

Kerberos and Windows 2000

Ronald L. Mendell 2001-10-11

Kerberos and Windows 2000

by *Ronald L. Mendell*

last updated October 11, 2001

Computer security theorists ranging from Bruce Schneier (in *Secrets & Lies*) to Charles P. Pfleeger (in *Security in Computing*) recognize that digital security has three pillars: authentication, authorization, and auditing (AAA). We must be able to identify who is entering our digital space, determine what legitimate privileges they have, and to trace what objects they may be able to access. Kerberos, as implemented in Microsoft, does an excellent job in all three areas. Yet, the technology is far from bulletproof. This article will offer a brief overview of Kerberos in the Windows 2000 environment and will examine some of its potential shortcomings.

A Quick Overview

Before we look at possible avenues of attack against Kerberos, let's briefly examine its history and basic design in order to identify some of the assumptions that underlie the design of Microsoft's Kerberos in Windows 2000.

Kerberos' story derives from the myth of Cerberus (Kerberos in Greek), the three-headed guard dog who allowed people into Hades but not out until Aeneas, desiring a two-way ticket, bribed the beast with a cake. (Proving that even the most watchful security can be circumvented.) A product of MIT's Athena project, Kerberos, the security software, was initially intended as a tool for Authentication, Authorization, and Auditing. Yet, only one "head" of Kerberos, Authentication, became active in the Athena project. However, Microsoft saw compelling security advantages in adding the other two "heads" of Authorization and Auditing. Working as an SSP (a Security Support Provider) with Windows 2000, Kerberos interacts with Active Directory (AD), the Key Distribution Center (KDC), and with PKI (Public Key Infrastructure) to implement all three aspects of AAA.

Authentication, Authorization and Auditing

Kerberos is based upon cryptography and begins its work with an exchange between a client and a server. The client and the server share a secret, a symmetric cryptographic key. Inside

the key is a data structure known as an authenticator that contains information about the client. The client then sends the authenticator to the domain controller that contains the KDC. In examining the authenticator from the client, the KDC extracts information about the client and a timestamp from the client.

Since the client requests access to a server, the KDC uses the server's session key to encrypt the client's timestamp and then send it back to the client. A mutual authentication occurs. The KDC verifies the identity of the client, and the client is able to verify the identity of the server. All this activity takes place within the authentication service of the KDC. The KDC's ticket-granting service function actually grants the session ticket for access to the server. In other words, the KDC handles the server's entire security overhead. The server doesn't have to go to the KDC to verify the client, and the client doesn't have to go to the KDC every time it wishes to visit the server.

The critical controlling factor is time. A session ticket usually expires in about eight hours. So, the client gains access to the server throughout the workday without having to bother the KDC again. As long as the client submits the authenticator and the session ticket, the client gets in. Kerberos therefore establishes a transitive two-way trust relationship between the client and the server.

What safeguard then prevents replays of the client's credentials? The timestamp mechanism works against replays on the network. Frequently, all timestamps in the domain are synchronized with five minutes of each other. If a new authenticator yields a time later than the time stamp of the last authenticator, then it must be from the client, not an old authenticator simply resent. Since Kerberos establishes transitive trust, the client's logon works throughout the tree. The client may travel a trust path from server to server. It can pass its credential onto other servers or services without going back to the KDC. Kerberos also allows Delegated Authentication to allow a service to impersonate its client when accessing other services.

Because Kerberos uses both encryption and time stamping, it presents formidable defenses against attempts to spoof identity. And, since it works in conjunction with the AD, Microsoft Kerberos not only authenticates users, clients, or services but also prevents them from gaining access beyond their permission rights. If group membership and policies are set correctly, then a user or client cannot exceed declared bounds.

What assumptions underlie the security design of Windows 2000? First, it is assumed that the

network administrator will have applied Group Policies and Access Control Lists (ACL) correctly. It is also assumed that all secret keys will remain secured. Also, the network administrator should have prevented the reuse of SIDs (Security Identifiers). Finally, the admin should understand the inherent weakness of symmetric keys and, consequently, implement safeguards such as preauthentication, which should minimize the danger of password guessing attacks. Deviations from these procedures will obviously diminish the security of the system on which Kerberos is working.

Threats to Kerberos on Windows 2000

Evaluating threats is the computer security professional's bread and butter task. The challenge escalates constantly since the threats are myriad. To stay a step ahead of potential attackers, a conceptual model is necessary. As Windows 2000 faces security threats at three levels, when examining Kerberos, the development of a three-ring conceptual model will help us group the wide range of security features into more manageable units for discussion.

At the very center of the three rings is the network that is to be protected. The innermost ring recognizes threats at the code level. The middle ring contains threats arising from policy flaws. And, the outermost ring represents external attacks. These attacks range from those of sophisticated crackers or hackers to script kiddies to an outsider finding a back door into your network. Vandals who disseminate Denial of Service (DOS) attacks also fall into this category.

The Outer Ring of Threats - External Threats

Fortunately, most security professionals understand external attacks, but they may miss the importance of how these attacks endanger Kerberos. These attacks include those on DNS, Smart Cards, PKI (which supports Smart Cards), and by out-of-date users. There are also, unfortunately, attacks that Kerberos is ineffective against: these include attacks via Telnet and FTP, and denial of service attacks.

DNS

External attacks can threaten DNS (Domain Name System) servers and services like SRV. Since Kerberos relies on DNS SRV records to locate domain controllers, any attacks on DNS may endanger Kerberos functioning. The Windows 2000 domain controllers disseminate SRV links for Kerberos. So, protecting the DNS namespace can be an important point to consider as a

security countermeasure to buttress Kerberos. Proper configuring of DNS services and their directories by restricting access to trusted servers and processing becomes absolutely essential. Even the best configuration of Kerberos itself will be useless if supporting services such as DNS do not receive equal care and consideration. Developing restrictions regarding network addresses and domain names requires the same rigor as setting up a firewall: 'Default Deny' is the best methodology to use. Unless an address, domain name, or process is clearly and expressly authorized, then its access to the DNS services is automatically barred.

The DNS SRV records are available in the domain controller at:

```
%Windir%\System32\Config\Netlogon.dns
```

These records enable Windows 2000 to locate the KDCs. Obviously, an attack rendering DNS blind would cripple Kerberos.

PKI - Smart Cards

Smart cards, an added feature to Windows 2000 over that of Windows NT, combine convenience and high-level security. Windows 2000 extensions permit public key cryptography for smart cards. This innovation allows the introduction of PKI (Public Key Infrastructure) into Kerberos functioning. Since smart cards may travel (or stumble) into the outside world, they become an external avenue of attack: as such, a strong policy for the physical security of smart cards needs to be in place.

While the technical details of smart card technology may seem dreary, understanding how it interacts with PKI illustrates the flaws that can compromise Kerberos. Look at Kerberos as the horse, with the smart card as the horseshoe, and PKI as the nail. Your network (the rider) will be lost if PKI fails. So, understanding the following chain of relationships is essential.

Smart card logons follow this sequence after the user enters the PIN or password to release the digital certificate (x.509 version 3) to Windows 2000:

The KRB_AS_REQ (Kerberos Authentication Service Request) goes to the KDC.

Using PKI, the KDC verifies the digital certificate all the way to the root Certificate Authority (CA).

At the same time the KDC verifies the authenticator supplied by the smart card. The KDC also does the timestamp-checking routine to rule out the replay of the authenticator. Security information retrieved from the AD serves as a check against the data in the x.509 version 3 certificate. This data includes the security ID (SID) and group membership information. If everything matches correctly, the user or client receives a ticket to present to servers or services.

The reliance on PKI technology does create some additional security concerns for Kerberos. A primary cause for concern is the password for the smart card itself. No matter how authoritative the x.509 version 3 certificate on the card, it remains worthless if the password is easy to guess. Or, if the password has very low entropy (disorder or randomness), it becomes subject to brute force password attacks.

Another security consideration is the Certificate Authority (CA) granting certificates for the Windows 2000 environment. Normally, the root CA resides on the Certificate Server, and the managing the certificates takes place through the Certificate Services Manager MMC (Microsoft Management Console.) Proper management of the certificate-granting function includes revoked certificates, issued certificates, and failed or pending certificates. If certificates are not issued according to security guidelines set by the enterprise, the quality of the PKI database will be degraded. How does the CA verify the identity of the users? What safeguards prevent fraudulent users from obtaining certificates? How are certificates revoked? If CRLs (Certificate Revocation Lists) do not issue regularly, and expired or invalid certificates may remain in use. These questions need addressed in the set-up and implementation of PKI for Kerberos.

Additional issues involve questions about the public key length. Careful consideration about whether to use a 40-bit, a 56-bit, or a 128-bit key should govern any PKI project. Prime factors involve the sensitivity of the data protected, the lifetime of the data, and the exposure of encrypted data to the outside world. Recognizing that the protection Kerberos offers may depend upon the length of a key begins to open one's eyes. Any automated security system has great fragility, no matter how strong it may be against the frontal attack. The collateral attacks due to ineffective key lengths may render the entire structure weak and prone to collapse.

Out-of-Date User Accounts

The final external threat that Kerberos can protect against is disabled or out-of-date user accounts. Security admins should enable the Kerberos policy, "Enforce Logon Restrictions",

which will keep disabled accounts from gaining access to the network. If it is in place, unauthorized or out-of-date users will not be able to get session tickets. Kerberos will check their credentials every time they ask for a ticket.

The Middle Ring of Threats - Policy Level

A great amount of press coverage concentrates on external threats perpetrated by inventive hackers. In reality, however, most security compromises succeed not because of the perpetrator's cleverness, but because available protection measures are not properly or adequately implemented. This is often the result of poorly designed or poorly implemented policies. Some of the policy weaknesses that may weaken the effectiveness of a Kerberos implementation include:

- Failing to enable preauthentication;
- Allowing outdated security rights to recycle in Security Identifiers;
- Not keeping Kerberos Policies such as clock synchronization and ticket lifetimes current;
- Not preventing the Delegation of Authentication on sensitive accounts; and,
- Not updating to Group Membership and Access Control Lists fail to happen so that Kerberos enforces outdated authentication permissions.

Monitoring these features may be technical and tedious, but failing to keep them up-to-date diminishes Kerberos's strength drastically. Preauthentication, for example, should never be turned off. This security feature makes offline password guessing immensely harder for a cracker. Disabling preauthentication invites attacks on logon passwords.

Users should make sure that their Windows 2000 configuration does not recycle SIDs (Security Identifiers). Access Control Lists (ACLs), the data structures that authorize access to an object or process, attach to SIDs. Reusing SIDs may permit out-of-date ACLs to come into play within Kerberos. Such a scenario could elevate the privileges of users whose security level has been lowered or stopped.

A clock synchronization scheme needs regular maintenance. For the scheme enables Kerberos to discern slight, normal variances in timestamps as opposed to the time inconsistencies of replay attacks. Preventing clock synchronization drift is very important. It is one of the main ways Kerberos establishes the validity of tickets. The tolerance varies though, depending upon the needs of your enterprise. It is set in the Kerberos policies found in the Default Domain

Group Policy object.

Other important Kerberos Policies include:

- Enforce User Logon;
- Maximum Life for a Service Ticket;
- Maximum Life for a User Ticket Renewal; and,
- Maximum Life for a User Ticket.

Many Windows 2000 references mention eight (8) hours as the norm for ticket lifetimes. Decisions in this area depend upon balancing your security needs against the additional overload that frequent ticketing would place on your system. Networks with highly sensitive data may require shorter lifetimes for tickets.

Kerberos in Windows 2000 has the Delegation of Authentication feature. While this powerful tool can streamline processing and file access, it should not be used on sensitive accounts. In Active Directory (AD) Users and Computers, use the option 'Account is sensitive and cannot be delegated' to protect accounts whose compromise would be critical (for example, the sys admin's account.)

Another critical area is Group Membership. Windows 2000 works best by assigning users to groups. Then, the admin attaches access privileges to groups, for it is much easier to administrate access at the group level. Of course, keeping track of whether a user's group memberships remain reasonable and necessary can become a real challenge. Periodic auditing of users' group memberships is a necessary adjunct to good Kerberos Security. Kerberos cannot determine whether a user should still have access to an object - only the admin can do that. Set a regular schedule and procedure for reviewing group memberships.

Enterprises constantly change. Their boundaries alter on a daily basis. Outdated Access Control Lists (ACLs) can creep in on the reliability of the Kerberos database. Protecting against "identity drift" due to rapid change requires strong guidelines for the entire organization. Identity drift can corrupt the ACL database indirectly. If it is inaccurate, its value becomes questionable. A secure reporting system needs to be in place to alert Security immediately to the hiring, firing, promotion, or transfer of users.

The Innermost Ring - Code-level attacks

Despite the best efforts at implementation and controls, Kerberos will fall victim to weaknesses that are present in other software in general. Hidden in the nooks and crannies of code lie unknown exploits waiting to happen. So, it is very important that security specialists keep their ears open to intelligence channels for software flaws that come to notice.

For instance, Kerberos began in the UNIX environment where it encountered many trials. Despite its great strength as a security tool, Kerberos fell prey to granular level attacks. Kerberos 5 release 1.2 sought to remedy problems such as buffer overflows, leakage of temporary files, ASN.1 parsing problems, and unexpected core dumps of passwords.

Simply because Kerberos has been through this period of refinement and is now in the Windows world, does not mean it is invincible. Constant intelligence gathering via Windows 2000 user groups, Microsoft Security Updates, [BugTraq](#) and [ARIS alerts](#) will help keep a pulse on what's happening at this level. Buffer overflows are not going away for any type of security-related code. Windows 2000 is far too complex for its designers to know its whole story line. The same goes for a security library like Kerberos, which operates as a dynamically linked library in the Win2000 OS.

Conclusion

Kerberos represents a unique partnership of security tools, utilizing DNS, AD, KDC, and PKI. Making it a successful tool rests upon accepting its dependency with regard to these other components. Each component can have its own story line leading to compromise. Failure in one can cause Kerberos to fall short of its promise.

Ronald Mendell is an independent writer specializing in investigative and security topics. He currently does technical support for a high-tech company in Austin, Texas. He reviews computer security titles for www.securitymanagement.com. His latest review is on "RSA Cryptography" in October 2001.

Relevant Links

[Kerberos V5 UNIX User's Guide Release 1.2](#)

[Microsoft Windows 2000 Server White Paper - Windows 1000 Kerberos Authentication](#)

[Privacy Statement](#)

Copyright 2006, SecurityFocus