CONF-940873--5

PNL-SA-24744

IMPLEMENTING A SECURE CLIENT/SERVER APPLICATION

B. A. Kissinger

August 1994

Presented at the
11th Office Information Technology Conference
August 23-25, 1994
Chicago, Illinois

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

Implementing a Secure Client/Server Application

Bruce A. Kissinger, Senior Research Scientist, Pacific Northwest Laboratory

Abstract

There is an increasing rise in attacks and security breaches on computer systems. Particularly vulnerable are systems that exchange user names and passwords directly across a network without encryption. These kinds of systems include many commercial-off-the-shelf client/server applications. A secure technique for authenticating computer users and transmitting passwords through the use of a trusted "broker" and public/private keys is described in this paper.

Introduction

Recent news headlines have reported the increased rise in attacks on computer security by hackers utilizing "sniffer" software to capture user names and passwords from applications that transfer this information in clear-text. For instance, Michael Higgins, Deputy Director for Information Systems Security at the Defense Information Systems Agency (DISA) was quoted in the July 11, 1994 issue of *Federal Computer Week* as saying in recent attacks that "a major portion of the international commercial service infrastructure [was] compromised affecting the U.S. National Information Infrastructure." He added that "... major portions of the Defense Information Infrastructure were [also] compromised, adversely affecting DOD military readiness." The same article said that "...the number of captured passwords has now climbed into the million-plus level and that the sniffer attacks continue." "The hacker attacks have reached such a scale over the past few months that on any given day DOD literally does not have control of five or six of its computer systems; the hackers do." These attacks are not limited only to Department of Defense systems. Most of the major Department of Energy National Laboratories, including Pacific Northwest Laboratories (PNL), have been flooded by attacks.

---

[1] Pacific Northwest Laboratory is operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC06-76RLO 1830.

Particularly vulnerable to these kinds of attacks are client/server applications involving commercial-off-the-shelf software systems like database management systems (e.g., Oracle and Sybase) that transmit user names and passwords without any encryption.

This paper describes a technique PNL has developed for authenticating computer users and transmitting passwords between client/server applications. This technique is currently being used to secure several production applications including the Tank Waste Information Network System (TWINS).

Principal Components

There are three principal components used in the authentication process. There components are:

- Client Application - the computer software application that wishes to access computer resources.
- Server Application - the computer software application that provides services to the Client application. For example, this might be a database management system (DBMS) or a file server.
- Broker - a "trusted" computer software application that issues public keys and synchronizes access to the desired Server applications.

Typically, each of these components reside on physically separate computer systems; however, for smaller applications they may be combined on one hardware platform.

Authentication Process

The data and control flow used in the process is illustrated in Figure 1.

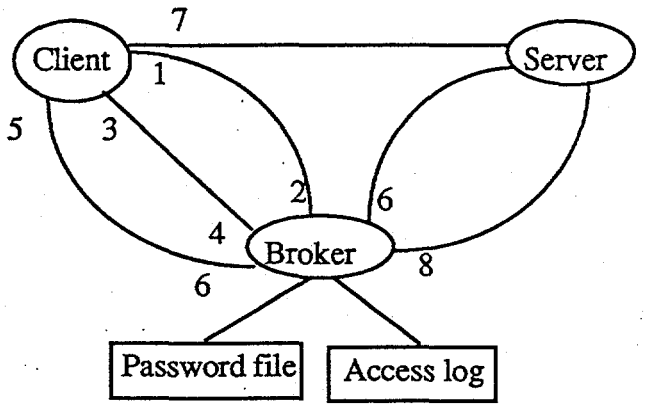


Figure 1

A detailed discussion of each of these steps is listed below. Note that steps 1-7 are all conducted in a single session with the Broker. If this session is broken during the process, the access attempt is rejected.

1. The Client initiates a session with the Broker and presents its application name, version, and IP address.

2. The Broker verifies that the Client's name and version number are correct and that the IP address is well formed, then sends back an acknowledgment.

3. The Client requests a public key.

4. The Broker returns a public key.

5. The Client uses the public key to encrypt the user name and password that the Server requires and passes this back to the Broker. In our scheme the encryption is done using the Digital Encryption Standard (DES) algorithm.

6. The Broker decrypts the information and checks for a match in a password file that only the Broker can access. If the match fails the Broker sends a

"failed" response back to the Client. If the match succeeds, the Broker takes the following steps:

A. Makes a new password.
B. Encrypts the user name and password using a different encryption key.
C. Changes the user name and password in the Server application to match the new user name and password.
D. Sends the new password back to the Client.
E. Sends the IP address of the Server(s) back to the Client.
F. Logs the access attempt.

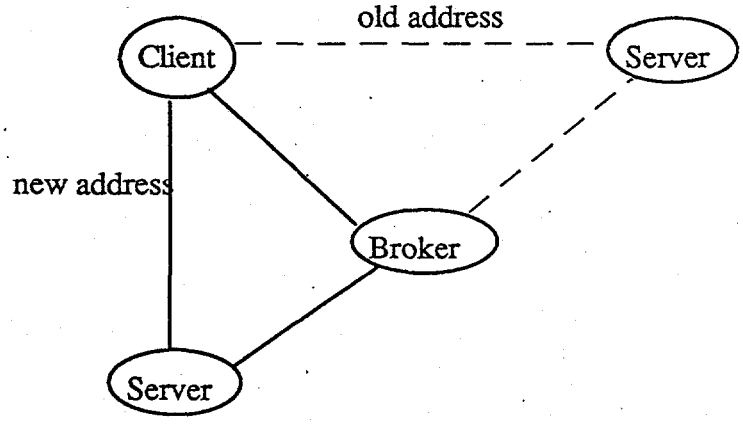
7. The Client decrypts the new password, disconnects from the Broker session and connects to the Server application using the new user name and password.

8. After a short duration (20-60 seconds), the Broker automatically makes a new dummy password for the user name and changes the user's password in the Server application to this dummy password.

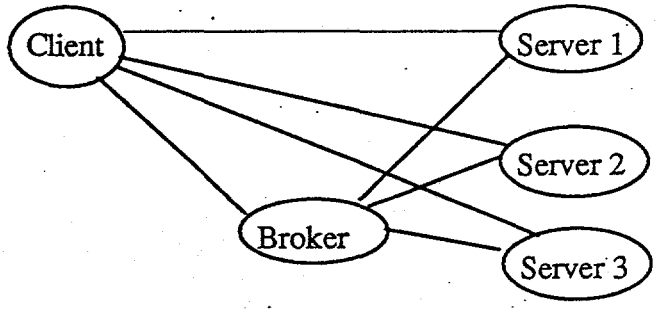
Additional Benefits

In the process of implementing this security scheme, several important side benefits were discovered by using a centralized broker to handle all connection requests. These benefits include:

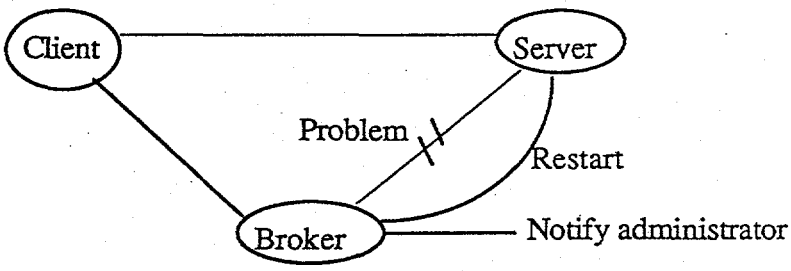
- The ability to change Server locations (IP addresses) without requiring that potentially thousands of Client applications be modified. This is made possible by having the Broker send the Server's address back to the Client during the session.



- The Broker can balance the load between multiple Servers by telling the Client to use a Server that is less loaded.



- Server applications that have crashed can be automatically restarted by the Broker if it encounters an error in trying to connect with it. In addition, the Broker can send email messages to the appropriate system administrators that an error condition was encountered.

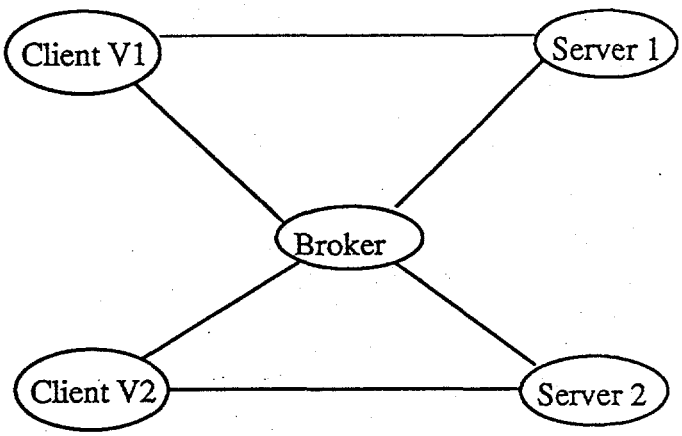


- The difficulty in keeping user names and passwords synchronized between multiple Servers is removed because the Broker automatically generates the passwords and configures the Server(s) to accept them. With applications that have very dynamic user bases, this can be a significant time savings for a database or system administrator.

- Application and Server usage metrics are easy to capture and view because the Broker maintains a central log of all access attempts.

- Certain classes of IP addresses can be easily disallowed from accessing Server applications. For example, the Broker could disallow access to an application from any computer coming from a university (i.e. with a *.edu IP name).

- Because the application version is presented to the Broker when an attempt is made, out-of-date Client applications can be restricted to certain Servers or disallowed entirely.



Acknowledgments

This technique was developed by several PNL staff members including: Bruce Kissinger, William T. Valdez, Cullen Tollbum, John McCoy, Gary Danielson, J.D. Fluckiger, and Jim Brown.