

Harsh Vardhan Kushwaha

17BCN7017

Secure Coding Theory Assignment – 2

1. What is COW?

Copy-on-write, an optimization strategy used in computer programming. The fundamental idea is that if multiple callers ask for resources which are initially indistinguishable, you can give them pointers to the same resource. This function can be maintained until a caller tries to modify its "copy" of the resource, at which point a true private copy is created to prevent the changes becoming visible to everyone else. All of this happens transparently to the callers. The primary advantage is that if a caller never makes any modifications, no private copy needs ever be created.

2. Brief description on Dirty COW (Dirty Copy on Write) vulnerability.

Dirty COW vulnerability is a type of privilege escalation exploit, which essentially means that it can be used to gain root-user access on any Linux-based system. While security experts claim that such kinds of exploits are not uncommon, its easy-to-exploit nature and the fact that it has been around for more than 11 years is pretty worrisome.

Dirty COW gets its name from the copy-on-write (COW) mechanism in the kernel's memory management system. Malicious programs can potentially set up a race condition to turn a read-only mapping of a file into a writable mapping. Thus, an underprivileged user could utilize this flaw to elevate their privileges on the system.

By gaining root privileges, malicious programs obtain unrestricted access to the system. From there on, it can modify system files, deploy keyloggers, access personal data stored on your device, etc.

3. What is Hal.dll? How to fix hal.dll errors?

Windows Hardware Abstraction Layer (HAL), a file that hides hardware complexities from Windows applications. Hal.dll is a system process that is needed for your PC to work properly. It should not be removed. The hal.dll is an executable file on your computer's hard drive. This file contains machine code. If you start the software Microsoft Windows Operating System on your PC, the commands contained in hal.dll will be executed on your PC. For this purpose, the file is loaded into the main memory (RAM) and runs there as a Microsoft Windows Hardware Abstraction Layer DLL process (also called a task).

Hal.dll Win XP error screen:

```
Windows could not start because the following file is missing or corrupt:  
<Windows root>\system32\hal.dll.  
Please re-install a copy of the above file.
```

Hal Win Vista+ error screen:

Windows Boot Manager

Windows failed to start. A recent hardware or software change might be the cause. To fix the problem:

1. Insert your Windows installation disc and restart your computer.
2. Choose your language settings, and then click "Next."
3. Click "Repair your computer."

If you do not have this disc, contact your system administrator or computer manufacturer for assistance

File: \Windows\system32\hal.dll

Status: 0xc0000221

Info: Windows failed to load because the HAL is missing, or corrupt.

Causes for this error –

1. Invalid BOOT.INI configuration file.
2. BCD Configuration is incorrect.
3. The Hal.dll is in fact corrupted or missing.
4. File system integrity compromised.

Fix –

1. Rebuild BOOT.ini or BCD with Easy Recovery Essentials –

Download Easy Recovery Essentials. Burn the image on any portable storage device. Boot your PC from the storage device. Choose “automated repair” on the OS drive. Once the process is finished. You can boot your PC normally and your error should be gone.

2. Manually attempt rebuild of boot.ini.

Insert the Windows Installation CD and boot into it. Start the recovery console and enter the following command –

```
bootcfg /rebuild
```

This will scan the computer for any other OSes. Assign a name to the OS and enter the following command when prompted for OS Load Options –

```
/fastdetect
```

Restart your computer. This should fix your errors.

4. What is significance of 787 Integer Overflow? Where was it first observed and how it can be rectified?

Integer overflow is the result of trying to place into computer memory an integer that is too large for the integer data type in a given system.

787 integer overflow is an integer overflow problem in the software of the Boeing 787 Dreamliner. It is such a software bug that can lead to the loss of control of the plane. A Model 787 airplane that has been powered continuously for 248 days can lose all the controls as a signed 32-bit integer happens to be the number of seconds in 248 days multiplied by 100 (i.e. a counter in hundredths of a second). Currently, the solution to this problem is to reboot the system of 787 airline every 248 days.

5. Write a program for integer overflow and test it in 32 bit and 64-bit compiler and look for Errors in 32-bit compiler when 64-bit program is run.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      int a, b, c;
8
9      cout << "Enter first number : ";
10     cin >> a;
11
12     cout << "Enter second number : ";
13     cin >> b;
14
15     cout << "Enter third number : ";
16     cin >> c;
17
18     if (a > b && a > c){
19         cout << a << " is the greatest.";
20     }
21     else if (b > c){
22         cout << b << " is the greatest.";
23     }
24     else{
25         cout << c << " is the greatest.";
26     }
27 }
```

In 32-bit compiler –

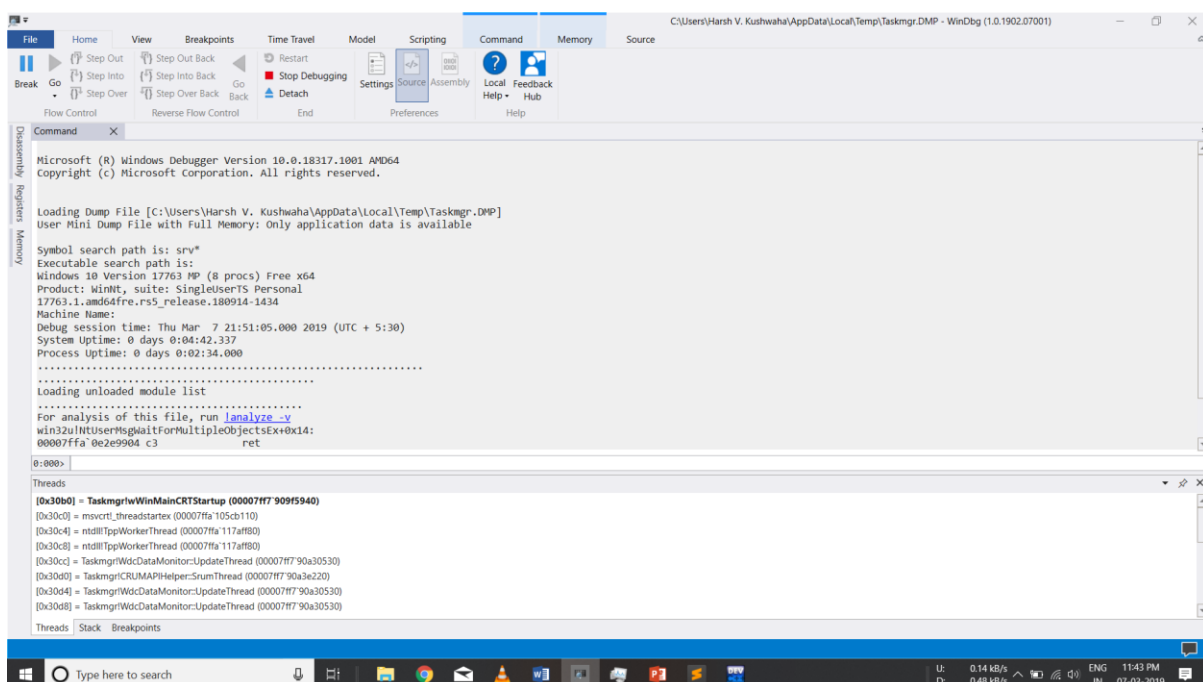
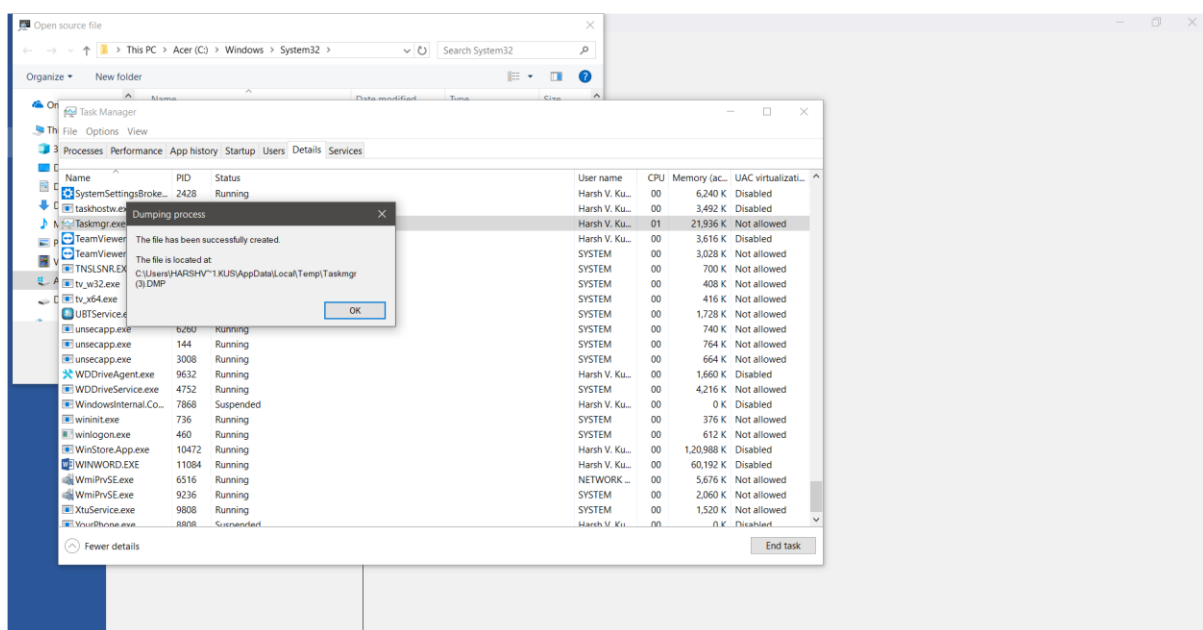
```
-----
Failed to execute "D:\College docs\projects\2nd year\sc\Untitled2.exe":
Error 1450: Insufficient system resources exist to complete the requested service.
Press any key to continue . . .
```

In 64-bit compiler –

```
Enter first number : 10
Enter second number : 15
Enter third number : 5
15 is the greatest.
-----
Process exited after 294.5 seconds with return value 0
Press any key to continue . . .
```

6. Open task Manager dump for any of the processes. Load dump. Use lm to list.

Going to Task Manager and creating a dump file for TaskMgr.exe



7. What is Init level 0-6? Where is it used? Significance of init in boot procedure.

A runlevel is one of the modes that a Unix-based operating system will run in. Each runlevel has a certain number of services stopped or started, giving the user control over the behaviour of the machine. Each runlevel has a certain number of services stopped or started, giving the user control over the behaviour of the machine. Conventionally, seven runlevels exist, numbered from zero to six.

After the Linux kernel has booted, the init program reads the `/etc/inittab` file to determine the behaviour of each runlevel. Unless the user specifies another value as a kernel boot parameter, the system will attempt to enter the default runlevel.

Run Level	Mode	Action
0	Halt	Shuts down system
1	Single-User Mode	Does not configure network interfaces, start daemons, or allow non-root logins
2	Multi-User Mode	Does not configure network interfaces or start daemons.
3	Multi-User Mode with Networking	Starts the system normally.
4	Undefined	Not used/User-definable
5	X11	As runlevel 3 + display manager(X)
6	Reboot	Reboots the system

8. What does Ring 0-3 mean in context of Operating systems?

In computer science, hierarchical protection domains, often called protection rings, are mechanisms to protect data and functionality from faults and malicious behaviour. This approach is diametrically opposite to that of capability-based security.

Computer operating systems provide different levels of access to resources. A protection ring is one of two or more hierarchical levels or layers of privilege within the architecture of a computer system. This is generally hardware-enforced by some CPU architectures that provide different CPU modes at the hardware or microcode level. Rings are arranged in a hierarchy from most privileged (most trusted, usually numbered zero) to least privileged (least trusted, usually with the highest ring number). On most operating systems, Ring 0 is the level with the most privileges and interacts most directly with the physical hardware such as the CPU and memory.

Special gates between rings are provided to allow an outer ring to access an inner ring's resources in a predefined manner, as opposed to allowing arbitrary usage. Correctly gating access between rings

can improve security by preventing programs from one ring or privilege level from misusing resources intended for programs in another. For example, spyware running as a user program in Ring 3 should be prevented from turning on a web camera without informing the user, since hardware access should be a Ring 1 function reserved for device drivers. Programs such as web browsers running in higher numbered rings must request access to the network, a resource restricted to a lower numbered ring.

A privilege level in the x86 instruction set controls the access of the program currently running on the processor to resources such as memory regions, I/O ports, and special instructions. There are 4 privilege levels ranging from 0 which is the most privileged, to 3 which is least privileged. Most modern operating systems use level 0 for the kernel/executive and use level 3 for application programs. Any resource available to level n is also available to levels 0 to n, so the privilege levels are rings. When a lesser privileged process tries to access a higher privileged process, a general protection fault exception is reported by the OS.

It is not necessary to use all four privilege levels. Current operating systems with wide market share including Microsoft Windows, macOS, Linux, iOS and Android mostly use a paging mechanism with only one bit to specify the privilege level as either Supervisor or User (U/S Bit). Windows NT uses the two-level system. The real mode programs in 8086 are executed at level 0 (highest privilege level) whereas virtual mode in 8086 executes all programs at level 3.

9. What is Nessus? How is it used to assess the vulnerabilities? What does it do to find the vulnerabilities?

Nessus is a proprietary vulnerability scanner developed by Tenable Network Security. Nessus allows scans for the following types of vulnerabilities:

1. Vulnerabilities that allow a remote hacker to control or access sensitive data on a system.
2. Misconfiguration (e.g. open mail relay, missing patches, etc.).
3. Default passwords, a few common passwords, and blank/absent passwords on some system accounts. Nessus can also call Hydra (an external tool) to launch a dictionary attack.
4. Denials of service against the TCP/IP stack by using malformed packets
5. Preparation for PCI DSS audits

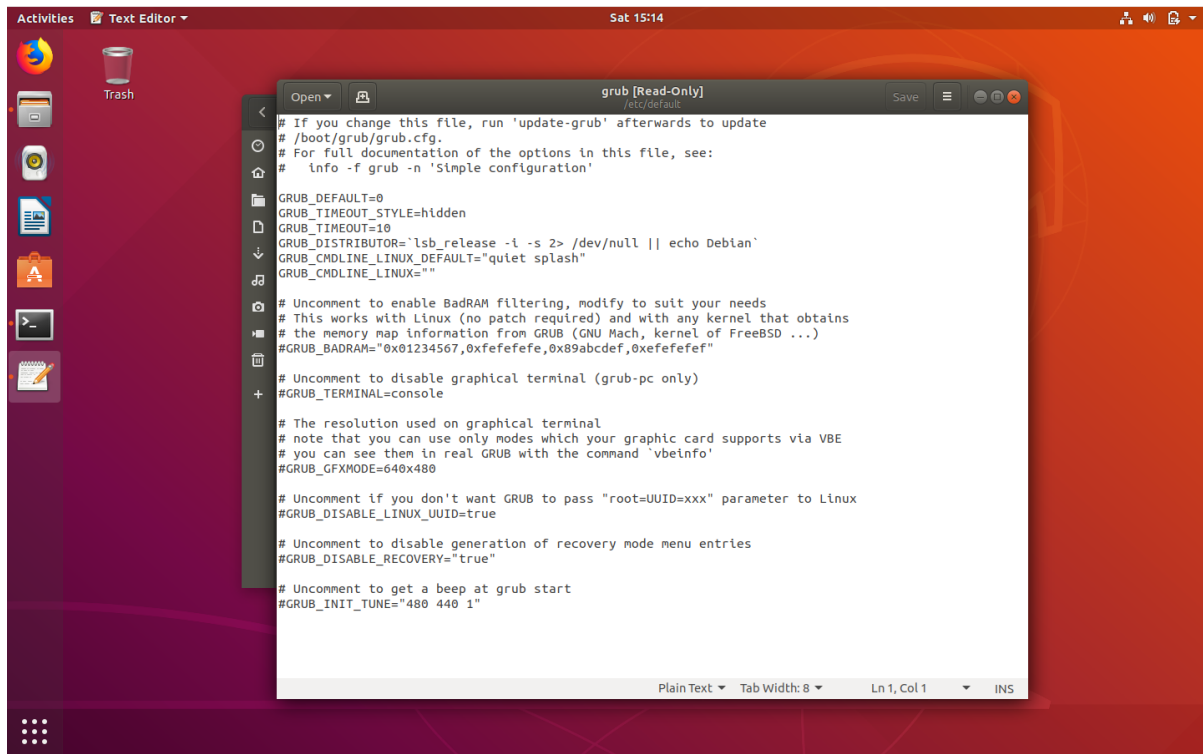
Most high-level network traffic, such as email, web pages, etc reach a server via a high-level protocol that is transmitted reliably by a TCP stream. To keep different streams from interfering with each other, a computer divides its physical connection to the network into thousands of logical paths, called ports. So, if you want to talk to a web server on a given machine, you would connect to port #80 (the standard HTTP port), but if you wanted to connect to an SMTP server on that same machine you would instead connect to port #25.

Each computer has thousands of ports, all of which may or may not have services (ie: a server for a specific high-level protocol) listening on them. Nessus works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack. Nessus is called a "remote scanner" because it does not need to be installed on a computer for it to test that computer. Instead, you can install it on only one computer and test as many computers as you would like.

After a scan, Nessus clients typically offer to means to analyse the result. The client itself will often list each vulnerability found, gauging its level of severity and suggesting to the user how this problem could be fixed.

10. Try to boot Ubuntu or any linux in cli (UNIX) without GUI and analyze the init level.

Original Grub File –



A screenshot of a Linux desktop environment. A text editor window titled 'grub [Read-Only] /etc/default' is open, displaying the contents of the GRUB configuration file. The desktop background is a red Ubuntu logo. The text editor shows the following configuration:

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

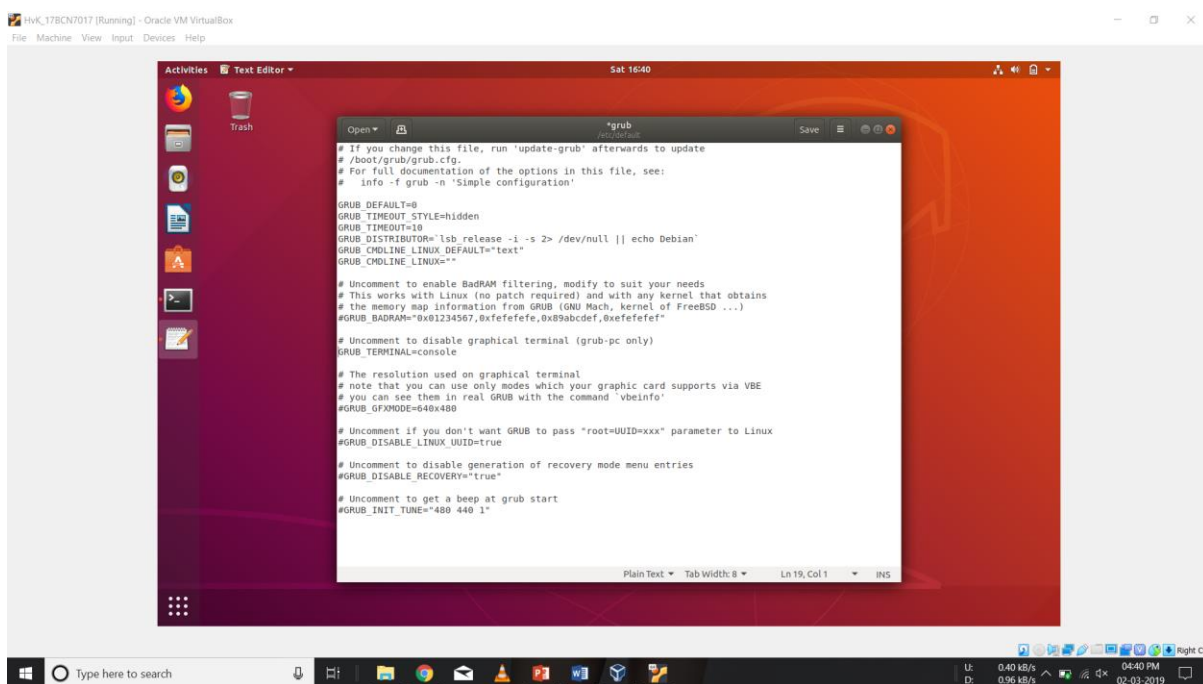
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```



A screenshot of a Windows desktop environment. A text editor window titled 'grub [Read-Only] /etc/default' is open, displaying the contents of the GRUB configuration file. The desktop background is a red Ubuntu logo. The text editor shows the same configuration as the previous screenshot, but with some changes highlighted in yellow:

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="text"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

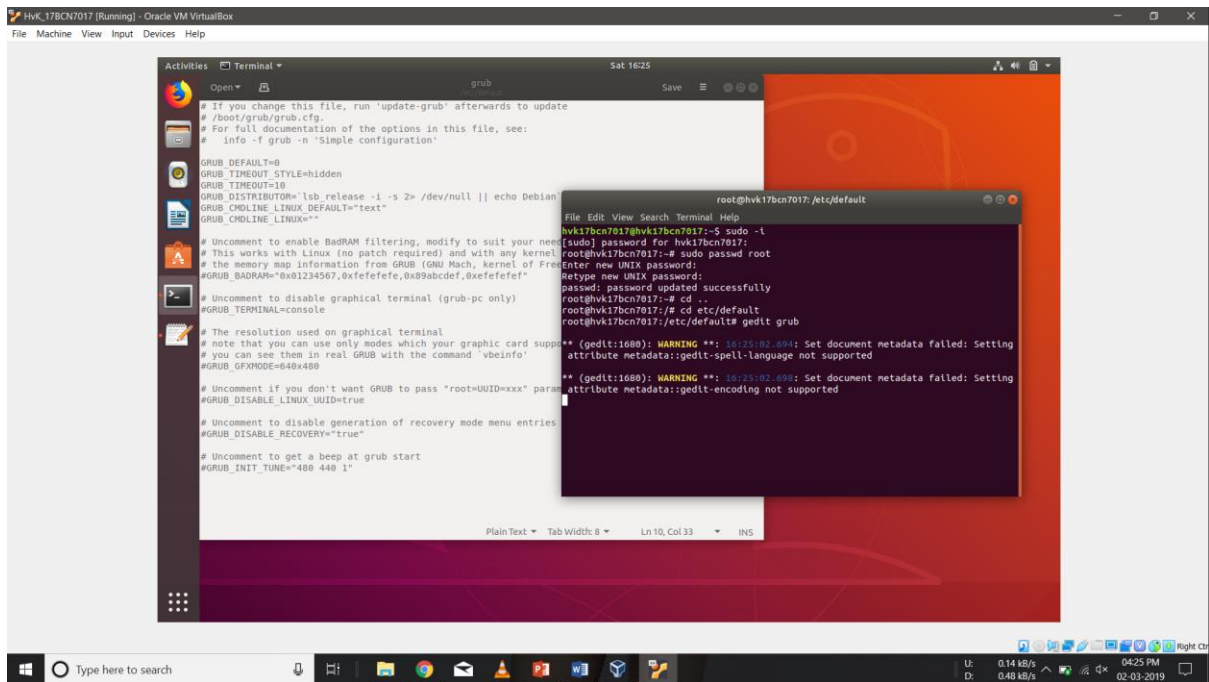
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

Now committing some changes –

```
GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="text"
GRUB_CMDLINE_LINUX=""
```



Update your GRUB using the following commands –

```
sudo update-grub
```

```
sudo systemctl set-default multi-user.target
```

The next time you boot your PC it will boot in CLI mode.

11. Why RC4 is vulnerable to WEP attacks?

Wired Equivalent Privacy (WEP) is a security algorithm for IEEE 802.11 wireless networks. Introduced as part of the original 802.11 standard ratified in 1997, its intention was to provide data confidentiality comparable to that of a traditional wired network. WEP, recognizable by its key of 10 or 26 hexadecimal digits (40 or 104 bits), was at one time widely in use and was often the first security choice presented to users by router configuration tools.

In cryptography, RC4 (Rivest Cipher) is a stream cipher. While remarkable for its simplicity and speed in software, multiple vulnerabilities have been discovered in RC4, rendering it insecure. It is especially vulnerable when the beginning of the output keystream is not discarded, or when non-random or related keys are used. Particularly problematic uses of RC4 have led to very insecure protocols such as WEP.

Weakness in WEP –

1. Key Management and Key Size
2. The Initialization Vector (IV) is too small
3. The Integrity Check Value (ICV) algorithm is not appropriate
4. Authentication Messages can be easily forged

And lastly, usage of RC4 in WEP –

RC4 in its implementation in WEP has been found to have weak keys. Having a weak key means that there is more correlation between the key and the output than there should be for good security.

Determining which packets were encrypted with weak keys is easy because the first three bytes of the key are taken from the IV that is sent unencrypted in each packet. This weakness can be exploited by a passive attack. All the attacker needs to do is be within a hundred feet or so of the AP.

Out of the 16 million IV values available, about 9000 are interesting to the most popular attack tool, meaning they indicate the presence of weak keys. The attacker captures “interesting packets”, filtering for IVs that suggest weak keys. After that attacker gathers enough interesting packets, he analyses them and only has to try a small number of keys to gain access to the network. Because all of the original IP packets start with a known value, it’s easy to know when you have the right key. To determine a 104-bit WEP key, you have to capture between 2000 and 4000 interesting packets. On a fairly busy network that generates one million packets per day, a few hundred interesting packets might be captured. That would mean that a week or two of capturing would be required to determine the key.

The best defence against this type of attack is not to use those weak IV values. Most vendors are now implementing new algorithms that simply do not choose weak IVs. However, if just one station on the network uses weak keys, the attack can succeed.

12. How existing crypto is vulnerable to KRACK and how KRACK attack works?

KRACK, an acronym for Key Reinstallation Attack is a tool used for cracking WPA2 by exploiting the vulnerability of Wi-Fi. The attacker is able to decrypt all data that the victim transmits. For an attacker this is easy to accomplish, because our key reinstallation attack is exceptionally devastating against **Linux and Android 6.0 or higher**. This is because Android and Linux can be tricked into (re)installing an all-zero encryption key. When attacking other devices, it is harder to decrypt all packets, although a large number of packets can nevertheless be decrypted.

Our main attack is against the 4-way handshake of the WPA2 protocol. This handshake is executed when a client wants to join a protected Wi-Fi network, and is used to confirm that both the client and access point possess the correct credentials (e.g. the pre-shared password of the network). At the same time, the 4-way handshake also negotiates a fresh encryption key that will be used to encrypt all subsequent traffic. Currently, all modern protected Wi-Fi networks use the 4-way handshake. This implies all these networks are affected by (some variant of) our attack. For instance, the attack works against personal and enterprise Wi-Fi networks, against the older WPA and the latest WPA2 standard, and even against networks that only use AES.