The Department of Electrical and Computer Engineering

The University of Alabama in Huntsville

CPE 435 Lab-11 Part 2

Introduction to Google Two Factor Authentication

In the last lab assignment, we saw how we can hack into a vulnerable account using multiple retries of passwords. But as an engineer with security in mind, we must be able to protect such vulnerable systems.

In this assignment, we will study one such method of protection, namely **Google Two Factor Authentication**. **You are required to take at least one screenshot per each step. Please paste them with description in your report.**

PreTask I

1. How many possible protection strategies can you think of in order to protect the system that you hacked into during 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 intruder (which you were in last assignment). **[10]**

Pluggable Authentication Module, PAM is an authentication infrastructure for user authentication. *Google Two Factor* is an example of PAM. We will install Two Factor Authentication in second user account, that you previously created in Guest machine. This time again we will try to hack into this second guest user account using same strategy we used before and see if we can hack or not.

*Warning: All the task that you perform in this assignment should be performed in a single sitting. Following the exact order of instructions is compulsory. Please do not close any open terminals unless specifically told to. This lab module is designed to follow **Learning By Doing** paradigm, so there might not be so many statements ending in **?** sign. You should prepare the report by keeping in mind that you are preparing a tutorial highlighting the necessity of Two Factor authentication and steps involved in doing so.*

Subtask I (Installation)

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 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.

4. In your **First Terminal** as *root* in guest

   1. Make sure that you are in your root user account in 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.

Hacking I

6. Open a new terminal. This will be called **Hacking Terminal**. Log into your odroid Host. Use Hydra attack to attack the second user in guest as you did in previous lab. Explain the procedure and result. Are you able to hack into the account? Post the screenshot.

Subtask II (Configuring Google Auth)

7. Now we will set up two factor authentication for guest. And again try to hack to 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 play store on your phone and look for 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 **First terminal**. Log into the second user account in guest. One you are logged into the second user in guest, run 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 loose it**. Your screenshot should contain the six emergency codes also. Save them in a safe place.

11. You will be presented will a series of Yes/No questions.

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

12. After you scan the QR code in your phone, you should get a number in your app. Put a screenshot of your code.

14. You can change the *username@machine_name* according to your wish so that what you feel easy to remember can be named 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 accout 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.

15. Log out of the second user account in your First Terminal. Try loging back in.

    1. What is the difference that you noticed?
    2. Were you able to log into the second user account? Why/Why not?

Hacking II & III

16. Go to your hacking terminal. You should be at host machine. Use hydra attack to guess the password of the second user in guest machine. Make sure that the text file that you provide has the correct password.

    1. Were you able to hack? Why/ Why not?
17. Log into the root user of guest machine and perform the hydra attack again. Write your results here.

Subtask III

18. 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.

19. We will basically edit two files from 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?

Restart the ssh daemon using the following command.

`sudo systemctl restart sshd.service`

Subtask IV (Testing)

20. Now go back to **First Terminal**. Log out of it if you are logged in as second user in guest. Try logging back in as second user account. Explain your experience in a paragraph (Verification code is the code in the app)

Hacking IV

21. Go to **Hacking Terminal**. Log into your odroid Host.
22. Use Hydra attack to attack the second user in guest. Explain the procedure and result. Are you able to hack into the account? Why/WhyNot?

Tampering

23. **Write a short note on NTP Server?**

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

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

25. 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.

26. 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 same as your machine). But, NTP server in your guest synchronizes the time before you log back in again after you change the time.

27. Change the time in your machine so that 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 in guest. Explain your experience.

    [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.

28. Go to your **Backup Terminal** where you should be logged in as root in 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.

29. Edit file */etc/pam.d/sshd* and comment the statement that we added. save and close the file.

30. Restart the ssh service.

31. Go to Hacking Terminal and try to run hydra for second user in guest. Are you able to run hydra and guess the correct password?

32. Go to your **Backup Terminal** and perform

    `sudo apt-get remove libpam-google-authenticator`

Now you can gracefully close all the terminals.