Global Information Assurance Certification Paper

Terrence K Hudgen
GSEC Version 1.3

Network Security: Authentication Applications Kerberos and Public Key Infrastructure

Abstract

Networks that are used to store, process, or transmit sensitive information must provide appropriate protection to prevent undesirable events such as compromise of information or denial of service. This document will show how to improve the security of a network through the use of authentication applications. Kerberos and Public Key Infrastructure (PKI) are the two applications that will be examined. The pros and cons of each application will be discussed. Solutions to overcome the limitations of each application will be identified.

Network Security: Authentication Applications Kerberos and Public Key Infrastructure

INTRODUCTION

The most commonly known network is the Intranet, which is made up of hundreds of thousands of smaller networks. Other networks, such as local area networks (LANs) and wide area networks (WANs) can also be connected to the outside world. Once a computer is hooked up to devices outside of your office walls, you and your data become vulnerable. People may try to steal data, use your accounts for inappropriate purposes, or simply damage your system because they can. The same can be said for internal networks, such as Intranets, and semi-internal networks, such as Extranets, although to a somewhat lesser degree due to their limited public exposure.

The effects of a security compromise, whether by interception of sensitive transactions, information being modified in transit, or transmissions being fabricated could devastate an organization. As long as organizations rely upon public networks to support transactions without a widely adopted, open, and interoperable security solution, any transaction is susceptible to security breaches.

Security solutions for distributed networks are ever changing; the most discussed solutions in the network security field are: Kerberos and PKI. They both have their strengths and weaknesses. They both rely upon a centralized trust figure: the Key Distribution Center (KDC) for Kerberos and the Certificate Authority (CA) for PKI, and both use the application of cryptographic algorithms to secure network-based communications. They also differ from one another in numerous ways, both in the way they secure communications and the capabilities they bring to the table. This paper discusses the strengths and weaknesses of each technology and will provide solutions to their limitations.

INTRODUCTION TO KERBEROS

Kerberos is an authentication protocol that lets clients and servers reliably verify each other's identity before establishing a network connection. Developed at MIT in the late 1980s, Kerberos takes its name from the three-headed dog on Greek mythology that guards the entrance of

Hades. Instead of guarding the underworld, today's Kerberos brings a measure of security to a distributed computer environment, where one computer can access the resources of any other machine on a network. [Tipton, 1999] Like Kerberos the dog, Kerberos the protocol has three heads; two principals and one trusted third party. [Conray 2001]

A distributed security system provides a wide range of security services for distributed environments. Those services include authentication and message protection, as well as providing the ability to securely carry authorization information needed by applications, operating systems, and networks. Kerberos also provides the facilities necessary for delegation, where limited-trust intermediaries perform operations on behalf of a client.

Kerberos differs from many other distributed security systems in its ability to incorporate a very wide range of security technologies and mechanisms. That flexibility allows a mixture of security technologies and mechanisms to be used, as narrowly or broadly as required, while providing the economies of scale that a common, reusable, and technology-neutral Kerberos security infrastructure.

The basic Kerberos authentication process proceeds as follows: A client sends a request to the authentication server (AS) requesting 'credentials' for a given server. The AS responds with these credentials, encrypted in the client's key. The credentials consist of 1) a 'ticket' for the server and 2) a temporary encryption key (often called a "session key"). The client transmits the ticket (which contains the client's identity and a copy of the session key, all encrypted in the server's key) to the server. The session key (now shared by the client and server) is used to authenticate the client, and may optionally be used to authenticate the server. It may also be used to encrypt further communication between the two parties or to exchange a separate sub-session key to be used to encrypt further communication. [Neuman 2000]

The implementation can consist of one or more authentication servers running on physically secure hosts. The authentication servers maintain a database of principals (i.e., users and servers) and their secret keys. Code libraries provide encryption and implement the Kerberos protocol. In order to add authentication to its transactions, a typical network application adds one or two calls to the Kerberos library, which results in the

transmission of the necessary messages to achieve authentication.

The Kerberos protocol consists of several exchanges. A client can ask a Kerberos server for credentials by two methods. In the first approach, the client sends a clear text request for a ticket for the desired server to the AS. The reply is sent encrypted in the client's secret key. Usually this request is for a ticket-granting ticket (TGT) that can later be used with the ticket-granting server (TGS). In the second method, the client sends a request to the TGS. The client sends the TGT to the TGS in the same manner as if it were contacting any other application server that requires Kerberos credentials. The reply is encrypted in the session key from the TGT.

Once obtained, credentials may be used to verify the identity of the principals in transaction, to ensure the integrity of messages exchanged between them, or to preserve privacy of the messages. The application is free to choose whatever protection may be necessary.

To verify the identities of the principals in a transaction, the client transmits the ticket to the server. Since the ticket is sent in the clear and might be intercepted and reused by an attacker, additional information is sent to prove that the message was originated by the principal to whom the ticket was issued. This information is encrypted in the session key, and includes a timestamp. The timestamp proves that the message was recently generated and is not a replay. Encrypting the authenticator in the session key proves that a party possessing the session key generated it. Since no one except the requesting principal and the server know the session key this guarantees the identity of the client.

The integrity of the messages exchanged between principals can also be guaranteed using the session key. This approach provides detection of both replay attacks and message stream modification attacks. It is accomplished by generating and transmitting a hash function of the client's message, keyed with the session key. Privacy and integrity of the messages exchanged between principals can be secured by encrypting the data to be passed using the session key passed in the ticket, and contained in the credentials.

ADVANTAGES OF KERBEROS

As a distributed security service, Kerberos provides authentication, confidentiality, and integrity security capabilities. Those three elements of security are essential to any organization that wants to operate a secure environment. Kerberos uses encryption to provide each of these three security services and supports both public key cryptography (asymmetric) and secret key (symmetric) encryption for authentication; however, the core functionality of any Kerberos implementation relies on secret key encryption. [Hynes 2001] As a side-effect of the dual-key encryption scheme employed in the Kerberos model, a service-session key is generated which constitutes a shared secret between a particular client system and a particular service. This shared secret may be used as a key for encrypting the conversation between the client and the target service, further enhancing the security of Kerberized transactions.

One of the most significant benefits of using Kerberos is its standing as a relatively mature, IETF standards-based protocol (RFC 1510) supported by Windows 2000, Linux, and Windows platforms. Unlike many of its proprietary counterparts, Kerberos has been scrutinized by many of the top programmers, cryptologists and security experts in the industry. This public scrutiny has ensured and continues to ensure that any new weaknesses discovered in the protocol or its underlying security model will be quickly analyzed and corrected.

Another advantage is the tickets passed between clients and servers in the Kerberos authentication model include timestamp and lifetime information. This allows Kerberos clients and Kerberized servers to limit the duration of their users' authentication. While the specific length of time for which a user's authentication remains valid after his initial ticket issued is implementation dependent, Kerberos systems typically use small enough ticket lifetimes to prevent brute-force and replay attacks. In general, no authentication ticket should have a lifetime longer than the expected time required to crack the encryption of the ticket.

DISADVANTAGES OF KERBEROS

Kerberos was designed for use with single-user client systems. In the more general case, where a client system may itself be a multi-user system, the Kerberos authentication scheme can fall prey to a variety of ticket-stealing and replay attacks. The overall security of multi-user Kerberos client systems (file system security, memory protection, etc.) is therefore a limiting factor in the security of Kerberos authentication. No amount of cleverness in the implementation of a Kerberos authentication system can replace good system administration practices on Kerberos client and server machines.

Because Kerberos uses a mutual authentication model, it is necessary for both client machines and service providers (servers) to be designed with Kerberos authentication in mind. Many proprietary applications already provide support for Kerberos or will be providing Kerberos support in the near future. Some legacy systems and many locally-written and maintained packages, however, were not designed with *any* third-party authentication mechanism in mind, and would have to be re-written (possibly extensively) to support Kerberos authentication.

The Kerberos authentication model is vulnerable to brute-force attacks against the KDC (the initial ticketing service and the ticket-granting service). The entire authentication system depends on the trust ability of the KDC(s), so anyone who can compromise system security on a KDC system can theoretically compromise the authentication of all users of systems depending on the KDC. Again, no amount of cleverness in the design of the Kerberos system can take the place of solid system administration practices employed in managing the Kerberos KDC(s).

INTRODUCTION TO PKI

A PKI (public key infrastructure) enables users of an insecure public network such as the Internet to securely and privately exchange data using a public and a private cryptographic key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for digital certificates that can identify individuals or organizations and directory services that can store and, when necessary, revoke them. PKI is the underlying

technology that provides security for the SSL and HTTPS protocols, which are used extensively to conduct secure eBusiness over the Internet.

The public key infrastructure assumes the use of public key cryptography, which is the most common method on the Internet for authentication of a message sender or encryption of a message. Traditional cryptography involves the creation and sharing of a secret key for the encryption and decryption of messages. This secret key system has the significant flaw that if the key is discovered or intercepted by someone else, messages can easily be decrypted.

For this reason, public key cryptography and the public key infrastructure is the preferred approach on the Internet. A public key infrastructure consists of:

1. A certificate authority that issues and verifies digital certificates.
2. A registration authority that acts as the verifier for the certificate authority before a digital certificate is issued to a requestor.
3. One or more directories where the certificates (with their public keys) are held.
4. A certificate management system

In public key cryptography, a public and private key are created simultaneously by a certificate authority (CA). The private key is given only to the requesting party and the public key is made publicly available in a directory that all parties can access. The private key is never shared with anyone or sent across the Internet. By first encrypting data using an organization's or individual's public key it is possible to safely send the data across an insecure network such as the internet since only the holder of the associated private key will be able to decrypt it.

In addition to decrypting messages, private keys may also be used to digitally sign data as a way of both authenticating its origin and proving that data has not been tampered with. To create a digital signature the sender passes the data through a "hashing" algorithm, which returns a value known as a one-way hash. The one-way hash is unique to the data, but cannot be used to reproduce it. The sender encrypts the one-way hash using their private key to create a digital signature. The digital signature is sent in addition to the data.

The digital signature can be decrypted to obtain the one-way hash using the sender's public key (available from

a global directory of public keys). By passing the data through the same hashing algorithm and comparing the result with the one-way hash extracted from the digital signature the recipient can prove that the data was sent by the owner of the public key and has not been tampered with.

This dual use of keys leads to a conflict of interest in terms of key lifetime. For example if a private key used for digital signatures were stolen it could be used to forge the owner's identity and so should be destroyed as soon as its active life is over. In contrast, with a key pair used for encryption the private key should be archived for as long as possible, because if the private key were ever lost it would be impossible to retrieve messages encrypted with its public counterpart. It is therefore sensible to keep multiple copies of this private key.

ADVANTAGES OF PKI

The PKI approach to security does not take the place of all other security technologies; rather, it is an alternative means of achieving security. The following advantages of PKI have led to its emergence as an industry standard for securing Internet and e-commerce applications. PKI is a standards-based technology and it allows the choice of trust provider. It is highly scaleable. Users maintain their own certificates, and certificate authentication involves exchange of data between client and server only. This means that no third party authentication server needs to be online. There is thus no limit to the number of users who can be supported using PKI.

PKI allows delegated trust. That is, a user who has obtained a certificate from a recognized and trusted certificate authority can authenticate himself to a server the very first time he connects to that server, without having previously been registered with the system.

Another advantage of PKI is it offers non-repudiation whereas Kerberos does not. Non-repudiation ensures that strong and substantial evidence is available to the sender of message that the message has been delivered, and to the recipient, of the sender's identity, sufficient to prevent either from successfully denying having sent or received the message. This includes the ability of a third party to verify the integrity and origin of the message.

DISADVANTAGES OF PKI

PKI acceptance is a controversial issue. As explained by cryptography expert Bruce Schneier [2000], PKI increases some risks and fails to address some others. Drawbacks to implementing PKI include: client software trust, private key safekeeping and abuse, very high security requirements for verification servers, unclear certificate use guidelines, etc. If a user's private key is compromised, the user can reject transactions signed by it. Thus, the burden of protecting user private keys is the full responsibility of the user.

As stated earlier PKI is not a technology solution in and of itself. It is an enabling technology. It is also an investment, and the benefits of it do not immediately out weigh the cost of deployment. Because of this PKI deployment is a slow task and the number of fully functioning PKIs in existence today is quite small. While the early bird may catch the worm, it is usually not the first person through the mine field that survives, and that sentiment is seen in PKI development. PKIs are complex, and many thorny issues arise when they are built.

An additional disadvantage of PKI is that the standards supporting PKI are still developing. While a number of standards have distinguished themselves as the standards that will be used going forward, namely the Simple Certificate Enrollment Protocol (SCEP) and Certificate Management Protocol (CMP) standards, there are still too many being followed, with no single set achieving universal acceptance and application. The lack of approved standards could negatively affect the ability of organizations with a different interpretation of the standards to interoperate with other PKI implementations.
Solutions

SOLUTION

A way to overcome the limitations of Kerberos and PKI are to combine the two services. The new service will be bases on the PKI structure. The PKI structure would be used because the main disadvantage of Kerberos is the difficulty in administering and maintaining a Kerberos implementation; the entire idea behind PKI is to enable easier management of an organization’s cryptographic keys.

In addition, PKI can thrive in a widely distributed environment such as the Internet or Extranets due to cross-

certification and its design flexibility. PKI uses a distributed trust so that the day-to-day distribution of keys is conducted from a publicly accessible certificate repository. Whereas in a Kerberos implementation all the keys are stored on the KDC server (or set of KDC servers) and the KDC must always be available for authentication. The risk of compromise of the private key can be mitigated through the use of token or smart card-based technology, which recently has been made available for Kerberos implementations.

CONCLUSION

The need for an open and interoperable distributed network security "solution" is evident in today's computing environment. While a number of alternative solutions have been considered over the years, none appear as ready to address this need as Public Key Infrastructure and Kerberos do today.

While this paper has outlined the strengths and weaknesses of both PKI and Kerberos for internal use, and provided a solution those limitations. As shown, the two technologies are not mutually exclusive. Recent implementations of Kerberos, specifically the Windows 2000 implementation, support authentication via PKI-based digital certificates.

While Kerberos and PKI share some of the same advantages, each falls prey to some of the same disadvantages as well. Neither password-based authentication (Kerberos) nor certificate-based authentication (PKI) addresses security issues surrounding physical access to individual machines or passwords. Public-key Cryptography can verify that a private key used to sign some data corresponds to the public key in the certificate, but is unable to always maintain a tight association between the user and their certificate. [Hynes, 2001] The two technologies should be combined in order to take advantage of the advantages of each technology and reduce their limitations.

REFERENCES

[Bradsheer 1995] Bradshear, K. (1995, May) Kerberos Reference Page. Retrieved March 1, 2001, from Carnegie Mellon University Web site: http://www.contrib.andrew.cmu.edu/~shadow/kerberos.html.

[Conroy 2001J] Conray-Murray, A. (2001, July). Kerberos: Computer Security's Hellhound. Network Magazine, Vol. 16 No. 7, 40-45.

[Conray 2001N] Conray-Murray, A. (2001, November). Strategies & Issues: Public Key Infrastructure Nuts and Bolts. Retrieved April 5, 2001 from Network Magazine website: http://www.networkmagazine.com/article/NMG20011102S0008.

[Ellison 2000] Ellison, C. & Schneier. (2000). Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure. Retrieved March 3, 2001 from Counterpane website: http://www.counterpane.com/pki-risks.html.

[Hynes 2001] Hynes, M. (2001, April). Analyzing Kerberos and Public Key Infrastructure (PKI). Retrieved March 1, 2001 from CISCO World Magazine website: http://www.ciscoworldmagazine.com/webpapers/2001/04_guardent.shtml.

[PKI] Introduction to PKI. Retrieved March 1, 2001, from Baltimore Technologies website: http://www.baltimore.com/library/pki/index.html.

[Kerberos] Kerberos: The Network Authentication Protocol. Retrieved March 1, 2001, from Massachusetts Institute of Technology website: http://web.mit.edu/kerberos/www/.

[Kohl 1993] Kohl, J. & Neuman, C. (1993, September). The Kerberos Network Authentication Service (V5). Retrieved March 1, 2001 from Information Science Institute Web site: FTP://ftp.isi.edu/in-notes/rfc1510.txt.

[Neuman 2000] Neuman, Clifford (2000, August). Section 1 Kerberos Revisions for comment. Retrieved April 2, 2002 from Massachusetts Institute of Technology website: http://diswww.mit.edu:8008/menelaus.mit.edu/kprot/487.

[Schneier 200] Schneier, Bruce, (2000). Secrets & Lies. Wiley Computer Publishing.

[Tipton 1999] Tipton, H. F. & Krause, M. (1999). Information Security Management Handbook 4th Edition. CRC Press LLC.

[TUNG 1996] Tung, B., (1996, December). The Moron's Guide to Kerberos, Version 1.2.2. Retrieved March 1, 2001 from Information Science Institute Web site: http://www.isi.edu/gost/brian/security/kerberos.html.

Last Updated: March 19th, 2018

Upcoming Training

Mentor Session - AW SEC401	Mayfield Village, OH	Mar 21, 2018 - May 23, 2018	Mentor
SANS Boston Spring 2018	Boston, MA	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS 2018 - SEC401: Security Essentials Bootcamp Style	Orlando, FL	Apr 03, 2018 - Apr 08, 2018	vLive
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201804,	Apr 09, 2018 - May 16, 2018	vLive
Community SANS Charleston SEC401	Charleston, SC	Apr 09, 2018 - Apr 14, 2018	Community SANS
SANS Zurich 2018	Zurich, Switzerland	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
Mentor Session - AW SEC401	Memphis, TN	Apr 17, 2018 - May 17, 2018	Mentor
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
Baltimore Spring 2018 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Apr 23, 2018 - Apr 28, 2018	vLive
SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
SANS Riyadh April 2018	Riyadh, Saudi Arabia	Apr 28, 2018 - May 03, 2018	Live Event
Automotive Cybersecurity Summit & Training 2018	Chicago, IL	May 01, 2018 - May 08, 2018	Live Event
Community SANS Houston SEC401	Houston, TX	May 07, 2018 - May 12, 2018	Community SANS
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Northern VA Reston Spring 2018	Reston, VA	May 20, 2018 - May 25, 2018	Live Event
University of North Carolina - SEC401: Security Essentials Bootcamp Style	Charlotte, NC	May 21, 2018 - May 26, 2018	vLive
SANS Atlanta 2018	Atlanta, GA	May 29, 2018 - Jun 03, 2018	Live Event
SANS London June 2018	London, United Kingdom	Jun 04, 2018 - Jun 12, 2018	Live Event
SANS Rocky Mountain 2018	Denver, CO	Jun 04, 2018 - Jun 09, 2018	Live Event
Community SANS New York SEC401	New York, NY	Jun 04, 2018 - Jun 09, 2018	Community SANS
SANS Crystal City 2018	Arlington, VA	Jun 18, 2018 - Jun 23, 2018	Live Event
Community SANS Madison SEC401	Madison, WI	Jun 18, 2018 - Jun 23, 2018	Community SANS
Community SANS Portland SEC401	Portland, OR	Jun 18, 2018 - Jun 23, 2018	Community SANS
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Cyber Defence Japan 2018	Tokyo, Japan	Jun 18, 2018 - Jun 30, 2018	Live Event
SANS Cyber Defence Canberra 2018	Canberra, Australia	Jun 25, 2018 - Jul 07, 2018	Live Event
Community SANS Nashville SEC401	Nashville, TN	Jun 25, 2018 - Jun 30, 2018	Community SANS
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Vancouver 2018	Vancouver, BC	Jun 25, 2018 - Jun 30, 2018	Live Event