Implementing a Secure Authentication System

BRUNO CARPENTIERI
Dipartimento di Informatica
Università di Salerno
Via Giovanni Paolo II
ITALY
bc@dia.unisa.it

Abstract: One of the most used techniques for ensuring the safety of a system is authentication. The login/password mechanism that is widely used for authentication on PC or on general purpose computer systems it is not always adequate to ensure security in the authentication phase.
In this paper we show the implementation of a secure authentication system based on One Time Passwords and Face Recognition. The system has been implemented in C language on a Unix like environment and it is currently used and tested in our data compression lab in Salerno.

Key-Words: - Security, Operating Systems, Unix, One Time Passwords, Face Recognition.

1 Introduction

A system is safe if its resources are used and accessed only in the manner prescribed by a fixed set of access rules.

One of the most used techniques for ensuring the safety of a system is authentication. Authentication might be based on one or more elements:

- It might be based on the ownership of real objects: for instance access keys or magnetic cards.
- It might be based on the user's knowledge by using, for instance, user names and passwords.
- It might be based on physical attributes of the user such as fingerprints, facial recognition and biometric data in general.

The password mechanism, if it is not used jointly with other authentication techniques, it is not always adequate to ensure security in the authentication phase (see [1]). The main reasons for this inadequacy do not reside in the mechanism itself but in the way the passwords are managed by the agents that use them: the network protocols and the users.

When a network user accesses a remote computer he must be previously authenticated and his password is first encrypted and then it is sent via the network to the remote computer that shall check the validity of his UserID - password pair.

To beat this mechanism hackers have developed programs (*sniffers)* that enable an intruder to intercept and decrypt these passwords while they travel on the network so to use them to illegally access the network resources.

On the other hand, the users that are required to use passwords to access computing resources often choose easy-to-remember (and easy-to-guess) passwords and it is easy to develop computer programs that use specific dictionaries to try to break these passwords.

To address these problems other authentication mechanisms have been developed to make more secure the authentication phase (see for example [2], [3], [4], [5]).

These mechanisms are mainly based on the possess by the user of devices such as smart cards or pocket calculators that are able to generate a one time password that identifies the user properly.

These systems are called One-Time Pad or One-Time Password Systems and they are characterized by the fact that the password is never reused.

Consequently, the intruder has no historical information on which he can guess the new password. The one-time password systems therefore avoid the problems arising from the possible use of sniffers. The principle on which this mechanism is based is that each user of a system is provided of a list of passwords and each time you connect to a computer using a password on the list then that password is erased. The next connection shall need the next password on this list.

There are different implementations of this mechanism and some of them require the aid of appropriate devices to automatically execute the authentication function that produces the next password.

Another important authentication strategy can be based on biometrics. The use of biometrics requires a biometric recognition system.

2 System Architecture

Our combined approach to a secure authentication system consists of two modules that have the purpose of allowing only secure accesses to the system: the access is guaranteed only if both modules provide a positive feedback on the user.

When a new user registers, the system generates a series of N password (also called *code book*), these passwords will provide at most N future logins.

A new file containing the passwords is created and associated to the new user name. A copy of this file is given to the user.

When a user logs in, he provides user name and the first password of his code book. If the password is accepted then both sides, the system and the user, eliminate that password occurrence in their code book.

In our system, the one-time password module is implemented in C language for UNIX environments with standard libraries.

The purpose of the second module, called the face detection module, is to provide access only if the system recognizes the presence of a human user being in proximity to the associated camera.

The human face has very specific characteristics that have been studied widely, therefore we can used well tested algorithms for face recognition.

It is possible to distinguish two possible types of usage of this module in the system:

- To memorize in an information log file a snapshot from the camera containing a face, by allowing the access if a human face is recognized.
- To identify the user by using biometric informations, so to allow the access only if the face has been univocally identified as the user's face.

The Face Detection module is realized in C language for UNIX environments with image processing libraries *Intel Integrated Performance Primitives v5.3 for Mac OS*. The code runs also on other Unix like operating systems like Linux (where it is necessary only to recompile the source code).

Our System Architecture is pictorially described in Figure 1.

3 Project Specifications

The authentication system has been implemented and tested. Here we discuss the main issues of its implementation.

3.1 One Time Passwords

Upon registration in the system, each user is given a user ID and a USB device containing his code book. It is the user's responsibility to preserve the privacy of the device and to remember his username.

The system has available a number of resources in which he stores the data relating to the registered users:

- The text file /etc/Authenticate/ElencoUtenti.dat: contains the list of registered userid
- /etc/Authenticate/UserID/libroMaster.dat is structured in directories where each UserID is associated with one or more code books. It is necessary to maintain information also on the previous code books, so each user is associated with a directory.

Figure 2 shows the flow chart of the one time password module.

3.1.1 The Check Registration sub-module

The user must have one UserID that he acquires when he registers. This UserID is coupled to a personal profile, that might include also pictures of his face, via a data base.

The system has a text file containing the UsedIDs of all the registered users at the address:

/etc/Authenticate/ElencoUtenti.dat.

The submodule is built with two management functions.

- CheckUser (): Operates in contact with the user. It is called from main ().
- FindUser (): Work within the system searching for the presence of the user ID, it is called from CheckUser ().

3.1.2 The One Time Password sub-module

This part is the core of the One Time Passwords module because it implements the password management policy.

It uses two important resources: the system's and the user's code books containing the passwords that have to be matched.

The user's code book is available directly in the root directory of the user's USB device. The operating system auto-mounts the device in /Volumes/.

Having recognized the USB device, the system applies a sequential search algorithm comparing the first password in the code book of the client with the passwords enabled by the server.

If the first password is recognized as valid, the user gets access to the system and the password is deleted from the USB device and from the system's code book. Otherwise, if the password is not accepted, it might reveal a malicious attempt to login by a non authorized user or that the user's code book has been corrupted due to independent external factors.

In both cases the access is not authorized and the user is asked to contact the system administrator that eventually shall provide a new code book.

3.2 Face Detection

This module is in charge of detecting regions of the image containing a human face. If a face is found it records the entire frame on disk (for a possible post login analysis).

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in [6].

The approach in OpenCV stretches from the original studies of Viola and Jones, later modeled and extended by Lienhart (see for instance [7]).

Two API are provided, for high or low level face recognition.

The openCV classifier using Haar Cascades is stored in XML format and can be invoked through the procedure *cvHaarDetectObject*.

3.2.1 False Positives

In low-light environments or for digital frames acquired with capture devices that introduce noise in the image, it might occur the phenomenon of false positives in which some areas of the image, while not really presenting human faces, are recognized as such.

To avoid this problem we can take various countermeasures, as for example:

- **Act on the cascade**: Adapt the cascade model to the type of video shooting. Normally the production of the cascade takes place in a time and at a different location from the environment where the application will be executed, and this can often lead to errors induced by the quality of the device. One type of solution consists in producing the test set directly into the environment of application. This solution leads to a reduction of false positives but it is not always a viable solution because it might require a time disproportionate compared to the program's life cycle.

- **Background subtraction**: If it is possible to detect a static background, then by subtracting this static background to the frame in which we search for faces brings to excluding areas that are definitely not interesting and reduces the possibility of false positives.

- **Check the radius**: Eliminate ray observations that are not compatible to face dimensions. Recognition leads to the production of circles with center in the center of the faces and radius equal to average size of the contour. Often the radius of false positives is of different size with respect to the estimated viable possible radiuses.

4 Conclusions

In this paper we have described the implementation of a secure authentication system based on One Time Passwords and Face Recognition.

The system has been positively implemented by using the C language on a Unix like environment and it is currently used and tested in our data compression lab in Salerno.

We are currently working on improving the system by adding new features, like a better face recognition system, a possible third phase in which, when the identification is not 100% secure because of artifacts that do not allow an effective and unique face recognition result, then the system interacts with the user in order to ask him questions that will certify his/her identity.

A newer implementation of the system by using the Python programming language is also on its way.

4 Acknowledgements

We need to thank Carlo Santoro, Domenico Procaci and Fabrizio Bentivoglio, that were students of my Operating Systems class in Salerno, for implementing, testing and debugging over the years parts of this authentication system.

References:

[1] C. Herley, P. C. Oorschot, and A. Patrick, Passwords: If We're So Smart, Why Are We Still Using Them? International Conference on Financial Cryptography and Data Security, pp 230-237, Springer, Berlin, Germany, 2009.

[2] A. De-Angeli, M. Coutts, L. Coventry, G. Johnson, D. Cameron, and M.-H. Fischer, "VIP: a visual approach to user authentication," in Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 316–323, Trento, Italy, 2002.

[3] L. Y. Por, Mitigation of Shoulder-Surfing Attack on Picture-Based Passwords Using Falsifying Authentication Methods, Faculty of Computer Science and Information Technology, University of Malaya, 2012.

[4] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "PassPoints: design and longitudinal evaluation of a graphical password system," International Journal of Human Computer Studies, vol. 63, no. 1-2, pp. 102–127, 2005.

[5] Sang Guun Yoo, "E- SAS: Enhanced Secure Authentication System for Healthcare Applications using Wireless Medical Sensor Networks", WSEAS Transactions on Systems, pp. 309-320, Volume 15, 2016.

[6] Paul Viola and Michael Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*.

[7] Rainer Lienhart and Jochen Maydt, An extended set of Haar-like features for rapid object detection, *Proceedings of the International Conference on Image Processing. 2002. ICIP 2002*.

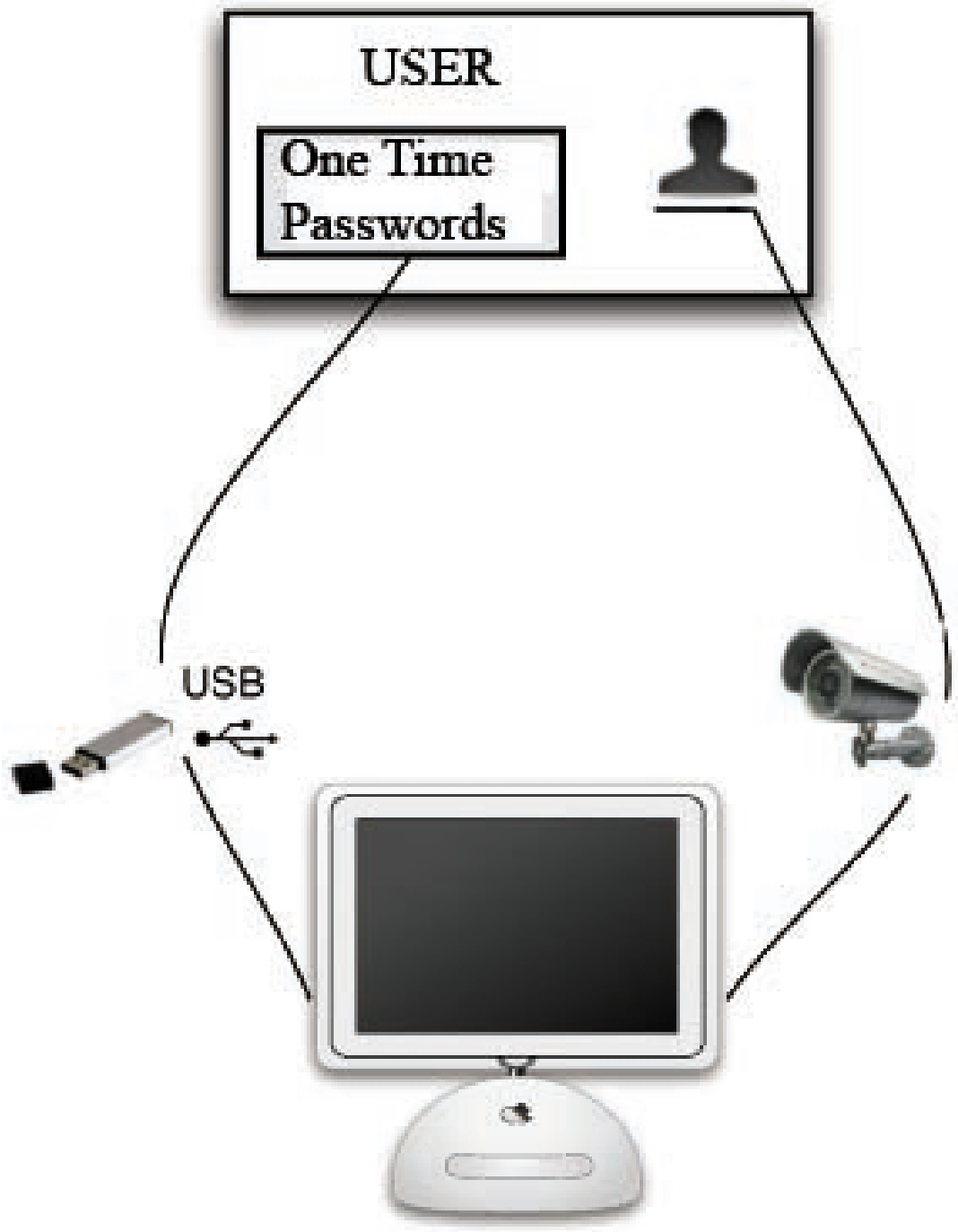

Figure 1: System Architecture

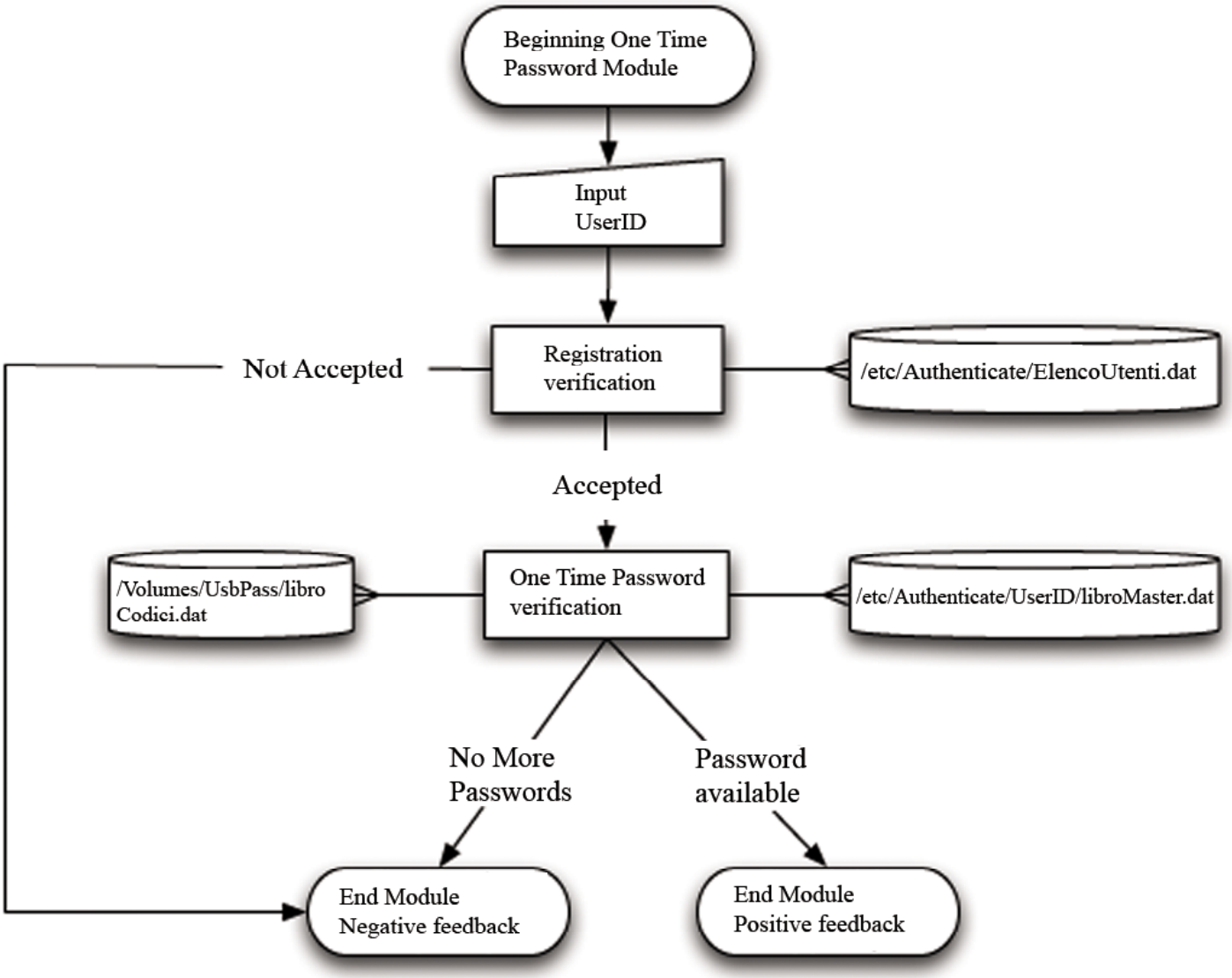

Figure 2: One Time Password Module