Enumeration

2021年7月30日 11:47

1: Use autorecon to enumerate its services, **22, 80, 111, 2049, 7742, 8080, 43329, 45519, 54375, 55683** are open
2: Http service running on 80 does not have any content.
3: **showmount -e 192.168.61.100**, and there is no **public share** can be mounted. This is also not the right direction
4: Enumerate Http service which runs on **7742**, it has a hidden folder: **zipfiles**. It contains **4 zip files**. One of it, **max.zip** contains max's **home folder**, including his **ssh directory**. It is definitely the way to get a initial shell
5: Even though, check **8080**, it does not have anything accessible

Foothold

2021年7月30日 11:48

1: Max's home folder contain **Tomcat's config** file, it has **Tomcat manager's credential**. However, I cannot access **Tomcat manager**, so it does not help me
2: There is an interesting file: **scp_wrapper.sh**. According to the content, it **restricts** me to use only **SCP** command when I connect to target's via Max's SSH private key.
3: Check his **authorized_keys**, it is the case. authorized_keys **invokes** scp_wrapper.sh to **make the restriction**.
4: To **bypass** the restriction, I can **modify scp_wrapper.sh** and then **upload** it to Max's home folder. But there is a **more perfect** way, write **my public key** to **Max's authorized_keys**.
5: Create an authorized_keys file, it contains my public key. **scp authorized_keys max@192.168.61.100:/home/max/.ssh/authorized_keys**
6: Now I can directly log in target server as Max **without any restriction**
7: The local flag locates in **dennis's** home folder

Privilege Escalation

2021年7月30日 11:48

1: Check **SUID** file, I find **start-stop-daemon** is set SUID
2: Execute **/usr/sbin/start-stop-daemon -n $RANDOM -S -x /bin/sh -- -p**
3: Get the root shell

Review

2021年7月30日 11:49

1: Target **SSH** and **HTTP** service
2: Find the **hidden directory** and download all zip files
3: Among four zip files, find the one contains **more juicy content**
4: Understand the **restriction** by analyzing **scp_wrapper.sh** and original **authorized_keys**, and then upload a **new authorized_keys** to **bypass the restriction**
5: Find the **SUID binary**, exploit it and get root shell