Now days, almost all networks have firewalls installed to protect them from the dangers of the un-trusted outside world of the Internet. When firewalls first came to the scene, they were nowhere near good enough to protect the Network completely. However, with the passage of time, the quality of firewalls has increased to such a level that the present day firewall systems make the internal trusted network almost 100% safe.

They can easily be configured to allow only certain kinds of data to pass through and even can be used to set which ports can be accessed from the un-trusted network (Internet) and which ports are accessible from the internal trusted network. Some good ones also scan all attachments going in and out for viruses and ensure that no confidential data is going out of the company. The present day firewalls have really made life quite easier for the system administrating by giving more than a little protection from the Outside world. However, one area where the firewalls falter is if the attach is from within the trusted internal network or in other words, the attacker is doing something wrong, something which he is not supposed to do from within the network and not through the Internet.

Say for example, you have a well configured; firewall installed at your company's main server and it scans all incoming email attachments for viruses. Now, if you get a virus attach from outside the internal trusted network and though the Internet, then normally the firewall will either delete or warn you about it. However, if the virus coder, is working for you and is within the internal trusted network, then a firewall would not be able to do anything about it and the virus will spread quite easily.

NOTE: The above is just an example taken to ensure that you understand.

So, now, I hope you realize that only a Firewall is not sufficient for a network and it also requires something for attacks from internal systems.

This is where the Kerberos comes in. Kerberos is a network authentication protocol, which provides for the verification of identities within a heterogeneous distributed networked environment. It is the de facto standard for authentication, which gets it name from the three-headed dog in Greek Mythology.

For complete reference and details about Kerberos authentication protocol, refer to the RFC 1510

Now, within an internal network, the greatest danger lies in the fact that anyone can easily pick up or sniff out confidential data like company plans, passwords and even credit card numbers while this data is being transferred from one system to another within the same network.

Let us take an example, to understand better. Say, you are on a client, which is connected to the main server, which provides services to all clients connected to it. Now, when you connect to the server to check your mail, then your email client sends your Username and Password to the internal network server, so that you can be authenticated. You may say that this is pretty much safe and how can it possible harm me? Well, you are wrong. Now, when your machine sends your Username and Password to the server, then this information does not reach the destination server directly. The data has to pass through other machines and sometimes if your network is large, then even through other servers before it reaches the destination. Now, anyone having access to those systems through which your data passes through, can easily sniff out your data and in this case can find out your Username and Password, using which he can check your mail.

To solve all the above problems (and many more) there is the Kerberos. The Kerberos not only ensures that no one sniffs data out, but also ensures the integrity of the client and server to prevent impersonation. This basically means that it ensures that no one can fool the server into thinking that it is some other system.

Now, to understand how exactly, Kerberos proves to be as good as it is, let us learn how the Kerberos protocol works.

The most popular Network Authentication providing software, Kerberos is constituted of 3 main parts or sections-:

1.) The Authentication Server or AS

2.) The Ticket Granting Server or TGS

3.) The Actual Encryption process or algorithm

In a Network with Kerberos installed or enabled, the Authentication Server or the AS acts as the head or the central unit, which ensures the authenticity of the client and server and also prevents data sniffing.

One good thing about the Kerberos Authentication system is that is a dual-authentication system, which means that it not allows the server to verify the identity of the client but also vise-a-versa.

The Authentication Server acts as the secretary of both the client and the server. The client and the server in order to communicate with each other, have to have a connection with the AS. (The AS verifies identities of both the client and the server.) Only once the AS has verified the authenticity of both the client and the server, can they start to communicate with each other.

Kerberos: The Working
So when does Kerberos jump in? Well, as soon as you want to login and type in your Username in the space provided. As soon as you type the confidential information, the Kerberos sends it to the Authentication Server or the AS. Then, the AS replies to the client with the session key and something called the Ticket Granting Ticket. Both the session key and the Ticket Granting ticket are encrypted by the user's key. Now, before we go on, I think there is need to explain certain things involved in the above process.

Now, you must remember that the client and the Authentication Server and the client share an encryption key, which is used to encrypt data. This encrypted data is understandable (de-cryptable) by only the AS and the client. This encryption key is generated from the User's Password. This means that, passing the User's password through a certain predefined formula derives this encryption key. Similarly all Servers, which provide services to clients, share an encryption key with the AS.

So this system of client-AS and server-AS encryption keys ensures that no one else can sniff the data.

Now, we come to the Ticket Granting Ticket, which is sent along with the session key[The session key is sent to the client by the AS, so that the client can start to communicate with the Ticket Granting Server or TGS.] to the client by the AS. A ticket is nothing but a certificate of authenticity given to the client by the AS to prevent impersonation. The ticket is readable only by the client system for which it is meant to be and the AS. The Ticket Granting Ticket also makes the Kerberos system efficient as it removes the need of repeating the initial process again and again.

Now, once the client receives the session key and the TG Ticket, it derives the client's key from the user's password and tries to use this generated key to decrypt the TG Ticket and the Ticket Granting Server key or the TGS key. If the client is able to decrypt these two, then the password is correct else wrong.

Now, say you want to then, use the POP services of the mail server to read your mail, then what happens is that, the client sends a request to The Ticket Granting Server or TGS. The client encrypts important network information and details about the request with the TGS key and sends this encrypted data to it.

If this is found to be valid, then TGS issues a ticket to the client which contains the following-:

1.) Username

2.) Address

3.) Service Name

4.) Lifespan

5.) Timestamp

6.) Other Session Key details.

An important thing to note here is that, for communication between the client and the server to actually take place, they should share the same key.

The TGS generates two copies of this session key, one encrypted with TGS key for the client and the other with application server key. Using the TGS key, the client then, decrypts the session key meant for it, and the session key for the application server is sent to the destination.

When the server receives the session key, and once it is decrypted, it knows that this particular client is trying to contact it. The server too has a procedure to ensure the authenticity of the client. It sends a random number in plain text to the client. The client then decrypts it with the session key (which they both have in common) and sends it back to the server. On receiving this encrypted text, it can ensure that the client is not an impersonator, as some other client cannot perform the same encryption.

Kerberos ensures that Sniffing out data is not that easy, as transfer of all data, even the Keys, is done in encrypted form. The encryption technique used by Kerberos is Data Encryption Algorithm or DES.

However, there is still a slight hole in the Kerberos system. You see, during the time when the Password is sent to the AS in the first step, it travels through the Network in unencrypted form. This is one time, when the Kerberos system can be exploited.

Windows 2000 is I think the first Operating System, which uses Kerberos as the standard authentication method. Anyway, now that you know how exactly, the Kerberos Authentication System works, let us move on to how to find out if your ISP is running it or not? Also, I highly recommend reading the RFC 1510. This is for those who want even the tiniest of details about this system.

How do I find out if my ISP is running Kerberos?
NOTE: In this section, I am assuming that you have enabled the Bring Up the Post Dial Up Screen option.

Well, the router of your ISP to which you initially connect to, holds the key. Almost all of you must have seen the Post Dial Up Screen, which comes up, where you have to enter your Username and Password. Now, this Post Dial Up Screen is actually your ISP’s router prompt. There is a secret (Well, not exactly secret) router command, which will let you find out if your ISP has implemented the Kerberos protocol.

The following is a log which contains my comments of what I did to find out whether my ISP is using Kerberos or not.

```
User Access Verification

Username: ankit
Password:

NP-NAS3>help
Help may be requested at any point in a command by entering
a question mark '?'.  If nothing matches, the help list will
be empty and you must backup until entering a '?' shows the
available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a
   command argument (e.g. 'show ?') and describes each possible
   argument.
2. Partial help is provided when an abbreviated argument is entered
   and you want to know what arguments match the input
   (e.g. 'show pr?'.)
```

[Ankit: help is not the right command, let me try ‘?’]

```
NP-NAS3>?
Exec commands:
```

```
access-enable    Create a temporary Access-List entry
access-profile   Apply user-profile to interface
attach           attach to system component
clear            Reset functions
connect          Open a terminal connection
disable          Turn off privileged commands
disconnect       Disconnect an existing network connection
enable           Turn on privileged commands
exit             Exit from the EXEC
help             Description of the interactive help system
lat              Open a lat connection
lock             Lock the terminal
login            Log in as a particular user
logout           Exit from the EXEC
mrinfo           Request neighbor and version information from a multicast
                 router
mstat            Show statistics after multiple multicast traceroutes
mtrace           Trace reverse multicast path from destination to source
name-connection  Name an existing network connection
pad              Open a X.29 PAD connection
ping             Send echo messages
ppp              Start IETF Point-to-Point Protocol (PPP)
resume           Res
```