Heimdal and Windows 2000 Kerberos — how to get them to play together

Assar Westerlund
Swedish Institute of Computer Science
assar@sics.se
Johan Danielsson
Center for Parallel Computers, KTH
joda@pdc.kth.se

Abstract

As a practical means of achieving better security and single sign-on, the Kerberos network authentication system has been in wide use in the Unix world for many years.

Microsoft has included its own implementation in Windows 2000, replacing the NTLM authentication system from older Windows NT versions. This facilitates sharing account information between Unix and Windows machines, as there is no need to keep different passwords.

Although Microsoft's Kerberos implementation mostly follows the specification, there are a number of deviations and extensions, not all of which are well documented. Consequently, it is not always obvious how to fit Windows 2000 clients and servers into an existing Kerberos environment. In this paper we discuss the differences between the two systems and describe how we got our Kerberos implementation, Heimdal, to work with Windows 2000.

1 Introduction

Ever since Microsoft announced that Windows NT 5 (later renamed to Windows 2000) would be using Kerberos for network authentication, there have been questions as to how that implementation would interoperate with existing implementations. Considering Microsoft's bad reputation of "embracing and extending" other systems, people feared that what eventually came out would be something that would at best be similar to Kerberos. As it turns out, these fears are mostly unfounded.

While it mostly follows the specification, the Kerberos in Windows 2000 has some small implementation differences and undocumented extensions to the protocol. This makes writing a replacement for the Windows 2000 Kerberos server hard. However, we feel that there are good reasons for using a Windows 2000 Kerberos server to support Windows clients, so this might not be a big problem.

Heimdal[1] is an implementation of Kerberos 5 that we have been working on for some time. To make it work better with Windows 2000, we have made a number of changes. These include adding RC4 encryption, configurable salting of keys (which is required by some other systems as well), and crude support for referrals.

This paper starts with an introduction to the relevant Kerberos concepts in section 2. Section 3 explains the difference between database organisations. Section 4 discusses the different issues that come up when trying to interoperate between Heimdal and the Windows 2000 Kerberos. Section 5 explores different scenarios on how the two systems can be integrated, and finally conclusions and future work are presented in sections 6 and 7.

2 Kerberos

Kerberos is a network security system for authentication. It allows users and services, collectively called *principals*, to authenticate to each other over an insecure network.

Kerberos relies on a central server (the *Kerberos server*) which is trusted by all principals. Starting with this trust relationship, the Kerberos server can securely introduce the communicating parties to each other. The Kerberos server is also called the Key Distribution Centre (KDC). All principals have a secret password or key that they share with the Kerberos server. This allows them to verify that they communicate with the correct Kerberos server, since no other entity should know their password or key.

Although each user has a secret password, the passwords are actually stored as encryption keys in the database. These keys are derived from the passwords with one-way functions (*string-to-key* functions). For services, the keys are stored in a location (typically in a file) where the server program can access them.

A client authenticates to a server by providing the server with a piece of data (the *ticket*) generated by the KDC and encrypted in the server's key. This ticket proves the client's identity to the server. The server may also prove its identity to the client by showing that it can decrypt the ticket. Each ticket contains a session key (also sent to the client) which allows the client and server to encrypt their traffic.

Single sign-on is achieved in Kerberos by using a special *ticket-granting ticket*, that is obtained when a user logs in. This ticket can later be used to get more tickets from the KDC, without having to enter the password again. The ticket identifies the holder as a particular user, so anyone with access to the ticket can impersonate that user. To lessen the damage if a ticket is stolen the ticket has a limited lifetime.

The Kerberos world is divided into *realms*, where each realm is an administrative domain. A realm's name will normally be the same as the site's DNS domain name. The name of a principal is a list of strings, separated by slashes, followed by the realm name. A typical user would be named *nisse@FOO.SE* and a service *host/bar.foo.se@FOO.SE*.

This paper discusses Kerberos 5, the current protocol version. Version 4 was the first to be publicly available and had a sizable installed base when version 5 reached maturity. There are still version 4 based applications and clients in use. Thus, most current version 5 implementations have functionality for handling version 4 clients. Version 5 is reasonably similar to version 4, except that it is more parameterised, including support of several types of encryption algorithms.

Kerberos is described in more detail in [2, 3, 4, 5].

3 Kerberos databases

Every key that the Kerberos server keeps must be stored in some kind of database. The database needs to contain at least the names and keys of the principals. Additional information stored and the organisation of the database can vary quite a lot between different implementations.

3.1 Heimdal's database

On traditional Unix systems, account and password information is stored in a local database (such as `/etc/passwd`) or some distributed database (such as NIS).

With a typical Heimdal setup the key database is separate from the account database and password information is not directly available to the clients.

This means that the name space for Unix users and Kerberos principals *can* be different, though normally they are not. Users might have several principals for different roles. For example a user might authenticate as the principal *nisse/root* when acting as super-user and as the principal *nisse/admin* when doing administrative functions with the database. Services have principals in the Kerberos database but not necessarily any corresponding Unix accounts. Even if they do, there is not necessarily a one-to-one mapping between principals and accounts. Services' principals are usually named *service/hostname*. The basic fields of a database entry are shown in Figure 1.

Field	Type
Principal name	list of strings
Principal expiration	date
Password expiration	date
Attributes	flags
Key version	integer
Keys	(encryption type, salt, key)...

Figure 1: A basic Heimdal database entry

There are several different string-to-key functions, so what particular function was used for a key has to be stored along that key. Some functions also take a known string as input, known as *salt*. The reason for the salt is to make comparing keys and performing dictionary attacks harder (if the same password is used in different realms, the resulting keys will not be identical).

The typical way a realm is set up is with one master server where all modifications to the database are performed, and a number of slaves that maintain read-only copies of the same database. Changes are propagated either periodically or incrementally from the master to the slaves. This is similar to the common DNS server configuration with one primary name server and zero or more secondary name servers.

3.2 Windows 2000's database

Windows 2000 uses a data repository called the Active Directory[6] for most of the domain data. This includes the users and machines, and their keys.

The active directory is a hierarchical directory service which stores different kinds of data, each identified by a particular schema. It is distributed among the domain controllers of a domain with multi-master replication. Thus, changes made to any of the servers will be propagated to the other servers.

4 Protocol and implementation issues

4.1 Encryption types

The original Kerberos protocol specification (RFC1510[5]) made DES the required encryption type to implement. Windows 2000 implements this encryption method, and it interoperates with other Kerberos implementations.

When upgrading an NT 4 domain to Windows 2000, there are only MD4 keys for all users, so there is no way to use DES. To support this common case, Microsoft included its own RC4 based encryption algorithm (rc4-hmac-md5) that make use of the MD4 keys. This algorithm is described in a series of drafts[7] published by Microsoft. Heimdal also has an implementation of it, which we have tested against Windows 2000.

4.2 Salting

Normally keys are salted with the principal name, but there are situations when a different salt is used. One example is when converting an existing Kerberos 4 realm to Kerberos 5. In Kerberos 4, the keys are not salted (the salt string is empty). Another is when a principal is renamed, since the principal name will change, but the key will remain the same.

When the salt is non-standard, it has to be stored in the database, and sent to the client. Windows 2000 can do this, but for unknown reasons it does not handle the empty salt.

The Heimdal key database can keep several keys with different salting information (both type and string). The point at which more keys can be added is when the user's password is changed, so there is configuration support for specifying what types of keys should be created whenever a password is changed.

4.3 Limitations and problems

Windows 2000 does not implement all of the functionality required by the Kerberos specification. One of the required checksum types is not actually implemented (rsa-md5-des). This is a problem because there is no negotiation or possible way of knowing this beforehand.

When a user's password has expired, the Kerberos server will return an error and only allow the user to change their password. Windows 2000 erroneously gave the same error when the user was actually trying to change the password which resulted in an infinite loop in our client. This bug has been fixed in Service Pack 1.

Unfortunately Windows 2000 does not support looking up KDC information for non-2000 realms using DNS, therefore, configuration information has to be added manually.

4.4 Authorisation data (or PAC)

The Kerberos protocol only provides authentication, it proves the identity of a communicating party, and not authorisation, or telling what rights and privileges they might have. The common way of implementing authorisation is to look up the identity in a separate list or database and see what they are authorised for in this context. Microsoft instead tried adding this to Kerberos.

The Windows 2000 KDC adds extra authorisation data to the tickets it generates. This data is called the Privilege Access Certificate (PAC). It includes some information about the user and group memberships. Both users and groups are represented by their Security IDs (SIDs), which is a unique number for every Windows 2000 object. All of this information is stored in the active directory, and the application server should be able to look it up from the client name in the ticket, instead of getting it from the PAC. However, it is unknown whether servers will do that if they get tickets without the PAC.

The PAC data format has been partially reverse-engineered. We wrote code as part of Heimdal to dump the authorisation data and then were assisted by people with much more familiarity with NT data structures. The format has also been documented in a Microsoft document[8] that has a *trade secret* license that prohibits anyone from implementing it.

4.5 Applications

Without applications that use Kerberos, a working infrastructure is not very useful. However, the traditional Kerberos applications on Unix (such as telnet, rsh, and ftp) are not available on Windows 2000 or do not support Kerberos. Applications on Windows 2000 use Kerberos for a number of different protocols such as LDAP, SMB, and COM. Unix counterparts of these applications are only somewhat available.

4.6 Administration

There is no standardised protocol for administering a Kerberos database. Windows 2000 uses the Active Directory to store the database which can be accessed through Kerberised LDAP. Consequently there is a possibility of using non-Microsoft tools to maintain the database information.

Password changes by users are done with a different protocol[9] and it works fine between Heimdal and Windows 2000.

Database propagation between the Kerberos servers is also rather different. Making the active directory distribution and the different propagation methods work together is non-trivial. This makes mixed realms with both Windows 2000 and other servers quite unlikely.

4.7 Referrals

When getting tickets the client has to know what principals to request them for. The traditional way of doing this is to use the user's login-name and the client machine's pre-configured realm name. Microsoft has proposed a draft[10] to extend this mechanism, so it would, for instance, be possible to use an e-mail address as login name. The extension requires some changes to the protocol, since the KDC is not allowed to return a ticket for a different principal than requested.

Referrals can also be used to remove the need for host name lookups on the client, somewhat like turning the KDC into a secure DNS server.

It is unclear how much of this draft has been implemented in any released Windows version. It is a fact that a Windows client will only talk to KDCs in the realm it currently belongs to, unless it gets a referral to another KDC. Thus the KDC needs to have some support for referrals or cross-realm authentication will not work. We have added functionality for referrals to the Heimdal KDC that is sufficient for Windows clients.

4.8 APIs

Windows 2000 does not support any of the traditional Kerberos 5 library functions, that many Unix applications use.

The GSS-API[11] protocol with a Kerberos mechanism is implemented, but not the API part of it. Kerberos application programming under Windows 2000 is done with the Security Service Provider Interface (SSPI) which is quite similar to GSS-API. Thus only a small effort is required to write code that works with both Windows 2000 and other Kerberos implementations. Because Windows 2000 implements the GSS-API protocol, applications written against SSPI will interoperate with GSS-API applications using other Kerberos implementations.

5 Scenarios

There are different ways to integrate a Windows 2000-based infrastructure with other Kerberos realms. Some of these are discussed here and the interoperability of each of them is explained. Some more details on the exact commands to run are available in [12].

5.1 A Windows 2000 client in a non-2000 realm

A standalone Windows 2000 workstation (a member of a workgroup but not of a domain), can be configured to use a non-2000 realm for login authentication. The `ksetup` program (which is unfortunately not installed by default but supplied on the Windows 2000 install CD) can be used to configure what realms should be used by a particular workstation. The DNS names of the KDCs also have to be configured (see 4.3).

The workstation must have a key in the Kerberos database. Also the mapping of Kerberos principals to local users (typically one-to-one) has to be configured. It is worth noting that the workstation will use the configured KDC for all its requests, independent of what realms the application servers belong to, so this KDC has to be able to handle these requests (see 4.7). If the configured KDC handles these requests, the workstation can connect to remote Windows 2000 domains.

Sites with a small number of Windows 2000 machines most likely want to use this configuration. It is being used mainly by sites that do not want to have a Windows 2000 domain or do not want all of their machines to be members of their 2000 domain.

5.2 A non-2000 client in a Windows 2000 realm

Clients using other Kerberos implementations should not need very many changes to interact with a Windows 2000 KDC. Key installation is of course different. The client must have a user in the active directory, created with the normal Windows 2000 tools. Then a mapping between this user and the principal name (*host/fully.qualified.hostname*) has to be installed with the `ktpass` command. The key that resides in the active directory also has to be copied to a file on the client.

We are not aware of anyone using this configuration.

5.3 A Windows 2000 domain with a non-2000 KDC

The problem using a non-Microsoft KDC for a Windows 2000 domain is that the KDC is very much integrated with the other domain controller servers. All of these servers would have to be replaced at the same time, much as Samba [14] can act as a domain controller for an NT 4 domain. It is unclear how much work is needed to make Samba a replacement for the Windows 2000 domain servers but it is probably a large amount. And, of course, the Heimdal KDC would also have to be integrated.

The PAC data also make things more complicated. If there are no native 2000 servers being run in the domain, PACs will not be needed. And it has yet to be determined if native 2000 servers use PACs as optimisations or if they are actually required. The worst case would be having to reverse engineer the complete format and then sew the KDC and the domain controller together.

Getting this working reliably is still far in the future.

5.4 Inter-connected Windows 2000 and non-2000 realms

A Windows 2000 realm can be integrated with an existing non-2000 realm by allowing clients in the Windows 2000 realm to authenticate to the existing realm. When a client wants to authenticate to a server in a different realm, the two realms must share a key, either directly or indirectly through other realms. Windows 2000 and non-2000 realms can share a key. All the involved Windows 2000 clients need to have configuration information about the foreign KDCs. Once the foreign KDC information is stored on the domain controller of the Windows 2000 domain, a key can be configured with the GUI administrative tool. This key also needs to be added to the other realm. Kerberos authentication can then take place between the two realms. The domain can also be configured with the same GUI tool to allow users to login when they are authentication in the other realm, similar to configuration of a standalone workstation.

This is likely to be the most common configuration, since there is only one password database and all Windows applications that rely on or make use of the domain infrastructure still work. There are several large sites that have their realms set up this way, keeping all their users in one realm and all the Windows machines in a Windows realm.

6 Conclusions

While Windows 2000 Kerberos is different, getting it to work with other Kerberos implementations is not that hard. The documentation is not always sufficient, and sometimes experiments have to be performed to figure out how things actually work.

7 Future work

Windows 2000 uses an extension for using public key cryptography in initial authentication[13]. An implementation of this in Heimdal would be useful, not only for use with Windows.

Assuming that there is a usable specification of the PAC format (see 4.4), integrating the Heimdal KDC with Samba[14] to create an entire domain controller would be useful.

8 Availability

Heimdal is freely available from
`http://www.pdc.kth.se/heimdal/`.

References

[1] Assar Westerlund and Johan Danielsson, *Heimdal — an independent implementation of Kerberos 5*,

In proceedings of the Usenix 1998 Annual Technical Conference, New Orleans, USA, June 1998, Freenix Track

[2] Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller, *Kerberos: An Authentication Service for Open Network Systems*, Proceedings Winter USENIX Conference, Dallas (1988)

[3] John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so, *The Evolution of the Kerberos Authentication System*, Distributed Open Systems, pages 78-94. IEEE Computer Society Press (1994).

[4] B. Clifford Neuman, and Theodore Ts'o, *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications, 32(9), pages 33-38 (1994)

[5] John Kohl and Clifford Neuman, *The Kerberos Network Authentication Service (V5)*, Network Working Group (1993), RFC1510

[6] Microsoft, *Directory Services*, `http://www.microsoft.com/windows2000/library/technologies/activedirectory/default.asp`

[7] M. Swift and J. Brezak, *The Windows 2000 RC4-HMAC Kerberos encryption type*, Work In Progress, draft-brezak-win2k-krb-rc4-hmac-02.txt

[8] Microsoft, *Kerberos PAC specification*, `http://www.microsoft.com/technet/security/kerberos/default.asp`

[9] M. Horowitz, *Kerberos Change Password Protocol*, Work In Progress, draft-ietf-cat-kerb-chg-password-02.txt

[10] M. Swift, J. Brezak, J. Trostle, and K. Raeburn, *Generating KDC Referrals to locate Kerberos realms*, Work In Progress, draft-ietf-krb-wg-kerberos-referrals-00.txt

[11] J. Linn, *Generic Security Service Application Program Interface Version 2*, Network Working Group (2000), RFC2743

[12] Microsoft, *Step-by-step guide to Kerberos 5 (krb5 1.0) Interoperability*, `http://www.microsoft.com/windows2000/library/planning/security/kerbsteps.asp`

[13] Brian Tung, Clifford Neuman, John Wray, Ari Medvinsky, Matthew Hur, and Jonathan Trostle, *Public Key Cryptography for Initial Authentication in Kerberos*, Work In Progress, draft-ietf-cat-kerberos-pk-init-06.txt

[14] Samba Team, *Samba*, `http://www.samba.org/`