COMP 4632
Practicing Cybersecurity: Attacks and Counter-measures

Week 10 Lab Exercise

Topic: *Buffer Overflow 101*

Lab Objective

This lab would introduce the mechanics of stack buffer overflow vulnerabilities:

- Authentication Override
- Buffer Overflow to gain privilege of another user

Task 1 – Authentication Override

A simple authentication program was provided. Students must find ways to overcome the authentication program by providing the correct password (or by any other means necessary)

Username and password for the VM: comp4632

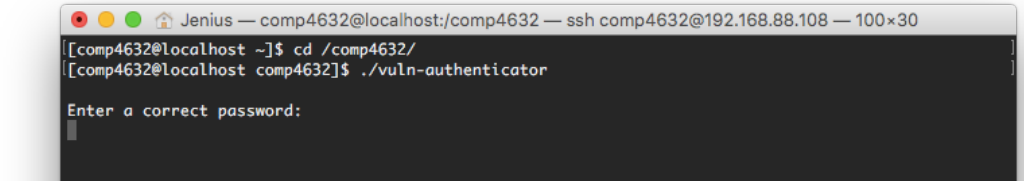
Task 1.1 Inspect the Source Code of the Authentication Program

- Using your preferred editor / viewer, inspect the source code of the authentication program located at "/comp4632/vuln-authenticator.c" (Example, using vim: ***vim /comp4632/vuln-authenticator.c***)

Question 1: What are the condition to pass authentication? (0.5 mark)

Task 1.2 Run the program and pass authentication

- Go into the folder "/comp4632" by executing ***cd /comp4632***
- Run the program by executing ***./vuln-authenticator***



- By providing a password input, get the program to return "Authentication Successful"

Question 2: What did you enter to gain success message? (1 mark)

Task 2 – Getting Obi-Wan's Secret

We want to gain access to the secrets of the user ***obiwan***, which is also stored in the same folder under the filename 'kenobi-secret'

Task 2.1 Identify files with SUID set

- Files with SUID set will be executed with the privilege of the owner of the file, not the current user
- Such files are denoted with the an 's' bit in permissions of the files, denoted by '---s------' (where – are any character)
- A full list of files, their respective permissions and owners can be listed using the command ***ls -al***

Question 3: List the files in the folder that has SUID set? Which file will be run with *obiwan*'s privilege? (1 mark)

Task 2.2 Inspect the source code of vuln-kenobi.c

- The source code of the vulnerable program can give us insights on how it can be exploited
- Review the source code of vuln-kenobi.c

Question 4: What is the variable that you would select to be attacked? What size in bytes is the variable? (0.5 mark)

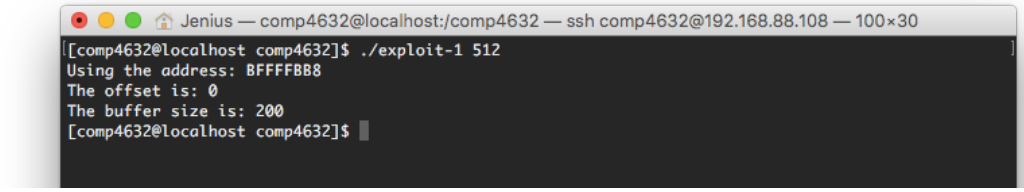
Task 2.3 Inspect the source code of exploit-1.c

- To attack the ***vuln-kenobi*** program, an "egg shell" exploit program would be used. Inspect the program's source code using your preferred editor / viewer. (e.g. ***vim exploit-1.c***)
- The program can estimate the return pointer address, and building a payload including the estimated return pointer address, the payload (a shellcode) and NOP instructions
- NOP instructions help by providing a bigger "hit" area for the return address
    - Without them, the exact address of the shellcode's location in the stack must be used
    - With NOP instructions, landing the return address to any of the NOP instructions would yield the same result

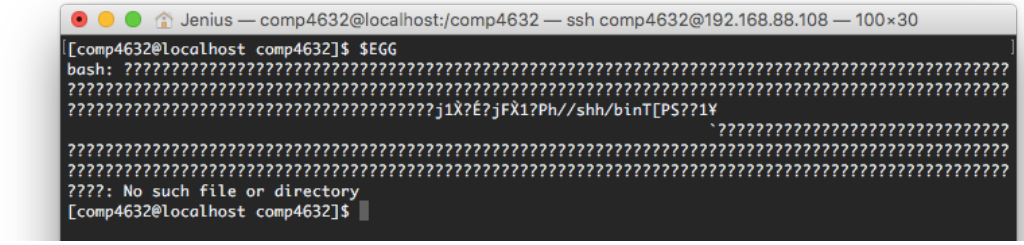
- The "egg shell" program will spawn another bash shell, but with a environmental variable called ***$EGG***

Task 2.4 Execute the Egg Shell program, and providing the parameter on the buffer size

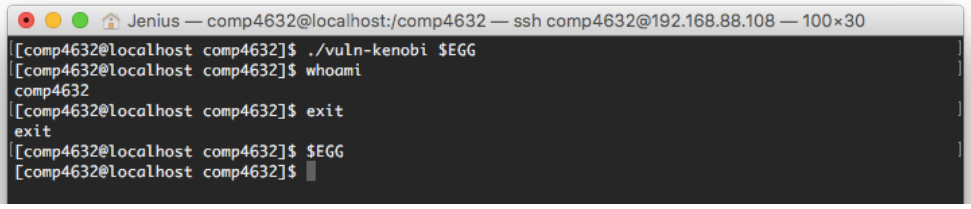
- To perform buffer overflow on ***vuln-kenobi***, the parameter provided to ***exploit-1*** should be larger than the variable to be attacked
- Start with the exact size of the variable in ***vuln-kenobi***
- Generate this $EGG by the command ***./exploit-1 <size>***



- Verify that a new bash shell has been spawned, but with the ***$EGG*** environment variable by the command ***$EGG***

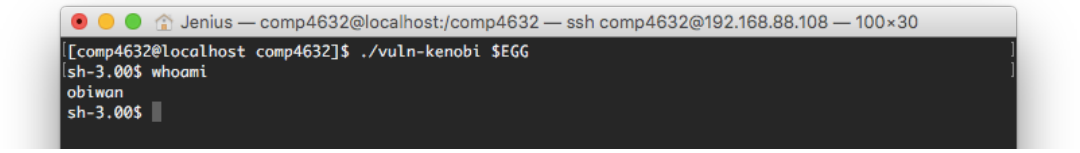


- Try exploiting vuln-kenobi by using the command ***./vuln-kenobi $EGG***
- Verify if you have obtain the privilege of *obiwan* by using the command ***whoami***
- If the exploit was unsuccessful, then ***whoami*** would return your own username
- Clean up the environment by exiting the egg shell by the command ***exit***
- Verify that you are no longer in the egg shell by verifying the output from the command ***$EGG***



Task 2.5 Continue the testing process until the exploit is successful

- Retry the attack process by repeating the process, each time incrementing the starting number by 100
- Stop when ***whoami*** return a privilege that is not your own



- Using your new found powers, view the secrets of Obi-Wan by viewing the content of ***kenobi-secret***

HINT: Ensure that you always start from the real shell, not the egg shell by ensuring that the $EGG environment variable is not present before you run the egg shell program.

Bonus Question 5: What is the content of kenobi-secret? (1.5 mark)

Task 2.6 Gain knowledge to Qui-Gon Jinn's Secret *(Bonus)*

- Using your new found knowledge of the dark side, perform an exploit on ***vuln-jinn***

HINT: No modifications to any of the programs are required. Only the parameters are different. Review to source code of vuln-jinn.c to find Qui-Gon's weakness.

Bonus Question 6: What is the content of jinn-secret? What is the numeric value used to generate the $EGG? (1.5 mark)

End of Lab