Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 7, July 2013, pg.350 – 355

RESEARCH ARTICLE

An Enhanced Authentication Scheme Using Kerberos with Hash-Based Message Authentication Code (HMAC)

R. Kogila

Department of Computer Science, Periyar University, TamilNadu, India

kogimca@gmail.com

Abstract— In this thesis, present a new secure authentication system is proposed that significantly reduces the possibility of frauds. This scheme is primarily designed for organizations. It is based on the Kerberos cryptographic framework that has been proven to be secure after being used in real world for decades. The proposed system allows tokens to be exchanged between the server and clients. The token is generated from the client information. A token is cryptographically secure and valid only for the designated client, and therefore it is robust against eavesdropping. The token is compared of Hash Message Authentication Code (HMAC) patent. It is symmetric key cryptography used to describe the underlying cryptographic schemes, the operating principles, and the system design. The proposed system provides authentication, for a more secure protection against the middle attacks. Also increased the timestamp since transports such as Remote Procedure Call (RPC) and Hypertext Transfer Protocol (HTTP) rely on the maximum token size. Show the authentication of system, and discuss the performance.

I. INTRODUCTION

1.1 Background

Originally, computer systems were very open in the sense that they were stand-alone machines with physical access control to decide who may access the data stored within the computing environment. With the arrival of terminal connections to mainframes, it became a necessity to develop the multi-user environment. The need for simple access control solutions to limit user access to various resources was subsequently developed. This worked well with plain user-id/password pair in order to login to systems, because computers were still scarce and there were only a limited number of users. In time, local area networks began to connect the computers to each other resulting in multiple systems that require authentication. The number of passwords one had to memorize began to grow.

First problem is the internet and global connectivity to various computing systems together with the abundance of computers in a typical corporate network, the number of different user-ids and passwords has grown tremendously. In addition, every user has to change their passwords at least twice a year. The passwords are made long and difficult to remember, because well-administered computer systems enforce strict password quality requirements. Passwords are easily misplaced or forgotten when the number of credentials the users has to use grows. It also consumes precious working time when one has to login to multiple systems manually, because it requires some seconds to remember and type in the user-id/password combination on each system when access is needed. It has also been noticed that login time increases with every failed authentication attempt.

Another problem with multiple computer systems is management. When the number of systems and users grow, the task of keeping track of authorized users and the termination of intruders becomes unbearable without good tools to automate the process of adding users to, and deleting users from, all of these systems.

II. REVIEW LITERATURE

2.1 Review Literature

It said that, the misuse of the Internet has increased the need for a centralized access control and strong authentication methods. Because web-services have become increasingly distributed, and the need for both a centralized administration and a single sign-on solution have become even more relevant. The instigator of this study Elisa Communications Corporation (Elisa) participates in the Pro FINEID–project that promotes the use of FINEID -certificate based digital identity in dealing with the authorities and e-business services.

The aim of this study is to select the best access control and administration solution for protecting the extranet of the Elisa Research Centre, based on the requirement specification dictated by the author. The basic requirements for both single sign-on and smart card support are required from all candidate products.

Both the tested, and many other products were able to satisfy the single sign-on requirement, but only one of the tested products was able to fulfill successfully the basic requirement of FINEID smart card utilization for strong authentication [1].

Kerberos is a secure, industry-standard protocol. Currently, Kerberos operates as a closed system; all users must be specified upfront and managed on an individual basis. An ant scheme EPAK (Extensible Pre-Authentication in Kerberos). EPAK explained the mutual benefits of enhancing the flexibility of Kerberos and increasing the viability of alternate authentication systems as they move to the enterprise [2].

The World Wide Web is an increasingly popular method for distributing content both public and private. To reliably deliver secure content requires a scalable authentication system and a highly available content store. This thesis will discuss the development of architecture for providing scalable authenticated web service. The architecture utilizes the Open Software Foundation (OSF) Distributed Computing Environment (DCE) to provide a secure and scalable authentication system while the OSF Distributed File System (DFS) is leveraged as a high-performance reliable content store. DCE authentication and access to DFS are integrated into the open source Apache web server through the development of an Apache module, mod_auth_dce [3].

2.2 Kerberos Authentication

Kerberos is an authentication protocol, and at the same time a (KDC), that has become very popular. Several systems including Windows 2000 use Kerberos. Kerberos is named after the three-headed dog in Greek mythology that guards the gates of Hades. Originally designed at Massachusetts Institute of Technology (MIT), it has gone through several versions. It was developed as a part of Project Athena at MIT to provide a solution to network security problems. Consider a distributed environment having many users on different workstations and services, available on servers distributed across the network. An unauthorized user may be able to gain access to services and data that he or she is not authorized to access. Instead of building elaborate authentication protocols at each server, Kerberos provides a centralized authentication server, whose function is to authenticate users to servers and servers to users.

Kerberos uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server have used Kerberos to prove their identity, they can also encrypt all their communications to assure privacy and data integrity, as they go about their business.

Kerberos involves three servers in addition to the client workstation as shown in **Fig. 2.1** an Authentication Server (AS), a Ticket-Granting Server (TGS), and a Data Server (DS) that provides services to others. In our example, Server is the real server, Client is the user/client server and Eve is the intruder.

1. Authentication server (AS): It verifies the users during the login process. It stores a secret password for every user. The AS is the KDC in the Kerberos protocol. Each user registers with the AS and is granted a user identity and a password. The AS has a database with these identities and the corresponding passwords. The AS verifies the user, issues a session key to be used between Client and the TGS, and sends a ticket for the TGS.
2. Ticket granting server (TGS): It issues 'proof of identity tickets'. These tickets are used to tell the other servers that the bearer of the TGS ticket is actually the person who he or she claims to be. The TGS issues a ticket for the real server. It also provides the session key (kAB) between Client and Server. Kerberos has separated the user verification from ticket issuing. In this way, although Client verifies her ID just once with AS, she can contact TGS multiple times to obtain tickets for different real servers.

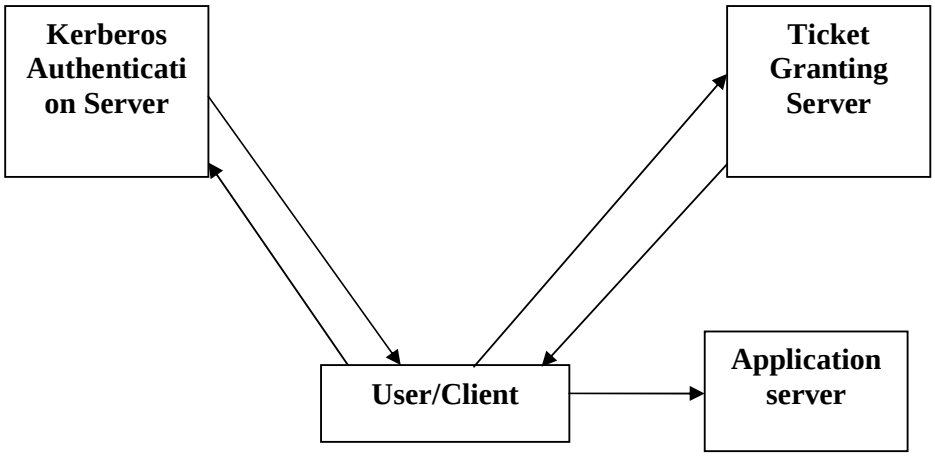

Fig.2.1 Kerberos authentication protocol

2.2.1 The client computer

Client computers are regarded insecure, because the user has full control over them. The client is usually a general-purpose computer with a Kerberos enabled OS and applications installed.

2.2.2 The authentication server

The authentication servers' role is to decide whether a user is who he claims to be i.e. authenticating the user. It also acts as an exchange, exchanging weak secrets (user ID and password) to a strong secret (a cryptographic ticket). With this ticket, the user can prove his identity to the ticket granting server (TGS).

2.2.3 The ticket granting server

Once the user has obtained a ticket granting ticket (TGT) from the AS, he is able to request an authenticator and a session key to a specific service S from the TGS. The TGS is able to authenticate the user based on his authenticator received with the TGT.

2.2.4 The application server

The application server can provide a multitude of services to the client, once the client is authenticated with the application server. This is done also vice versa, if mutual authentication is requested. The messages between them can be protected cryptographically providing confidentiality and integrity.

2.2.5 Kerberos Version 5

The minor difference between version 4 and version 5 are briefly listed below.

1. It can accept any symmetric-key algorithm.
2. It uses a different protocol for describing data types.
3. It has more overhead than version 4.

2.2.6 Realms

Kerberos allows the global distribution of ASs and TGSs, with each system called a realm. A user may get a ticket for a local server or a distant server. In the second case, for example, Client may ask her local TGS to issue a ticket that is accepted by a distant TGS. The local TGS can issue this ticket if the distant TGS is registered with the local one. Then Client can use the distant TGS to access the distant real server.

2.3 Operations on Kerberos Systems

Kerberos operates by encrypting data with a symmetric key. A symmetric key is a type of authentication where both the client and server agree to use a single encryption/decryption key for sending or receiving data. When working with the encryption key, the details are actually sent to a key distribution center, or KDC, instead of sending the details directly between each computer. The entire process takes a total of eight steps, as shown below.

1. – The **authentication server**, or AS, receives the request by the client and verifies that the client is indeed the computer it claims to be. This is usually just a simple database lookup of the user's ID. This is shown in **Fig.2.2**

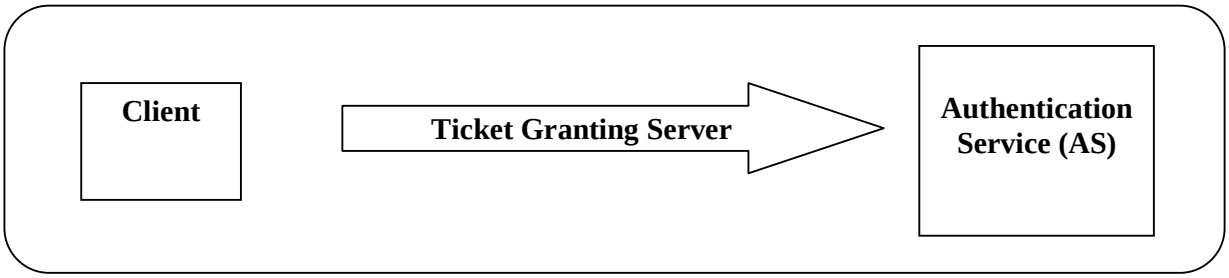

Fig.2.2 Client sends a login request to AS

III. KERBEROS WITH HMAC SYMMETRIC

3.1 Proposed System

In the proposed System, All the encryptions could be done using the HMAC algorithm and also increased the timestamp since transports such as remote procedure call (RPC) and HTTP rely on the max token size. We show the authentication of the system, and discuss the performance.

3.1.1 Advantages of proposed system

1. All the encryptions could be done using the proposed cryptographic algorithm. Since the current Kerberos system uses a standard symmetric key encryption algorithm, it is easy for an intruder to find out the key and decrypt. But when the proposed system is used, only the authorized persons, who have the decryption algorithm, could only decrypt the encrypted text. Any other intruder, who wants to perform off-line attack, will not be able to do so because this algorithm protects the message in a much stronger way using variable block cipher with cipher block chaining mode. It is very difficult to decrypt the message even with the algorithm available. Because this algorithm gives an extra layer of protection with a password. The chances of password guessing approach for any intruder are nullified because the proposed system does not store the password of the client anywhere in the hard disk. Hence no attempt can be made to find it out.

2. By integrating the proposed system with the smart card technology, some of the Kerberos systems problems may be overcome. The whole idea is to enhance the security of Kerberos authentication by authenticating the user directly at the beginning and before the granting of the initial ticket, so that one user cannot have the ticket of another. And, the use of smart card requires user logging into the system not only by recalling a password, but also to be in possession of a token.

Another way to enhance security is to use biometric technology with the proposed system in the smart card. Biometrics information of the cardholder can be placed on the card, so that the smart card can corporate with biometrics scanner to authenticate the user directly at the first stage of processing. Before granting the initial ticket, this authentication could take place, to avoid any intruder to pretend as the cardholder. The proposed system, which combines the techniques of cryptography and steganography, could be applied to embed the biometrics information of the cardholder into his photograph in the smart card. Since this algorithm provides a robust protection to the information against attacks, the biometrics details could not be easily trapped by any fraudulent.

IV. RESULT

4.1 Implementation

The HMAC algorithm is specified for an arbitrary Approved cryptographic hash function, H. With minor modifications, an HMAC implementation can easily replace one hash function, H, with another hash function, H'.

Conceptually, the intermediate results of the compression function on the B-byte blocks (K0 □ ipad) and (K0 □ opad) can be precomputed once, at the time of generation of the key K, or before its first use. These intermediate results can be stored and then used to initialize H each time that a message needs to be authenticated using the same key. For each authenticated message using the key K, this method saves the application of the hash function of H on two B-byte blocks (i.e., on (K □ ipad) and (K □ opad)). This saving may be significant when authenticating short streams of data. These stored intermediate values shall be treated and protected in the same manner as secret keys.

4.2 Performance Comparison of Kerberos on various groups of users

Here take various groups of users and check performance of their request per second, shown in Table4.1

Table.4.1. Performance Comparison of Kerberos Authentication

Group size	Request / Second
25	150
50	107
100	83

Use Table.4.1. Draw the column chart shown in Fig.4.1.

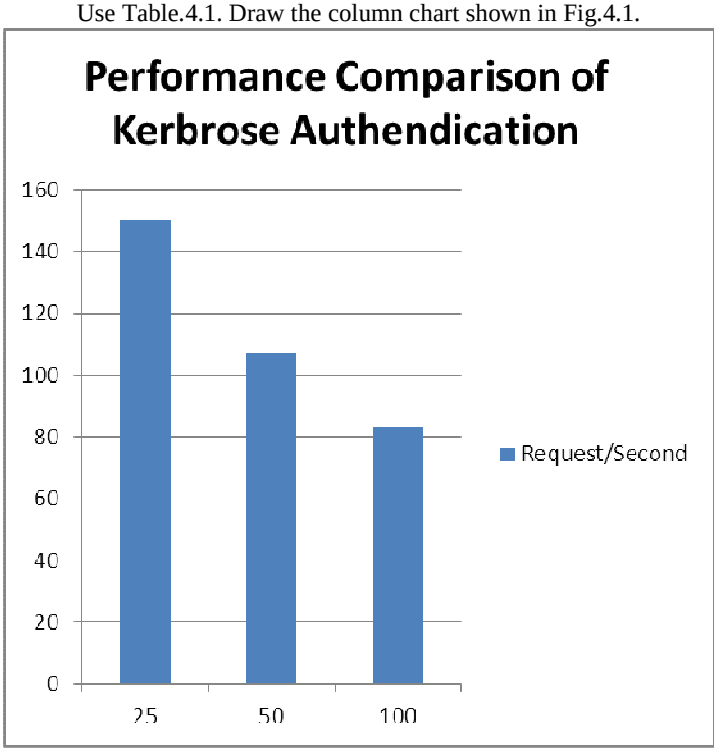

Fig.4.1 Performance Comparison of Kerberos Authentication

V. CONCLUSION

The protocol assumes a secure storage for the passwords in the AS, compromising them would provide entrée to all the services. Further the tickets and the session keys must not be cached in the system. A solution to this weakness could be the use of the challenge/response mechanism, where the server would generate the timestamp value encrypted using key center and the client would respond with some function of the timestamp value proving its veracity. The use of one-time pass-codes and the above technique during authentication could immune the system from these kinds of attacks. Major gains could also result from decoupling the protocol from the encryption algorithm used. The Cryptographic algorithm cold form the underlying layer on top of which Proposed System would run. By making Proposed System independent and token size has been increased so that transports such as remote procedure call (RPC) and HTTP rely on the maximum token size. The above change would be seamless. In conclusion Proposed System is a robust protocol for authentication and security though not without drawbacks.

REFERENCES

[1] "Raymond Philip Causton", "Smart card usage for authentication in web single sign-on systems", February 2002, (thesis)
[2] "Phillip Hallowell", "Extensible Pre-Authentication in Kerberos", August 2007, (thesis).
[3] "Paul B. Henson", "Scalable Authenticated Web Service", 2001, (thesis).
[4] "Jung Eun Kim", "A Secure on-line credit card transaction method based on Kerberos Authentication protocol", University of Nevada Las Vegas, May 2010, (thesis).
[5] "Jennifer G. Steiner, B. Clifford Neuman, and Jeffrey I. Schiller", "Kerberos: An authentication service for open network systems", In USENIX Technical Conference, 1988, (paper).
[6] "Neuman, B.C." "Kerberos: an authentication service for computer networks", Inf. Sci. Inst., Univ. of Southern California, Marina del Rey, CA 2008, (thesis).
[7] "S.Uma Maheswaran, C.S.Balakrishnan", "Internet Security – Kerberos Authentication With Blowfish Encryption

