


Lab 05 - Authentication in Linux

Objectives

- Authentication protocols
- Linux PAM
- Multi-Factor Authentication

Preparation

You may use the UPB's  OpenStack cloud to instantiate a Virtual Machine. Read these [instructions](#) if you wanna know how!

Overview

PAM (Pluggable Authentication Modules) is a collection of libraries that allows you to decide how you authenticate your users to different applications on your Linux [OS](#).

Tasks

In the current security lab, we will set up a Python-scripted PAM for user authentication, expose its vulnerability by finding the password, and then secure it by adding MFA with Google Authenticator.

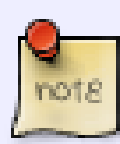
01. [5p] Setup

We use Docker (and we need to hack the cloud networking):

```
sudo vim /etc/docker/daemon.json
{
  "mtu": 1450
}

sudo systemctl restart docker

docker pull ghcr.io/cs-pub-ro/isc-auth-pam:latest
mkdir ~/auth-lab
docker run --rm --name auth-lab -v $(pwd)/auth-lab:/home/hacker/auth-lab -it ghcr.io/cs-pub-ro/isc-auth
```

 The ~/auth-lab folder is used as persistent volume so you won't lose + sync your work inside the container!

Download the lab archive.

Analyse the Python script and the users and groups on the system. What user are we interested in?

02. [5p] Python PAM

Download the latest [deb](#) file from <https://sourceforge.net/projects/pam-python/files> and install it.

Try to download it from inside the container using `wget` ;)

03. [20p] Linux PAM


Modify a single Linux PAM configuration file (look in `/etc/pam.d`) so that *authentication* is done using the Python module with our script.

Hint: Since the Python script is not done, it should be *sufficient* to authenticate using it, but not *required*.

References:

- <https://pam-python.sourceforge.net/doc/html>
- <https://docs.oracle.com/cd/E19253-01/816-4557/pam-15/index.html>

04. [30p] Python script

 Because of the pam_python PAM module, we need to use the obsolete Python 2.7.

Fill in `TODO(1-5)`. You'll know it's correct if you get the correct prompt.

References:


- <https://pam-python.sourceforge.net/doc/html>
- <https://docs.python.org/2.7/library/grp.html>

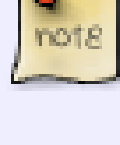
05. [10p] Exploit

Log in to the user account.

06. [30p] Multi-Factor Authentication

Install the needed dependencies using `sudo pip2 install pyotp==2.3.0 pyqrcode`.

 We need to use an older version of pyotp because of Python2.7.

 The packages need to be installed as root because the Python script will be run as root for authentication.

Solve `TODO(6-9)` and log in to the account using Google Authenticator.

References:

- <https://pyauth.github.io/pyotp/#module-pyotp>
- <https://pyqrcode.readthedocs.io>

Lectures

- Lecture 01 - Introduction
- Lecture 02 - Cryptography
- Lecture 03 - Hardware Security
- Lecture 04 - Access Control
- Lecture 05 - Authentication and Key Establishment
- Lecture 06 - Application Security
- Lecture 07 - Operating System Security
- Lecture 08 - Network Security
- Lecture 09 - Web Security
- Lecture 10 - Privacy Preserving Technologies
- Lecture 11 - Forensics

Labs

- **kernel**
 - Lab 01 - Introduction
 - Lab 02 - Cryptography
 - Lab 03 - Authentication and Key Establishment Lab
 - Lab 03 - Hardware Security
 - Lab 04 - Access control
 - Lab 05 - Authentication in Linux
 - Lab 06 - Application Security
 - Lab 07 - Operating System Security
 - Lab 08 - Network Security
 - Lab 09 - Web Security
 - Lab 10 - Forensics
 - Lab 11 - Privacy Technologies
 - Lab 12 - Security and Machine Learning

Support

- Useful resources
- Virtual Machine

Table of Contents

- Lab 05 - Authentication in Linux
 - Objectives
 - Preparation
 - Overview
 - Tasks
 - 01. [5p] Setup
 - 02. [5p] Python PAM
 - 03. [20p] Linux PAM
 - 04. [30p] Python script
 - 05. [10p] Exploit
 - 06. [30p] Multi-Factor Authentication