Security in the PASTA project

Tage Stabell-Kulø
Department of Computer Science
University of Tromsø
e-mail: Tage@ACM.org

CS-TR 96-27
October 15, 1996
Revised February 4, 1997

Abstract

This report presents the system model for the security effort in the PASTA project. We present the objectives of the security effort, the threats we will consider, and those we will not consider. As such, the report describes the environment which applications must be prepared to face, and still provide users with the desired degree of privacy.

1 Introduction

The PASTA project is concerned with the introduction of personal, portable machines in a distributed system. The project studies a wide range of issues that arises from this point of view.

In PASTA we plan to investigate how users can ensure that access to data entrusted to the File Repository (called FR, see below) is granted in accordance with the owners policy. This report describes the overall guidelines for this endevaur.

We start out, by presenting our objectives. The threats we consider, some we explicitly do not take into account and the assumptions we make are presented in Section 3. In Section 4 we give a short presentation of our research vehicle: the FR. We sum up in Section 5.

2 Objectives

In PASTA, we assume widespread usage of personal machines. This must be reflected in the way security is investigated. An ever increasing part of the machines in modern systems is under the full control of their owners, and official policies for security—or on any other issue, for that matter—can thus only be applied to a shrinking part of the overall resources. In particular, as computing becomes personal, the corporate model (one central agency that can enforce a security policy) of computing becomes out-dated. It is a goal of this project to understand how this development will effect issues related to security. And, since computing in our view is a personal activity, privacy will predominately have our attention. In particular, we will investigate whether it is possible to build a distributed system that enables each and every user to have, and enforce, his own security policy. Each user must make his own decisions regarding security and privacy. There will definitely not be a single, system-wide security policy at any level.

Based on this line of argument, we have established the following as the objectives of the security effort:

- It must be possible to formulate a trust relation, and to verify that a particular relation holds.

  Trust (and trust relations) is one of the fundamental building blocks of any system that provides security. In order to give each user the ability to maintain and enforce his own security policy, the system must provide the means for each user to distinguish those subjects he do trust from those that he do not trust.

- Any user must, in isolation, be able decide upon whom to place trust.

  It follows from the above that there can be no "trusted third party" in this system; at least, not one that everybody must trust. On the other hand, any set of users must be able to establish a server they trust to do (or not to do) something. The infrastructure must make it possible to fully exploit such a (trust) resource. Naturally, not placing trust in providers of important services may lead to denial of service, but that is an different issue altogether.

- It must be possible to delegate to someone (a principal) some authority over some object.

  Communication between any two users may be direct, in which case they themselves decides on the policy they want to enforce. If they choose to communication through FR, the server(s) involved should enforce the policy the users specify.

- The system should be structured so that other types of secrets than shared or public encryption keys may be used to identify a channel as originating from a particular user or server.

- Every part of the system must be structured so that it is possible to understand the effect of the compromise of any secret.

  In other words, only those who have chosen to place trust in something (a secret, for example) are hurt if the secrecy is compromised. It is thus an objective to minimize the threat to security that is caused if the secrets of a principal are compromised.

In general, we believe that decisions related to security should be visible and that trust relations should be explicit.

3 Threats and Assumptions

Throughout our investigation the following is asserted to be true:

- The confidentiality of any datum kept within a users machine is only so good as the ability of that particular user to protect it.

  In general, users have their own, personal machines. This implies that the user is fully in control of the machine, and also that they are fully responsible for the maintenance of the security features of the machine. There is a clear distinction between the users intentions and their actual ability.

- In general, one can not trust a machine, with the possible exception of one's one.

  In particular, it might be difficult to trust a certificate which, for example, states "machine X says user U is logged on", without any further proof of this claim.

- Any datum sent across a computer network can be seen, altered, withheld and/or replayed by some malicious third party.

- One can not trust the network's notion of the origin of any datum.

On the other hand, we also make two simplifications:

- Communication is either possible (connected) or not at all possible (disconnected).

  This does not imply that messages are assumed to arrive in its entirety since communication links may break.

- We assume encryption is perfect.

  Perfect encryption holds both for shared-key encryption systems such as IDEA and DES, and public-key encryption systems such as RSA. This includes the impossibility of generating a message that matches a given checksum provided by algorithms such as MD5.

  Only the holder of the correct key can obtain any information about the contents of an encrypted message. Naturally, we do not assume that only legitim users have access to keys, machines can be compromised, for example. Perfect encryption can be viewed as a "black box approach."

  We will, however, be careful to avoid interference between different cryptographic tools.

Discarding the assumption on perfect encryption might be natural extension of the project at a later stage.

4 File Repository

When users utilize more than one machine, they experience a consistency problem. If some of these machines are portable, and therefore often disconnected, this can be a nuisance. Within PASTA, we are concerned with building an infrastructure to ease these problems.

FR provides services to users through a set of servers. These cooperate, and maintain a distributed repository in which users can store files. Research into the area of replication protocols and policy is part of the project at large, and will not concern us here. We assume that any datum can be replicated if need be, with the limitations that arises from an unsecure and possibly malicious environment.

Clients, e.g., users' machines, interact with servers by means of some standard Internet communication protocol. In practice, this protocol will be the TCP/IP protocol suite, but we do not believe this choice has any security implications.

About the scale of the system, we assume the following:

- A relatively small set of "sites" will together support a FR connected through a wide-area network. In the order of ten sites will participate.
- In general, we envision that each site will run a (small) set of servers, we assume in the order of three servers per site.
- Associated with each set of servers, there will be a number of users. There will be in the order of one hundred users per site.
- Every user will own more than one machine. One of them will presumably be a high performance workstation, and one will probably be a small, mobile, palm-top-sized computer.

We make these numbers explicit, but security must not be violated even if any of them are grossly under (or over) estimated. However, we allow performance to depend on them.

5 Conclusions

In this report we have outlined the functional requirements of a security model for FR. The main point is that we aim not for a fully secure system, but one in which each user may decide, in isolation, which risks he is prepared to take. Work on the different subareas of the project at large should adhere to these requirements.

6 Acknowledgments

Arne Helme provided valuable input; to the security effort in PASTA in general, an to this report in particular. Alberto Bartoli helped with the presentation.

The Pasta project has received funding from several sources, mostly as support for smaller, focused, sub-projects. Among these are the MobyDick project (Esprit-IV LTR 20422) and the "Global Distributed Diary" project (Research Council of Norway project number 112577/431).

The development of the ideas that are fundamental in Pasta, was partially funded by "The Radical Initiative" of CaberNet, Esprit BRA 6568; it is gratefully acknowledged.