Command Injection via `issue_comment.body` in GitHub Actions Workflow

Critical **njzjz** published **GHSA-r4fj-r33x-8v88** 4 days ago

Package

njzjz/wenxian (GitHub Actions)

Affected versions

<= latest

Patched versions

none

Description

Summary

A GitHub Actions workflow uses untrusted user input from `issue_comment.body` directly inside a shell command, allowing potential command injection and arbitrary code execution on the runner.

Details

The workflow is triggered by `issue_comment`, which can be controlled by external users.
In the following step:

```
echo identifiers=$(echo "${{ github.event.comment.body }}" | grep -oE '@njzjz-bot
```

the value of `github.event.comment.body` is directly interpolated into a shell command inside `run:`.

Since GitHub Actions evaluates `${{ }}` before execution, attacker-controlled input is injected into the shell context without sanitization. This creates a command injection risk.

Additionally, the extracted value is later reused in another step that constructs output using backticks:

```
echo '@${{ github.event.comment.user.login }} Here is the BibTeX entry for `${{ st
```

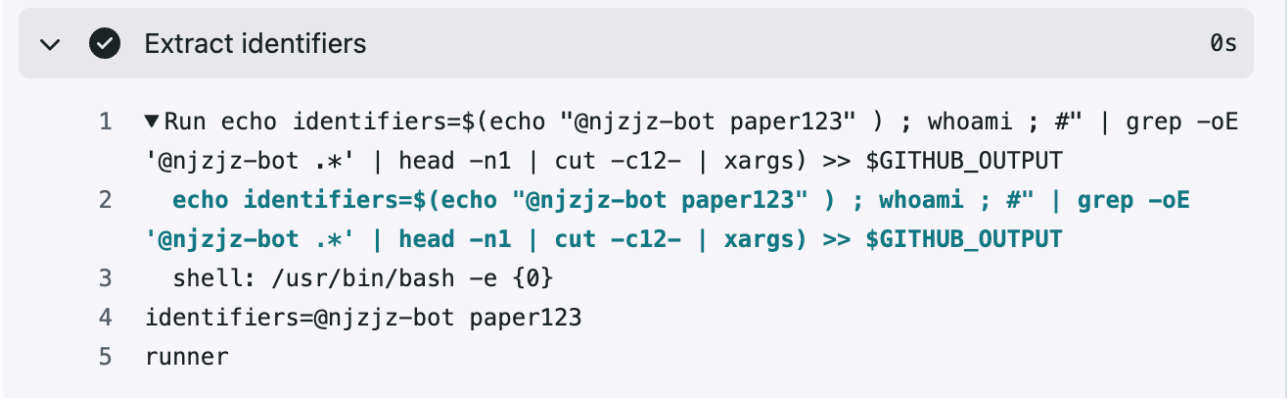
which may further propagate unsafe content.

PoC

1. Go to an issue in the repository
2. Post a comment such as:

```
@njzjz-bot paper123" ) ; whoami ; #
```

3. Observe whether the command is executed or reflected in logs/output



The injected payload successfully breaks out of the quoted context and executes arbitrary shell commands.

As shown in the workflow logs, the injected `whoami` command is executed, and the output (`runner`) is printed. This confirms that attacker-controlled input from `github.event.comment.body` is interpreted as shell commands.

This demonstrates a clear command injection vulnerability in the workflow.

Impact

- Remote attackers can inject arbitrary shell commands via issue comments
- Potential impacts:
  - Execution of arbitrary commands in GitHub Actions runner
  - Access to `GITHUB_TOKEN`
  - Exfiltration of repository data
  - CI/CD pipeline compromise

This issue affects all current versions of the repository as the vulnerable workflow is present in the main branch.

Suggested Fix

Avoid directly interpolating untrusted user input into shell commands.

Instead, pass `github.event.comment.body` through an environment variable and reference it safely within the script:

```
- name: Extract identifiers
  id: extract-identifiers
  env:
    COMMENT_BODY: ${{ github.event.comment.body }}
  run: |
    identifiers=$(echo "$COMMENT_BODY" | grep -oE '@njzjz-bot .*' | head -n1 | cut -c12-
    echo "identifiers=$identifiers" >> $GITHUB_OUTPUT
```

Severity

Critical 9.8 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVE ID

CVE-2026-34243

Weaknesses

▶ CWE-77

Credits

 **choseogyeong** Reporter