

The Kerberos Authentication Service

By: Cule Stevan
ID#: 0047307

SFWR 4C03

April 4, 2005

The Kerberos Authentication Service

The need for security:

A majority of businesses in North America are heavily dependent on software for day to day and long term strategic operations eg: banks. This gave rise to software solution suites such as the enterprise resource planning, supply chain management, customer management systems, human resource enterprise system and many other aimed at helping organizations gain competitive advantage. This reliance on software means that company sensitive data is now floating around in cyberspace available to anyone who can find it. For these reasons the companies are focusing on security more and more. In recent past the common business term location, location, location can be now translated into security, security, security because the location is less and less physical.

Introduction to Kerberos:

Most of the experts would agree that many of the protocols used in the internet today are not secure. There exist a large number of applications which do not encrypt sensitive data such as passwords which are sent over the internet. Some applications that I have seen embed passwords inside an html page or do a post on an asp page. Needless to say, such data is almost guaranteed to be compromised by even a novice hacker. There has been an attempt to rectify this problem through the use of firewalls. Now, firewalls can do a fair job in protecting an organization's intranet from the dangers of malicious hackers coming from the internet. Nevertheless, there is a terrible flaw with this concept since most of the damage is historically done from inside the company's intranet.¹ Even if you attempt to prevent "inside hackers" from damaging your organization with firewalls, you will end up restricting everyone including honest people and dramatically reduce the productivity levels. Kerberos was created precisely to correct the above mentioned problems. Kerberos is a network authentication protocol specifically designed for the purpose of providing powerful authentication for client/server applications. It uses strong secret-key cryptography in order to allow the client to prove its identity to the server across a connection which is almost entirely insecure. Kerberos is a software based network authentication protocol created by MIT and is really available free of charge.² Suppose that you use the rlogin unix command to login onto your account on another machine. Now if the .rhosts file has been set correctly the rlogin program will not require the user to enter the password any more! (we did this in the SFWR 4F03 labs) This means that an attacker has a great chance of logging in as that user. To fix this you can force the user to enter the password at all times but this is time consuming and the password is still unencrypted.

How Kerberos attempts to solve the security problem?:

The key phrase in Kerberos is “the key.” It is used to encrypt messages before they are sent across internet. The Kerberos works similarly to the class discussion where by the encryption routine uses the key to create ciphertext and the decryption routine uses the key and the ciphertext to provide the plain text. Before we start explaining the inner details of Kerberos there are two assumptions we must keep in mind. The first assumption is that the user will not make a poor selection of its password since then the hacker can try every word in the dictionary to decrypt the message. The second assumption is that only the network connections are not secure and workstations are fairly secure.³ A client needs to access some service and it first sends out the request for a ticket to the Key Distribution Center (KDC) which creates a Ticket-Granting-Ticket (TGT) which it immediately encrypts using the client’s password as the key and sends the encrypted TGT back to the client. This is a smart way of making sure that only the original client gets the TGT. So at this point the client attempts to decrypt the Ticket-Granting-Ticket using its password and if he is successful it keeps the TGT as a proof of its identity. A really smart thing that the developers of this system invented is that the Ticket-Granting-Ticket expires after a specified time interval to prevent the possibility of a hacker from obtaining the client’s identity and using it to access various services. Now this TGT allows the client to ask for and hopefully obtain more tickets depending on the service required. The client uses TGT to talk to the Ticket-Granting-Service which also runs on KDC. Once the Ticket-Granting-Service verifies the user using the TGT a ticket for the desired service is issued. (Exhibit 1) In case you have not figured it out yet, if you possess the TGT that means that you have been authenticated by the Kerberos and you now have the right to ask for various services. Kerberos negotiates authenticated and possibly encrypted communications among any two points on the internet which means that our previous concern of using firewalls is now pretty much solved. Another really neat thing is that Kerberos is a single-sign-on system where you only need to type password once per session and still enjoy safe communication.⁴ A good analogy of the Ticket-Granting-Ticket is suppose that you buy a ticket to the big amusement park which allows you to go onto any ride. When you want a ride inside an amusement park you show your TGT to get another ride ticket which will allow you to enter the roller coaster. When you exit the park your TGT has expired and nobody else can use your TGT to enter the park again.

Architecture Details:

You might have wondered what is the KDC or the Key Distribution Center mentioned above. “Kerberos Server” refers to the Key Distribution Center. It implements the authentication service (AS) and the Ticket-Granting-Service (TGS). This KDC stores a copy of every password associated with every principal (client/user). There is also an administration server which runs on top of the KDC and allows remote manipulation of the Kerberos database.⁵ Kerberos

uses a DES (data encryption standard) algorithm for encryption. A DES is a symmetric cipher defined in the federal information processing standard number 46.⁶ This algorithm uses a 56 bit key and is vulnerable to brute force attacks. Kerberos also supports other algorithms such as CRC-32, MD4, MD5, and DES for checksum.

Kerberos Benefits:

Delegated authentication where services impersonate the client when accessing services on their behalf. More efficient authentication to servers since Kerberos does not need to go to the domain controller. Mutual authentication benefit since a party at either end of the network can identify the other party.⁷

Kerberos Weaknesses:

Kerberos assumes that it is running on a trusted host on a non-trusted network. So if your host is compromised in any way, so is Kerberos. One possibility is that the attacker steals user's tickets from a machine. He can use them until they expire. From above we know that Kerberos uses the user's password which is the encryption key as a primary security measure. If a user's password is stolen then Kerberos can not know the difference. Perhaps one of the main weaknesses is if a host on which KDC and its entire password database are running is compromised then the entire realm (unique login space) is also compromised. This can be done as easily as obtaining administrative privileges to the system. Another weakness is that a hacker can perform an off-line attack on a ticket by trying to decrypt it with a password. Fortunately Kerberos' designers have identified this and are using preauthentication to solve it which is simply places one security measure before the actual TGT is given.⁸ Furthermore, Kerberos' tickets are cached on the client's system. If an attacker can gain access to the cache memory, he can steal the tickets.

Conclusion:

The core of Kerberos architecture is the Key-Distribution-Center which holds the user authentication information. Kerberos authentication is secure because it does not use plain text, does not rely on the authentication by the operating system, does not base trust on IP addresses, and it does not require physical security of network hosts.⁹ You might want to know that the word Kerberos itself refers to the three headed Greek mythological creature

¹ http://web.mit.edu/kerberos/www/#what_is

² [ibid](#)

³ <http://www.isi.edu/gost/brian/security/kerberos.html#whatis>

⁴ <http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4/doc/krb5-user.html>

⁵ <http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html#whatis>

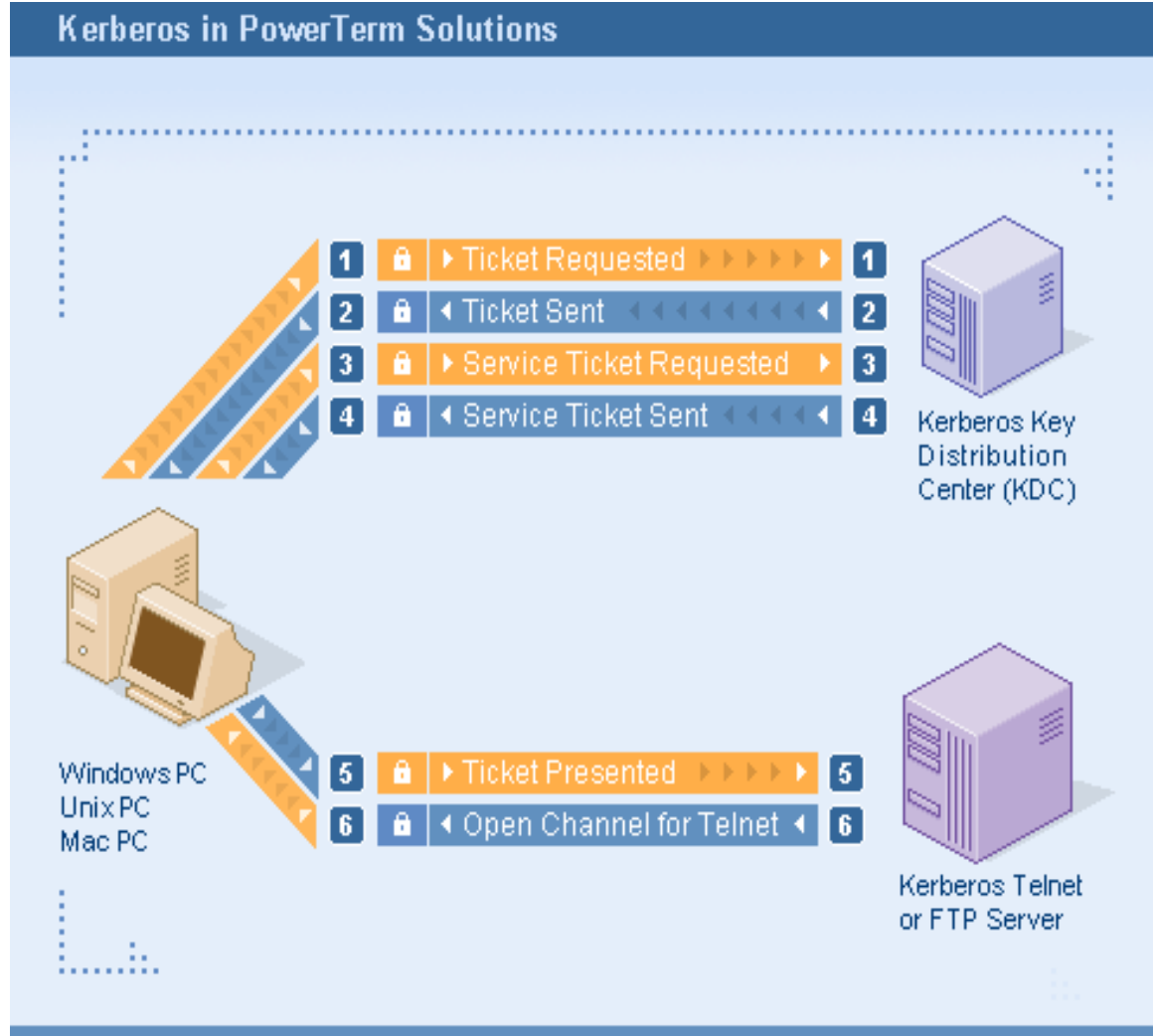
⁶ <http://www.tech-faq.com/des-data-encryption-standard.shtml>

⁷ http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/techref/en-us/Default.asp?url=/Resources/Documentation/windowsserv/2003/all/techref/en-us/w2k3tr_kerb_what.asp

⁸ <http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html#preauth>

⁹ <http://corky.net/2600/cryptology/kerberos.shtml>

Exhibit 1:



Reference:

<http://www.ericom.com/imgs/diagrams/kerberos.gif>

Appendix A – Useful Terminology

Terms	
Authentication	Verification, or proof, of identity.
Authorization	Assigning privileges, or allowing access to resources, based on identity.
Principal	<p>The Kerberos identifier of a person or process. Kerberos V5 principals are of the form <i>primary/instance@realm</i>. Usually, ordinary user principals have null instances, while the principal of a person with a special role has an instance that indicates the nature of that role. Application servers often have principal names whose primary component defines the service, with the instance equal to the fully-qualified domain name of the host it's running on.</p> <p>Examples:</p> <p>UCB staff member with employee ID 012345678 would have a principal name of <i>012345678</i>.</p> <p>If that same staff member happened to be system administrator for a number of hosts, it might be useful also to have a separate principal of <i>012345678/root</i>, for use when performing privileged operations on those hosts. But this would be just for convenience and is by no means required.</p> <p>A kerberized telnet service at <i>socrates.berkeley.edu</i> would be registered with a Kerberos principal name of <i>host/socrates.berkeley.edu</i>.</p> <p>Kerberos administrators, by convention, have a principal name with an instance of <i>admin</i> (for example, <i>xxxxx/admin</i>).</p> <p>(The realm portion of a principal name may normally be omitted if the principal is in the local Kerberos realm, which is determined from a configuration file on the principal's host machine).</p>
Passphrase	A user secret which, unlike a password, may contain blanks.

KDC	Key Distribution Center = Kerberos servers + database of principals.
AS	Authentication Service. The component of Kerberos that initially authenticates a principal.
Realm	The "jurisdiction" of a Kerberos database. The realm includes the KDC plus the set of all principals registered in the database. A realm name is normally a DNS domain or subdomain name, in upper-case.
Ticket	A data structure obtained from Kerberos, which is presented to a service or application to authenticate the ticket holder. A ticket contains information about the client principal plus a session key that was randomly-generated by Kerberos, all encrypted in the secret Kerberos key of the server to which it will be presented. The client has its own copy of this session key, which was contained in the credentials that included this ticket. Only the server can decrypt the ticket to receive its copy of the session key; once this is done, the server and client share a common secret key. That key is used to complete the authentication process and may also be employed by client and server to encrypt their entire session. (See Credentials below).
TGT	Ticket-Granting Ticket. A special ticket issued by the AS, allowing a principal to authenticate itself to the TGS , for the purpose of obtaining service credentials .
Credentials	The combination of a ticket and its session key. The client sends the ticket to the server in order to distribute the session key and also sends an authenticator which is encrypted in that session key.
Service Ticket Service Credentials	This refers to a ticket or credentials used with an application server, as opposed to a TGT or initial credentials.
TGS	Ticket Granting Service. The component of Kerberos that issues service credentials.

Authenticator

A short-lived packet of information that is generated by a kerberized client each time it connects to a server. The authenticator is encrypted in the session key the client shares with the server and has a very short lifetime (normally 5 minutes). It is presented to the server along with the service ticket (from which the server extracts the session key). The authenticator (rather than the ticket) is what really proves the client's identity, since an imposter may have stolen the ticket in transit from a prior session, but only the client who possesses the session key can generate a valid authenticator. To prevent an imposter from simply reusing a stolen authenticator (within its 5 minute lifetime), servers should refuse to accept the same authenticator more than once.

Reference: <http://www.net.berkeley.edu/kerberos/k5concepts.html>