MIECT: Security 2015-16

Practical Exercise: Pluggable Authentication Modules

November 18, 2015 Due date: no date

Changelog

- v1.0 - Initial Version.

1 Introduction

The goal of these exercises is to explore the functionalities of PAM (*Pluggable Authentication Modules*) infrastructure present in current Linux distributions.

These exercises will also make use of the Portuguese Citizen Card as a way to authenticate users and to provide detailed logs.

Use the virtual machine that was provided in the beginning of the classes. If you need you can download a new one from:
`http://www.joaobarraca.com/page/teaching/security/2015/`

Errors in PAM configuration files may block further access to the system. Beware! Do not use your own system!

2 Weak Passwords

By default, current Linux distributions don't verify the strength of the passwords in use. A weak password is a password which can be found through an intelligent search attack where a set of passwords are tested (dictionary attack). The widespread availability of dictionaries with words and even well known passwords, makes this attack very simple to be executed. Once the password is found, the security of the entire system is compromised as it can allow the execution of local exploits and access private data.

This policy can be changed so that the password modification procedure verifies the strength of the keys in use. (In Linux) These procedures are controlled using PAM (*Pluggable Authentication Modules*), which is also used to control several other authentication and access control aspects of any Linux system.

Install `libpam-cracklib` using `apt-get`.

Take a look at the manual of pam_cracklib (`man pam_cracklib`) and understand the purpose of the module.

Create a test user named `secuser` and set a password for this user. Verify that you can login into his account.

The file `/etc/pam.d/common-password` stores the rules controlling how users can change passwords. This is a good candidate for enforcing strong passwords and is typically used for that purpose.

- Propose a reasonable model for password security and enforce it. Consider password length, existence of symbols, lower case characters, upper case characters and digits.
- In order to better enhance the difficulty of finding the password, enforce a policy so that common passwords found in the system dictionary are forbidden. You can use the dictionary files available at `/usr/share/dict`. In the current Debian distributions this is done automatically. Please check the files in `/var/cache/cracklib` and `/etc/cracklib`.

Using the user recently created, test the correct enforcement of the password safety model you implemented.

In the end, please restore all PAM-related configurations! Otherwise you may be unable to log in into the system.

3 One Time Passwords

One Time Passwords (OTP) can also be used for authenticating users. As expected, PAM supports OTP by means of a library and a definition in the configuration files. In Ubuntu systems, it is required to install the packages `libpam-otpw` and `otpw-bin`.

The next step is to add the `pam_otpw.so` module to the approporiate place in the PAM authentications stack. For the purpose of this laboratory guide, add it to the `common-auth` file, as an alternative authentication method besides the ones already existing:

```
auth sufficient pam_otpw.so
```

You should take in consideration the the order is important! Place `pam_otpw.so` after `pam_unix.so`. Also, the usage of `pam_unix.so` should also be declared as `sufficient`.

After PAM is configured for using `pam_otpw.so` users can choose to use OTP by running the `otpw-gen` command. This will generate a file named `~/.otpw` containing some metadata, as well as a list of one-way hashes for the purpose of verifying the response to challenges.

Run the command in order to create the file. You will need a password prefix. Choose any string and remember it. The prefix will be required for authentication purposes. Also a table will be generated. This table provides the responses to the several challenges iniciated by the PAM OTPW module. In the real world these challenges would be printed to a paper ou a card.

To test if the system is working, as `root` execute the `login` command. Provide no password for the first password prompt. You sould be presented with a new password prompt carrying a number. The correct answer is composed by the prefix and the challenge. If multiple numbers are provided, you must provide the answer to all challenges. Spaces are ignored.

4 Authentication using Smartcards

An example source code in C of a PAM module implementing local authentication with the CC can be found at:
`http://code.ua.pt/projects/ccpam`

Before you can compile the module, you should install the PAM library development files (`libpam0g-dev`) using `apt-get`.

Compile the module by executing `make` and install it with `make install`. Edit the PAM configuration file `/etc/pam.d/common-auth` and add the line `auth sufficient pam_PTEIDCC.so /etc/CC/keys` as the first rule in the authentication stack.

Using the `addCCuser` tool bind a Citizen Card (C) to the test user (the default file is `/etc/CC/keys`).

Verify that you can use the CC to authenticate the test user.

Modify the library so that the actual name and BI of the user are logged to the `auth.log` file. You will have to consult the CC SDK documentation in order to determine the relevant PTEID API functions to use.

The following code may be helpful to log messages to syslog:

```
#define PACKAGE "pam_PTEIDCC"

static void pam_cc_syslog(int priority, const char *format, ...) {
        va_list args;
        va_start(args, format);

        openlog(PACKAGE, LOG_CONS|LOG_PID, LOG_AUTHPRIV);
        vsyslog(priority, format, args);
        closelog();
        vfprintf(stderr, format, args);
}
```

Listing 1: Helper Function to write a string to syslog

Usage:

```
pam_cc_syslog(LOG_ALERT,"Name: %s BI: %u\n","CARLA VALIDO", 1234);
```

Listing 2: Example usage of the helper function

Once again, verify that this information is indeed present in the system auth log file: `/var/log/auth.log` .

Note: The test cards fail the integrity verification. In order to use these cards with the PAM library, you must remove the call to the functions `PTEID_SetSODCAs` and `PTEID_SetSODChecking` in the `addCCuser.c` and `pam_PTEIDCC.c` files.

5 References

- Portuguese Citizen Card web site: http://www.cartaodecidadao.pt
- Linux PAM web site: http://www.linux-pam.org
- Linux PAM OTPW web site: http://www.cl.cam.ac.uk/ mgk25/otpw.html