

CPE 435: OPERATING SYSTEMS LABORATORY.

Lab11- Part 2.

Introduction to Google Two Factor Authentication.

Submitted by: Dan Otieno.

Date of Experiment: 03/31/23.

Report Deadline: 04/07/23.

Demonstration Deadline: 04/07/23.

Introduction:

The purpose of this lab was to understand how Google's two factor authentication works.

Pretask 1:

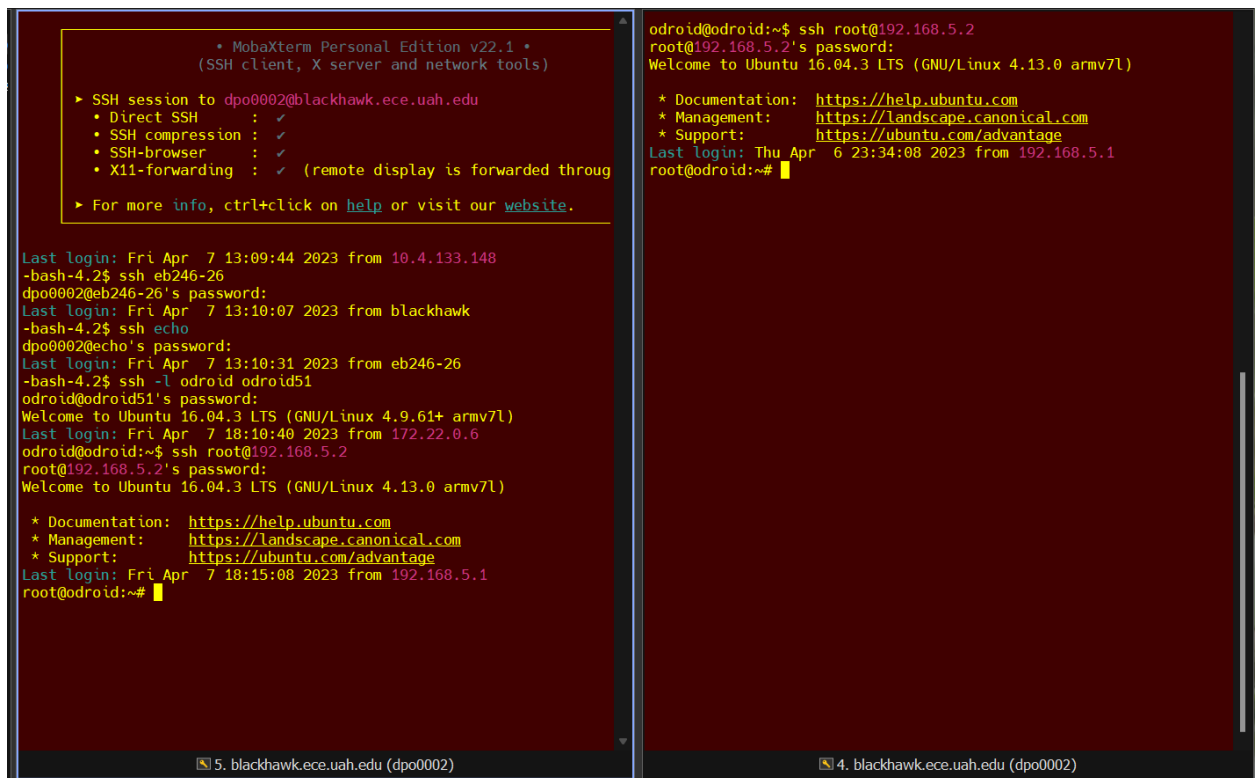
1. How many possible protection strategies can you think of in order to protect the system that you hacked into during the last lab assignment? Please think about at least two such strategies. More than two are always welcome. You do not have to be specific about any product or tool, but just think of the possible strategies that can be adapted to protect from an intruder (which you were in last assignment)
 - a. **Firewall: Firewall restricts unauthorized access to a device by verifying the safety of network data coming into it.**
 - b. **Using SSH (secure shell) to generate a key that allows access to the system.**
 - c. **Authentication software (eg. Microsoft or Google authenticator), that allow for 2-factor authentication setup to keep out hackers.**

Results & Observation:

Subtask 1:

Description:

1. Open a terminal. Log into echo and log into your odroid machine using the credentials given to you. This terminal will be called First Terminal.
2. Log into the root user of the guest using ssh.
3. Open a second terminal. Log into echo, log into your odroid machine and log into guest as root. This terminal will be called Backup Terminal. All your following operations will be performed in the previous First Terminal unless specifically mentioned, but you will leave Backup Terminal open through the entire duration of this lab session.



```
* MobaXterm Personal Edition v22.1 *
(SSH client, X server and network tools)

> SSH session to dpo0002@blackhawk.ece.uah.edu
  • Direct SSH      : ✓
  • SSH compression : ✓
  • SSH-browser     : ✓
  • X11-forwarding  : ✓ (remote display is forwarded through X server)
> For more info, ctrl+click on help or visit our website.

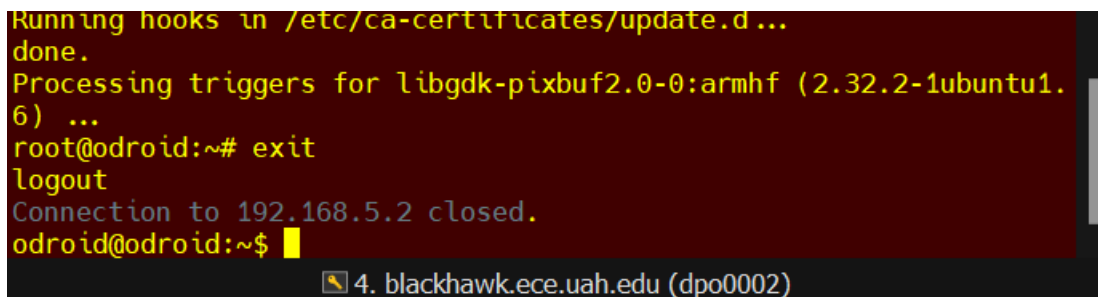
Last login: Fri Apr  7 13:09:44 2023 from 10.4.133.148
-bash-4.2$ ssh eb246-26
dpo0002@eb246-26's password:
Last login: Fri Apr  7 13:10:07 2023 from blackhawk
-bash-4.2$ ssh echo
dpo0002@echo's password:
Last login: Fri Apr  7 13:10:31 2023 from eb246-26
-bash-4.2$ ssh -l odroid odroid51
odroid@odroid51's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.9.61+ armv7l)
Last login: Fri Apr  7 18:10:40 2023 from 172.22.0.6
odroid@odroid:~$ ssh root@192.168.5.2
root@192.168.5.2's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0 armv7l)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
Last login: Fri Apr  7 18:15:08 2023 from 192.168.5.1
root@odroid:~#
```

4. In your First Terminal as root in guest

1. Make sure that you are in your root user account on the guest. Perform the following command in terminal as a sudoer: **apt-get install libpam-google-authenticator**. This will install google authenticator in your machine.

5. Although you have installed two factor authenticator in your guest machine, you can still run attacks since we have not configured it to work. Log out of the root user and Log back in to see it for yourself. **No changes seen here.**



```
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for libgdk-pixbuf2.0-0:armhf (2.32.2-1ubuntu1.6) ...
root@odroid:~# exit
logout
Connection to 192.168.5.2 closed.
odroid@odroid:~$
```

```

odroid@odroid:~$ ssh root@192.168.5.2
root@192.168.5.2's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Fri Apr  7 18:19:16 2023 from 192.168.5.1
root@odroid:~# █

```

Hacking I

6. Open a new terminal. This will be called the Hacking Terminal. Log into your Odroid HOST. Use Hydra attack to attack the second user on the GUEST as you did in the previous lab. Explain the procedure and result. Are you able to hack into the account? Post the screenshot. **The text containing the second user's password is still present in the home directory, therefore, Hydra attack was successful, I was able to access the account, as shown in the screenshot below.**

```

odroid@odroid:~$ hydra -l labdpo0002 -P password.txt ssh://192.168
.5.2 -s 22
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in milit
ary or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2023-04-07 18:36:
25
[WARNING] Many SSH configurations limit the number of parallel tas
ks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 64 tasks, 11 login tries
(l:1/p:11), ~0 tries per task
[DATA] attacking service ssh on port 22
[22][ssh] host: 192.168.5.2  login: labdpo0002  password: odroid
1
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2023-04-07 18:36:
35
odroid@odroid:~$ █

```

7. blackhawk.ece.uah.edu (dpo0002)

Subtask 2:

7. Now we will set up two factor authentication for the guest. And again try to hack into the GUEST using Hydra and the same password file that you created earlier. Make sure you have not changed the password of the second guest. If you have, then make sure to change the password in the password file also.

8. Go to the play store on your phone and look for an application called Google Authenticator and install it on your phone. Read carefully all the instructions that you see while installing the app.

9. Come back to your computer again in the First terminal. Log into the second user account on the GUEST. Once you are logged into the second user on the GUEST, run the following command. You do not have to be sudo for this operation. `google-authenticator` Once you press enter, you will be asked a question something like the following. Answer 'yes' to that. Thereafter you will be presented with a QR code, scan this with your app that you installed previously.

10. Take a screenshot and save it in a safe place. Do not lose it. Your screenshot should contain the six emergency codes also. Save them in a safe place.



```

Your new secret key is: J5FQIXTGMJE2Q5BE
Your verification code is 458379
Your emergency scratch codes are:
  46264844
  22450543
  19748459
  58313619
  66744590

Do you want me to update your "/home/labdpo0002/.google_authenticator" file (y/n) y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) y

By default, tokens are good for 30 seconds and in order to compensate for
possible time-skew between the client and the server, we allow an extra
token before and after the current time. If you experience problems with poor
time synchronization, you can increase the window from its default
size of 1:30min to about 4min. Do you want to do so (y/n) n

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting (y/n) y
$ █

```

11. You will be presented with a series of Yes/No questions. (See screenshot right above).

1. How many questions were there? **4 Questions.**
2. Describe each one of them in at least two sentences for each. (What each does and why is it important?) *********

Q1: Prompts to allow Authenticator configuration in the user's home directory. This would be achieved by updating the file certificate using the authentication tokens.

Q2. Prompts to disallow reusing a token multiple times. I picked yes for this question because generating a fresh token with each login ensures more system security.

Q3. Prompts to increase time allowed for verification to be completed using a token. This would increase the default window to 4 minutes to compensate for poor time synchronization.

Q4. By enabling rate-limiting, a computer that is not configured to handle brute-force hacking attempts can be further secured. This restricts login attempts to 3 within 30 seconds.

12. After you scan the QR code in your phone, you should get a number in your app. Put a screenshot of your code. (Google Authenticator DOES NOT allow screenshots from the app due to security reasons, that is the pop-up I got when I tried to take one).

13. You can change the username@machine_name to be something that is easy to remember in the app. The six digit number is the verification code that you will enter when prompted to enter while logging into the second user. The circle shows the time limit remaining for the validity of code. At this moment, you have enabled two factor authentication for your second user account in your guest machine. Your root account is still without two factor authentication. This is the configuration that we will be using for this lab.

14. Log out of the second user account in your First Terminal. Try logging back in.

1. What is the difference that you noticed? **No changes observed at this step.**
2. Were you able to log into the second user account? Why/Why not? **Yes, because the authenticator configuration was not completed, within the second user directory.**

Hacking II & III

15. Go to your hacking terminal. You should be on the HOST machine. Use hydra attack to guess the password of the second user on the GUEST machine. Make sure that the text file that you provide has the correct password.

1. Were you able to hack? Why/ Why not? **Yes, because the authenticator was not fully configured.**

16. Log into the root user of the GUEST machine and perform the hydra attack again. Write your results here. **Command not found, Hydra does not exist in the root directory.**

```
odroid@odroid:~$ hydra -l labdpo0002 -P password.txt ssh://192.168.5.2 -s 22
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2023-04-07 19:11:47
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 64 tasks, 11 login tries (l:1/p:11), ~0 tries per task
[DATA] attacking service ssh on port 22
[22][ssh] host: 192.168.5.2  login: labdpo0002  password: odroid1
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2023-04-07 19:11:58
odroid@odroid:~$ ssh root@192.168.5.2
root@192.168.5.2's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Fri Apr  7 19:08:46 2023 from 192.168.5.1
root@odroid:~# hydra -l labdpo0002 -P password.txt ssh://192.168.5.2 -s 22
-bash: hydra: command not found
root@odroid:~#
```

Subtask 3:

17. Although you have enabled the two factor mechanism in the second user, you need to configure the ssh engine to use two factor authentication. So now we will be using the Backup Terminal that you had open and unused for a long time.

18. We will basically edit two files from the root user.

1. Open the file /etc/ssh/sshd_config. In the file find the following statement and change the option no to yes towards the middle of the file. Save and close the file **ChallengeResponseAuthentication yes**

2. Open the file `/etc/pam.d/sshd`. In the file add the following statement in the bottom of the file. `auth required pam_google_authenticator.so nullok`

Save and close the file.

The statement that you just added makes the `pam_google_authenticator` as a required module for authentication for users. The term `nullok` means that it is not required for those users that do not have authenticator enabled. Which users in our case would require the google authentication and which do not in our case? **The second guest user account (labdp0002) would require authentication, but the HOST (Odroid) and GUEST (root) would not require authentication.**

Restart the ssh daemon using the following command: `sudo systemctl restart sshd.service`

Subtask 4:

19. Now go back to First Terminal. Log out of it if you are logged in as the second user on the GUEST. Try logging back in as the second user account. Explain your experience in a paragraph (Verification code is the code in the app). **I was able to log in the second user, however, I had to use the first available code in the app. If I waited for a new code to be generated, I would be looped back to the prompt for verification code. As long as I entered the first available code after entering my password, I was able to log in every time, with no issues.**

```
root@odroid:~# ssh labdp0002@192.168.5.2
Password:
Verification code:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0 armv7l)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
Last login: Fri Apr  7 19:10:49 2023 from 192.168.5.2
$ █
```

Hacking IV

20. Go to the Hacking Terminal. Log into your odroid Host.

21. Use Hydra attack to attack the second user on the GUEST. Explain the procedure and result. Are you able to hack into the account? Why/Why Not? **Login failed, Hydra did not successfully "hack" into the second user account, the authenticator was fully configured to prevent unauthorized login.**


```

root@odroid:~# exit
logout
Connection to 192.168.5.2 closed.
odroid@odroid:~$ hydra -l labdpo0002 -P password.txt ssh://192.168.5.2 -s 22
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2023-04-07 19:35:26
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 64 tasks, 11 login tries (l:1/p:11), ~0 tries per task
[DATA] attacking service ssh on port 22
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2023-04-07 19:35:34
odroid@odroid:~$ █

```

Tampering:

22. Write a short note about the NTP Server.

Network Time Protocol (NTP) is an internet protocol used to synchronize with computer clock time sources in a network. It belongs to and is one of the oldest parts of the TCP/IP suite. The term NTP applies to both the protocol and the client-server programs that run on computers. [SOURCE](#)

23. Go to the Backup Terminal. You should be logged in as root on the GUEST. Set the time in your guest so that it is not equal to the UTC time in your Phone (choose any random time).

1. In your Hacking Terminal type in date to see the UTC time in host (make sure you are logged in to host).

```

odroid@odroid:~$ date
Fri Apr 7 19:40:42 UTC 2023
odroid@odroid:~$ █

```

2. Go to your Backup Terminal (you should be logged in as root in guest), and perform the same operation. What do you see?

```

root@odroid:/# date
Fri Apr 7 19:40:08 UTC 2023
root@odroid:/# █

```

3. To set the date in guest in your Backup Terminal, do the following: date -s "19 APR 2012 11:14:00"

```

root@odroid:/# date -s "19 APR 2012 11:14:00"
Thu Apr 19 11:14:00 UTC 2012
root@odroid:/# █

```

24. Go to the Hacking Terminal and check to see if you can log into the second guest account from the HOST (Normal ssh, not hydra). Explain your experience. **Logged in successfully.**

```

odroid@odroid:~$ ssh labdpo0002@192.168.5.2
Password:
Verification code:
Password:
Verification code:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Fri Apr  7 19:37:44 2023 from 192.168.5.2
$ █

```

25. In the above step, you should be able to login after using Verification code. Google authentication module is time dependent (this means that the UTC time in your phone should be the same as your machine). However, the NTP server in your GUEST synchronizes the time before you log back in again after you change the time.

26. Change the time in your machine so that the NTP server does not synchronize it. For this we have to disable the synchronization. Run the following command as root in guest from Backup Terminal. `timedatectl set-ntp 0` Now change the time to any random time you want. Check if the time changed or not. Try to log into the second user account on the GUEST. Explain your experience. *****

```

root@odroid:/# date -s "19 APR 2015 12:14:00"
Sun Apr 19 12:14:00 UTC 2015
root@odroid:/# █

```

[Make sure you change the set-ntp to 1 once you are done with this]

Finishing:

Here we will get rid of google authenticator essentially undoing what we just did.

27. Go to your Backup Terminal where you should be logged in as root on the GUEST. Edit the file `/etc/ssh/sshd_config` file and undo the change that you previously made. ie change the option `ChallengeResponseAuthentication` yes to `ChallengeResponseAuthentication no`

1. Save and close the file.

28. Edit file `/etc/pam.d/sshd` and comment out the statement that we added. save and close the file.

29. Restart the ssh service.

30. Go to the Hacking Terminal and try to run hydra for the second user on the GUEST. Are you able to run hydra and guess the correct password?

```
odroid@odroid:~$ hydra -l labdpo0002 -P password.txt ssh://192.168.5.2 -s 22
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for ill
egal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2023-04-07 19:59:24
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use
-t 4
[DATA] max 11 tasks per 1 server, overall 64 tasks, 11 login tries (l:1/p:11), ~0 tries per task
[DATA] attacking service ssh on port 22
[22][ssh] host: 192.168.5.2  login: labdpo0002  password: odroid1
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2023-04-07 19:59:35
odroid@odroid:~$
```

31. Go to your Backup Terminal and perform `sudo apt-get remove libpam-google-authenticator`

I forgot to screenshot this part but successfully completed the steps.

Now you can gracefully close all the terminals.