GOVERNMENT COLLEGE OF ENGINEERING, JALGAON
(An Autonomous Institute of Government of Maharashtra)
Department of Computer Engineering

Name: Vaibhav Bharat Sontakke	PRN No:
1942207 Class: T. Y. B. Tech.	Batch: L4
Date of Performance: __________	Date of Completion: ____________
Subject: DSL	Sign. of Teacher with Date: __________

Experiment No.: 09

Title: To study Kerberos

Theory:

Kerberos

System security and integrity within a network can be unwieldy. It can occupy the time of several administrators just to keep track of what services are being run on a network and the manner in which these services are used.

Further, authenticating users to network services can prove dangerous when the method used by the protocol is inherently insecure, as evidenced by the transfer of unencrypted passwords over a network using the traditional FTP and Telnet protocols.

Kerberos is a way to eliminate the need for protocols that allow unsafe methods of authentication, thereby enhancing overall network security.

What is Kerberos?

Kerberos is a network authentication protocol created by MIT, and uses symmetric-key cryptography[18]to authenticate users to network services, which means passwords are never actually sent over the network.

Consequently, when users authenticate to network services using Kerberos, unauthorized users attempting to gather passwords by monitoring network traffic are effectively thwarted.

Advantages of Kerberos

Most conventional network services use password-based authentication schemes. Such schemes require a user to authenticate to a given network server by supplying their username and password. Unfortunately, the transmission of authentication information for many services is unencrypted. For such a scheme to be secure, the network has to be inaccessible to outsiders, and all computers and users on the network must be trusted and trustworthy.

Even if this is the case, a network that is connected to the Internet can no longer be assumed to be secure. Any attacker who gains access to the network can use a simple packet analyzer, also known as a packet sniffer, to intercept usernames and passwords, compromising user accounts and the integrity of the entire security infrastructure.

The primary design goal of Kerberos is to eliminate the transmission of unencrypted passwords across the network. If used properly, Kerberos effectively eliminates the threat that packet sniffers would otherwise pose on a network.

Disadvantages of Kerberos

Although Kerberos removes a common and severe security threat, it may be difficult to implement for a variety of reasons:

o Migrating user passwords from a standard UNIX password database,such as**/etc/passwd** or **/etc/shadow**, to a Kerberos password database can be tedious, as there is no automated mechanism to perform this task. Refer to Question 2.23 in the online Kerberos FAQ: http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html

o Kerberos has only partial compatibility with the Pluggable AuthenticationModules (PAM) system used by most Red Hat Enterprise Linux servers.Refer
to Section 42.6.4, “Kerberos and PAM”for more information about this issue.

o Kerberos assumes that each user is trusted but is using an untrusted host on an untrusted network. Its primary goal is to prevent unencrypted passwords from being transmitted across that network. However, if anyone other than the proper user has access to the one host that issues tickets used for authentication — called the *key distribution center* (*KDC*) — the entire Kerberos authentication system is atrisk. o For an application to use Kerberos, its source must be modified to make the appropriate calls into the Kerberos libraries. Applications modified in this way are considered to be *Kerberos-aware*, or *kerberized*. For some applications, this can be quite problematic due to the size of the application or its design. For other incompatible applications, changes must be made to the way in which the server and client communicate. Again, this may require extensive programming. Closed-source applications that do not have Kerberos support by default are often the most problematic.

o Kerberos is an all-or-nothing solution. If Kerberos is used on the network,any unencrypted passwords transferred to a non-Kerberos aware service is at risk. Thus, the network gains no benefit from the use of Kerberos. To secure a network with Kerberos, one must either use Kerberos-aware versions of *all* client/server applications that transmit passwords unencrypted, or not use *any* such client/server applications atall.

Kerberos Terminology

Kerberos has its own terminology to define various aspects of the service. Before learning how Kerberos works, it is important to learn the following terms.

authentication server (AS)

A server that issues tickets for a desired service which are in turn given to users for access to the service. The AS responds to requests from clients who do not have or do not send credentials with a request. It is usually used to gain access to the ticket-granting server (TGS) service by issuing a ticket-granting ticket (TGT). The AS usually runs on the same host as the key distribution center (KDC).

ciphertext

Encrypted data.

client

An entity on the network (a user, a host, or an application) that can receive a ticket from Kerberos.

credentials

A temporary set of electronic credentials that verify the identity of a client for a particular service. Also called a ticket.

credential cache or ticket file

A file which contains the keys for encrypting communications between a user and various network services. Kerberos 5 supports a framework for using other cache types, such as shared memory, but files are more thoroughly supported.

crypt hash

A one-way hash used to authenticate users. These are more secure than using unencrypted data, but they are still relatively easy to decrypt for an experienced cracker.

GSS-API

The Generic Security Service Application Program Interface (defined in RFC 2743 published by The Internet Engineering Task Force) is a set of functions which provide security services. This API is used by clients and services to authenticate to each other without either program having specific knowledge of the underlying mechanism. If a network service (such as cyrus-IMAP) uses GSS-API, it can authenticate using Kerberos. **hash**

Also known as a hash value. A value generated by passing a string through a hash function. These values are typically used to ensure that transmitted data has not been tampered with.

hash function

A way of generating a digital "fingerprint" from input data. These functions rearrange, transpose or otherwise alter data to produce a hash value.

key

Data used when encrypting or decrypting other data. Encrypted data cannot be decrypted without the proper key or extremely good fortune on the part of the cracker.

key distribution center (KDC)

A service that issues Kerberos tickets, and which usually run on the same host as the ticket-granting server (TGS).

keytab (or key table)

A file that includes an unencrypted list of principals and their keys. Servers retrieve the keys they need from keytab files instead of using kinit. The default keytab file is /etc/krb5.keytab. The KDC administration server, /usr/kerberos/sbin/kadmind, is the only service that uses any other file (it uses /var/kerberos/krb5kdc/kadm5.keytab).

kinit

The kinit command allows a principal who has already logged in to obtain and cache the initial ticket-granting ticket (TGT). Refer to the kinit man page for more information.

principal (or principal name)

The principal is the unique name of a user or service allowed to authenticate using Kerberos. A principal follows the form root[/instance]@REALM. For a typical user, the root is the same as their login ID. The instance is optional. If the principal has an instance, it is separated from the root with a forward slash ("/"). An empty string ("") is considered a valid instance (which differs from the default NULL instance), but using it can be confusing. All principals in a realm have their own key, which for users is derived from a password or is randomly set for services. **realm**

A network that uses Kerberos, composed of one or more servers called KDCs and a potentially large number of clients.

service

A program accessed over the network.

ticket

A temporary set of electronic credentials that verify the identity of a client for a particular service. Also called credentials.

ticket-granting server (TGS)

A server that issues tickets for a desired service which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.

ticket-granting ticket (TGT)

A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

unencrypted password

A plain text, human-readable password.

How Kerberos Works

Kerberos differs from username/password authentication methods. Instead of authenticating each user to each network service, Kerberos uses symmetric encryption and a trusted third party (a KDC), to authenticate users to a suite of network services. When a user authenticates to the KDC, the KDC sends a ticket specific to that session back to the user's machine, and any Kerberos aware services look for the ticket on the user's machine rather than requiring the user to authenticate using apassword.

When a user on a Kerberos-aware network logs in to their workstation, their principal is sent to the KDC as part of a request for a TGT from the Authentication Server. This request can be sent by the log-in program so that it is transparent to the user, or can be sent by the kinit program after the user logs in.
The KDC then checks for the principal in its database. If the principal is found, the KDC creates a TGT, which is encrypted using the user's key and returnedto thatuser.

The login or kinit program on the client then decrypts the TGT using the user's key, which it computes from the user's password. The user's key is used only on the client machine and is not transmitted over the network.

The TGT is set to expire after a certain period of time (usually ten to twenty four hours) and is stored in the client machine's credentials cache. An expiration time is set so that a compromised TGT is of use to an attacker for only a short period of time. After the TGT has been issued, the user does not have to re-enter their password until the TGT expires or until they log out and log in again.

Whenever the user needs access to a network service, the client software uses the TGT to request a new ticket for that specific service from the TGS. The service ticket is then used to authenticate the user to that service transparently.

Conclusion :-

In this Practical, we studied about Kerberos

DOC: 04/05/2021

Miss. Archana Chitte

Name & Sign of Course Teacher