ONLINE FILE ACCESS AND AUTHENTICATION SYSTEM IN CAMPUS NETWORK

Nu Nu Lwin, Yin Ko Latt
University of Computer Studies, Magway
nunulwin158@gmail.com

Abstract

The organizational systems are replaced with the computer networking system all over the world. File systems are responsible for the organization, storage, retrieval, naming, sharing and protection of files. The system mainly uses the network file system for online file accessed in campus network. It is implemented to check the user allowed time and access permission for the files and folders in the system. Also it is essential to implement security measures to avoid the access by unauthorized users. In this system, security is the key to authenticate client requests so that access control at the server is based on the correct user identities and to protect the contents of files and folders with encrypted passwords. The system used password hashing process to get the hashed password for authorized users of the system. So, Message Digest 5 (MD-5) is used to create hash of passwords allowing an encrypted form of password to be sent over network or stored in file system. Therefore, this file accessed system is useful for everyone who wants to access its own files to multiple users in the local area network.

1. Introduction

Many early network systems provided file transfer services that permitted users to move a copy of a file from one machine to another. More recent network systems provide file access services that permit an application program to access a file from a remote machine. Online File Access system is built on top of a web based files sharing service. This makes the file system available online and to anyone anywhere to access files with one another. The additional benefit of this is only the end-users are aware of the file system organization and contents.

The main goal of this system is to protect the important files in the campus network. The important files may be protected against the unauthorized user and unauthorized access using password authentication and user allowed time. In this system, there are six types of users according to the user role. In each user, there is an administrator or head of network, a super user (teacher), first year (students) second year (students), third year (students) and viewer (view only). Among these, super user and administrator are control in this system. Super users in file access system may create their files with owner access and the other users can not delete those files.

Network file system allows computers to mount a disk partition on a remote computer as though it were on a local hard drive. Adopting client/server architecture, NFS enables a fast, seamless sharing of files across a network. An important goal of NFS is to achieve a high level of support for hardware and operating system heterogeneity. NFS provides access transparency, user programs can issue file operations for local or remote files without distinction. To provide remote access to some or all of the files that reside on a computer, the system manager must arrange for the computer to run a server that responds to access requests. The server checks each request to verify that the client is authorized to access the specified file, performs the specified operation, and returns a result to the client.

Password authentication is an important mechanism for remote login systems, where only authorized users can be authenticated via using their passwords. Cryptographic hash functions are an important building block for a wide range of applications such as the authentication of information, digital signatures and the protection of pass-phrases. The most popular hash functions are the custom designed iterative hash functions from the MD4 family. In this system, Message Digest MD-5 algorithm is used in password authentication. It is required that only authorized users can be authenticated by the server, and then are granted to access the resources and/or services provided by the server.

2. Related Work

Networked file systems allow for distributed storage and access of data. The Sun Network File System (NFS) [16], originally designed for sharing among a small set of mutually-trusting workstations, developed over several versions to now support access over wide-area networks with strong security mechanisms for end to end mutual authentication and integrity. Other network file systems like the Andrew File system [10] and CIFS [6] developed in a similar

fashion and also now scale to service large distributed networks of computers.

As the web grew to become one of the most pervasive computing platforms, network file systems were created or adapted to fulfill the needs of this new area. WebFS [3] provides a global file system over HTTP with support for functionality needed by many distributed Internet applications. WebDAV [11] focuses on web authoring specifically and extends HTTP to a read-write platform for web clients. WebDAVA [2] provides for file sharing over HTTP with flexible access controls using user issued access credentials. DavFS allows to mount files from a WebDAV server on a local driver. WebDAV is an extension of http that allows remote collaborative authoring of web resources. DavFS allows a remote web server to be edited simultaneously by a group using standard applications [8]. WebNFS [2] adapts the NFS protocol for the web, by extending the semantics of the NFS protocol to support web browser clients by creating a lightweight binding mechanism. Even though these protocols target web browsers as clients, they are not fully supported by current web browsers. They require additional software to enable the browser to communicate with the file system servers. Web applications running in the browser do not interface with these network file systems in through the browser.

Menagerie [12] provides a virtual file system composed of data from heterogeneous web applications by providing an interface for a web application to export their data into a namespace and a file system interface that combines these namespaces. A user can then mount this virtual file system on his local computer and manipulate it with standard file system commands. We take a user-centric approach where data is stored separately from the application and provide web applications with interfaces to the storage. The user then also retains control over the sharing of his data. O.Kiselyov describes the Httpfs, is a network file system that provides access to files on a remote machine using the http protocol. It requires a component to run on the remote server, from where documents can be fetched on the client. This is similar to the network file system implementation but using http[7].. A.Adya provides the Web File system that interface to the World Wide Web. This file system allows the user to browse the web as different files that are downloaded on the local hard drive [1].

The Distributed Credential File System (DisCFS) uses credentials to identify both the files stored in the file system and the users that are permitted to access them, as well as the circumstances under which such access is allowed. Furthermore, users can delegate access rights simply by issuing new credentials, providing a natural and very scalable way of sharing information. This is not the case for SFS, where access control relies on user and group ID, which are translated from one machine to another. This forces users to have accounts on file servers to access protected files, and defeats the purpose of a truly distributed file system [15].

3. Background Theory

3.1 Password Authentication

In today's world, security is a very big concern, and, passwords are the primary method of authentication for computer systems [14]. A multi-user system requires users to prove their identity before accessing system resources. A user typically begins a session by providing user name and secret password to login program. This program verifies the password using system-wide password file [13]. Given the importance of keeping passwords secret, the program does not store plaintext passwords in this file. To verify a password the login program hashes the password and compares the result to the appropriate hash in the password file [9].

3.2 Cryptographic Goals

Five main categories of Cryptographic goals are:

Authentication

This means that before sending and receiving data using the system, the identity of receiver and sender should be verified.

Secrecy or Confidentiality

Usually this function is how most people identify a secure system. It means that only the authenticated people are able to interpret the message content and no one else.

Integrity

Integrity means that the content of the communicated data is assured to be free from any type of modification between the end points (sender and receiver).

Non-Repudiation

This function implies that neither the sender nor the receiver can falsely deny that they have sent a certain message.

Service Reliability

Since secure systems usually get attacked by intruders, which may affect their availability and type of service to their users [5].

3.3 Cryptographic Hash Basics

A cryptographic hash algorithm takes a message of arbitrary size and produces an output of fixed size. The output is the result of a one-way function, which cannot be reversed. These hash algorithms have several other desirable properties. Given a message *M*, hash (*M*) will always produce the same result. Given a hash *h* it should not be possible to determine the message *M*, such that *h* = hash (*M*). The output of hash function should look random, so that the hashes of two similar messages look very different.

Common cryptographic application of these hash functions is for data integrity and for authentication.

3.4 Message Digest 5 (MD5)

Message Digest 5 (MD5) hash was developed by Rivest as an update to his previous MD4 hash and published in 1992 [2]. MD5, like other cryptographic hash algorithms, takes a message of arbitrary size and produces an output of fixed size (128 bits). Figure 1 shows how the MD5 algorithm works. A given message is divided into 512-bit chunks and each chunk is processed as a single MD5 operation. The input to the first operation is an initialization vector and the output is used as the starting point for the next chunk's operation. The last part of the message is padded and appended with the length of the message to form the final 512-bit chunk. The output of this last operation is the hash result.

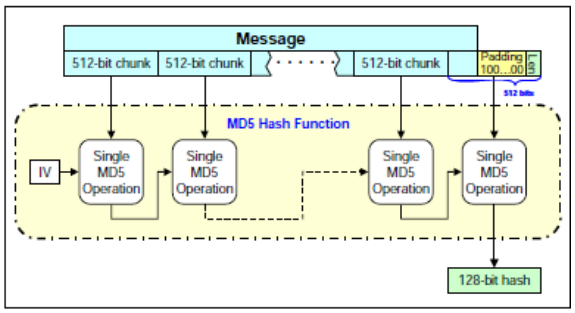

Figure 1. Illustration of MD-5 Hash Function

Within a single MD5 operation, there are 4 rounds of processing, with each round having 16 steps and using a different compression function. MD5 is optimized for 32-bit processors, so the initialization vector and the working state each consist of 4 32-bit words (represented as A, B, C and D) and the 512-bit message chunk is divided into 16 32-bit words. For each round, a different function is performed on 3 of the 4 words (B, C, D) in the state. That result is added (modulo 2^32) with the fourth state word (A), one of the 16 message chunk words and a constant based on the Sine function (which contributes to the randomness of the output). A bitwise shift of the result is performed and then added to state word B. The values of state words B, C, and D are moved to C, D and A respectively and the next step is performed for a total of 64 iterations (4 rounds, 16 steps each). Figure 2 illustrates a single MD5 operation.

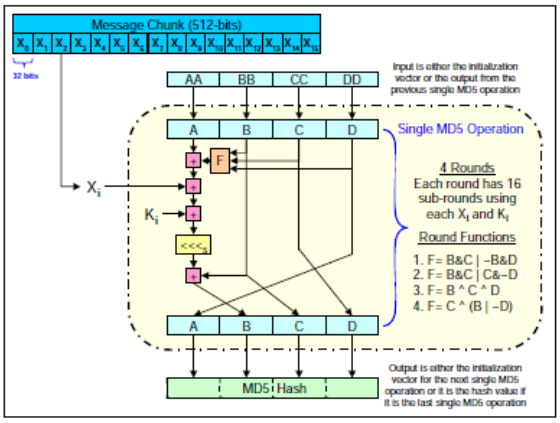

Figure 2. Single MD-5 Operations Illustrated

3.5 Uses for MD-5 Hashes

MD5 hashes are used to verify data integrity by providing check-sums of files and with digital signatures of messages. Instead of digitally signing a message under PKI, often a hash of the message is signed instead. MD5 is also used commonly in authentication by hashing passwords. Instead of storing clear text passwords on a system, a hash of the password is stored and during authentication, the system hashes the input password and compares it to the stored hash value. For client server applications, the client sends a hash of the password instead of a clear text password and the server compares that to a hash of the stored password.

3.6 Advantages of Message Digest MD-5

The generation of a digest is very fast and the digest itself is very small and can easily be encrypted and transmitted over the internet. It is very easy and fast to check some data for validity. The algorithms are well known and implemented in most major programming languages, so they can be used in almost all environments.

3.7 Access Check Algorithm

When the user requests the desired file/folder in the system, the system determines the access rights of user and their allowed time on the user data. To check Read, Write, Delete, Control Permission, the access check is performed on the Access Control Entries using the following algorithm:

- If (user ID && request Access) ="Yes", then Access is allowed.
- If (Read) = "Yes", then Access is allowed.
- If (Write) = "Yes", then Access is allowed.
- If (Copy) = "Yes", then Access is allowed.
- If (Control Permission) = "Yes", then Access is allowed.

- If (Read && Write) = “Yes”, then Access is allowed.
- If (Read && Copy) = “Yes”, then Access is allowed.
- If (Read && Control Permission) = “Yes”, then Access is allowed.
- If (Write && Copy) = “Yes”, then Access is allowed.
- If (Write && Control Permission) = “Yes”, then Access is allowed.
- If (Copy && Control Permission) = “Yes”, then Access is allowed.
- If (Read && Edit && Copy &&Control Permission) =“Yes”, then Access is allowed.

After the system checks the access right for the requested user, the system will configures the items to use the folder. If the read access is ‘Yes’, the system allows to open the folder and files. If the write and copy access is ‘Yes’, the system allows to edit the files. If the control permission access is ‘Yes’, the system allows to control the accesses and the control permission item is true.

4. Proposed System Overview

4.1 Use Case Diagram

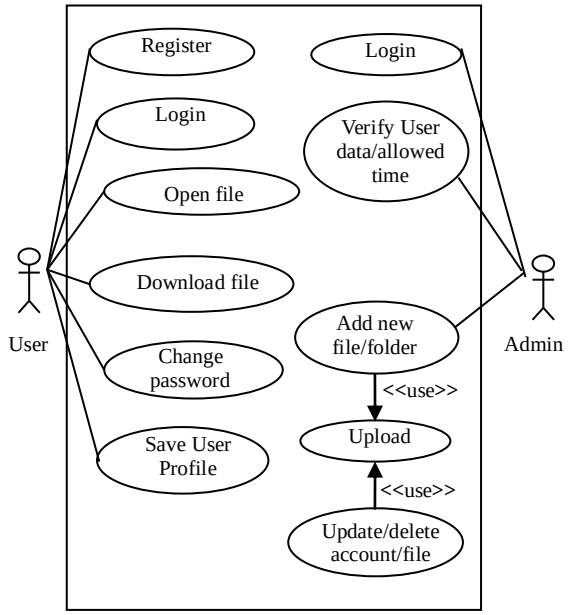

Figure. 1. Use Case Diagram of the System

This file accessed system is built for two users such as the authorized person or administrator and user or student. Administrator needs password to enter the system because it controls the system’s accessed. The administrator can add new files and folders. User has “login” process. If the user is new, he/she needs to be registered.

4.2. System Flow Diagram

Figure 2 represents the system flow diagram for online file access and authentication system. This system has two parts; these are user part and the admin part. In the user part; he/she must register to create the user account to enter the system. If the user is registered, then he/she can enter the system by typing user's name and password. If the user is new, he/she must fill register form to create a new user account. The user’s password is invalid, the system shows unauthorized person. If the user’s password is valid, he/she is authorized person. Then, the user can enter the system. The user can access two sections; the first is user can view all file and the second is downloading the required file.

If the user selects the administrator’s part, the user must enter the login process. If the user’s password is invalid, the system shows unauthorized person. If the user’s password is valid, he/she is authorized person. The administrator can view the user account list. The administrator can choose to update or delete user data and add or delete file or folder.

In this system, all password of the login process are encoded with MD-5 digest function and the hashed password is stored in password file. If the login password and the stored password are matched, the user can enter this file access system.

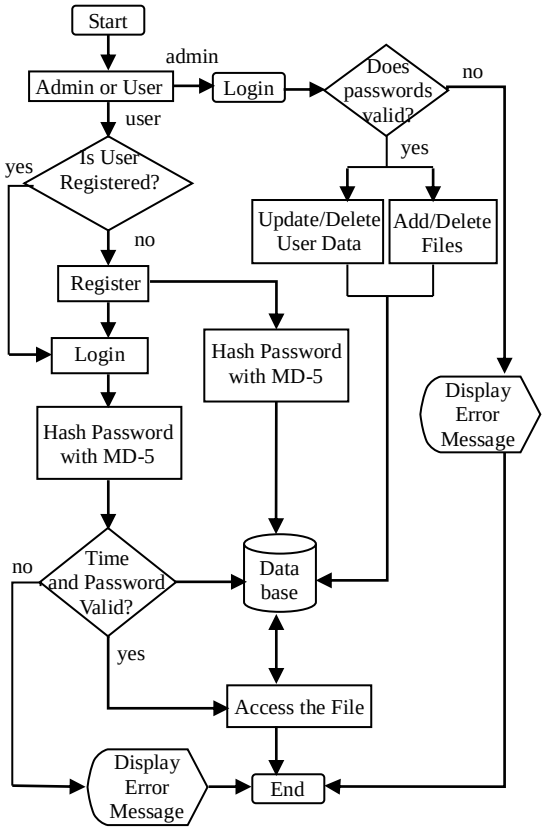

Figure. 2. System Flow Diagram

5. Implementation System

5.1 Home Page

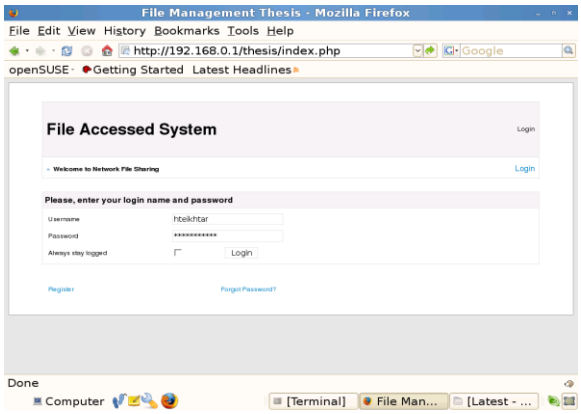

Figure. 3. Home Page

The system home page screen composites with interfaces that the user friendly tools and object to ease of use and easy to understand. This page has two parts. They are: User Login, Register and Forgot Password. If the user is registered user, than he/she can enter the system by typing user name and password and then the user click 'Log in' button.

5.2 User Registration Page

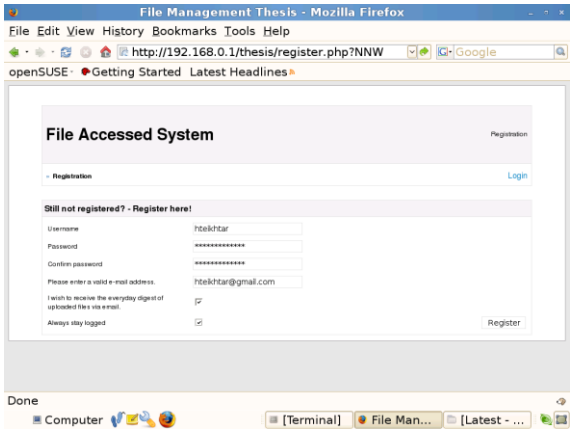

Figure. 4. Create User Account Form

This page is used for student registration. If the student is a registered user, he/she can enter the system by typing user name, password, confirm password and e-mail. And then the user click 'Register' button. When registration process is completed, the user can enter the system.

5.3 User Page

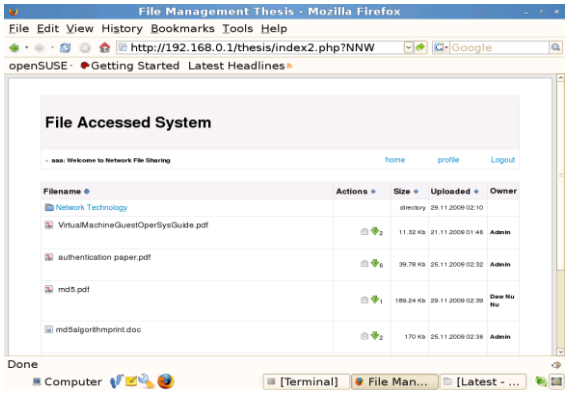

Figure. 5. User Page Form

In this page, the user can view all file from this system and open or download the required files. Then the user can view the last upload and top download files on this system.

5.4 Admin Page

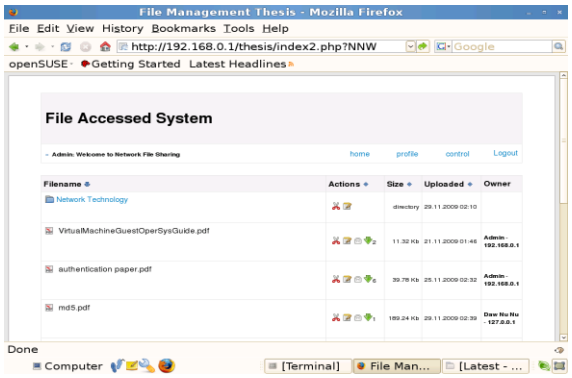

Figure. 6. Admin Page Form

In this system, admin is provided with setting page to add new files and folders data and delete user account and files.

5.5 Admin Control Page

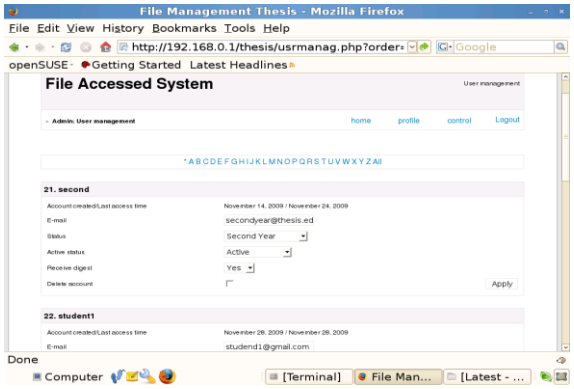

Figure. 7. Admin Control Page Form

In figure 7, the admin can view the user account list, update or delete user data and user account from this system. The admin can verify the login time according to the user type.

5.6 Upload File Page

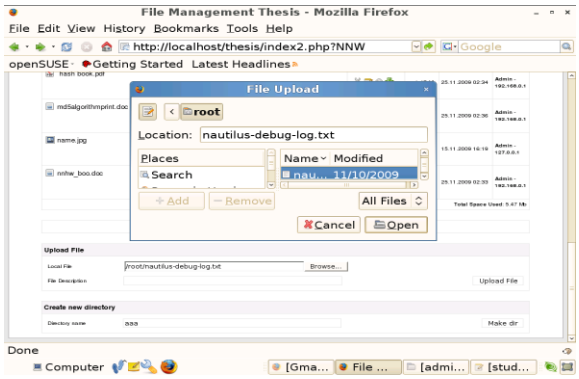

Figure. 8. Upload File Page Form

In this page, the admin can create or delete the files. After creating the folder, the admin can upload the new file on this page.

6. Conclusion

This system is based on Web Technology and developed by web programming languages such as PHP and Java Scripts. The user name, password, and user data can be given by users. In this system, administrator takes responsibility for the access control to files, restricting files access according to users' authorizations and the type of users. Unauthorized persons are restricted by the security features like password authentication. The system mainly uses the password-based authentication for recognizing users and authorizing it. Message digest (MD-5) is used for secure passwords to ensure the authenticity of the user. And when the user uses the password mismatch, the systems can response the error message. And the system allows verifying various types of user status to use the system.

The file accessed system is useful for everyone who wants to share its own files to multiple users in the local area network. According to the user login time limits, the users can use systematically. Because of using Network operating system and web server software, saving database on server and then creating an Internet environment, multi-user can access this system through browser such as Mozilla Fire Fox, Internet Explorer and Netscape browser.

REFERENCES

[1] A.Adya, "Web File System: File-like access to the web", In 5th Annual MIT Student Workshop on scalable Computing, 1995.

[2] A. Levine, V. Prevelakis, J. Ioannidis, Webdava: An administrator-free approach to web file-sharing. In *WETICE '03: Proceedings of the Twelfth International Workshop on Enabling Technologies*, page 59, Washington, DC, USA, 2003. IEEE Computer Society.

[3] A. M. Vahdat, P. C. Eastham, and T. E. Anderson. Webfs: Aglobal cache coherent file system. Technical report, UC Berkeley, 1996.

[4] B. Callaghan. WebNFS Client Specification. RFC 2054 (Informational), Oct. 1996.

[5] B. Schneier and J. Wiley, "Applied Cryptography, second edition", New York, 1996.

[6]Common internet file system (cifs) technical reference. http://www.snia.org/tech activities/ CIFS/ CIFS-TR-1p00_FINAL.pdf.

[7] O.Kiselyov, "A Network File System Over HTTP: Remote access and modification of files ", pp. 75-80.

[8] http://dav.sourceforge.net/

[9] J. Habbits, "Passwords and Authentication," April 26, 2004.

[10] J. H. Howard, M. L. Kazar, S. G. Menees. Scale and performance in a distributed file system. *ACM Trans. Comput. Syst.*, 6(1):51–81, 1988.

[11] L. Dusseault. HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV). RFC 4918 (Proposed Standard), June 2007.

[12] R. Geambasu, C. Cheung, A. Moshchuk, S. D. Gribble, and H. M. Levy. Organizing and sharing distributed personal web-service data. In *Proceeding of the 17th international conference on World Wide Web*, pages 755–764, Beijing, China, 2008. ACM.

[13] R. Morris and K. Thompson, "Password Security: A Case History". Communications of the ACM, 22(11)"594-597, November 1979.

[14] S. Alexander, "Password protection for modern operating systems".

[15]S.Miltchev, V.Prevelakis, D.Keromytis, M.Smith, "Secure and Flexible Global File Sharing".

[16] S. Shepler, Network File System (NFS) version 4 Protocol. RFC 3530 (Proposed Standard), Apr. 2003.