Global Information Assurance Certification Paper

Win 2000 Interoperability and Single Sign On:
Future trends in Authentication and Authorisation
Michael Roylance
March 2002 GSEC

Introduction

Abstract
As information technology becomes more pervasive the need for authentication and authorisation management becomes more acute. System administrators spend their lives synchronizing accounts and resetting passwords. One solution to this problem is Single Sign On or SSO. Current offerings in this arena tend to be piecemeal and heavily customised. This situation may be about to change. Microsoft, a global leader in operating systems and office productivity tools has adopted an open source solution to address the issues of authentication and authorisation. In Kerberos, Microsoft may have found an SSO silver bullet.

Background
By definition Kerberos is the three-headed dog that guards the gates of Hades. In practice Kerberos is an authentication method that was developed at the Massachusetts Institute of Technology, to deal with communicating over an untrusted network through the use of trusted 3rd party machines and modified client/server applications.

The above is a direct quote from the paper by Richard Tufaro Jr. entitled 'Kerberos, A Typical Study', which appears on this site; http://rr.sans.org/authentic/kerberos.php In two sentences it captures the essence of Kerberos.

Kerberos might be the best model for secure authentication on an untrusted network but is of no use if it's an academic curiosity left in the cupboard. Enter Microsoft. Microsoft has the market power to make or break technologies. Consider how IBM legitimized the PC in 1980. Microsoft is in a similar position today. Somehow the company has been able to find the killer aps, either acquiring the IP or imitating them and promoting them by either push or pull. Push is where it uses its market power like a blunt instrument. Pull is where it entices users down a new path. Either way, being anointed by Microsoft often means a fast ride into the mainstream. Kerberos is just another in a long line of technologies, from the web browser to the GUI itself, which have been subsumed into Microsoft's architecture. The questions that will be addressed here are, how well does Win 2000 Kerberos interoperate with implementations on other systems, and does Win 2000 offer any advances toward SSO?

Kerberos Detailed Functionality

The fine details of Kerberos operation have been well documented and will not be covered in this paper. A list of references is provided in Appendix A which give detailed explanations of the nuts and bolts of Kerberos. A certain amount of familiarity with this material is presumed from here on. Kerberos is a very elegant solution to a perennial problem. It's well worth taking the time to understand its basics if you haven't already done so.

Win 2000 Interoperability

How the Kerberos Architecture fits into Win 2000

Figure 1 below shows the Kerberos model in a Win 2000 context. Look at this figure and ask yourself this question. What would happen if the KDC (Key Distribution Centre) was on a Windows Domain controller but the Client and the Target Service were running on Unix hosts? Would the Unix client be able to authenticate to and use the services of the Target Service?

Win 2000 Kerberos

Figure 1.

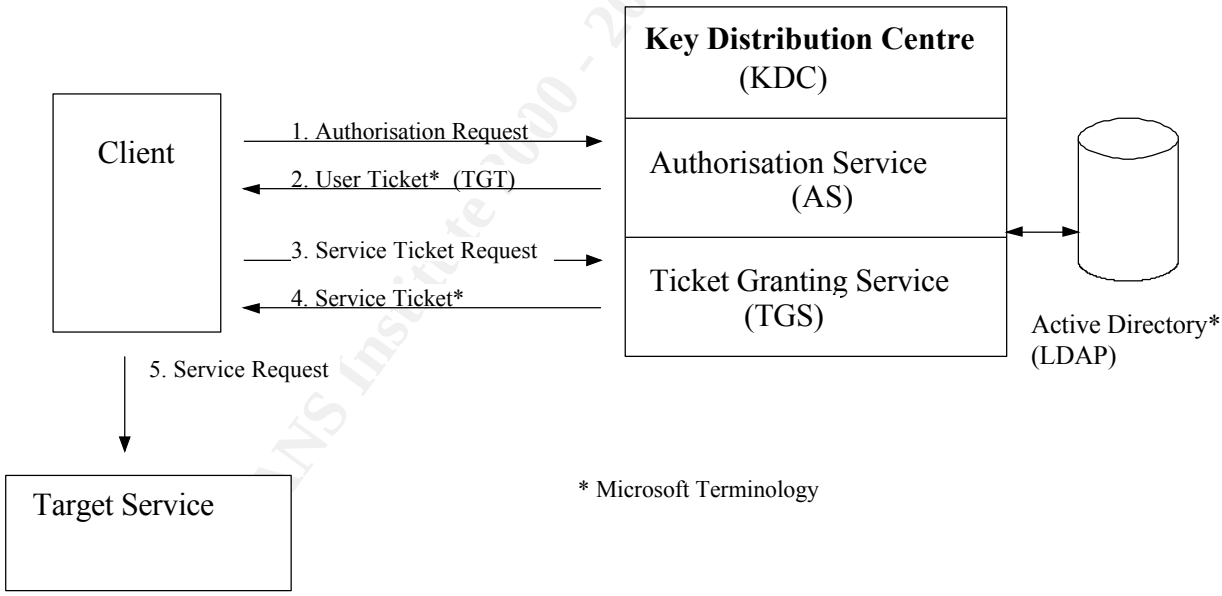
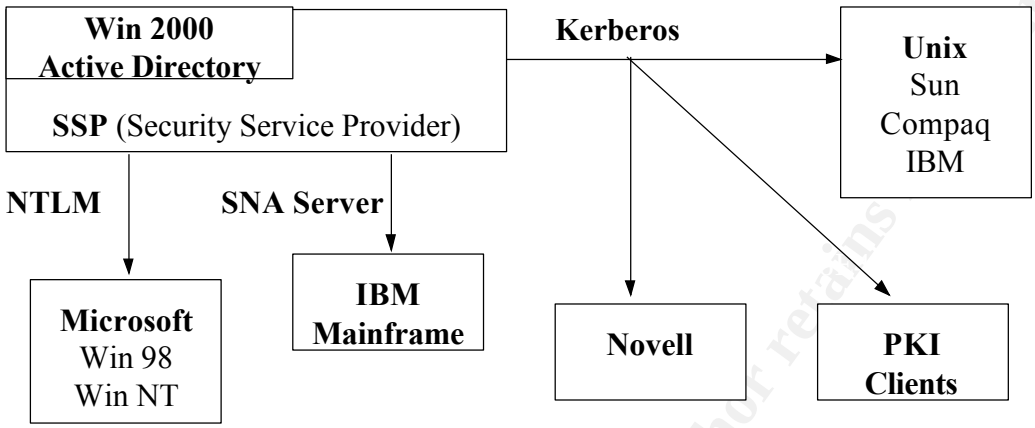


Figure 2 shows how Win 2000 authentication interoperates with other operating systems.

Active Directory Interoperation

Figure 2



For more details of the above concepts please visit this page:-
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/windows2000serv/evaluate/featfunc/nt2ksso.asp

Win 2000 Kerberos Interoperability with non-Windows systems

Following is a list of the different scenarios supported by the Win 2000 implementation of Kerberos.

1. **Windows KDC**. Non-windows Kerberos implementations can authenticate to the KDC in a Windows 2000 domain. Non-windows Kerberos users and hosts can authenticate to a domain controller by using **kinit** and DES-CBC-MD5 or DES-CBC-CRC encryption.
2. **Non-windows KDC**. Systems running Windows 2000 can authenticate to a host serving as the KDC of a Kerberos realm. In addition, a standalone Windows 2000 system can be configured so that local computer accounts map to Kerberos principals. This configuration allows users to log on simultaneously to both the computer and the Kerberos realm.
3. **Windows Client, Non-Windows Target Service**. This is far and away the most likely interoperation scenario you will see. Why is explained a bit later. Client applications running on Windows 2000 can authenticate to non-Windows Kerberos services if the services support the Generic Security Service Application Program Interface (GSS-API) defined in RFC 1964.
4. **Non-Windows Client, Windows Target Service**. Client applications

running on non-Windows Kerberos systems can authenticate to services running on Windows 2000 if the client applications support the GSS-API as defined in RFC 1964.

The above information is taken from this Microsoft URL :-
http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp

Specific Example: Windows KDC and Unix host
The following URL carries an explanation of how to install functions to a generic Unix machine to enable it to authenticate to a Win 2000 KDC. Sample source code for these functions is provided. Once the code is compiled and installed on the Unix machine then it can authenticate to the Win 2000 KDC.

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnactdir/html/kerberossamp.asp

The sample functions are;

- Kinit: commences the Kerberos authentication procedure
- Locator: queries the DNS for service location resource records (SRV RR)
- Kpassword: implements the Kerberos change password protocol
- Ksetup: used to set an account's password
- Pwdump: queries the Active Directory for an account
- Adduser: creates a user account
- Netjoin: creates a computer account

Interoperation with other architectures such as Novell
Mention is made by Microsoft of interoperation with other architectures such as Novell and SNA. However most of the attention is focussed upon interoperation with Unix platforms. This is not surprising since Unix was the parent environment where Kerberos was developed and it would indeed be strange if Win 2000 Kerberos did not interoperate with it. Unix is also an excellent target market from a strategic point of view. There are inevitably Unix machines on every major site today and will be for the foreseeable future. While Unix is not the all-singing-all-dancing OS that Win 2000 is, it has a reputation for speed and reliability that assures its future. However, proprietary implementations of Unix often have very poor user account management tools by comparison to general market standards. They present an ideal target market for upgrade to an SSO solution.

Single Sign On

Single Sign On (SSO)

SSO offers two advantages. First it holds the promise that you, Mr User, can log on once and access all the systems you need without having to enter you user name and password again - well, not until tomorrow at least. The other major promise is that all user authorisation data can be held centrally. This is an advantage for system administrators who currently struggle with islands of authorisation data sitting on boxes scattered across the network.

What if such a system offered improved security? Generally, users are not interested in security. The only time they notice security is when it restricts their activity in some way. Security is never a motivation to implement anything, rather it's the reverse. SSO on the other hand has features that users want. It provides convenience for both users and administrators. If security also happens to be increased along the way then this is just a bonus.

The concept of SSO is key to improving the chronic overuse of computer accounts and associated passwords we see in our daily lives. As computer hardware becomes cheaper and users purchase more of it, we roll out systems for everything, and passwords to match. Passwords for bank accounts. Passwords for email accounts. Network passwords applications passwords and so on. Some of these passwords we are forced to change on a regular basis. This leads to password fatigue. People begin forgetting their passwords. Administrators are forever changing and resetting passwords.

However the current entropic mess is not all bad. Multiple layers provide a kind of defence-in-depth. Consider the situation where a user needs a network logon followed by an application logon. This is cumbersome but it interposes two layers for attackers to crack. Combining these two steps into one, as SSO would do, reduces these layers to one. You might think that this is less secure, and you would be correct, unless the SSO system had better security than the two layer model.

Kerberos is actually a multi layer model but the mechanics of the layers are rolled into a single system and presented to the user as one process. The other key concept in the Kerberos design is that it keeps a centralised cache of authorisation data for a whole environment. This presents a single point of update when users are created, deleted or moved. Microsoft's adaptation of the open source technologies Kerberos and LDAP have been combined to offer these two features, a Single Sign On and a single repository of authentication data. Microsoft calls this Active Directory.

Does Win 2000 provide Single Sign On?

This question is the most critical. The fact that Kerberos offers excellent security is offset by its complexity. Ninety percent of users will not use security features by choice, especially if it involves a sacrifice of convenience, and it almost always does. But most users will adopt a system which increases convenience.

So, having set the scene, what capacity does Win 2000 have to deliver? The

answer to that depends on several things. First of all, remember that Kerberos is primarily a network logon system. It will authenticate the user onto a target box on the network. This may not authenticate the user onto applications running on that system although Kerberos is designed to support this function. Within the standard Kerberos Service Ticket there is a field called Authorisation-Data. Kerberos itself does not interpret the contents of this field but this information can be passed to a service or application on the Target. The concept is not unlike the payload encapsulated in a TCP/IP data frame or an API that developers can take advantage of to pass authentication data about the client across to the application. From here an automated logon to an application can be effected if that application is designed to interact with Kerberos. In fact Microsoft has taken advantage of Authorisation-Data field to transmit group SIDs within all Service Tickets. This supports Microsoft's drive and file permission structure. Microsoft designers have done this by introducing a security broker into the architecture call the SPP (Security Support Provider). This broker will negotiate the authentication protocol for each user. This allows Win 2000 to deal with legacy Win NT hosts as well as Novell and SNA systems. The SPP has a standard interface that third party software vendors can also use to build Win 2000 SSO interoperability into their products. Such products are intended to carry the BackOffice logo to denote their compatibility with the Win 2000 SSO regime. For legacy applications the outlook is not so good. Legacy in this context means NT 4.0, for it does not support Kerberos authentication as Win 2000 does. Such environments will be left in the old multiple logon world even though Win 2000 will still authenticate them.

The exciting prospect for the future is that designers will incorporate SSO interoperation into new applications if they know they only have to do it once and it's a standard. This is pretty much the equation that gave MS DOS such a meteoric rise some twenty years ago - this plus the cheap and ubiquitous Intel platform. Developers understood the huge market footprint DOS could deliver to their product. This logic has not changed and Microsoft has an even bigger market today. Because of this, developers may take advantage of the SSP interface (SSPI) when they design new applications or upgrade existing ones for Win 2000. This will mean a one-step logon for many applications. It will also mean a huge reduction in account and password administration. The only complicating factor is the success of Microsoft itself.

The undoubted market power wielded by the organisation has become a liability. When DOS was a struggling underdog, software developers willingly fostered it to advance their own chances in the market place. Bill Gates himself was something of a Linus Torvalds figure in those days - a revolutionary bringing computing to the masses. Today however, Microsoft is an established market heavyweight of which many developers are deeply suspicious. Software vendors have watched as Microsoft devoured or destroyed their brothers and sisters. They are caught in the invidious position of having to dance with the 800lb gorilla or go out of business. It is not beyond the bounds of possibility that a backlash will occur among software vendors whereby they will boycott Microsoft in favour of some alternative platform. In fact we see many of the established firms in the industry pouring

money into Linux distributions in an attempt to slow Microsoft down. Microsoft needs to be careful here because it is not difficult for programmers to reverse engineer the Active Directory solution in the same way the SMB protocol was reverse engineered to produce Samba. An open source imitation of Active Directory is a real prospect and would do the world no harm.

Changes made to Kerberos in Win 2000

1. PKI
   Normally the only encryption method supported by Kerberos is symmetrical encryption, sometimes called secret key encryption. This was the original MIT design. Microsoft has extended the MIT implementation of Kerberos to offer the option of asymmetric encryption at the very first stage of authentication. This may seem a strange departure at first glance, given that Kerberos is a complete authentication and authorisation system in its own right. Why change the details at all? The significance of this move is that it allows a logon to Win 2000 using PKI (Public Key Infrastructure). This is ideal for logging on to a system via a browser using a Smartcard.

   Smartcards have been around for a long time but never really taken off. Sometimes this happens. Technologies just sit on the shelf waiting for the killer ap to come along. The Internet and the World Wide Web are a good example. Debit cards, the first practical use proposed for Smartcards have never really taken off. While Smartcards have been struggling to find a niche they have been getting smarter. This plus the growing need for rationalisation in account management may be the catalyst to find a legitimate use for Smartcards. And Microsoft may just provide the logon gateway. While Smartcards do not constitute a Single Sign On solution in the traditional sense they do alleviate many of the problems of account management described in this document – especially password management.

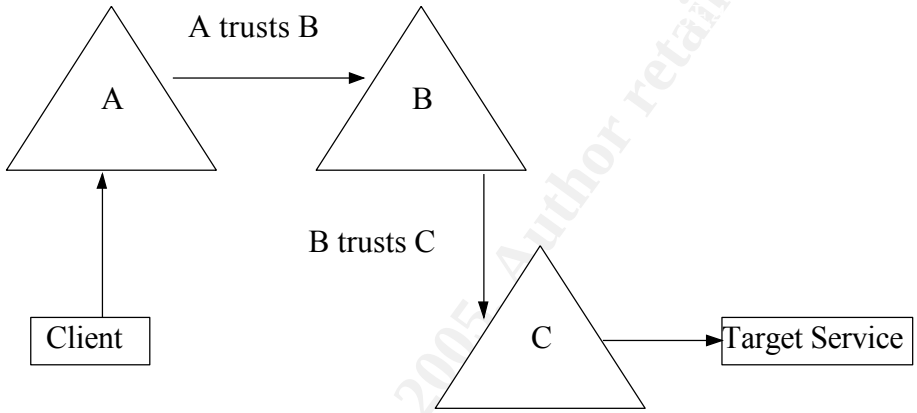
   Imagine a world where access to your corporate network was outsourced to a CA using PKI. It's not such a far fetched idea. When an employee joins the organisation a signed authority is sent to the CA indicating the employee's access rights. These are cached on the CA's server and used to create a smart card which is given to the employee. When the smart card is used to log on the authorisation data will be retrieved from the CA's server and the employee given the appropriate system access. This will allow a seamless global logon and dramatically reduce account admin within the organisation.

2. Transitive Trust
   Windows 2000 Kerberos offers the concept of transitive trust

relationships. The traditional version of Kerberos offers one for one trust mapping between domains (realms) but Windows 2000 goes a step further than this. The client in the figure below is able to access services on the Target Service despite the fact that domain A does not have a trust relationship with domain C. In Windows 2000 the fact that A trusts B, and B trusts C, means that A can access services on C. The trust relationship is transitive. This needs less administration because no trust relationship is required to be established between A and C. In large mutli-domain networks this can save a lot of time.

Transitive Trust

Figure 3



3. Use of TCP and not UPD on port 88
   Traditional implementations of Kerberos have always used UPD, and for good reasons. This tradition is maintained for communicating with non-Windows clients in the Win 2000 domain. However Microsoft's intensive use of the 'Authorisation-Data' field means that more than one dataframe is required to transmit the required information. As a consequence Microsoft has decided to use TCP for communication between Win 2000 systems in a Kerberos domain. Obviously this means non-Windows hosts sharing the authentication system won't have all the Windows 2000 bells and whistles.

4. Use of IPSEC
   The Win 2000 implementation of Kerberos allows for IPSEC to be used on session traffic between Clients and Target Services. This does not represent a major departure as the MIT incarnation of Kerberos always allowed for the Session Key to be used to encrypt traffic between a Client and a Target Service. This is a minor technological improvement.

5. SSP (Security Support Provider)
   This is Microsoft's security broker allowing Win 2000 to accommodate varying authentication protocols such as NTLM (LanManager) and SNA. SSP is not part of Kerberos but interfaces with it. It is also where the non-Windows hosts interface the Active Directory authentication system.

Use of open source code: Legal and Philosophical Considerations

For a background on open source licenses see:- http://www.opensource.org/

Kerberos is released under the MIT license which allows anyone to take the source code to change and resell. LDAP which forms the basis for Microsoft's Active Directory is released under a similar license by the University of Michigan. These licenses are not really licenses at all but disclaimers. A real license is the GNU General Public License (GPL). The GPL insists that if you take the source code and modify it you must make the modifications available with the binaries. Kerberos was not released under this license and Microsoft is under no obligation to release the modified source code. However security people being what they are, always want to look under the hood. In fact, on the MIT Kerberos home page it states that '*MIT provides Kerberos in source form so that anyone who wishes to use it may look over the code for themselves and assure themselves that the code is trustworthy*'.

So what does this mean? It means that Win 2000 fails a basic test from a strict security point of view. Security analysts want to see the source code that Microsoft has used to create their three headed monster. Does this really matter? For the FBI and the CIA it matters, but not for the rest of us. Firstly, Microsoft is big enough not to worry about what security analysts think. The very fact that Microsoft has actually gone to the trouble of incorporating real security into their flagship operating system is a step forward. Secondly, the changes Microsoft has made to the basic Kerberos system are not all that great, meaning it will be pretty close to the original MIT source code. Furthermore, Microsoft is at pains to point out that their implementation of the Kerberos protocol in Windows 2000 is fully compliant with the IETF Kerberos version 5 specification.

Microsoft do have a public license of their own called the 'Microsoft Code License'. A copy can be found in Appendix D. In fact it is no license at all but another disclaimer. Under this license Microsoft re-release code that will enable a Unix platform to authenticate to a Win 2000 domain controller. They have paid lip service to the values of the open source movement but their efforts are nothing more than token. This not surprising given Microsoft's position and history. It has a tendency to take what it needs, absorbing it into the Microsoft entity and not giving much in return, much to the irritation of

many in the open source world. This underscores the fact that open source is more and attitude than it is anything else. It is the desire to work cooperatively and to share ideas.

Conclusion

It is assumed that Win 2000 will supplant Win NT and become the dominant front line corporate operating system. While Kerberos it is not an SSO silver bullet for Win 2000 it does offer a coherent platform and some developer tools for the deployment of SSO. This is due to pure market power rather than any technology watershed created by Microsoft. MIT already provided most of the functionality currently enjoyed by users of Kerberos 5. Rather than setting the trend, Microsoft is following it. Most of the other SSO offerings in the market are using Kerberos as the authentication and authorisation engine. These solutions are currently confined to a small section of the market. For SSO to become a widespread reality applications must be designed and shipped with Win 2000 authentication compatibility. For Windows 2000 software this would provide plug and play SSO which would be in the reach of many organisations. While this is not currently the case it could easily become so. If it did, then boring and repetitious password maintenance would be lifted from the shoulders of systems administrators, and users would have access to everything with just one logon. And Microsoft would have even more market power than it has today.

Will developers who make applications include features to interoperate with Win 2000? Many of these people don't trust Microsoft but cannot afford to make the wrong decision. They are between a rock and a hard place. The collective decisions of this group will determine the success or failure of Microsoft's Win 2000 as an SSO platform. The answer to this question is indeterminate at this stage but should become clearer over the next year or two.

Scrutiny of the Win 2000 source code is impossible because no source code is provided by Microsoft. However this will not reduce the uptake of Win 2000. Most users are not that interested in security. The popularity of Win NT is a testament to this fact. Functionality is what users buy, security is an afterthought. Microsoft's main reason for adopting Kerberos is not for security at all. The ability to use PKI as an authentication method is probably of much more interest as it will position Win 2000 as an e-Business gateway. This has tremendous future growth potential. While Microsoft did not need to take Kerberos to create a PKI logon system, Kerberos offered a convenient platform onto which to add the PKI capability.

What about the fact that Microsoft will profit by the work of others and give nothing in the return. Not even the source code, which is what developers are most interested in. Cannibalism of ideas is the one true constant in the IT world. Microsoft just seem to do it better than most. The concept of open source was never intended to lock anyone out of the game – not even a global giant headed by the richest man in history. Provided it plays by the rules Microsoft has every right to incorporate this technology into its products. In fact, it might be a good thing. It might kill two birds with one stone,

providing better security at the same time as the convenience of SSO.

Appendix A – Kerberos Reference Material

http://web.mit.edu/kerberos/www/

http://web.mit.edu/kerberos/www/dialogue.html

http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/windows2000serv/maintain/featusability/kerberos.asp

http://www.microsoft.com/msj/defaultframe.asp?page=/msj/0899/kerberos/kerberos.htm

http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp

Appendix B – Windows Kerberos HOWTOs

http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnactdir/html/kerberossamp.asp

Appendix C – Open Source Homepage

http://www.opensource.org/

Appendix D – Microsoft Code License

Code License and Access to Samples
This MICROSOFT SOURCE CODE LICENSE is a contract that allows you to use the accompanying software. For short, we'll refer to the Microsoft Source Code License as the "License" and the accompanying software as the "Software."

This License governs use of the accompanying Software. Microsoft hopes you find this Software useful.

You are licensed to do anything you want with the Software.

In return, we simply require that you agree:

1. not to remove any copyright notices from the Software.
2. that the Software comes “as is”, with no warranties. None whatsoever. This means no implied warranty of merchantability or fitness for a particular purpose or any warranty of non-infringement. Also, you must pass this disclaimer on whenever you distribute the Software.
3. that we will not be liable for any of those types of damages known as indirect, special, consequential, or incidental related to the Software or this License. Also, you must pass this limitation of liability on whenever you distribute the Software.
4. that if you sue anyone over patents that you think may apply to the Software, your license to the Software ends automatically (this applies even when the rest of the License ends).
5. that the patent rights Microsoft is licensing only apply to the Software, not to any derivatives you make.
6. that your rights under the License end automatically if you breach this in any way.