IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, February 2022
Impact Factor: 6.252

A Study Kerberos Protocol: An Authentication Service for Computer Networks

Prof. Charushula D. Patil[1], Prof. Reena S. Sahane[2], Prof. Surbhi D. Pagar[3]
Assistant Professor, Department of Computer[1,2,3]
Guru Gobind Singh College of Engineering & Research Centre, Nasik, Maharashtra, India
charushila.patil@ggsf.edu.in[1], reena.sahane@ggsf.edu.in[2], surbhi.pagar@ggsf.edu.in[3]

Abstract: *The Kerberos Authentication Service, developed at MIT, provides a trusted third-party authentication to verify users' identity. This paper is an overview of this protocol. The article describes the protocol in the perspectives of the client and the server, focusing on how Kerberos achieve authentication. It also gives an idea of its limitations. This paper deals with practical arguments concerning Kerberos: it goes deep in some applications of Kerberos at two different levels: Cisco and the Operating System Windows 2000; and after that some results about performance are presented.*

Keywords: Kerberos, Key Distribution center, protocol, Ticket-Granting Server (TGS)

I. INTRODUCTION

Authentication is the process of proving one's identity to someone else. As humans, we authenticate each other in many ways: We recognize each other's faces when we meet; we recognize each other's voices on the telephone.

An authentication protocol would run before the two communicating parties in the system run some other protocol. The authentication protocol first establishes the identity of the parties to each other's satisfaction; only after authentications do the parties get down to the work at hand. It is a fundamental building block for a secure networked environment.

Kerberos [1] is an authentication service developed at in the mid-'80s as part MIT (Massachusetts Institute of Technology) that uses symmetric key encryption techniques and a key distribution center; it is an add-system that can be used with existing network.

Kerberos provides a means of verifying the identities of principals on an open (unprotected) network. This is accomplished without relying on authentication by the host operating system, without basing trust on host address, without requiring physical security of all the hosts on the network, and under the assumption that packets travelling along the network can be read, modified, and inserted at will.

Kerberos performs authentication under these conditions as a trusted third party authentication service by using conventional cryptography. It trusted in the sense that each of its clients believes Kerberos's judgments as to the identity of each of its other clients to be accurate.

The problem that Kerberos addresses is this: a distributed system in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restricted access to authorized users and to be able to authenticate requests for service.

In this system the following three threats exist:

1. A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
2. A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
3. A user may eavesdrop on exchanges and use a reply attack to gain entrance to a server or to disrupt operations.

In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access. Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. Kerberos provides the following requirements:

1. Secure: a network eavesdropper should not be able to obtain the necessary information to impersonate a user.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, February 2022
Impact Factor: 6.252

2. Reliable: Kerberos should be highly reliable and should employ distributed server architecture, with one system able to back up another.
3. Transparent: the user should not be aware that the authentication is taking place, beyond the requirement to enter a password.
4. Scalable: the system should be capable of supporting large numbers of clients and servers.

II. KERBEROS AUTHENTICATION DIALOGUE

A full-service Kerberos environment, consisting of a Kerberos server, a number of clients and a number of Application servers, requires that the Kerberos server must have the user ID (UID) and hashed passwords of all participating users in its database.

All users are registered with the Kerberos server. Such an environment is referred as a realm. Moreover, the Kerberos server must share a secret key with each server and every server is registered with the Kerberos server. A simple authentication procedure must involve three steps:

1. The client C requests the user password and then sends a message to the AS of the Kerberos system that includes the user's ID, the server's ID and the user's password.
2. The AS check its database to see if the user has supplied the proper password for this user ID and whether this user is permitted access to the server V. If both tests are passed, the AS accepts the user as authentic and must now convince the server that this user is authentic. Thus the AS creates and sends back to C a ticket that contains the user's ID and network address and the server's ID. Then it is encrypted with the secret key shared by the AS and the server V.
3. C can now apply to V for the service. It sends a message to V containing C's ID and the ticket. V decrypts the ticket and verifies that the user ID in the ticket is the same of the one which came with the ticket. If these two match, the server grants the requested service to the client.

This scheme is correct, but this is not enough in a distributed environment: it has some leaks related to security and requires that the user introduce the password every time he needs to access to a service. Indeed, the client sends its password unencrypted to the AS; an eavesdropper could capture the password and use any services accessible to the victim. Moreover, it is better to minimize the number of times that the user has to enter a password.

To solve these additional problems, we introduce a scheme for avoiding plaintext passwords and a new server, known as the **Ticket-Granting Server** (TGS). The new service issues tickets to users who have been authenticated to AS. Each time the user require access to a new service, the client applies to the TGS using the ticket supplied by the AS to authenticate itself. The TGS then grants a ticket to the particular service and the client saves this ticket for future use. The new scenario (Fig. 1) is as follow:

1. The client request a ticket-granting ticket on behalf of the user by sending its user's ID to the AS, together with the TGS ID, indicating a request to use the TGS service.
2. The AS responds with a message, encrypted with a key derived from the user's password that contains the ticket for the TGS. The encrypted message also contains a copy of the session key used by C and the TGS. In this way, only the user's client can read it. The same session key is included in the ticket, which can be read only by the TGS. Now C and the TGS share a common key.
3. Armed with the ticket and the session key, C is ready to approach the TGS. C sends the TGS a message that includes the ticket plus the ID of the requested service. In addition, C transmits an authenticator, which includes the ID and address of C's user and a timestamp; it is encrypted with the session key known only by C and TGS. Unlike the ticket, which is reusable, the authenticator is intended to use only once and has a very short lifetime. The TGS can decrypt the ticket with the key that it shares with the AS. This ticket indicates that the user C has been provided with the session key Kc. Thus the TGS decrypt the authenticator with Kc gained from the ticket and check the name and the address from the authenticator with that of the ticket and with the network address of the incoming message. If all match, then the TGS is assured that the sender of the ticket is indeed the ticket's real owner.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, February 2022
Impact Factor: 6.252

4. Then the TGS replies to the client, sending a message encrypted with the common key that they share. It includes a new session key to be used with the server V that must provide the service, the ID of V, the ticket valid for a specific service and the timestamp of the ticket. The ticket itself includes the new session key.
5. C now has a reusable service-granting ticket for V. When C presents this ticket, it also sends an authenticator. The server can decrypt the ticket, recover the session key and decrypt the authenticator. Finally, at the conclusion of this process, the client and the server share a secret key. This key can be used to encrypt future messages between the two or to exchange a new session key for that purpose.

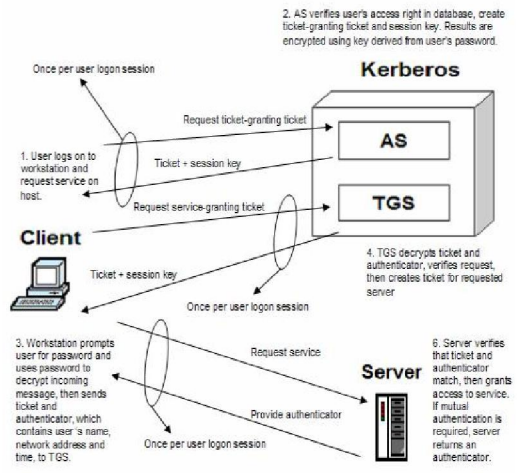

Fig. 1.1 Kerberos Authentication Dialogue

The Kerberos system is also able to manage more complicated situations which involve more than one realm. Usually, networks of clients and servers under different administrative organizations constitute different realms. Sometimes, user in one realm may need access to server in other realms and they need to be authenticated before to use services provided by those servers. Kerberos has a mechanism for supporting such inter-realm authentication. The only requirement requested is that the Kerberos server in each interoperating realm shares a secret key with the server in the second realm. The two Kerberos server are registered with each other.

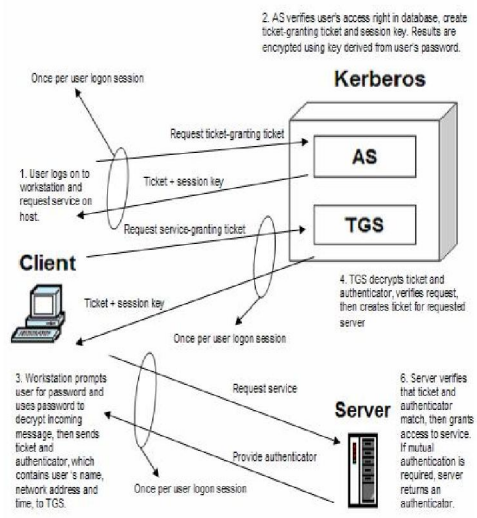

Fig. 1.2 Kerberos Authentication Dialogue

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, February 2022
Impact Factor: 6.252

The communication mechanism between a client and a server in two different realms is the following:

1. The client C in the α realm needs a ticket in order to communicate with the server in the β realm. Thus, the user's client follows the usual procedures to gain access to the local TGS and then request a ticket-granting ticket for the remote TGS in realm β.
2. The client can then apply to the remote TGS for a service-granting ticket valid for the desired server in that different realm.
3. The client can now use the ticket obtained from the remote TGS with the server V in the other realm.

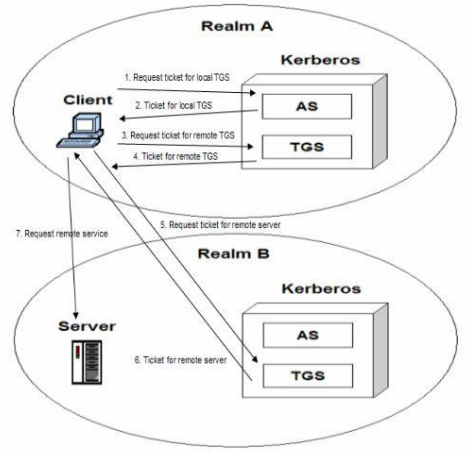

Fig. 2 Inter realm communication

III. KERBEROS LIMITATIONS

Since Kerberos has much strength, it has a number of limitations and some weakness in its protocol which has made Kerberos is not as extended as it should be. One of the main problems of Kerberos (which is not relative to security of the protocol) is that any application which wants to use the Kerberos protocol, have to be modified in the code in order to establish a secure communication. This means high costs in terms of time and money, and is not reliable for all applications nor enterprises.

The timestamps are used to establish secure communication. Therefore, servers have to be synchronized within no more than a few minutes, and for instance, times servers are required to synchronize often their clocks. But then, "if a host can be misled about the correct time, a stale authenticator can be replayed without any trouble at all". A solution to this problem could be performed by "adding challenge/response as alternative to time-based authentication".

Another problem relative to security relies on the high dependency on the Kerberos server. If this server goes down, then the whole network goes down as well. This is something very expensive and not desirable in a distributed system. In addition to this, all security relies on the server, so if someone can access the Kerberos server, then all network connection could be hacked.

There is another potential problem which is that someone could steal from the network the message from the AS to the client, this message, as we have seen above, is encrypted by the client key which is derivate from the client password. Therefore, someone could try to guess this key and then he would be able to identify himself as the original client. "Kerberos is not effective against password guessing attacks; if a user chooses a poor password, then an attacker guessing that password can impersonate the user". A solution proposed is the using of "exponential key exchange to provide an additional layer of encryption [...] it involves the two parties exchanging numbers that each can use to compute a secret key. An outsider, not knowing how the numbers were calculated, cannot easily derive the key".

Kerberos by itself cannot guarantee that the password will not travel along the net. It can happen that "the user enters a password to a program that has already been modified by an attacker (a Trojan horse), or if the path between the user and

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, February 2022
Impact Factor: 6.252

the initial authentication program can be monitored, then an attacker may obtain sufficient information to impersonate the user." Kerberos must be combined with other techniques to address these limitations.

Kerberos is designed to authenticate clients which are users, but "Kerberos is not a host-to-host protocol". Even thougth, in last version 5, it has been extended to user-to-user authentication.

These limitations have made Kerberos became a protocol not as extended nor used as it should be. Most of the networks only need an encrypted communication protocol to establish a minimum security; therefore sometimes Kerberos is replacing by programs as simple and transparent as SSH.

IV. KERBEROS' APPLICATIONS

Cisco's implementation of Kerberos client support

To implement a Kerberos security system with Cisco router remote, users attempting network services must pass through three layers of security before they can access network services.

The first one is called Authentication to the Boundary Router and describes the operations to follow in the authentication process. A remote user who successfully initiates a PPP session to the corporate site is prompted by the router in order to register him with login and password. Although in this phase the user is inside the firewall, to gain access to the network services, it still must authenticate to the Key Distribution Centre (KDC). This because the TGT issued by the KDC is stored on the router and is not useful for additional authentication unless the user physically logs on the router.

Therefore the next step is obtaining a TGT from a KDC. At this point, the client launches the KINIT program which is part of the software provided with Kerberos protocol. KINIT finds the user's identity and requests a TGT from the KDC. When the KINIT program receives the encrypt TGT, it prompts the user for the password to decrypt the ticket if it is successfully, the user has a TGT and can communicate securely with the KDC.

The third and last layer describes how the client, with a TGT, authenticates to network services within a given Kerberos realm. To make it possible the router must share a secret key with the KDC. To do this, you must give the router a copy of the SRVTAB you extracted on the KDC. This file contains the passwords or randomly generated keys for the service principals you entered into the KDC database.

The most secure method to copy SVRTAB files to the hosts in your Kerberos realm is to copy them manually in each host in turn. For the router, instead, you must transfer them via the network using the TFTP. Finally the user is authenticating to the network service. The entire process described above is repeated each time a user wants to access a network service in the Kerberos realm.

V. CONCLUSION

Kerberos isn't the only encryption protocol available. There are multiple ways to encrypt data and this holds true for many types of different applications. Email encryption protocols, for example, are a breed all of their own. With a product that has been researched and developed for over 8 years, it is generally expected that the product should be well polished. Kerberos doesn't fail to deliver, and this can be seen by looking at all the vendors who use it. Cisco, Microsoft, Apple, and many others rely on this faithful three-headed dog for network security. Authentication is critical for the security of computer systems. Without knowledge of the identity of a principal requesting an operation, it's difficult to decide whether the operation should be allowed. Traditional authentication methods are not suitable for use in computer networks where attackers monitor network traffic to intercept passwords. The use of strong authentication methods that do not disclose passwords is imperative. The Kerberos authentication system is well suited for authentication of users in such environments.

ACKNOWLEDGEMENT

I would like to show my sincere gratitude towards Prof. Sandip Shukla, HOD, Department of Computer Engineering, Guru Gobind Singh Polytechnic & Research Centre, Nashik for his valuable guidance throughout study of review papers and to keep me motivated. I am also thankful to Dr. N. G. Nikam, Principal, Guru Gobind Singh Polytechnic,& research Centre, Nashik for their great support.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, February 2022
Impact Factor: 6.252

REFERENCES

[1]. J. Kohl C. Neuman, RFC 1510, The Kerberos Network Authentication Service (V5), September 1995

[2]. William Stallings, Cryptography and Network security (Principle and Practice), Upper Saddle River N.J., Prentice Hall 1999

[3]. Steven M. Bellowin and Michael Merit from AT&T Bell Laboratories, Limitations of the Kerberos Authentication System, Winter '91 USENIX Conference Proceedings, USENIX Association, 1991

[4]. 1992--2001 Cisco Systems, Inc. All rights reserved http://www.cisco.com/univercd/cc/td/doc/product/software /ios121/121cgcr/secur_c/scprt2/scdkerb.htm, August 2001

[5]. Microsoft Corporation. All rights reserved, Windows 2000 Kerberos Authentication - White Paper http://www.microsoft.com/technet/treeview/default.asp?url =/TechNet/prodtechnol/windows2000serv/deploy/kerberos.asp

[6]. Mahmoud T. El-Hadidi, Nadia H. Hegazi, Heba K. Aslan, Performance Analysis of the Kerberos Protocol in a Distributed Environment, 2nd IEEE Symposium on Computers and Communication (ISCC 1997).