Authentication of different OS's

Remco Hobo

December 14, 2004

Preface

In this report, three different operating systems will be examined on their autenticating systems. We will look at:

- Gentoo/freeBSD
- MacOS X

Gentoo uses PAM MacOS X uses opendirectory

Gentoo/freeBSD

passwd

Gentoo is a widely user Linux distribution, which we also use at our lab. As all *nix distributions it uses an authentication system based on users, rights and groups. Normally users are stored in /etc/passwd, with a shadow file for the passwords in /etc/shadow. A shadow file is a file where the passwords are stored in hash format. The file is normally only readable by the root user. With freeBSD this will be the /etc/master.passwd file. Group information can be found in /etc/groups.

The MD5 algorithm is an algorithm that is used widely across different platforms, it is fast, and used to be safe. Unfortunately it is now considered [4]broken.

PAM

Another form of authentication is [2]PAM (Pluggable Authentication Modules)

PAM provides a way to develop programs that are independent of authentication scheme. These programs need ”authentication modules” to be attatched to them at run-time in order to work. Which authentication module is to be attatched is dependent upon the local system setup and is at the discretion of the local system administrator.

Figure 1 displays how PAM works.

```
+----------------+
| application: X |
+----------------+       /  +----------+     +================+
| authentication-[---->--\--] Linux-   |--<--| PAM config file|
|       +        [----<--/--]   PAM    |     |================|
|[conversation()][--+    \  |          |     | X auth .. a.so |
+----------------+  |    /  +----------+     | X auth .. b.so |
|                |  |       __|  |           |           _____/
|  service user  |  A      |     |           |____,-----’
|                |  |      V     A
+----------------+  +------|-----|---------+ -----+------+
                       +--------------+    |      |      |
                       |   auth....   |--[ a ]--[ b ]--[ c ]
                       +--------------+
                       |   acct....   |--[ b ]--[ d ]
                       +--------------+
                       |   password   |--[ b ]--[ c ]
                       +--------------+
                       |   session    |--[ e ]--[ c ]
                       +--------------+
```

Figure 1

The left of the figure represents the application; application X. Such an application interfaces with the Linux-PAM library and knows none of the specifics of its configured authentication method. The Linux-PAM library (in the center) consults the contents of the PAM configuration file and loads the modules that are appropriate for application-X. The A represents authentication, the V represents the validation. These modules fall into one of four management groups (lower-center) and are stacked in the order they appear in the configuration file. These modules, when called by Linux-PAM, perform the various authentication tasks for the application (a to e). Textual information, required from/or offered to the user, can be exchanged through the use of the application-supplied conversation function.

PAM uses is used for four different types of management:

- Authentication management
- Account management
- Session management
- Password management

The main advantage of PAM is it can easily be expanded with other modules. PAM can use MD5 and DES encryption out-of-the-box. PAM is considered to be stable and secure. PAM for BSD is still in [5]development.

MacOS X

MacOS X uses open [3]directory be default. It uses the "Open directory server", which uses LDAP (Lightweight Discovery Access Protocoll), the built-in authentication authority. In version 10.3, OpenLDAP is implemented to make authentication MacOS more platform-independent, it can be used to incorporate multiple platforms into one namespace, and use one authentication authority for the whole organisation. The "Open Directory" authentication is considered robust and with the use of the Berkley DB, the world's most scalabele database, it scales excellent. It can easily manage hundreds of thousands user records. Also replication can be easily done, making it fault tolerant. LDAP is also available for Gentoo/freeBSD. There are three ways to do authentication for LDAP

- Simple authentication: The simplest way to do authentication is clear text. The server attempts to match the password with the userPassword value in the directory service. If the password is stored in a hashed format, the server will hash the password by its self. Of course, it is a major drawback that the passwords are sent clear-text.
- Simple authentication over SSL/TLS: Passwords can be sent in two different ways:
    1. LDAP over SSL (LDAPS - tcp/636) is well supported by many LDAP servers, both commercial and open source. Although frequently used, it has been deprecated in favor of the StartTLS LDAP extended operation.

2. RFC 2830 introduced an LDAPv3 extended operation for negotiating TLS over the standard tcp/389 port. This operation, which is known as StartTLS, allows a server to support both encrypted and nonencrypted sessions on the same port, depending on the clients' requests.

- Simple Authentication and Security Layer (SASL): SASL is an extensible security scheme defined in RFC 2222 that can be used to add additional authentication mechanisms to connection-oriented protocols such as IMAP and LDAP. SASL is a pluggable authentication scheme by allowing client and server to negotiate the authentication mechanism prior to the transmission of any user credentials. There are several authentication schemes for SASL, including:

  1. Kerebos v4
  2. The Generic Security Service Application Program Interface, Version 2 (GSSAPI), RFC 2078
  3. The S/Key mechanism (SKEY), which is an one-time password scheme based on the MD5 message digest algorithm
  4. The External (EXTERNAL) mechanism, which allows an application to make use of a user's credentials provided by lower protocol layer, such as authentication provided by SSL/TLS

What authenticating system would I use?

When a network consists of little nodes and the data on the network is not really vital, the /etc/passwd encryption will do fine. It‘s lightweight, easy to configure and it will work out-of-the-box. When however the network has more nodes and/or the data is mission critical, a more advanced authentication system might be needed. For this, LDAP might be a good solution. It‘s password encryption is much harder to crack and the whole protocoll is more secure. When uses with Linux/freeBSD it can even be combined with kerebos.

References

[1] http://www.kernel.org/pub/linux/libs/pam/FAQ,
The PAM FAQ

[2] http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html,
The PAM overview

[3] http://www.apple.com/server/macosx/open_directory.html,
The Mac open directory homepage

[4] http://desigeek.com/weblog/amit/archive/2004/07/03/314.aspx,
A site about how to crack MD5

[5] https://listman.redhat.com/archives/pam-list/1997-July/msg00013.html,
A threat about PAM for BSD