Implement two-factor authentication for AIX using Kerberos

Achieve two-factor authentication using Kerberos GSS-API as the underlying security mechanism

Sandeep Ramesh Patil

November 04, 2008

In the ever-growing need for higher security systems, multi-factor authentication is preferred for network security. Since Kerberos is one of the most popular network authentication mechanisms, learn how to design a multi-factor authentication over the Kerberos protocol. Understand the use of One-Time Password (OTP) and GSS-API to achieve this.

Introduction
Authentication is one of the most crucial modules for any given system. Kerberos is a popular security mechanism used by systems for network authentication and secure transmission of data. Almost all UNIX®-based operating systems supporting pluggable authentication modules incorporates Kerberos-based authentication. IBM® AIX® V5.3 features a Kerberos-integrated login and has many of its utilities like SSH, rlogin, telnet, and more. IBM AIX ships with its own flavor of Kerberos called IBM Network Authentication Service, available in its expansion pack CD (see Related topics).

With an increase in the need for secure environments, the authentication modules of systems (including operating system) are moving from traditional one-factor authentication (usually based on a static password) to multi-factor authentication. Two-factor authentication is a popular method being practiced for increased security. This article describes how Kerberos can be used in two-factor authentication systems. It elaborates on how you can have a more highly secure system using Kerberos and One Time Password (OTP) in multi-factor authentication schemes. Finally, the article gives you the design on implementing a two-factor authentication system using Kerberos and Generic Security Service API (GSS-API).

Two-factor authentication using Kerberos and OTP
Authentication is a process to verify a person’s identity for security purposes and authentication factor is the piece of information used to achieve it. Two-factor authentication is a process in which two different factors are used for authentication and hence delivers a higher level of security. Typically, in a two-factor authentication system, the first factor used for authenticating the user
Implement two-factor authentication for AIX using Kerberos

is something the user knows, like a static password or a pin, and the second factor is something the user has, like a credit card, mobile phone, or a hardware security token that generate a One-Time Password (OTP) token. OTP systems are typically built on a client-server model, where at a given time, for a given user, the same one-time passwords are generated on the client side. The client may be a handheld hardware device, a hardware device connected to a personal computer through an electronic interface such as USB, or a software module resident on a personal computer, as well as the server side which does the verification. In these systems, the passwords are constantly altered, which greatly reduces the risk from an unauthorized intruder. Hence, its usage in two-factor authentication is considered safer and suits many systems requiring highly secure authentication modules. For more information on OTP, please refer to IETF RFC 2289 (see Related topics).

Kerberos is a popular security mechanism used for network authentication. Most of the login modules of operating systems such as AIX, Linux®, and Windows® support Kerberos-based authentication. In addition, many remote login applications available on UNIX-like Open SSH, telnet, rlogin, and more have their Kerberized versions. Kerberos-based authentication modules are implemented directly using a Kerberos API or by using the GSS-API interface. With the ever-increasing use of Kerberos, practitioners might be faced with the need to have Kerberos mechanisms be a part of their two-factor authentication systems using OTP.

In the proposed design for a two-factor authentication system, this article makes use of Kerberos for the first-factor authentication and then makes use of the Kerberos infrastructure (using the GSS-API interface) to securely achieve the second-factor authentication using the OTP token. In this way, you can achieve a secure two-factor authentication where the Kerberos protocol plays a dual role:

- Usage of Kerberos as the authenticating protocol in the first-factor authentication.
- Usage of Kerberos (using the GSS-API interface) as a secure channel to communicate with the OTP and its verification result between the client and the server during the second-factor authentication.

Apart from the existing Kerberized applications (with single-factor authentication) the design can also be incorporated in network-based solutions requiring two-factor authentications along with the Kerberos protocol.

**GSS-API**

As defined in IETF RFC 2078 (see Related topics),

> The Generic Security Service Application Program Interface (GSS-API) provides security services to callers in a generic fashion, supportable with a range of underlying mechanisms and technologies and hence allowing source-level portability of applications to different environments.

Kerberos is the most popular underlying security mechanism available with GSS-API. IBM Network Authentication Service V1.4 for AIX implements the GSS-API standard and exports its interfaces using GSS-API libraries, which support the Kerberos mechanism. Using these GSS-API interfaces,
you can implement secure Kerberized applications and systems. Typically, in GSS-API a secure connection between two communicating applications is represented by a data structure called a security context. The application that establishes the secure connection is called the context initiator. The application that accepts the secure connection is the context acceptor. The context establishment between the initiator and the acceptor is actually a handshake process that involves authenticating both the parties. On successful establishment of a GSS-API context, it is concluded that the user identification has been successfully verified.

**Important GSS-API interfaces**

Two vital aspects that can be programmed using Kerberos GSS-API is authentication followed by per-message encryption. Following are few of the vital interfaces that help achieve the stated:

- **gss_init_sec_context(…)**
  Establishes a security context between the context initiator and the context acceptor.

- **gss_accept_sec_context(…)**
  Accepts a security context created by the context initiator.

- **gss_wrap(…)**
  Cryptographically signs and optionally encrypts a message.

- **gss_unwrap (…)**
  Unwraps a message sealed by the gss_wrap subroutine and verifies the embedded signature. For more information on the GSS-API, please refer to IBM Network Authentication Service Version 1.4 Application Development Reference, which is shipped with the product.

**Design a two-factor authentication system using GSS-API and OTP**

Figure 1 shows a Kerberized client configured to a Kerberos server system. Either system can be an AIX box with IBM Network Authentication Service installed and configured.

**Figure 1. Design steps for implementing a two-factor authentication system using GSS-API and OTP**
The following explains the steps that can be implemented using a standard Kerberos GSS-API to achieve a two-factor authentication login module with OTP.

**Step 1** - Prompt the user to enter the Kerberos user name and Kerberos password.

**Steps 2, 3, 4, 5, 6, 7** - Use the Kerberos User Name and the corresponding Kerberos password (which the user has to remember) to acquire the Kerberos credential (TGT - Ticket Granting Ticket). If the password entered is incorrect, the authentication fails and the user is not allowed any access. On successful acquisition of the Kerberos ticket (TGT), the first-factor authentication is completed. This is similar to any regular login module that based on Kerberos authentication.

**Steps 9, 10, 11** - Use the above-acquired Kerberos credential and establish a secure GSS-API context with the GSS-API based OTP application server residing on the system (assuming the OTP server has been Kerberized using GSS-API). This involves a handshake between the client login module and the GSS-API-based one-time password application server. Note that here both the login module and the OTP application server are GSS-API-based applications running on the Kerberos mechanism. The handshake exercises the gss_init_sec_context() and related APIs at the client side and gss_accept_sec_context() and related APIs at the server side. On successful handshake, a secure authenticated GSS-API context is established over the underlying Kerberos security mechanism. This context helps the client login module and the GSS-API-based server to communicate securely. If the handshake fails, the login program is terminated.

**Step 12** - Enter the One-Time Password. The OTP is the authentication information that the user possesses (which differs from the earlier one where the user needs to remember the Kerberos password).

**Steps 13, 14** - Encrypt the Username and One-Time Password using the securely established GSS-API context and send across the encrypted information to the server. GSS-API interfaces like gss_wrap() assist in encrypting the information, which can be decrypted only by the GSS-API based OTP server application.

**Steps 15, 16** - GSS-API-based OTP server application decrypts the authentication information and present the actual OTP server with "Username" and "One Time Password." (Using the earlier-established secure GSS-API context, interfaces like gss_unwrap() assist in decrypting the information). The Username and One Time Password are verified against the server database and the result of the verification (either success or failure) is passed back to the GSS-API-based OTP server application.

**Steps 17, 18** - GSS-API-based OTP server application encrypts the authentication result and passes it back to the client login module/application.

**Step 19** - The client login module/application decrypts and interprets the result. (Using the earlier-established secure GSS-API context, interfaces like gss_unwrap() assist in decrypting the information). If the result is successful, it completes the second-factor authentication and allows the user to log into the system. Otherwise, the client login program terminates, cleaning the entire GSS-API context and both client and server side.
**Note:** These steps do not claim to have completely introduced two-factor authentication in a Kerberos protocol. They can be implemented by practitioners who require a two-factor authentication where the first factor needs to be a regular Kerberos authentication and the second factor needs to be OTP, or they can be implemented as a part of login modules of applications or secure systems.

**Conclusion**

This article highlighted the use of GSS-API and Kerberos. It emphasized the growing need of multi-factor authentication and the use of OTP in such a system. It finally explained the steps that can be used in implementing a two-factor authentication using Kerberos and OTP Password for highly secure systems.
Related topics

- **A Kerberos Primer** (developerWorks, Nov 2001): This article introduces Kerberos technology and Distributed Computing Environment-based applications.
- **A One-Time Password System**, IETF RFC 2289.
- **IBM Network Authentication Service for AIX**: Download the IBM Network Authentication Service for AIX from IBM AIX Web Download Pack Programs.
- **IBM Network Authentication Service for Linux, Solaris**: Download the IBM Network Authentication Service for Linux, Solaris from here.
- **IBM GUI-based Administration Tool for Network Authentication Service**: Experience the GUI to perform the IBM NAS related administration tasks. Download from the IBM alphaWorks today.