual Environments

Steps
Jobs have a number of steps that get executed sequentially

workflow.yml

```yaml
name: My custom workflow
on:
  pull_request:
    types: [opened]
jobs:
  build:
    outputs:
      build: ${{ steps.build.outputs.build }}
    steps:
    - uses: actions/checkout@v3
      with:
        ref: main
    - id: build
      run: |
       sudo ./build.sh
       if [ $? -eq 0 ]; then
         echo "build=true" >> $GITHUB_OUTPUT
       fi
  test:
    needs: build
    steps:
    - if: needs.build.outputs.build == 'true'
      run: |
        echo "Successfully built - ${{ github.event.pull_request.title }}"
```

Outputs/Envs
Jobs and Steps can share data using Outputs and Env Variables.

Dependencies
A Job executes only after all it's dependent Jobs have completed

Event Data
Workflows can access event data using special variables

Permissions

Controls the privileges of the GITHUB TOKEN used by the workflow to interact with Github API

static.yml

```yaml
# Simple workflow for deploying static content to GitHub Pages
name: Deploy static content to Pages
on:
  # Runs on pushes targeting the default branch
  push:
    branches: ["master"]
  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:
# Sets permissions of the GITHUB_TOKEN to allow
# deployment to GitHub Pages
permissions:
  contents: read
  pages: write
  id-token: write
```

Secrets

Encrypted secrets allow you to store sensitive information which can later be used in Github Workflows

build.yml

```yaml
# Simple workflow for building the environment
name: Deploy static content to Pages
on:
  # Runs on pushes targeting the default branch
  push:
    branches: ["master"]
jobs:
  build:
    steps:
      - name: Checkout code
        uses: actions/checkout@v2
        with:
          repository: my-org/my-repo
          ssh-key: ${{ secrets.SSH_PRIVATE_KEY }}
```

Threat Model

- Execute **Arbitrary Commands** without visible code changes
- Gain **Unauthorized Read/Write** access to repository
- Exfiltrate **Confidential Secrets** present in the pipeline

Challenges

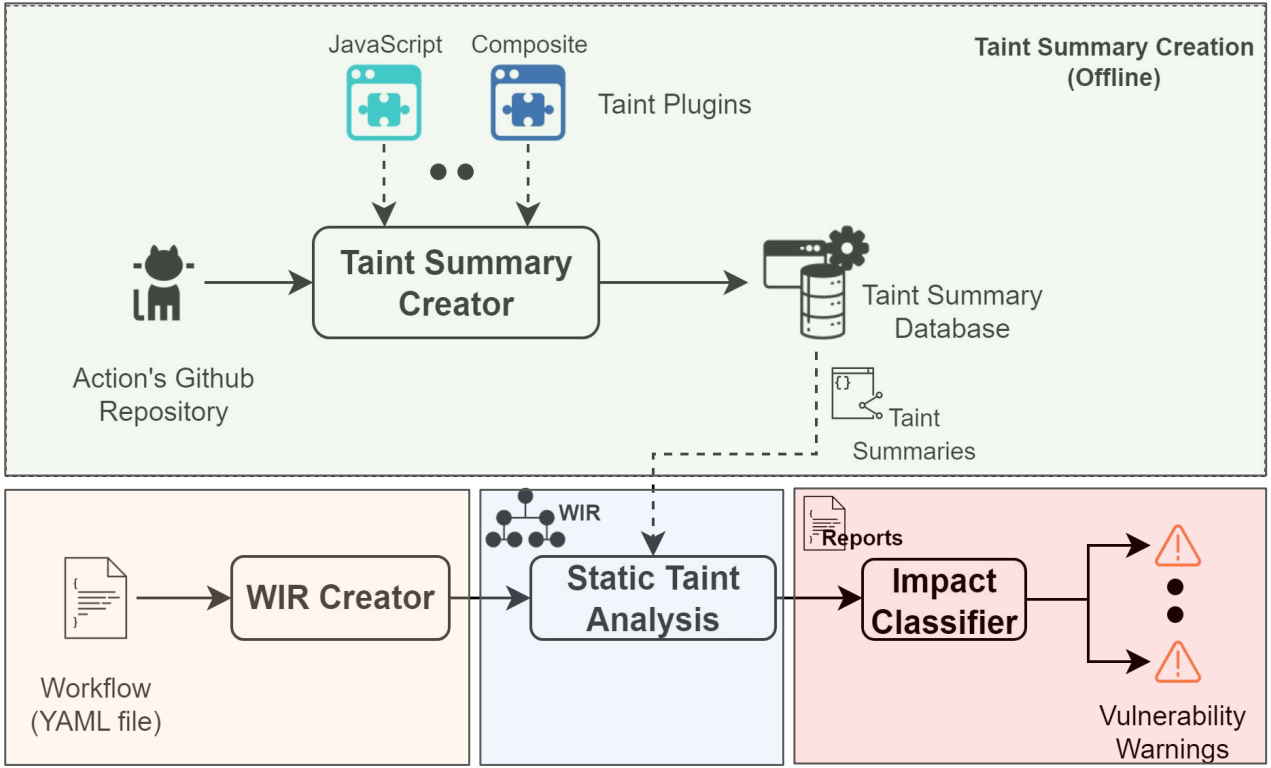
- Capture Workflow's **semantics and execution flow**
- Track **dataflow** across workflows and the actions
- Support multiple **programming languages**
- Predict the potential **impact** of identified vulnerabilities

Our Solution!

ARGUS

Staged Static Taint Tracking tool
for Github Workflows and Actions

Argus



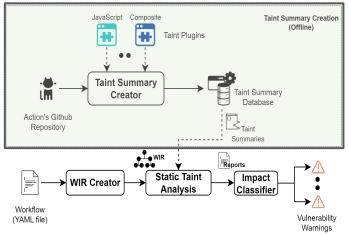
Action Taint Summaries

Workflow IR Generation

Static Taint Analysis

Impact Classifier

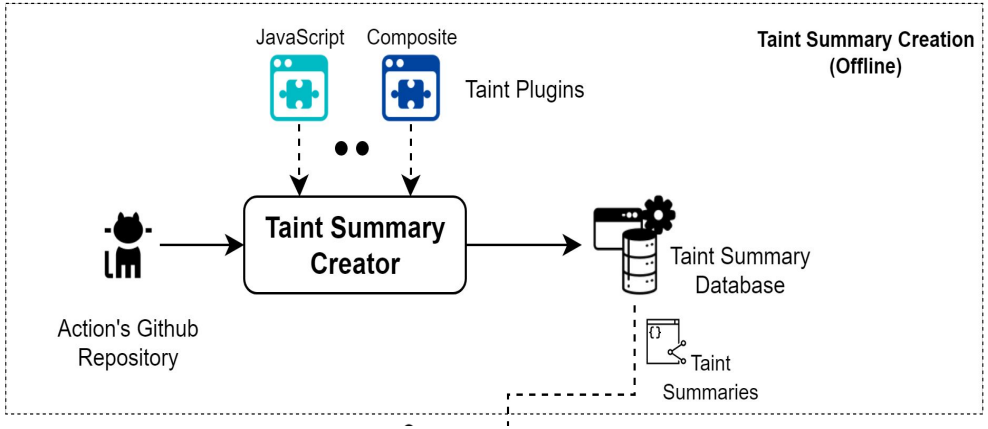
1 Generating Taint Summaries (Actions)



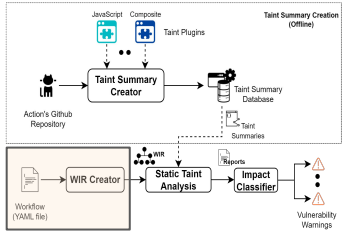
Javascript Actions

Composite Actions

~~Docker Actions~~



2 IR Generation



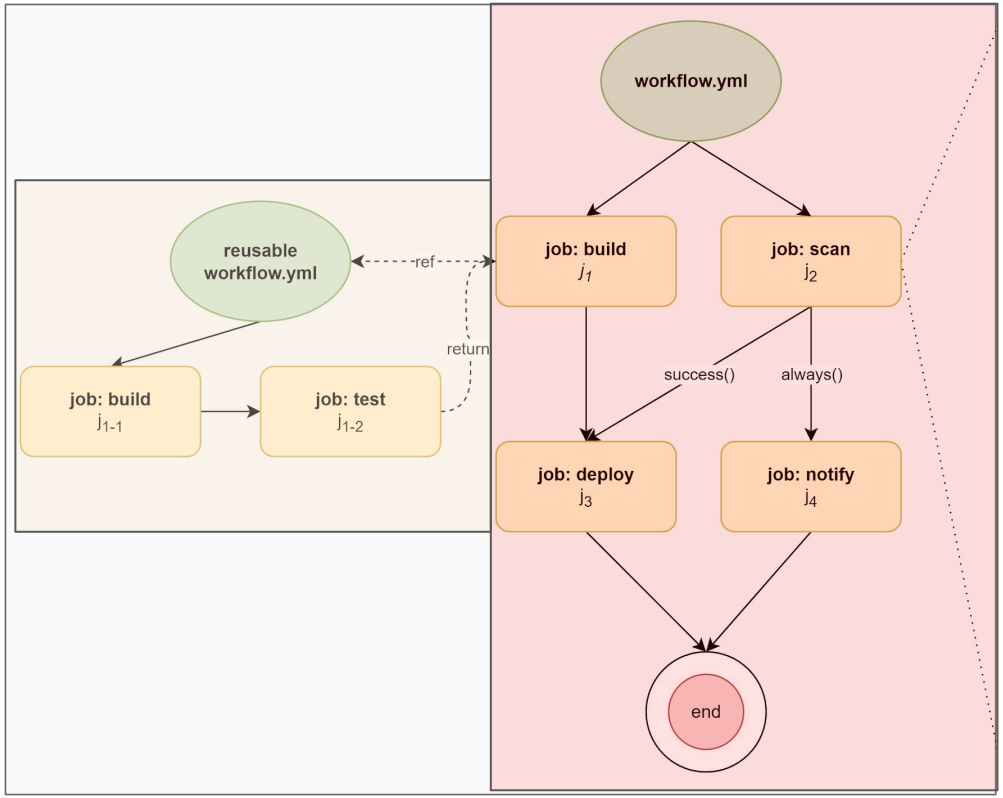

```
"build_step1" {
    exec {type: "gh_action", name: "action/checkout@v2"}
    execution_id: 0
    args {name: "token", value: "${{ secrets.GITHUB_TOKEN }}"}
    environment {}
    CIvars {name: "GITHUB_TOKEN", type: "secrets", ref:arg1 }
}
"build-proj" {
    exec {type: "shell_cmd", command: "./build.sh"}
    execution_id: 1
    args {}
    environment {name: "CFLAGS", value: "-Wall"}
    CIvars {}
}
```

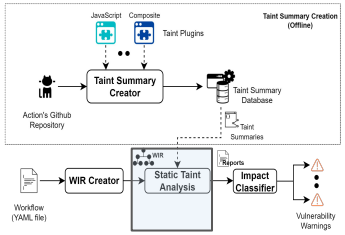
Ordering

Execution Environment

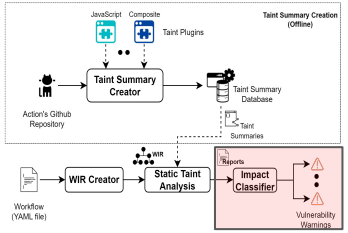
CI Variables

3 Workflow Dependency Graph

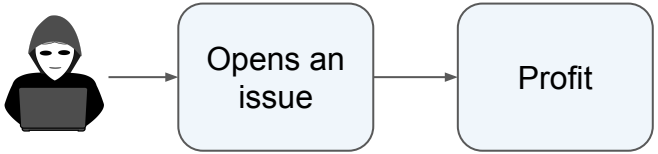




4 Impact Classifier

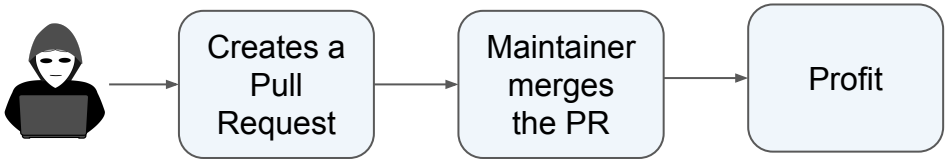


High Impact



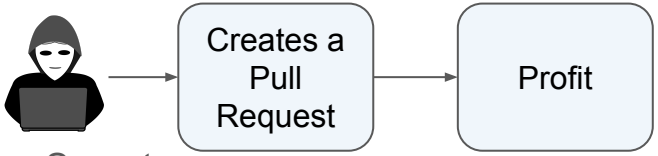
- Access to R/W GITHUB_TOKEN or Secrets
- No Maintainer Interaction Required

Medium Impact



- Access to R/W GITHUB_TOKEN or Secrets
- Requires Maintainer Interaction

Low Impact



- No Access to R/W GITHUB_TOKEN or Secrets

Case Study

```yaml
name: Issue Workflow
on:
  issues:
    types: [opened,edited]
jobs:
  #This job will check the issue to determine if it should be moved to a different repository
  redirectIssue:
    name: Check for issue transfer
    env:
      content_analysis_response: undefined
```

Dynamo Public

Open Source Graphical Programming for Design

C# 1,434 586 279 (3 issues need help) 36 Updated 22 minutes ago

```yaml
          replace-with: '-'
          flags: g
      - name: Check Information
        id: check-info
        run: |
          echo "content_analysis_response=$(pwsh .\\.github\\scripts\\title_analyzer.ps1 "${{
steps.remove_quotations.outputs.replaced }}" )" >> $GITHUB_ENV
      - name: Label issue
        if: env.content_analysis_response != 'Valid'
        run: |
          curl -v -u admin:${{ secrets.DYNAMOBOTTOKEN }} -d '{"labels": ["${{env.content_analysis_response}}"]}' ${{
github.event.issue.url }}/labels
```

Case Study

```yaml
name: Issue Workflow
on:
  issues:
    types: [opened,edited]
jobs:
  #This job will check the issue to determine if it should be moved to a different repository
  redirectIssue:
    name: Check for issue transfer
    env:
      content_analysis_response: undefined
```

Case Study

```yaml
- uses: actions/checkout@v2
- name: Remove conflicting chars
  env:
    ISSUE_TITLE: ${{github.event.issue.title}}
  uses: frabert/replace-string-action@v1.2
  id: remove_quotations
  with:
    pattern: "\""
    string: ${{env.ISSUE_TITLE}}
    replace-with: '-'
- name: Check Information
  id: check-info
  run: |
    echo "content_analysis_response=$(pwsh .\\.github\\scripts\\title_analyzer.ps1
    "${{ steps.remove_quotations.outputs.replaced }}" )" >> $GITHUB_ENV
- name: Label issue
  if: env.content_analysis_response != 'Valid'
  run: |
    curl -v -u admin:${{ secrets.DYNAMOBOTTOKEN }} -d '{"labels": ["${{env.content_analysis_response}}"]}'
    ${{ github.event.issue.url }}/labels
```

Taint Summary for replace-string-action @ v1.2		
Input Name	Sinks	Output Name
string	N/A	replaced

Evaluation

- **Accuracy** and **Precision** of taint tracking on **Actions**
- **Accuracy** and **Precision** of taint tracking on **Workflows**
- Effectiveness in **identifying security vulnerabilities**
- **Comparative evaluation** with existing state-of-the-art tools

Evaluation Dataset

Real World Dataset

2.8 million Workflows

1 million Repos

Vuln Bench

Collected a set of **24** previously reported vulnerable workflows

Breakdown of Workflows and Actions in the Dataset

No. of Workflows	No. of Repos	Actions		
		Type	**Count**	**Analyzable**
2,778,483	1,014,819	Javascript	22,433	22,433 (100%)
		Composite	9,292	9,292 (100%)
		Docker	13,445	0 (0%)
		Total	**48,369**	**31,725** (70.2%)

1 Taint Analysis on Actions

Input Flow

```
const property = core.getInput('property');
console.log(`property:${property}`);

const value = core.getInput('value');
console.log(`value:${value}`);

exec(`grep -r "^[#]*\s*${property}=.*" "${path}"`,
     (grepError) => {
      if(grepError != null) {
       ...
      } else {
       exec(`sed -ir \
         "s/^[#]*\s*${property}=.*/${property}=${value}/"
\        "${path}"`,
        (error, stderr) => {
         ...
     });
```

Reedyuk/write-properties

Direct Flow

```
function renderMark() {
  return > **${context.payload.action === 'reopened' ?
    context.payload.sender.login + 'issue' : 'issue'}**
    > **:** ${context.payload.issue.title}
   ..
}
const markdownString = renderMark();
exec(
    `curl ${wxhook} \
   -H 'Content-Type: application/json' \
   -d '
   {
     ...
    "markdown": {`
      `"content": "${markdownString.replaceAll('"', "'")}"
    }
   }'`...
  );
```

94dreamer/create-report

1 Taint Analysis on Actions

Precision of Taint Analysis by ARGUS on Actions

Type	Javascript			Composite		
	True Positives	False Positives	Precision	True Positives	False Positives	Precision
Input Flow	138	10	93.2%	46	1	97.9%
Direct Flow	27	0	100%	109	4	96.4%
Cumulative	175	10	**94.2%**	155	5	**96.8%**

94%
Precision in Javascript Actions

96%
Precision in Composite Actions

80
Unique Direct Flow Actions

2 Taint Analysis on Workflows

27,489
Vulnerable Workflows Identified

ALL
VulnBench Workflows Identified

3,643
High Impact Vulnerabilities

3 Vulnerability Identification

Severity Assignment of Vulnerabilities using the Impact Classifier

Flow Type	No. of Workflows				Num. Repos	Direct Flow Actions		Input Flow Actions	
	High (Total: 3,643)	Medium (Sampled: 1,000)	Low (Sampled: 1,000)	Total (Expected: 5,643)		Unique Root Cause	Unique Actions	Unique Root Cause	Unique Actions
Public Repositories									
Intra-WF	2,875	467	769	4,111	3,226	N/A			
Inta-WF-Ac	787	597	287	1,671	1,257	55	33	34	13
Total	**3,322** (91.18%)	985 (98.5%)	991 (99.1%)	5,298 (93.88%)	4,000	55	33	34	13

93%

Precision in finding vulnerabilities

5298

Zero day vulnerabilities found

4 Comparative Evaluation

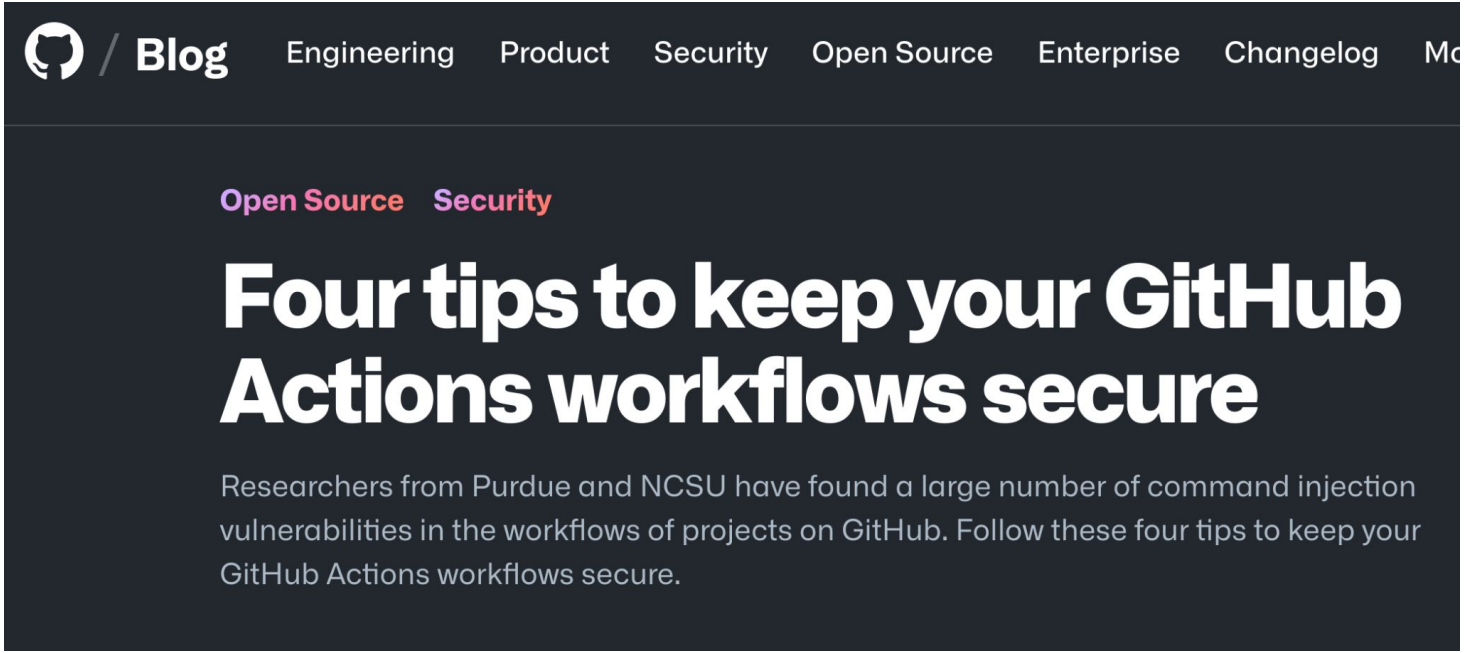
Comparative Evaluation of ARGUS with other state-of-the-art works in finding Code Injection Vulnerabilities

Tool	High/Medium				Low			
	TP	FP	FN	P	TP	FP	FN	P
GHAST	744	157	3,563	82.6%	331	363	660	47.7%
GITSEC	1,527	53	2,870	96.6%	204	3	787	98.5%
ARGUS	4,307	336	0	92.8%	991	9	0	99.1%

7x

More vulnerabilities discovered
Compared to State-of-the-art

Responsible Disclosure



150+ Vulnerable Workflows Fixed

20 Security Advisories

Conclusion

- Introduced ARGUS, the first static taint analysis system for GitHub Actions
- Our system can track taint across Workflows and Actions
- Conducted a Large scale evaluation of over 2.8million workflows identifying critical vulnerabilities

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

Website : secureci.org/argus
Code : github.com/purs3lab/argus