Secure Command Write-up

What is SSH?

SSH (Secure Shell) service is a protocol used to securely access another computer over a network. Basically, it allows users to access a remote server or computer over an encrypted network connection. This access is usually through the command line and allows the user to use the remote machine through the command line.

An SSH service consists of two main components: an SSH client and an SSH server. The SSH client runs on the computer where the user is located and is used to connect to the remote SSH server. The SSH server runs on the remote computer and accepts incoming SSH connections. The communication between client and server is encrypted to secure it. This encryption prevents the transmitted data from being read or modified by third parties.

SSH supports username and password authentication as well as the more secure key-based authentication. In key-based authentication, each user has a "private key" and a "public key". The user keeps the private key and authenticates with this key. The public key is pre-installed on the SSH server. When the user connects to the server, the server authenticates by checking if the public key matches with the private key.

SSH not only provides remote command line access, but is also used for file transfer. Tools such as SCP (Secure Copy) and SFTP (SSH File Transfer Protocol) allow to transfer files securely using the secure channel of SSH. This allows sensitive data or important files to be transferred securely.

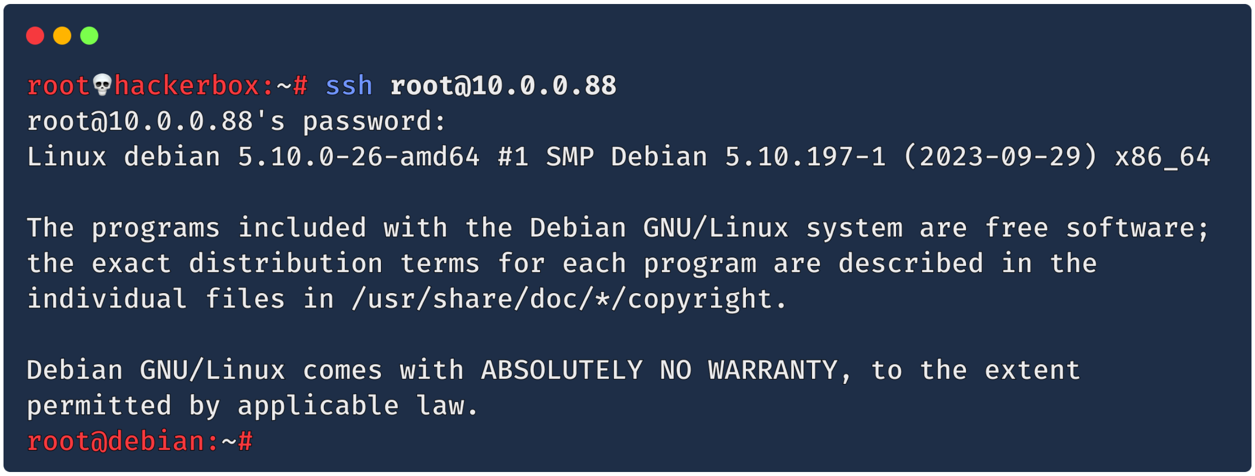
In summary, the SSH service is an important tool for secure communication and data transfer over the network. It enables users to securely access remote systems, transfer files and perform system administration tasks.

The following command string is used to connect to the SSH service.

```
ssh <username>@<hostname or ip address> -p <port-number>
```

Note: We do not need to specify a port number when connecting if the SSH service uses port 22 by default.

Let's try to connect with SSH to a remote computer.



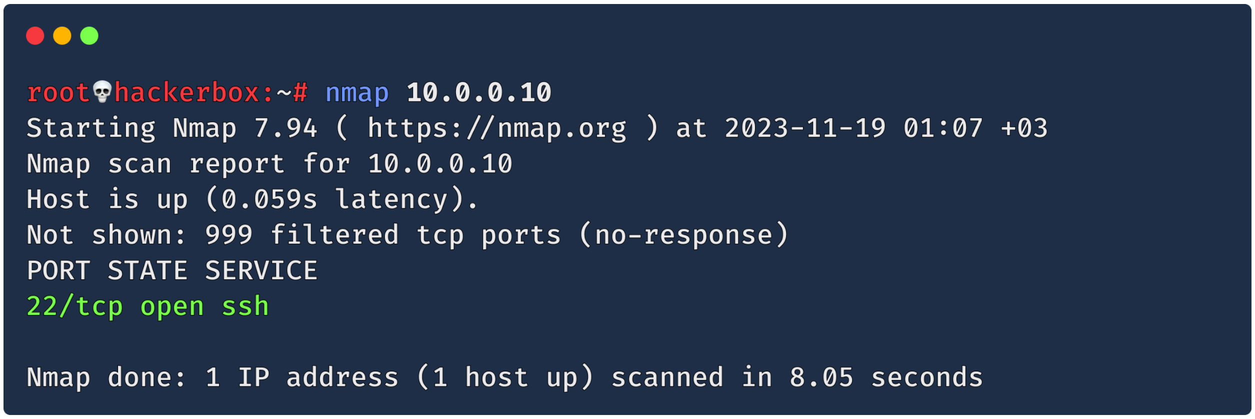
```
root💀hackerbox:~# ssh root@10.0.0.88
root@10.0.0.88's password:
Linux debian 5.10.0-26-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~#
```

As you can see in the example above, we connected to a remote computer with `SSH` service by entering username and password information.

Information Gathering

Let's run a port scan for our target machine.

Task 1, Task 2



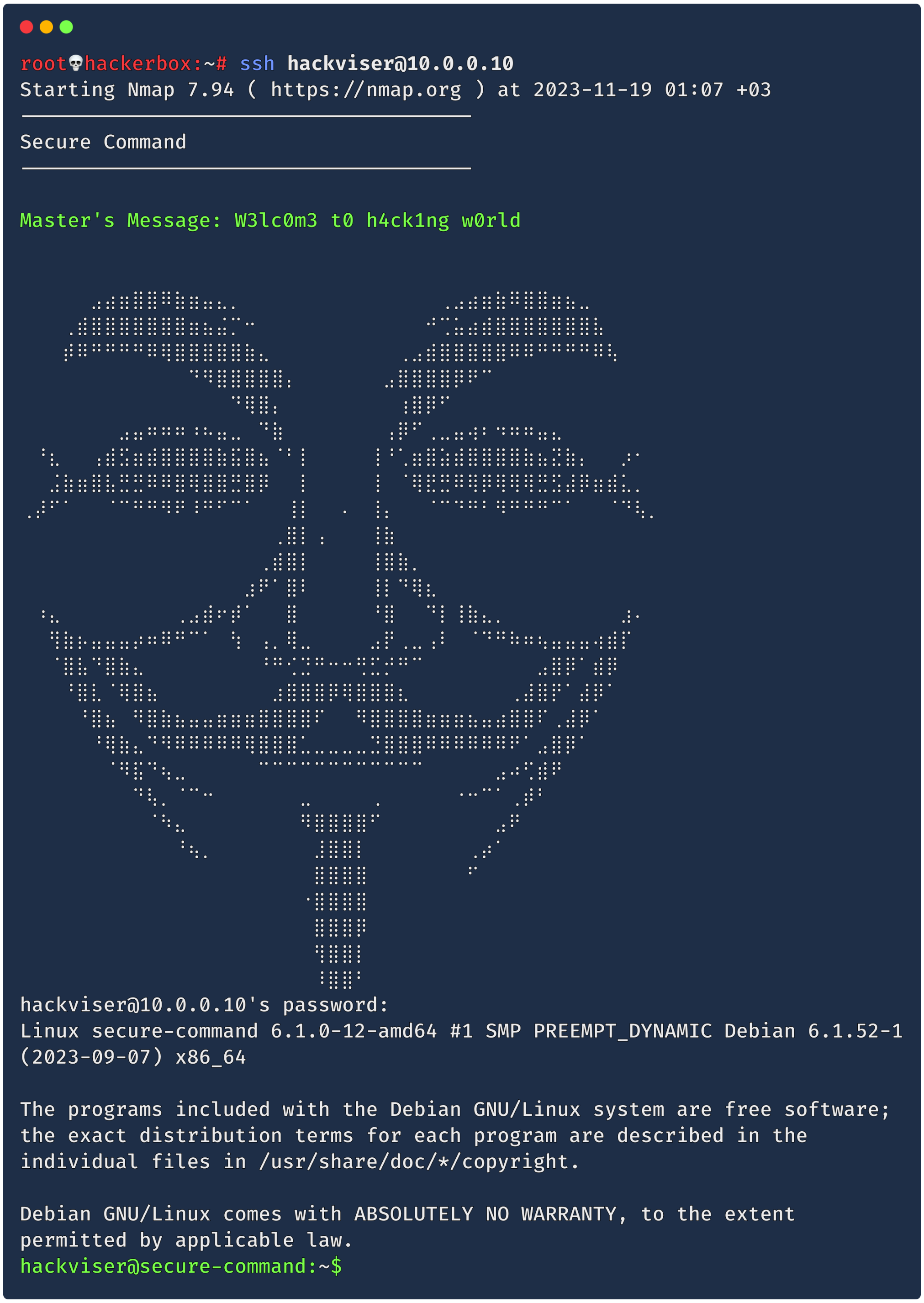
```
root💀hackerbox:~# nmap 10.0.0.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-19 01:07 +03
Nmap scan report for 10.0.0.10
Host is up (0.059s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
22/tcp open ssh

Nmap done: 1 IP address (1 host up) scanned in 8.05 seconds
```

System Access

Let's try to connect to our target machine with SSH.

Task 3

When connecting to our target machine via SSH, we need to use the `hackviser:hackviser` login credentials given to us in the task. When trying to login with this information, we see the `Master's Message`.

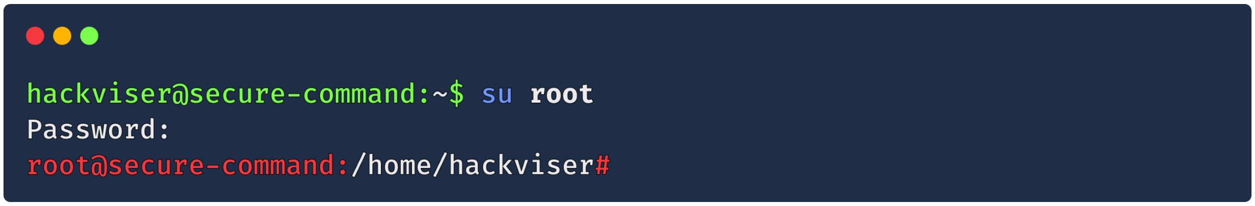


```
root💀hackerbox:~# ssh hackviser@10.0.0.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-19 01:07 +03

Secure Command


Master's Message: W3lc0m3 t0 h4ck1ng w0rld

hackviser@10.0.0.10's password:
Linux secure-command 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1
(2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
hackviser@secure-command:~$
```

At this stage, we are connected to our target machine as a **`hackviser`** user.

Task 4, Task 5

To escalate our privileges on the machine, let's try to switch to the root user, which is the most privileged user. In Linux distributions, the command we can use to switch users is **`su`**. This command stands for "Switch User".



```
hackviser@secure-command:~$ su root
Password:
root@secure-command:/home/hackviser#
```

When we wanted to switch to root user in the machine, it asked us for a password. Trying simple passwords or default passwords, which is a method that can always work, can be useful at this stage.

We tried the **`root`** password when connecting to the root user and this is how we succeeded in becoming the **`root user`**.

Task 6

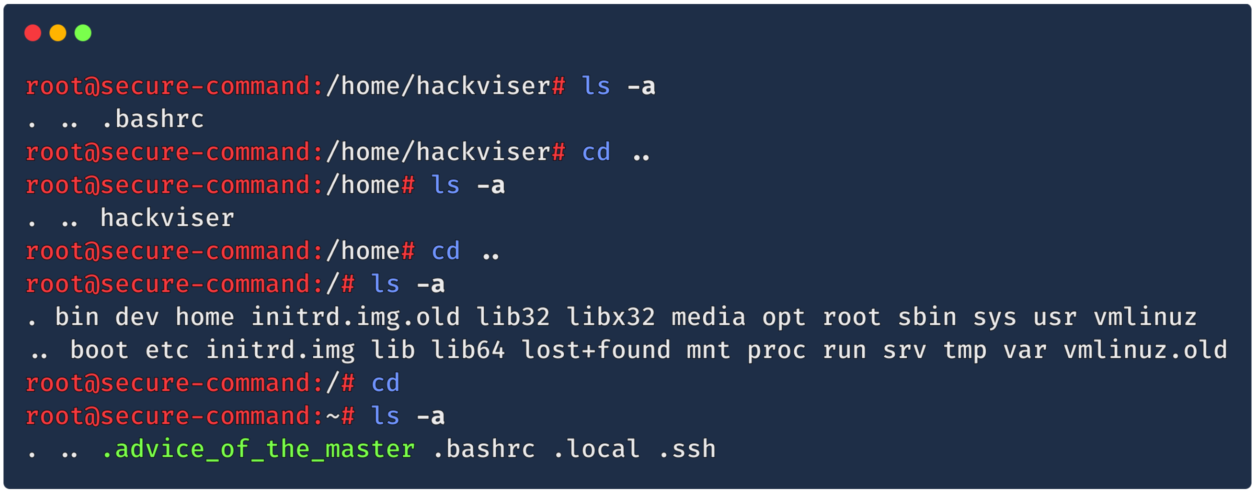
On Linux computers, files or folders with a **`.`** character at the beginning of their name are considered hidden files. When we run the **`ls`** command without the required parameter, we cannot view these hidden files.

To view hidden files, we need to add the **`-a`** parameter to the ls command.

```
ls -a
```

Task 7

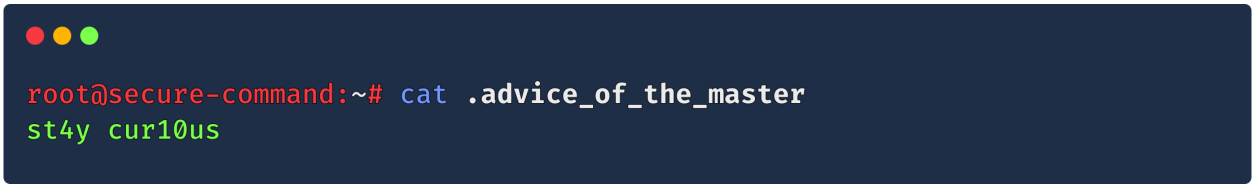
Our task is to get the master's advice. To do this, let's take a little check around the file system and try to catch something.



```
root@secure-command:/home/hackviser# ls -a
. .. .bashrc
root@secure-command:/home/hackviser# cd ..
root@secure-command:/home# ls -a
. .. hackviser
root@secure-command:/home# cd ..
root@secure-command:/# ls -a
. bin dev home initrd.img.old lib32 libx32 media opt root sbin sys usr vmlinuz
.. boot etc initrd.img lib lib64 lost+found mnt proc run srv tmp var vmlinuz.old
root@secure-command:/# cd
root@secure-command:~# ls -a
. .. .advice_of_the_master .bashrc .local .ssh
```

After a bit of browsing around, we found an interesting hidden file named "**`.advice_of_the_master`**" in the root user's **`home directory`**.

Now let's read the contents of this file with the cat command.



```
root@secure-command:~# cat .advice_of_the_master
st4y cur10us
```

-

Congratulations 🙌

✨ You have successfully completed all tasks in this warmup.