

# Exercise Class “Introduction into Cyber Security”

Winter Term 2018/2019



**Chair of IT Security**

Topic:

Introduction into Cyber Security

– Buffer Overflow Attacks –

Deadline: 9th January, 2019

**Supervised By:**

Torsten Ziemann

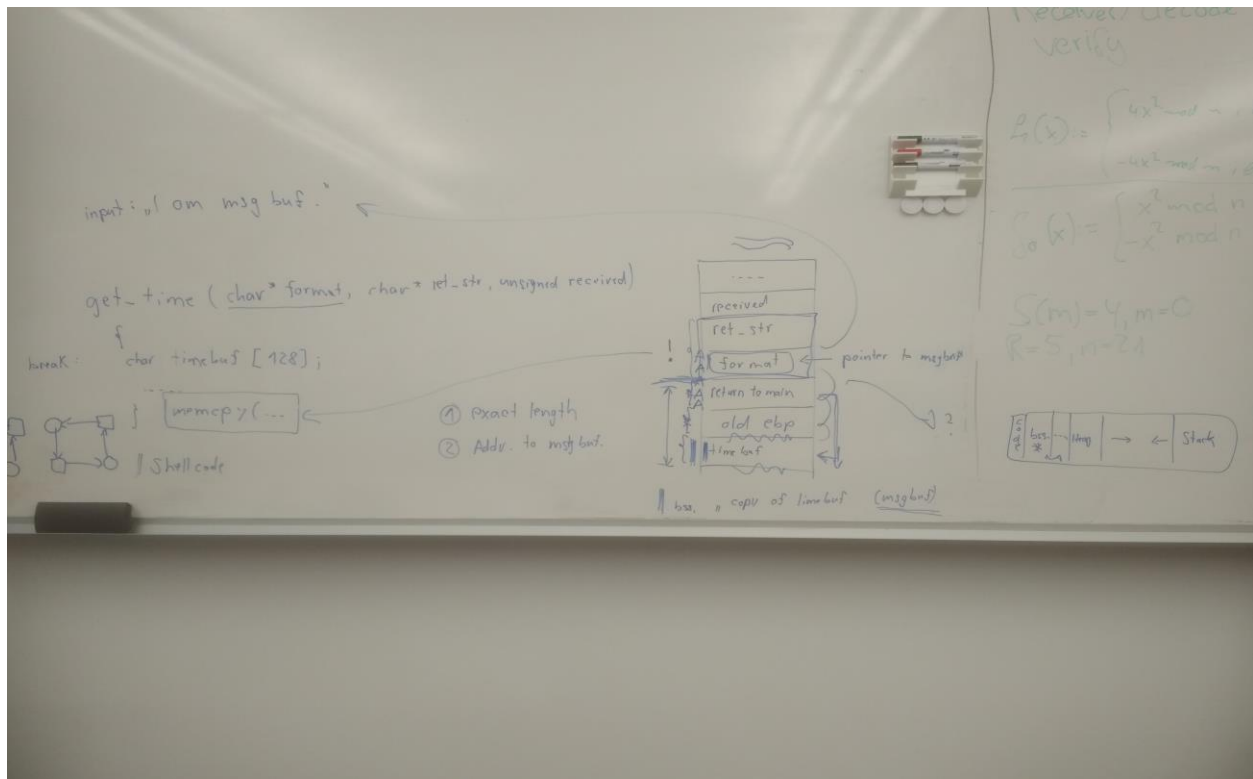
**Group: 14**

**Members:**

Siddique Reza Khan  
Sam Abubakar Tareque  
Fathima Shaik

## 1.0 Description of our Task:

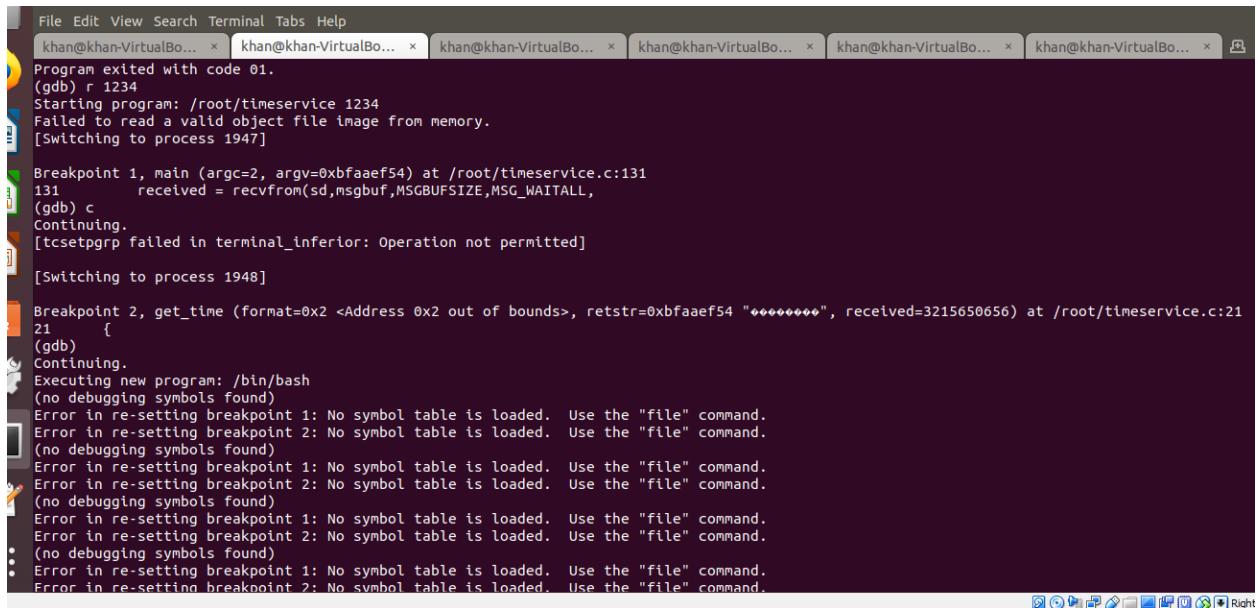
First, the task we have completed till now to find the exact length of the buffer and then overwrite the "get\_time function" call return address with the msgbuf, global variable, address.



As we discussed had with you, it can be possible for us to print a message after redirect the return address pointing to the msgbuf, a global variable.

```
khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  kl
(gdb) ni
0x08048833 43      strftime(retstr, TIMEBUFSIZE, format, localtime);
(gdb) ni
0x08048837 43      strftime(retstr, TIMEBUFSIZE, format, localtime);
(gdb)
0x0804883f 43      strftime(retstr, TIMEBUFSIZE, format, localtime);
(gdb)
0x08048842 43      strftime(retstr, TIMEBUFSIZE, format, localtime);
(gdb)
0x08048845 43      strftime(retstr, TIMEBUFSIZE, format, localtime);
(gdb)
46      }
(gdb)
Cannot access memory at address 0x41414145
(gdb)
0x08049e60 in msgbuf ()
(gdb) x/x msgbuf
0x08049e60 <msgbuf>: 0x73696854
(gdb) x/s msgbuf
0x08049e60 <msgbuf>: "This is message buffer"
(gdb) ni
0x08049e61 in msgbuf ()
(gdb) ni
0x08049e66 in msgbuf ()
(gdb) ni
0x08049e88 in msgbuf ()
(gdb) ni
0x08049e89 in msgbuf ()
(gdb)
0x08049e8a in msgbuf ()
(gdb)
```

However we could manage to get the shell program named `/bin/bash` in the `msgbuf` memory address but after continue running it was hanging for sometimes.

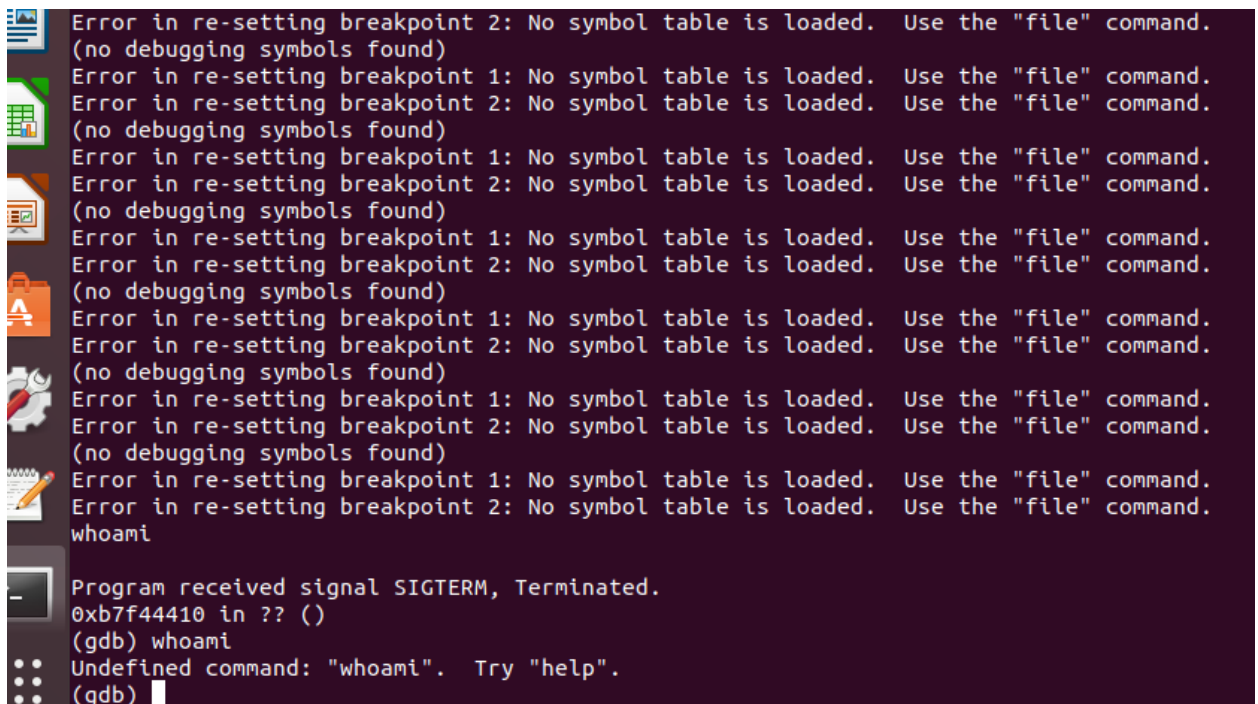


```
File Edit View Search Terminal Tabs Help
khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  khan@khan-VirtualBo... x  khan@khan-VirtualBo... x
Program exited with code 01.
(gdb) r 1234
Starting program: /root/timeservice 1234
Failed to read a valid object file image from memory.
[Switching to process 1947]

Breakpoint 1, main (argc=2, argv=0xbfaaef54) at /root/timeservice.c:131
131     received = recvfrom(sd,msgbuf,MSGBUF5SIZE,MSG_WAITALL,
(gdb) c
Continuing.
[tcsetpgrp failed in terminal_inferior: Operation not permitted]
[Switching to process 1948]

Breakpoint 2, get_time (format=0x2 <Address 0x2 out of bounds>, retstr=0xbfaaef54 "*****", received=3215650656) at /root/timeservice.c:21
21     {
(gdb)
Continuing.
Executing new program: /bin/bash
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
```

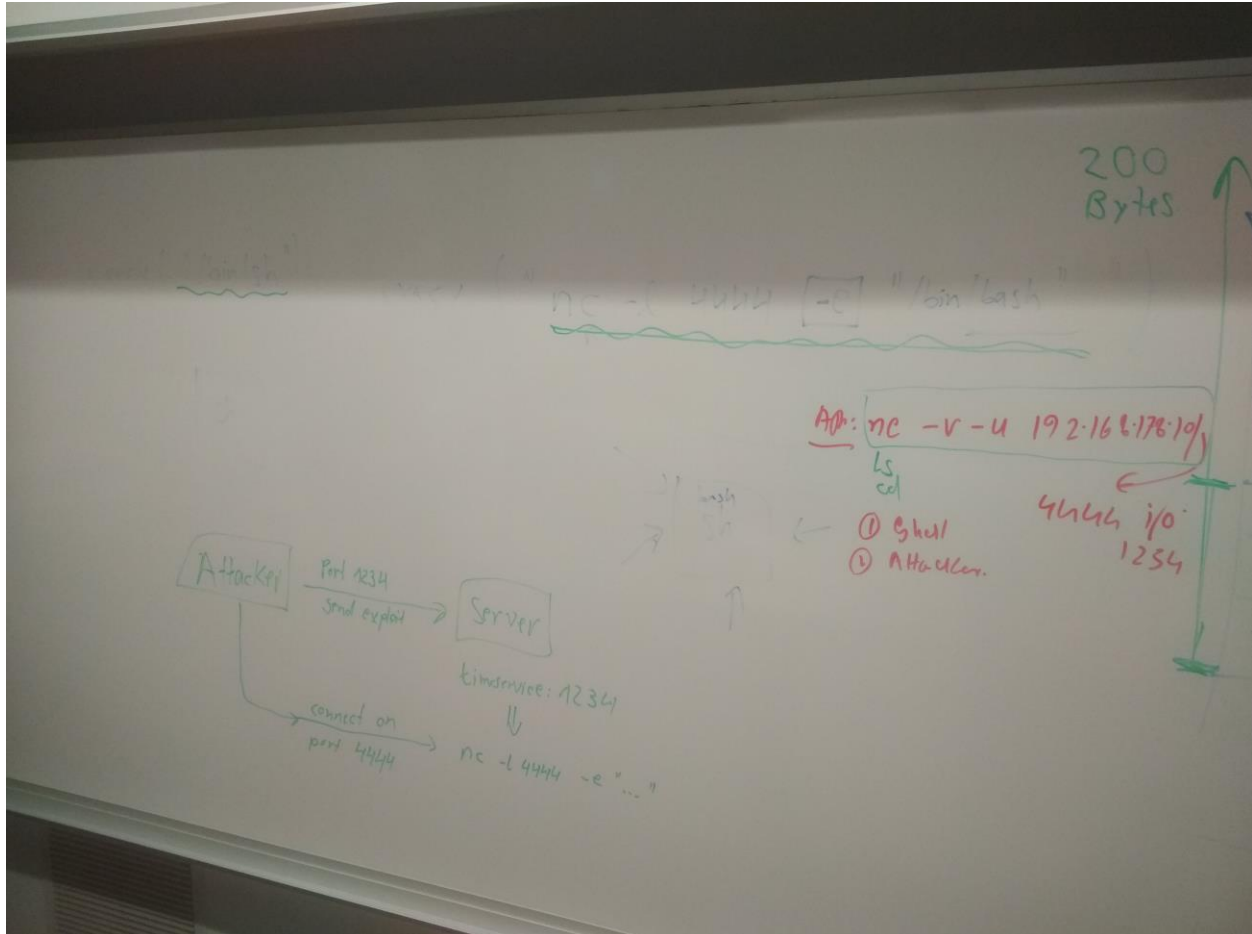
Last, we could get the shell. Whereas the `"shell program"` was not executed in our local attacker machine but we had a program crash with unusual termination.



```
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
(no debugging symbols found)
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 2: No symbol table is loaded. Use the "file" command.
whoami
Program received signal SIGTERM, Terminated.
0xb7f44410 in ?? ()
(gdb) whoami
Undefined command: "whoami". Try "help".
(gdb)
```

## 2.0 Problem:

As we discussed had with you in the lab, that we have to change our shell code with bind a port.



## 3.0 Future Work:

Our future goal to 1. Improve and generate the "shell code" as you suggested to us, 2. Exploit the local attacker machine 3. Run the shell into the server machine and 4. try to get the secret message from the server.

## 4.0 Special Thanks:

We are very much glad to get a wonderful guidance, support and cooperation from our supervisor, Mr. Torsten Ziemann.

## Bibliography:

1. <https://dhavalkapil.com/blogs/Buffer-Overflow-Exploit/>
2. <https://dhavalkapil.com/blogs/Shellcode-Injection/>
3. <https://dhavalkapil.com/assets/files/Shellcode-Injection/shellcode.asm>
4. <https://www.linuxnix.com/suid-set-suid-linuxunix/>
5. <http://www.primalsecurity.net/0x0-exploit-tutorial-buffer-overflow-vanilla-eip-overwrite-2/>
6. <https://sploitfun.wordpress.com/2015/05/08/classic-stack-based-buffer-overflow/>
7. <https://sploitfun.wordpress.com/author/sploitfun/>
8. <https://penturalabs.wordpress.com/2011/03/31/vulnerability-development-buffer-overflows-how-to-bypass-full-aslr/>
9. <https://www.slideshare.net/AlexandreMoneger/07-aslr-bypass>
10. <https://security.stackexchange.com/questions/157478/why-jmp-esp-instead-of-directly-jumping-into-the-stack>
11. <http://www.sheepshellcode.com/blog/2015/03/24/writing-buffer-overflow-exploits-with-aslr/>
12. <https://sploitfun.wordpress.com/2015/05/08/bypassing-aslr-part-i/>
13. <https://stackoverflow.com/questions/16110591/aslr-brute-force>
14. <https://hk.saowen.com/a/97794707e756f8e5724f107ea3f7d2ef7a388a157a8fb20080fee56f7b1ac646>
15. <https://cybersecurity.upv.es/attacks/offset2lib/offset2lib.html>
16. <https://security.stackexchange.com/questions/20497/stack-overflows-defeating-canaries-aslr-dep-nx>
17. <https://security.stackexchange.com/questions/18556/how-do-aslr-and-dep-work>
18. <https://dkalemis.wordpress.com/2010/10/27/the-need-for-a-pop-pop-ret-instruction-sequence/>
19. <https://www.corelan.be/index.php/2009/07/23/writing-buffer-overflow-exploits-a-quick-and-basic-tutorial-part-2/>
20. <https://www.corelan.be/index.php/2010/06/16/exploit-writing-tutorial-part-10-chaining-dep-with-rop-the-rubikstm-cube/>
21. <https://www.securitysift.com/windows-exploit-development-part-6-seh-exploits/>