Zhixing Jiang and Eli Arbogast
June 1, 2021
CS231

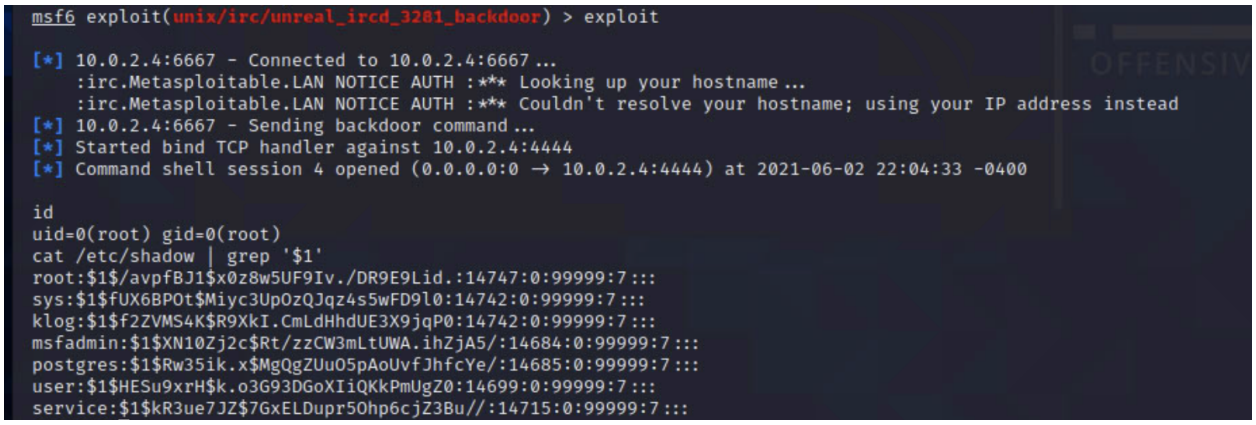
Penetration Testing #2

Part 2

We chose the service UnrealIRCd. By googling up, we found a module `unreal_ircd_3281_backdoor`, which "exploits a malicious backdoor that was added to the Unreal IRCD 3.2.8.1 download archive.", according to Rapid7 (https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor/). UnrealIRCd allows a user to run their own IRC server from their system, and the backdoor can be used to "run any command on a system running the compromised server."(https://311hrs.wordpress.com/2016/05/22/metasploitable-2-unreal-ircd-part-10/). "The vulnerability allowed an attacker to execute arbitrary code by sending the string "AB," which triggered the backdoor, followed by the payload. The command would run as whatever user the IRC daemon was running as, so root-level access could potentially be achieved."(https://null-byte.wonderhowto.com/how-to/hack-unrealircd-using-python-socket-programming-0198050/)

- To use the module, enter use exploit/unix/irc/unreal_ircd_3281_backdoor.
- Enter `show options.` Then enter `set RHOST 10.0.2.4`
- We first used the payload `cmd/unix/bind_perl`. Enter `exploit` to run it. This payload "listen for a connection and spawn a command shell via perl (persistent)." . Unlike reverse, it binds a command prompt to a port on the target machine, so that we can then connect to it. (https://www.infosecmatter.com/list-of-metasploit-payloads-detailed-spreadsheet/)
- We also used the payload "cmd/unix/reverse_bash_telnet_ssl" (https://www.infosecmatter.com/list-of-metasploit-payloads-detailed-spreadsheet/). It "creates an interactive shell via mkfifo and telnet." and it encrypts using ssl. A connection is made from the target to our machine.

We can obtain usernames and passwords by entering the command `cat /etc/shadow | grep '$1'` after a shell is started. This is what we got:



Part 3

The ps command gives us a list of activities, from which we can see which ones are suspicious

Part 4

As I was searching the internet for details on various exploits/modules we worked with, I came across the backdoor that was prevalent across multiple services that just required that the user input a smiley face ":)" after the prompt for username, and it would force a login to a local shell.