

ibm.com



e-business

Using EIM to Enable Single Sign-On for Your IBM **e**server iSeries Server

EP06

ITSO iSeries Technical Forum - 2003

Brian R. Smith



Redbooks

International Technical Support Organization

© 2003 IBM Corporation

Acknowledgments



Thanks to Erik Larsson (IBM Sweden) for his work on this presentation during an ITSO residency at the Raleigh, North Carolina center under the guidance and direction of Thomas Barlen (IBM Germany).

In addition, the ITSO would like to thank Pat Botz and Garry Sullivan of the Rochester Lab for their contributions to this project.

Objectives



Introduction to Enterprise Identity Mapping (EIM)

- Discusses EIM and what it is
 - Why do we need EIM, and what problems does it solve?
- Discusses involved components
 - LDAP Directory, Kerberos, APIs, etc.

Introduction to Kerberos

- Explains Kerberos, its purpose, and how it works

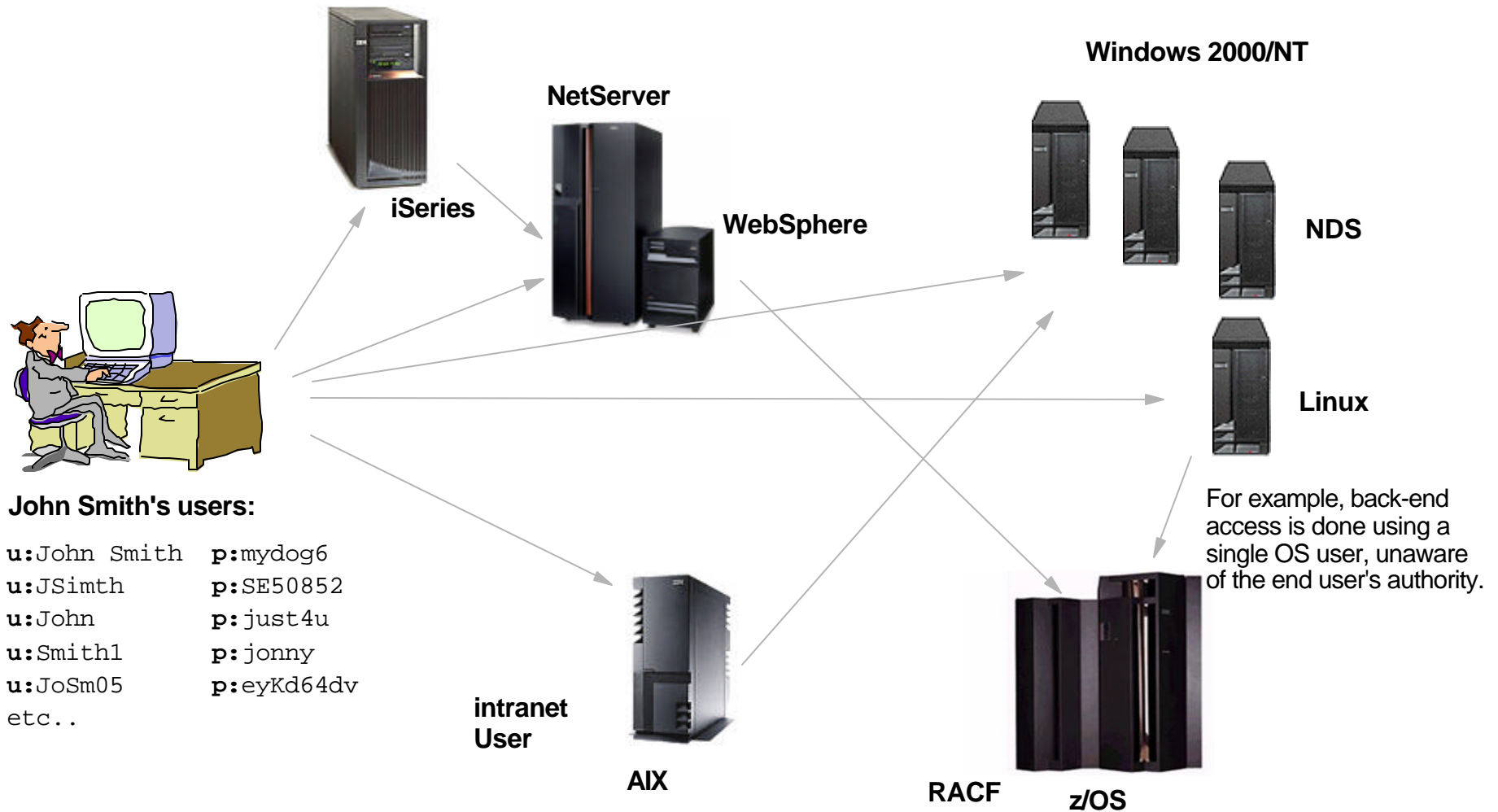
Implementing EIM

- Lists prerequisites
- Adding a principal to a KDC
- Setting up NAS and EIM on an iSeries server



Enterprise Identity Mapping (EIM)

Typical Environment Today



Notes Typical Environment Today



Each system has its own unique user registry, and most likely, its own rules for user IDs and passwords. Users end up with multiple user IDs and passwords. It is quite common that users try to simplify their own local environment by using the same password in multiple systems.

As an application developer, you know that the customer data is spread out across many different types of systems. All of them having their own user registries and associated security semantics. Your only chance of providing a distributed application that works is to provide a new user registry for your application, despite the impact it will cause on administration.

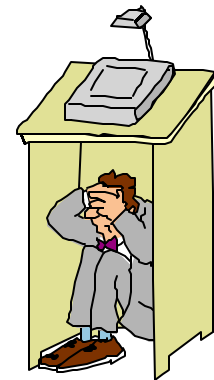
Cross platform distributed applications that span platforms often "agree" who a specific user is. When OS protected resources are accessed, the application projects (maps) the application view of a user into the OS view of the user. The back-end system is forced to trust the front-end servers.

Passwords are often transmitted in the clear.

Problems Today



- Every server platform has unique mechanisms for managing users (User Registries), making it complex for administrators
- Difficult to keep track of users in all systems
- Users have to remember user IDs and passwords to each system they use
- Application developers create their own user registries and use unsafe techniques for access to back-end systems
- Single point of management tools, like Tivoli, solve the problem for administrators, but not necessarily for users or ISVs



Notes Problems Today



In today's heterogeneous networks with partitioned servers and multiple platforms, administrators, users, and application developers all have to cope with the complexities that multiple user identities for individual users create within an enterprise. Users have to remember each user ID and password for each system they use. Administrators must perform password resets, attempt to synchronize user IDs and passwords, and remember every system in the network to which each individual has access. Application developers are often forced to use nonsecure techniques to solve this problem or to invest large amounts of money in writing applications that implement their own user registries and associated security semantics. These problems quickly become a large administrative problem for all parties involved.

One approach to handle a single sign-on environment is to create side-files containing all the users passwords and user IDs. This approach has several flaws. The passwords still need to be managed on these systems and the registries require that the user/password lists are synchronized. The passwords are commonly transmitted in clear-text, and also stored in clear-text or decryptable files directly accessible by the administrator. Finally, these approaches do nothing for third-party application providers wishing to provide heterogeneous, multi-tier applications.

Managing a multi-registry environment is also a burden on the budget. 20% to 40% of all calls to a help desk involve forgotten passwords costing a company \$14 to \$26 per reset.

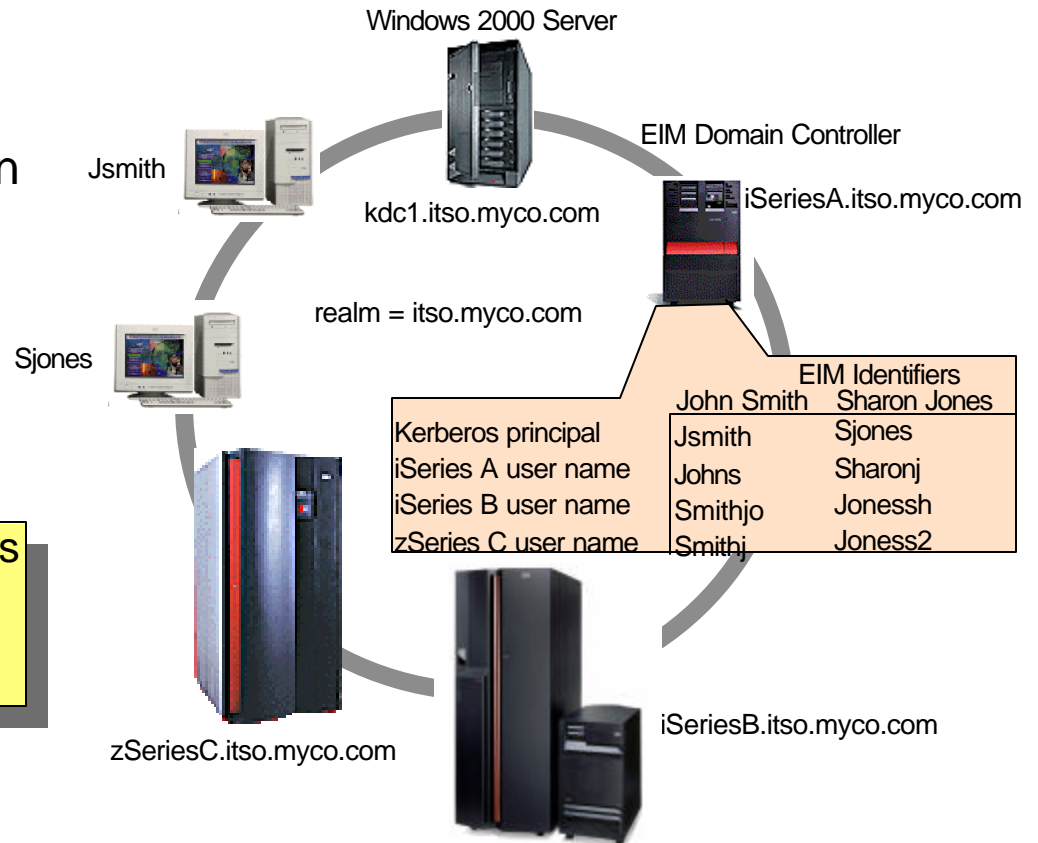
- Source: Gartner Group.

What Is EIM?



- Enterprise Identity Mapping (EIM) is a mechanism for mapping (associating) a person or entity to the appropriate user identities in various registries throughout the enterprise
- EIM provides an infrastructure that lowers the expense for application developers to provide single sign-on solutions
- Autonomic Computing Initiative

EIM defined: Identity associations across user registries associated with OS platforms, applications, and middleware.



Notes What Is EIM?



Enterprise Identity Mapping (EIM) provides an infrastructure that lowers the expense for application developers to provide single sign-on solutions. OS/400's exploitation of EIM and Kerberos, along with exploitation by other IBM platforms and IBM software, provides single sign-on capabilities. This, in turn, provides users, administrators, and application developers the benefits of easier password and user identity management across multiple platforms — without changing the underlying security schema.

EIM allows for OS programmers and ISVs to independently implement support for a single sign-on environment without having to wait for support from a specific product vendor.

EIM is a part of the IBM Autonomic Computing Initiative, a project which goal is to give businesses the ability to manage systems and technology infrastructures that are hundreds of times more complex than those in existence today.

Autonomic Computing Initiative represents the next stage of development under New Tools. Self-managing servers are the ultimate in new tools for our customers. They're self-optimizing, self-configuring, self-healing, and self-protecting.

What EIM Provides



- Enables single sign-on!
- Simplifies administration
 - Rely on existing security semantics already in place for existing data
 - Reduces load on administrators for "lost" passwords
 - Reduces client side risks (cached passwords, post-it notes, etc..)
- Better application design
 - No need to implement new user registries
 - No need to define or enforce additional security semantics
 - Provides maximum flexibility for distributed, multi-tier application developers
- Simplifies the process for the user; access is controlled under the covers
- The iSeries is the first IBM platform that provides EIM-enabled services
 - Shipped with AIX, OS/400, and zOS
 - Downloadable from the web for Linux and Windows



Notes What EIM Provides



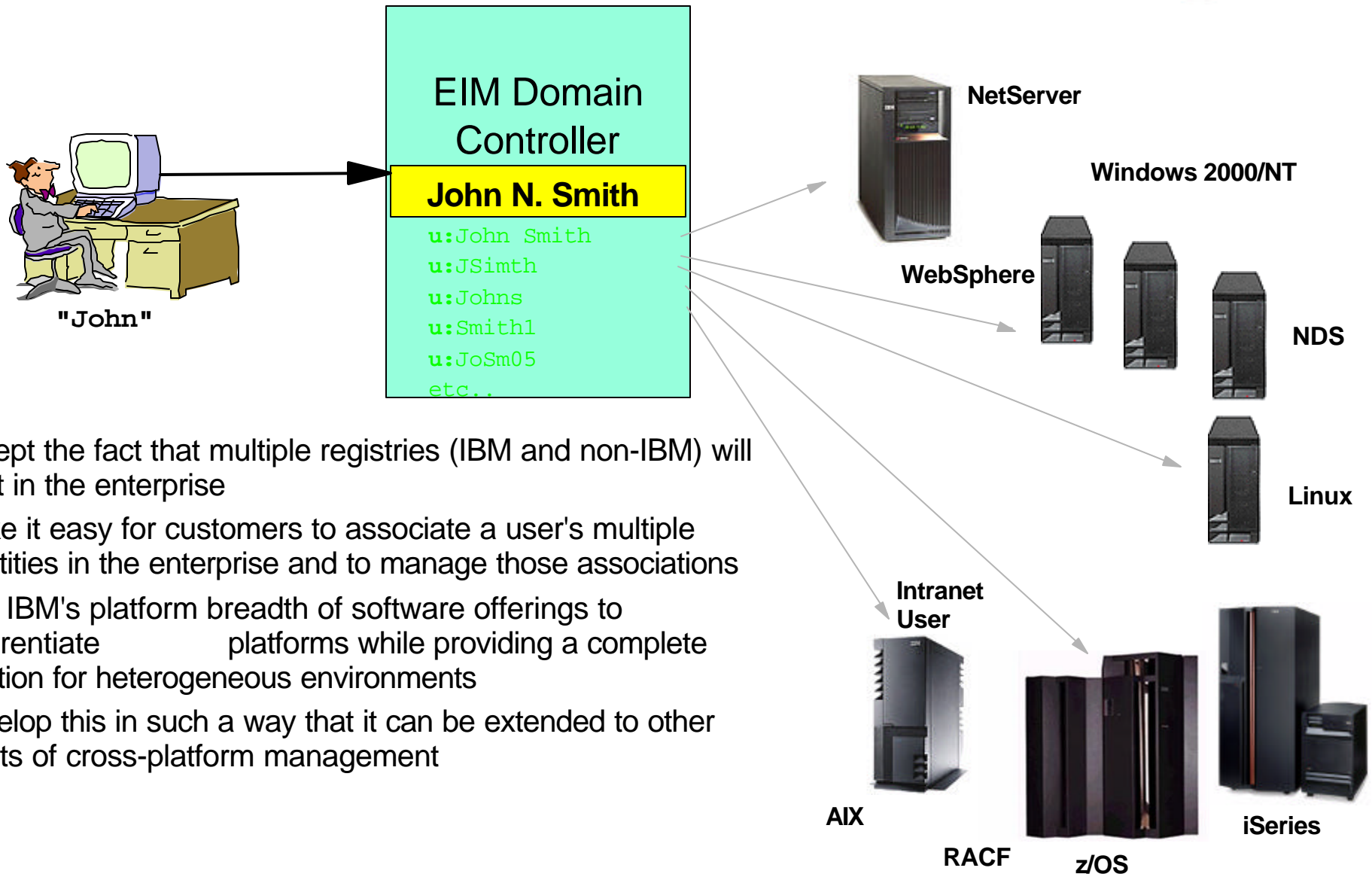
Enterprise Identity Mapping provides the mechanics for cross-platform single sign-on enablement. There are multiple benefits for users, administrators, and application developers alike when single sign-on is used in an enterprise.

The iSeries server uses EIM to enable OS/400 interfaces to authenticate users by means of Network Authentication Service (for example, Kerberos). Applications, as well as OS/400, can accept Kerberos tickets and use EIM to find the user profile that represents the same person as the Kerberos ticket represents.

The EIM infrastructure is now available for all platforms including Linux and Windows. EIM is shipped with AIX, OS/400, and zOS as part of the OS. It is downloadable from the web for Linux and Windows.

Or, more simply put: EIM addresses the run-time needs of applications and platforms which need to "translate" identity when crossing platform and registry boundaries with a set of common services.

Suggested Approach



- Accept the fact that multiple registries (IBM and non-IBM) will exist in the enterprise
- Make it easy for customers to associate a user's multiple identities in the enterprise and to manage those associations
- Use IBM's platform breadth of software offerings to differentiate platforms while providing a complete solution for heterogeneous environments
- Develop this in such a way that it can be extended to other facets of cross-platform management

Notes Suggested Approach



Rather than trying to invent a new user registry or ignore the fact that multiple user registries and their associated security semantics exist, and will continue to exist well into the future, use Enterprise Identity Mapping to coordinate the user identities across existing platforms.

If a user has already been authenticated in one user registry, you can determine which identity in another user registry represents that same person.

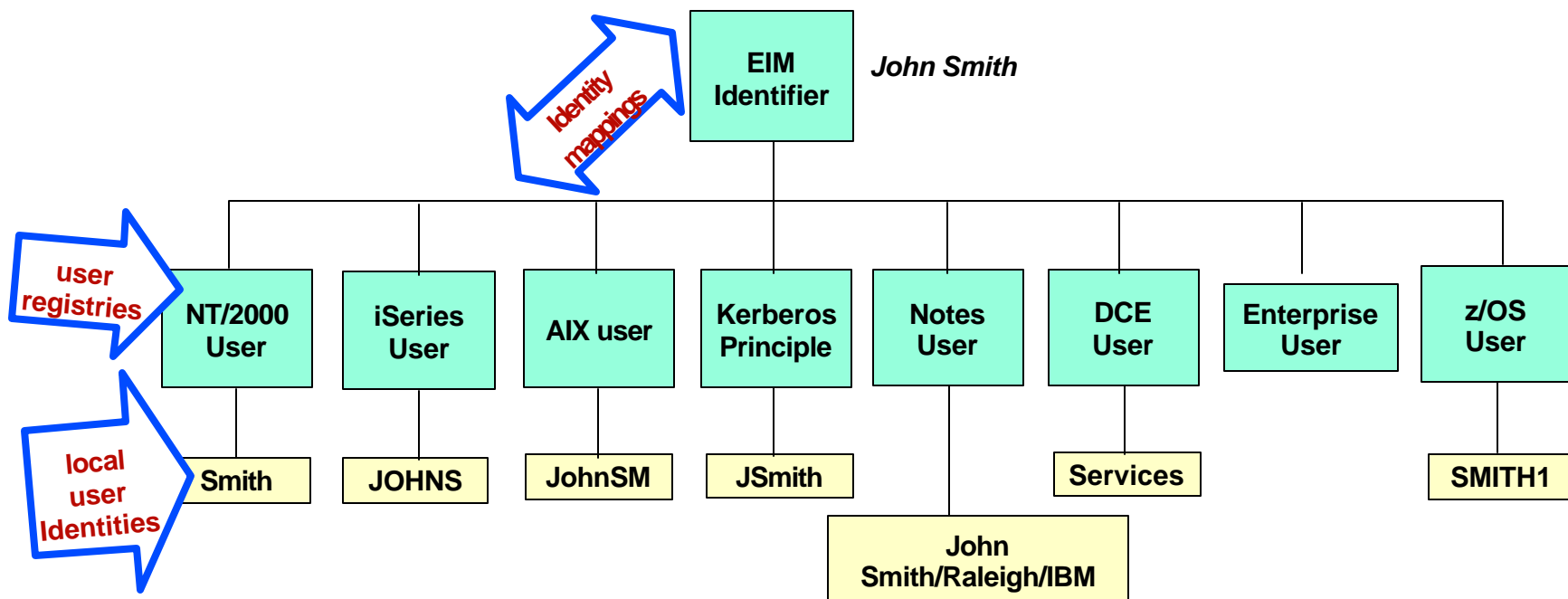
EIM is designed to manage these relationships between individuals or entities in the enterprise and their associated identities in the user registries.

EIM Components: EIM Identifier



An EIM identifier represents an actual person or entity in EIM

The identity associations (mappings) are stored in a well-known location, such as LDAP, with common services across platforms to access the mappings



Notes EIM Identifier



An EIM identifier represents an actual person or entity in EIM. User identities for that person or entity can be associated with the EIM identifier. These identity mappings help to simplify the administrative task of keeping track of all of the user IDs that this person or entity may have within the enterprise.

EIM identifiers can have a description, which can further define the person or entity it represents. You can also create aliases for the EIM identifiers, which can aid in locating a specific EIM identifier when performing a mapping lookup operation.

Quite often different individuals within an enterprise share the same name. EIM identifier names must be unique within the EIM domain and can be confusing as to which individual the identifier belongs. Aliases allow the EIM administrator to have arbitrary and unique EIM identifier names, and to provide additional information about the individual to which the EIM identifier belongs. This information can also be used in a mapping lookup operation.

For example, the EIM identifiers for two people named John S. Smith might be John S. Smith1 and John S. Smith2. The alias for John S. Smith1 could be John Samuel Smith and the alias for John S. Smith2 could be John Steven Smith.

Each EIM identifier can have multiple aliases that can be used to identify which John S. Smith the EIM identifier is representing. Another alias might be added to each of the EIM identifiers for the two individuals that contains their department numbers.

EIM Uses: LDAP



- EIM uses a Directory (LDAP Server) for storing identities along with EIM Domain data
- The Directory server also handles access control to the EIM Domain configuration
- A basic Directory configuration is required for creating an EIM Domain
- A user should never work directly with EIM domain data in the LDAP directory tree
 - EIM APIs are provided to manage the EIM domain



Notes LDAP



Enterprise Identity Mapping (EIM) requires that the Directory Services (LDAP) server is configured with at least a basic configuration. If one does not exist, the EIM wizard configures one for you. From an EIM management point of view, you do not need to access the directory directly.

But if you plan to use the directory for other functions, such as storing employee information, or configuring advanced functions, such as replication or SSL, you should first become familiar with the LDAP directory server. See "Plan your LDAP directory server" in the iSeries Information Center for planning information before you attempt to configure LDAP. If you are familiar with Directory Services and are past the planning stage for LDAP, see "Install and configure Directory Services" (also in the Information Center) to start the configuration process.

Another excellent resource for iSeries Directory Services implementation and use is the IBM Redbook *Implementation and Practical Use of LDAP on the IBM iSeries Server*, SG24-6193.

The directory server is the container for the EIM domain and domain controller information, authorities, as well as access control to the information contained in EIM.

For a production environment, we recommend that you configure the Directory Server to use SSL.

Do not attempt to alter the EIM information without using the EIM APIs.

EIM Uses: NAS



- Network Authentication Service (NAS) enables the iSeries to use Kerberos tickets for authentication instead of a user ID and password
- Applications can identify users and securely pass on the identity to other services
- NAS is built on the Kerberos Network Authentication Service (RFC1510)
- By using APIs, EIM can also be used without NAS for other purposes



Notes NAS



Many platforms including the iSeries server already support Kerberos (also known as Network Authentication Service) for authentication.

Network Authentication Service allows the iSeries and several iSeries services, such as iSeries Access for Windows, to use a Kerberos ticket as an optional replacement for a user name and password for authenticating a user. The Kerberos protocol allows a principal (a user or service) to prove its identity to another service within an insecure network.

Authentication of principals is completed through a centralized server called a key distribution center (KDC). The KDC authenticates a user with a Kerberos ticket. These tickets prove the principal's identity to other services in a network. After a principal is authenticated by these tickets, they can exchange encrypted data with a target service. Network authentication service verifies the identity of a user or service in a network. Applications can securely authenticate a user and securely pass on their identity to other services on the network. Once a user is known, separate functions are needed to verify the user's authorization to use the network resources. Network authentication service implements the following specifications:

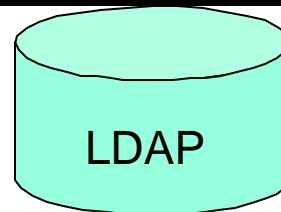
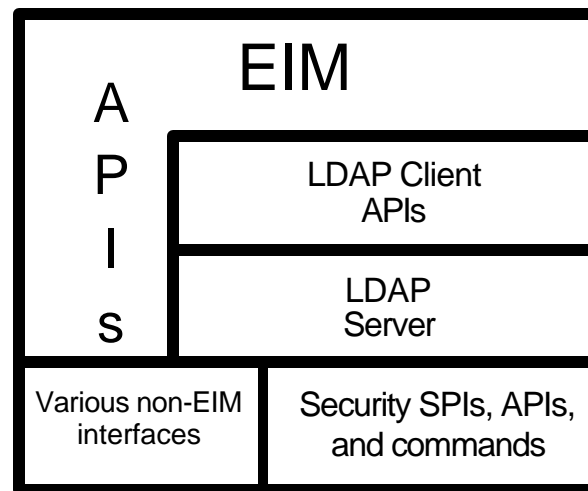
- Kerberos Version 5 protocol Request for Comment (RFC) 1510
- Many of the de facto standard Kerberos protocol APIs prevalent in the industry today
- Generic Security Service (GSS) APIs as defined by RFCs 1509, 1964, and 2743

Network authentication service on the iSeries interoperates with authentication, delegation, and data confidentiality services compliant with these RFCs, such as Microsoft's Windows 2000 Security Service Provider Interface (SSPI) APIs.

EIM APIs



- EIM provides a collection of APIs for OS, IBM and ISV applications
 - No new protocol introduced; uses LDAP
 - EIM APIs can be used for third-party products, for example, for GUI User Admin tools
 - IBM intends to freely distribute EIM APIs and Java packages for ISVs to bundle with their applications



Domain Management

- **eimGetHandle()**
- **eimConnect()**
 - authenticate caller (U1) in registry A (REGA)
 - ..
 - eimGetTargetFromSource(U1, REGA, REGB, associated_identity)
 - setuid(associated_identity)
 - perform task as local identity
 - get next request
- **eimDestroyHandle()**

Notes EIM APIs



Categories of APIs provided by EIM

- EIM "handle" operations - common
 - Manages a token that is an instance of the EIM services. Similar in concept to other services in which the invoker is responsible for hanging on to a "handle"
- Domain operations - EIM Admin
 - Creates a EIM domain, establishes the EIM "domain" controller...
- Registry operations - EIM Admin
 - System or application registries join EIM instance
- EIM Identifier operations - EIM Admin
 - Manages an "anchor" point for an enterprise user
- EIM Core Mapping operations - run-time
 - Supports determination of user's ID across disparate registries
- System operations - System/EIM Admin
 - Connection to an EIM domain
- User Management operations - Admin
 - Definition of this set of services is in progress
 - Direction is to define XML markup(s) that describe:
 - Users within registries and defines data passed on API
 - Allows add/modify/delete of users across multiple registries

Notes EIM APIs (cont)



EIM Exploitation Programming Model - If you intend to write an application that would use the EIM APIs to exploit EIM it might look something like this:

eimGetHandle()

eimConnect()

- authenticate caller (U1) in registry A (REGA)
- ...
- eimGetTargetFromSource(U1, REGA, REGB, associated_identity)
- setuid(associated_identity)
- perform task as local identity
- get next request

eimDestroyHandle()

Notes

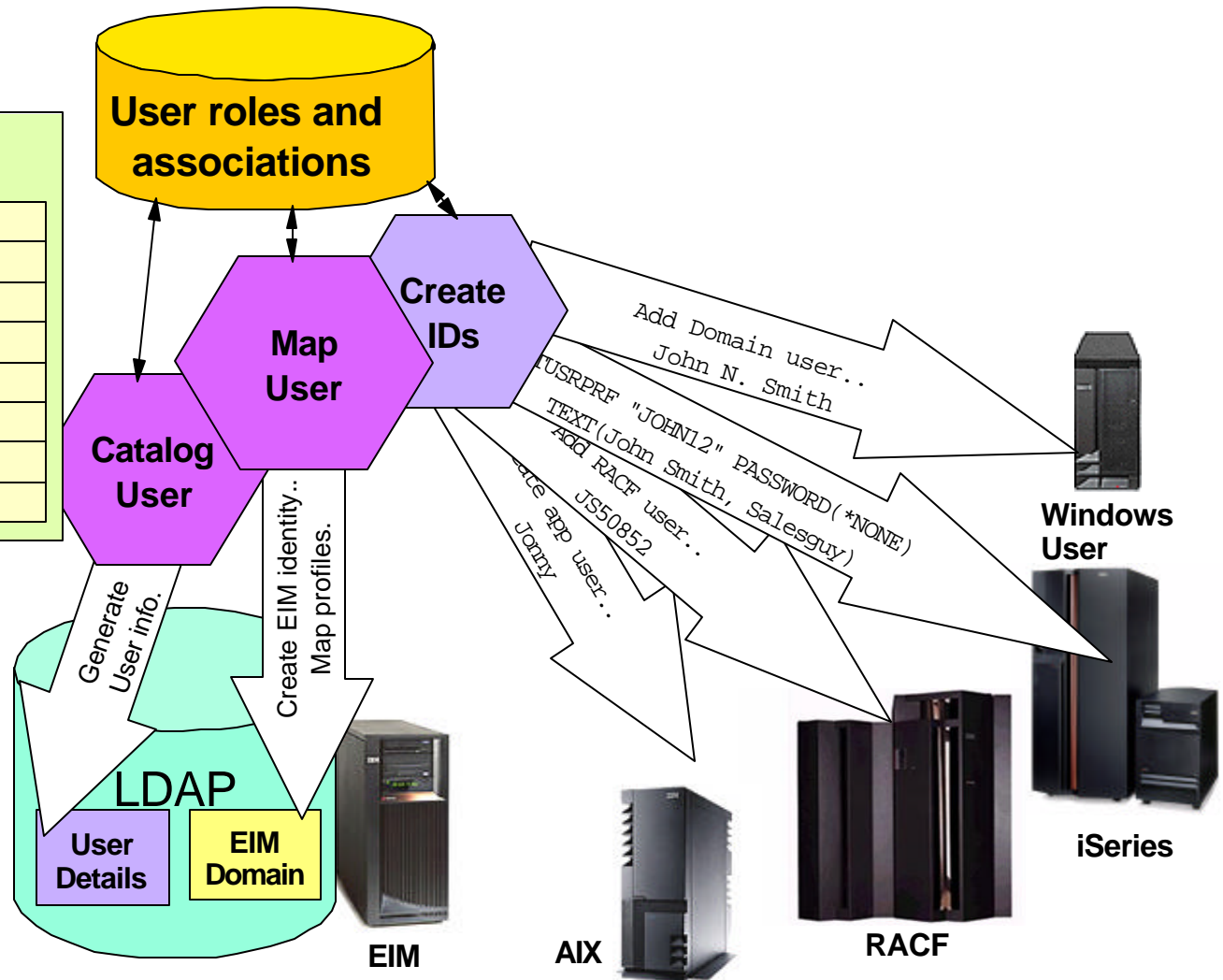


Using APIs to Create a 'User Admin Tool'



A possible use of EIM APIs is an Admin tool to manage enterprise users...

Add enterprise user	
User Name:	Smith, John. N.
Employee #	050852
Branch:	DummyBranch
Role:	Sales person
Start date:	2002 June 10
End date:	2003 June 9
Phone:	1-800-Security
e-mail:	john@dummycorp.com



Notes Using APIs to Create a 'User Admin Tool'



In this example, an administrator creates an enterprise user. This triggers the creation of profiles and IDs in existing environments, according to their "role". These IDs would be created with unique identifiers according to the rules of each registry.

This would also map the user IDs to their identity in the EIM domain database. Finally, this process could involve building an entry for the user in the company's directory. The roles, associations, and management rules could be defined in a single admin database.

These user IDs can be setup with very difficult passwords, or in the iSeries case, you can disable the use of password authentication (by setting the password to *NONE).

By using EIM, the admin tool could also have functions like removing/disabling a user or changing a user's role in the company.

Lets say if the user went to a manager role, it would enable access to administrative systems.

Note: The Java Jar file and the open source version of the Linux EIM APIs will be licensed to allow BPs to freely bundle the APIs with their applications.

Application Provider EIM Exploitation



Developers invest large amounts in building and maintaining for their applications:

- application specific user registries
- associated security semantics
- pair-wise application specific identity mapping

EIM:

- Reduces Development Cost
- Reduces Administrative Costs

ISV Product Opportunities:

- Products for managing EIM from enterprise view (eServer and beyond)
- Automation for "create association" and "define user registry" processes
- System and user management tools that exploit the identity relationship information in EIM (for example, delete all of a user's associated identities)
- Cheaper to build multi-tier, heterogeneous applications

© 2003 IBM Corporation

Notes



Developers invest large amounts in building and maintaining for their applications

- application specific user registries
- associated security semantics
- pair-wise application specific identity mapping

This increases the cost of the application and cost of administering the IT environments that deploy these applications

EIM Reduces Development Cost:

EIM Significantly Reduces Development Costs for Multi-tier, Heterogeneous Apps

- No need to implement new user registries
- No need to define or enforce additional security semantics
- Provides maximum flexibility for distributed, multi-tier application developers

EIM Reduces Administrative Costs:

EIM Significantly Reduces Administrative Costs -- Makes Security Easier to Administer

- Admins don't have to administer new user registries
- Rely on existing security semantics already in place for existing data
- Provides information about a person or entity and all of their associated identities

ISV Product Opportunities:

IBM will make EIM widely available on eServer and non-IBM platforms by:

- The EIM infrastructure is now available for all eServer platforms including Linux and Windows. EIM is shipped with AIX, OS/400, and zOS as part of the OS.
- It is downloadable from the web for Linux and Windows.

Licensed in a way to allow ISVs to freely bundle EIM APIs with their products!

IBM is also considering industry standards body for EIM

SSO enabled interfaces on OS/400



- On the iSeries server using OS/400 at V5R2, the following applications can be accessed through single sign-on:
 - iSeries Navigator
 - Host Servers
 - iSeries Access for Windows
 - PC5250 Emulator (Telnet server)
 - DRDA, ODBC, JDBC, DDM
 - NetServer
 - QFileSvr.400

- The following user registry types are predefined in EIM:
 - OS/400
 - AIX
 - Kerberos
 - Kerberos (case sensitive)
 - LDAP
 - RACF
 - Windows 2000
 - Novell Directory Services
 - Policy Director

Notes SSO enabled interfaces-OS/400



In V5R2 of OS/400, iSeries Navigator and host servers, ODBC/JDBC/DRDA, PC5250+Telnet Servers, NetServer, and QFileSrv400, are enabled for single sign-on via Kerberos and EIM.

This means:

A user can log into a Kerberos-enabled system (for example, Win2K) and never have to enter a user ID and password again. Further, a user ID and password never flows from the system. When the user clicks a system in iSeries Navigator, they are signed on to that system automatically under the appropriate OS/400 user profile. There is no synchronizing of user names or passwords. In fact, the OS/400 user profile can be configured with PASSWORD *NONE if the administrator chooses.

SQL can be submitted via iSeries Navigator (or any standalone ODBC- or JDBC-based application that uses Kerberos for authentication) to access data from iSeries and even connect to other platforms and access data from those machines. Again all of this is done without user IDs or passwords flowing or being coded in the SQL statement. And yet, the appropriate security is enforced at each system using the appropriate user identity and native security semantics. All of this works with no agent code on any of the platforms.

PC5250 allows bypass signon without using user IDs and passwords. This is the 5250 emulator that is part of iSeries Navigator. Note: Personal Communications (often called PCOMM) does not support Kerberos authentication as a client (at this time).

Using a NetServer configured to use Kerberos, users can map OS/400 file systems to their drives without providing a user ID/password. Again the appropriate security is enforced for that user.

QFileSrv.400 is also enabled. You can connect to a single iSeries server with iSeries Navigator and access a QFileSrv.400 mount point that actually points to a second iSeries server. You can have three different user IDs (windows log in, iSeries1 profile, and iSeries2 profile). Without ever being prompted for a user id and password, you can access the mount point (assume you are authorized to the mount point on iSeries1). You can also access the data in iSeries2 (assuming you are authorized to access the data in iSeries2), without ever having to re-enter a user ID and password.

These are the operating system level interfaces that exploit Kerberos and EIM in V5R2.

If you need to, you can define your own User Registry.

Notes EIM-enabled Functions (Cont'd)



User registry

A user registry contains a set of entries that represents a set of "User identity" an operating system or an application either knows or trusts, or both. The set of user identities can be a complete system user registry or a subset of a system user registry that is used with a particular application. A list of users defined for CICS in a particular RACF user registry is an example of such an application registry. When a user registry is created for an operating system to use, for example, the list of OS/400 user profiles on a particular iSeries server, this type of user registry is referred to as a system user registry within EIM. When a user registry is created for a particular application to use, this type of user registry is referred to as an "application user registry" within EIM. The majority of user registries that you work with in EIM are system user registries.

Notes





Kerberos

Notes



Kerberos



- Kerberos is a network authentication protocol
- Designed to establish secure authentication from client to server (and vice versa) on an untrusted network
- Can allow clients to enable cryptography to secure an established connection
 - Client dependent, iSeries Access currently does not support Kerberos encryption
- Widespread throughout the industry, allows for interoperability between platforms
- Simplifies trust management
- Outlined in RFC1510



RFC1510

The Kerberos Network Authentication Service

Notes Kerberos



The Kerberos system was designed and developed in the 1980s by the Massachusetts Institute of Technology (MIT), as part of the Athena project. The current version of Kerberos is Version 5, which is standardized in RFC 1510, The Kerberos Network Authentication Service (V5). For more details, see <http://www.ietf.org/rfc/rfc1510.txt>

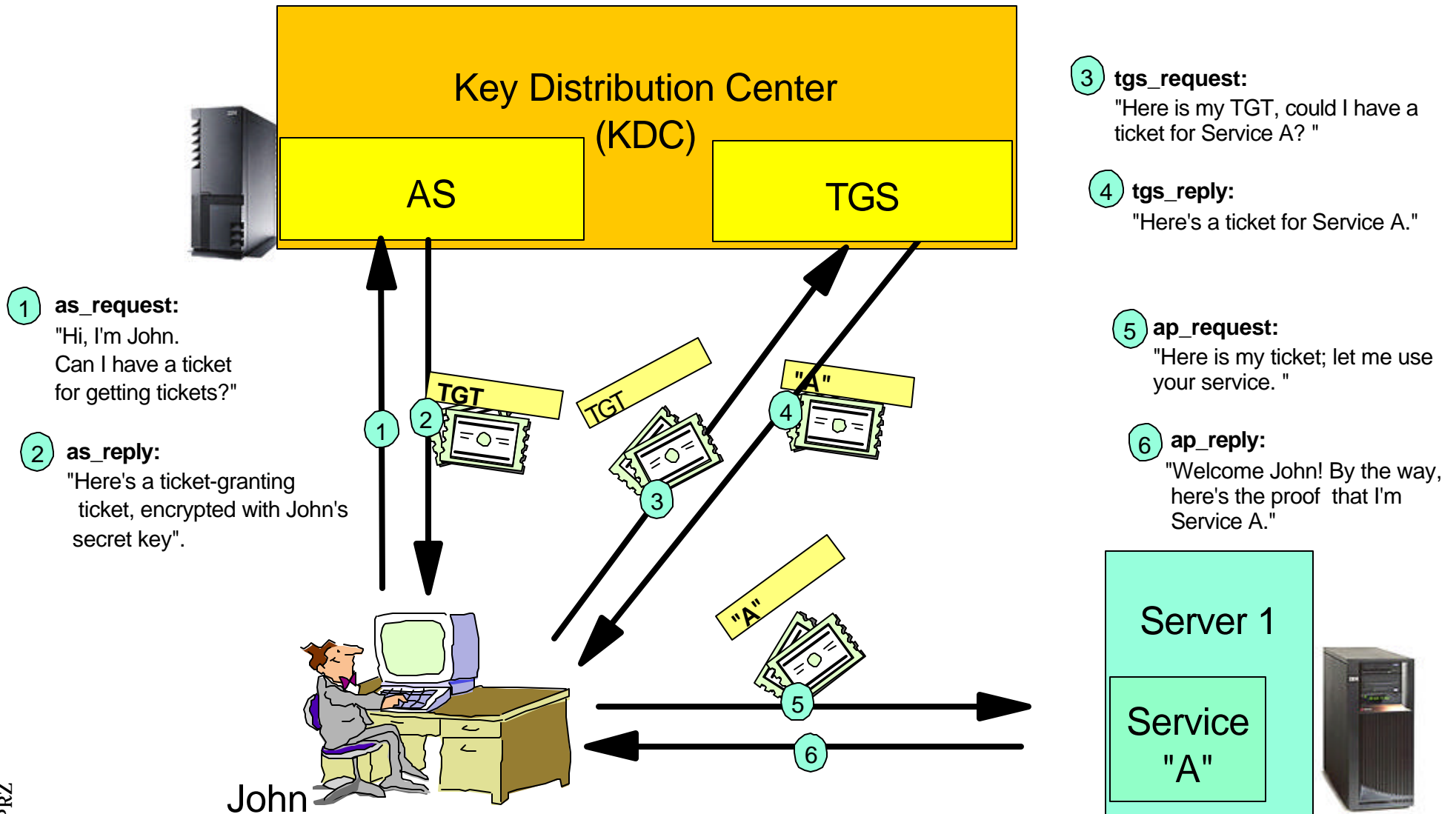
"Kerberos is freely available from MIT, under copyright permissions very similar to those used for the BSD operating system and the X Window System. MIT provides Kerberos in source form so that anyone who wants to use it may look over the code for themselves and assure themselves that the code is trustworthy. In addition, for those who prefer to rely on a professionally supported product, Kerberos is available as a product from many different vendors.

In summary, Kerberos is a solution to your network security problems. It provides the tools of authentication and strong cryptography over the network to help you secure your information systems across your entire enterprise. We hope you find Kerberos as useful as it has been to us. At MIT, Kerberos has been invaluable to our Information/Technology architecture." Source: MIT

Note: We would add that Kerberos is the solution to your network AUTHENTICATION problems (not security problems). Kerberos is not an authorization mechanism at all; it's just an authentication mechanism.

Kerberos authentication itself does not automatically imply that the rest of the session is encrypted. However, Kerberos enables a secure exchange of encryption keys that could be used by a client program for session encryption using the GSS APIs. iSeries Access, for example, does not implement the GSS APIs. However, iSeries Access traffic can be encrypted by SSL instead.

Kerberos Environment



Notes Kerberos



The Kerberos protocol consists of several sub-protocols (or exchanges). There are two methods by which a client can ask a Kerberos server for credentials. In the first approach, the client sends a clear text request for a ticket for the desired server to the Authentication Service (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 which requires Kerberos credentials. The reply is encrypted with the session key from the TGT.

The client and server do not initially share an encryption key. Whenever a client authenticates itself to a new verifier it relies on the authentication server to generate a new encryption key and distribute it securely to both parties. This new encryption key is called a session key and the Kerberos ticket is used to distribute it to the verifier.

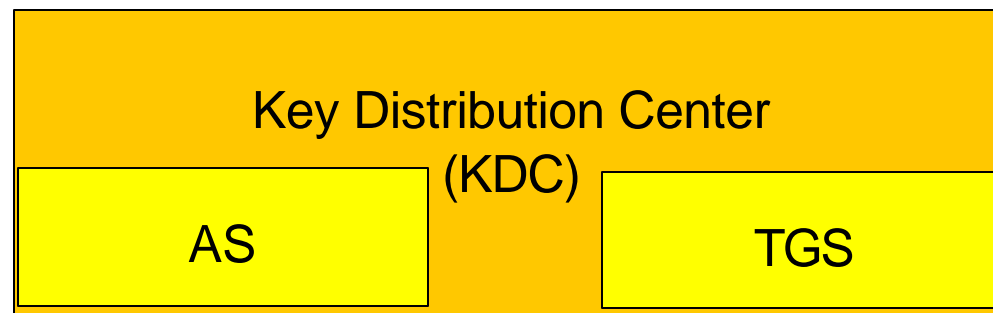
The Kerberos ticket is a certificate issued by an authentication server, encrypted using the server key. Among other information, the ticket contains the random session key that will be used for authentication of the principal to the verifier, the name of the principal to whom the session key was issued, and an expiration time after which the session key is no longer valid. The ticket is not sent directly to the verifier, but is instead sent to the client who forwards it to the verifier as part of the application request. Because the ticket is encrypted with the server key, known only by the authentication server and intended verifier, it is not possible for the client to modify the ticket without detection.

Components in Kerberos: KDC



The Key Distribution Center (KDC) has two primary services:

- **Authentication Server (AS)**
 - The AS contains the shared secret that is required to prove one's identity
 - Once authentication has been made, a TGT is issued
- **Ticket-Granting Server (TGS)**
 - Issues service tickets (containing session keys)
 - Does not keep track of ticket delivery



Notes Components in Kerberos: KDC



A Key Distribution Center (KDC) is a network service that provides tickets and temporary session keys. The KDC maintains a database of principals (users and services) and their associated secret keys. It is composed of the Authentication Server (AS) and the Ticket Granting Server (TGS). It is important that you use a secure machine to act as your KDC. If someone gained access to the KDC, your entire realm could be compromised.

Note: KDC support does not exist on the iSeries server.

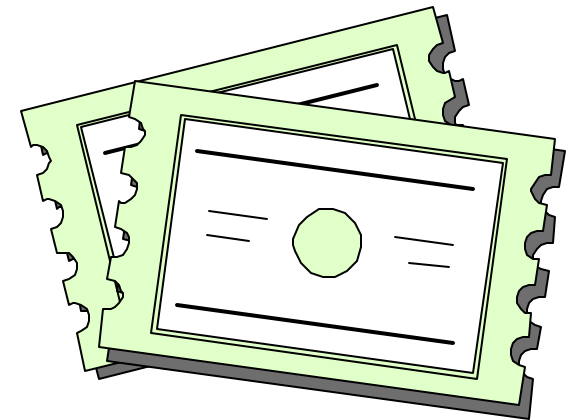
Components in Kerberos: Tickets



- **Ticket:** A record that helps a client authenticate itself to a server or service and establish a session.
- **Ticket-granting ticket (TGT):** A ticket used for requesting tickets subsequently used for sessions. A TGT is received once the proper credentials are given to the Authentication Server.

Some other tickets:

- **Proxiable/Proxy Ticket:** Ticket that can be used by servers to represent the client against a back-end server
- **Forwardable/Forwarded Ticket:** Ticket that delegates the task of obtaining service tickets on behalf of the client



Notes Kerberos Tickets



Ticket:

A record that helps a client authenticate itself to a server. It contains the client's identity, a session key, a timestamp, and other information, all sealed using the server's secret key. It only serves to authenticate a client when presented along with a recently created Authenticator*.

Ticket-granting ticket (TGT):

A ticket that is created once initial authentication has been made. This allows for using a temporary session key for further communication with the TGS instead of using the secret key for each request. The TGT usually has a time limit of 8 to 10 hours, where as the secret key, it would normally have a much longer lifetime. A TGT is also used by the client to obtain tickets to authenticate to services.

Proxiable and proxy tickets:

A proxiable ticket is a ticket-granting ticket (TGT) that allows you to get a ticket for a service with network addresses other than those in the TGT. Unlike forwardable tickets, you cannot proxy a new TGT from your current TGT; you can only proxy service tickets. Forwardable tickets let you transfer your complete identity (TGT) to another machine, where proxiable tickets only let you transfer particular tickets. Proxiable tickets allow a service to perform a task on the behalf of a principal. The service must be able to take on the identity of the principal for a particular purpose. A proxiable ticket tells the KDC that it can issue a new ticket to a different network address, based on the original ticket granting ticket. With proxiable tickets, a password is not required.

Forwardable tickets:

Forwardable tickets allow a server to pass on the credentials of the requester to another service. For this to happen, the initial TGT must have been requested with the forwardable option and the server is allowed to delegate credentials. An example where it might be used is when a user logs in to a remote system and wants authentication to work from that system as if the login were local.

* **Authenticator:** A record containing information that shows that this information (authenticator) has been recently generated using the session key, only known by the client and server. An authenticator consists of the fields listed in the table on the right (see RFC1510 for exact specification):

- Field -	Authenticator value
authenticator-vno	Version format.
crealm	Realm.
cname	Client name
cksum	Checksum of the application data in the request
cusec	Micro second of the timestamp, 0-999999
ctime	Time on host
subkey	Can contain a separate key for this specific session
seq-number	Sequence number

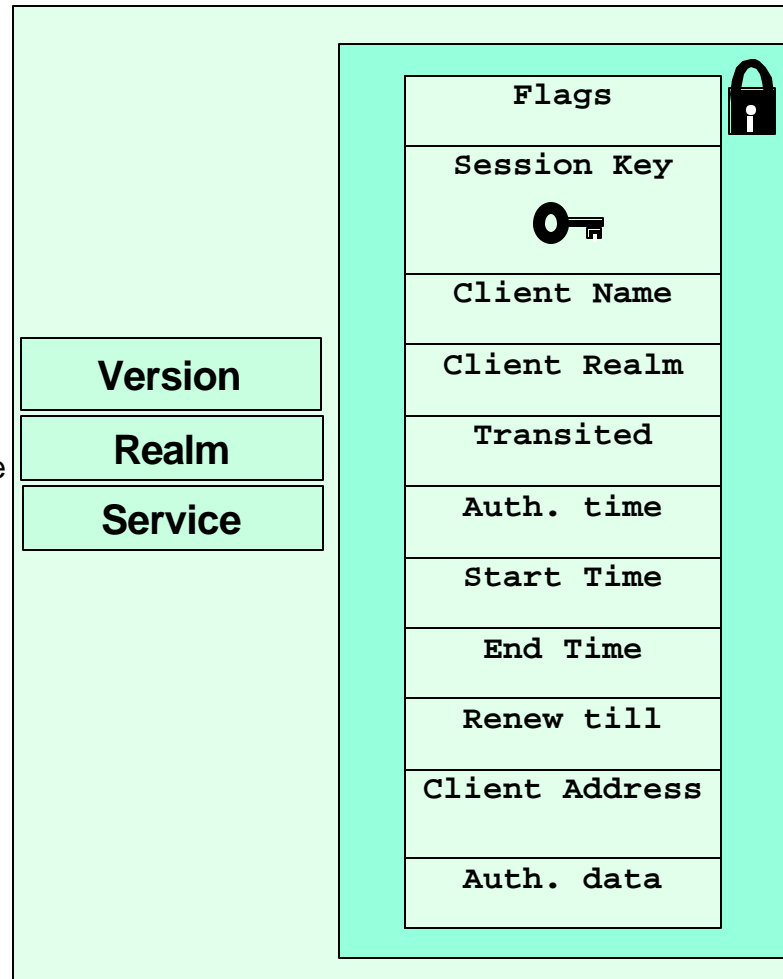
The 'Ticket'



This structure is the same for all tickets

(simplified; see RFC1510 for exact details)

- **tkt-vno:** Kerberos version used (v.5).
- **Realm:** Name of the realm that issued the ticket.
- **Sname:** Server/Service Name the ticket is intended for.



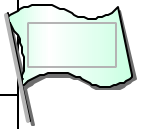
- **Enc-part:** Holds the encrypted part of the ticket. (Using the recipients secret key.)
- **Flags:** Various options that were set when the ticket was issued. Dictates what type of ticket this is.
- **Key:** The Session Key used between the client and server.
- **cname:** The name of the clients principal identifier.
- **crealm:** The realm that initially authenticated the client.
- **transited:** Realms that took part in authenticating the user.
- **Authtime:** Time of authentication
- **StartTime:** The Start time for which the ticket is valid.
- **EndTime:** The time when the ticket expires.
- **RenewTill:** The 'final' end time if the ticket is renewable.
- **CAddr:** Address from which the ticket can be used.
- **Authorization-data:** (Optional) Field used for passing data from issuer to server/service.

Notes Flags in a Ticket



The following table lists the possible flags in a Kerberos version 5 ticket.

Bit	Flag	Description
0	RESERVED	Reserved for future expansion of this field.
1	FORWARDABLE	The FORWARDABLE flag is normally only interpreted by the TGS and can be ignored by end servers. When set, this flag tells the ticket-granting server that it is OK to issue a new ticket-granting ticket with a different network address based on the presented ticket.
2	FORWARDED	When set, this flag indicates that the ticket has either been forwarded or was issued based on authentication involving a forwarded ticket-granting ticket.
3	PROXIABLE	The PROXIABLE flag is normally only interpreted by the TGS, and can be ignored by end servers. The PROXIABLE flag has an interpretation identical to that of the FORWARDABLE flag, except that the PROXIABLE flag tells the ticket-granting server that only non-ticket-granting tickets may be issued with different network addresses.
4	PROXY	When set, this flag indicates that a ticket is a proxy.
5	MAY-POSTDATE	The MAY-POSTDATE flag is normally only interpreted by the TGS, and can be ignored by end servers. This flag tells the ticket-granting server that a post-dated ticket may be issued based on this ticket-granting ticket.
6	POSTDATED	This flag indicates that this ticket has been postdated. The end-service can check the authtime field to see when the original authentication occurred.
7	INVALID	This flag indicates that a ticket is invalid, and it must be validated by the KDC before use. Application servers must reject tickets which have this flag set.
8	RENEWABLE	The RENEWABLE flag is normally only interpreted by the TGS, and can usually be ignored by end servers (some particularly careful servers may wish to disallow renewable tickets). A renewable ticket can be used to obtain a replacement ticket that expires at a later date.
9	INITIAL	This flag indicates that this ticket was issued using the AS protocol and not issued based on a ticket-granting ticket.
10	PRE-AUTHENT	This flag indicates that during initial authentication, the client was authenticated by the KDC before a ticket was issued. The strength of the preauthentication method is not indicated, but is acceptable to the KDC.
11	MAY-AUTHENT	This flag indicates that the protocol employed for initial authentication required the use of hardware expected to be possessed solely by the named client. The hardware authentication method is selected by the KDC and the strength of the method is not indicated.
12-31	RESERVED	Reserved for future use.

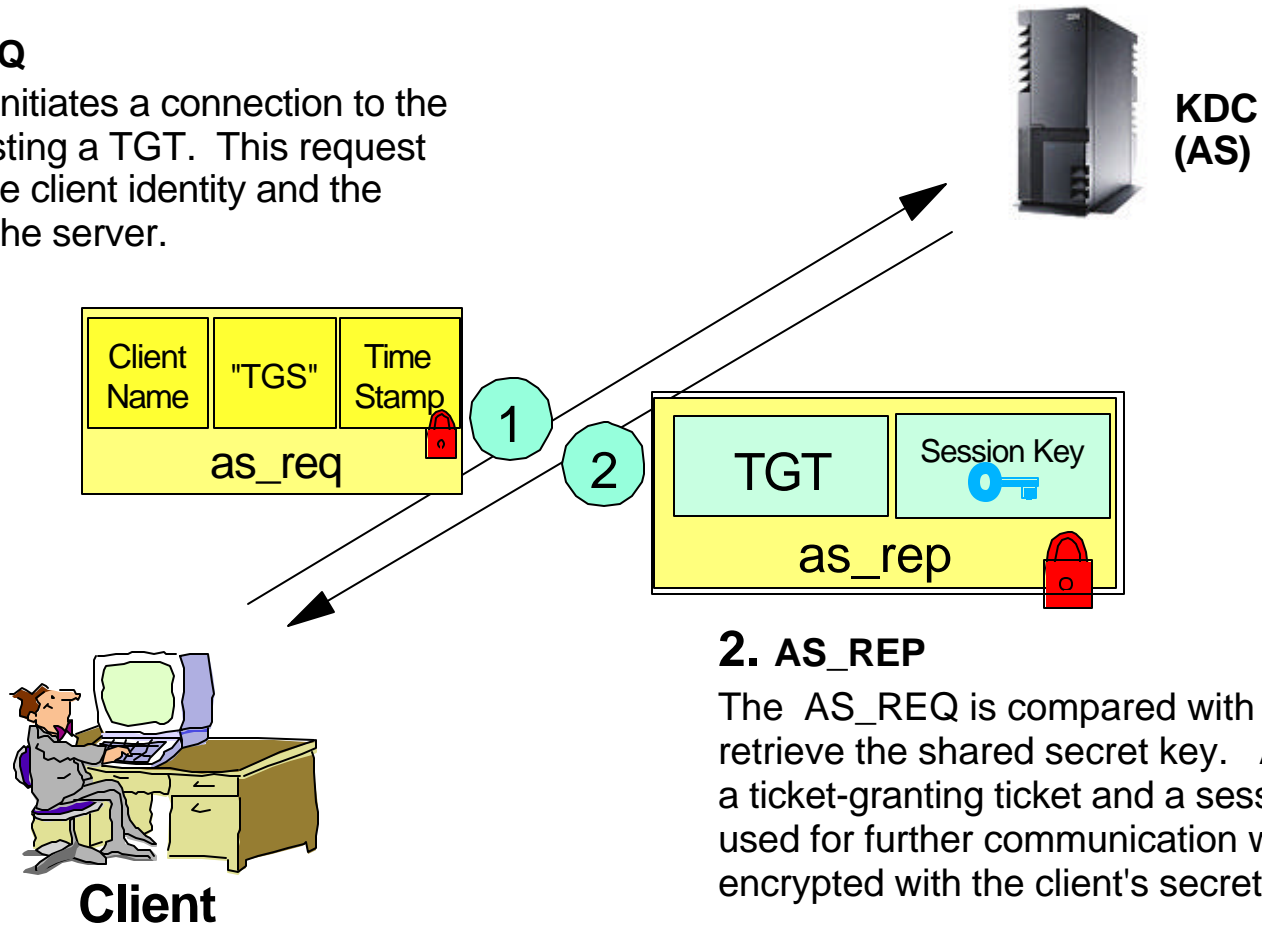


Example Session



1. AS_REQ

The client initiates a connection to the AS, requesting a TGT. This request contains the client identity and the identity of the server.



2. AS_REP

The AS_REQ is compared with existing principals to retrieve the shared secret key. A normal response is a ticket-granting ticket and a session key, which will be used for further communication with the KDC. All are encrypted with the client's secret key.

Notes Example Session



1. AS_REQ:

The client initiates a connection to the AS, requesting a TGT. Optionally, the server can require that the client preauthenticate themselves by using the secret key* to encrypt a timestamp. The request sent contains the client's identity and the identity of the server** in clear text and the optional encrypted timestamp.

2. AS_REP:

The AS_REQ is compared with existing principals to retrieve the shared secret key. A normal response is a Ticket Granting Ticket (TGT) and a Session Key, which will be used for further communication with the KDC. All are encrypted with the client's secret key.

By using a TGT, the client does not have to use its own secret key every time a request is made for credentials to a new service.

Usually the TGT has a lifetime of 8 to 10 hours.

*The *secret key* is derived from the password that the user enters the first time he signs in to the Kerberos *service*. In a Windows 2000 environment, the secret key is generated at the time of logging on to the Domain. Biometrics and smartcards can also be used to increase the security level of the client and storing the secret key.

** The TGS server's identity is "krbtgt".

Example Session (Cont'd)



3. TGS_REQ

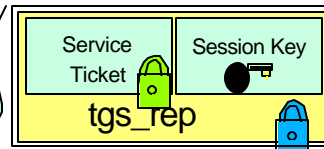
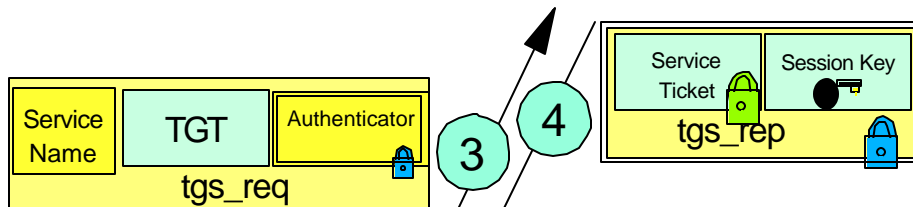
When the client wants to initiate a connection with a service, the client first requests a service ticket from the ticket-granting server. This request consists of the service name, the TGT, and an authenticator proving the identity of John.

KDC (TGS)



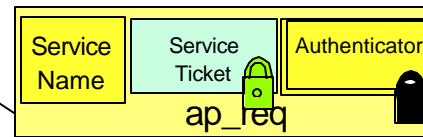
4. TGS_REP

The TGS responds with a service ticket for the requested service and a session key. This response is encrypted with the session key received earlier with the TGT.



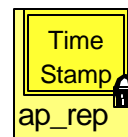
5. AP_REQ

The client can now forward the service ticket, along with an authenticator. After the server validates that the ticket came from the trusted third party, KDC, a session is established.



6. AP_REP

Optionally, the client could require the server to authenticate itself by using the session key to encrypt a timestamp.



Server_A

Notes Example Session (Cont'd)



These steps (3 through 5) are repeated for every new service requested.

3. TGS_REQ

When the client wants to initiate a connection with a service, the client first requests a service ticket from the ticket-granting server. This request consists of the service name, the TGT and an authenticator proving the identity of John. This transaction uses the session key the client received earlier from the AS_REP to encrypt the authenticator.

4. TGS_REP

The TGS responds with a service ticket for the requested service and a session key. This response is encrypted with the session key received earlier with the TGT. Except for the initial fields, the client is not able to decrypt the service ticket. The service ticket can only be used to be forwarded to the intended service. This is why the session key also is sent "outside" of the ticket for the client.

5. AP_REQ

The client can now forward the service ticket, along with an authenticator. After the server validates that the ticket came from the trusted third party, KDC, a session is established. The client used the session key to encrypt the authenticator, which the server can read once the ticket is decrypted with the server's shared secret.

6. AP_REP

Optionally, the client could require the server to authenticate itself by using the session key to encrypt a timestamp. This would prove that the server actually managed to decrypt the service ticket and used the session key for response.

Kerberos Limitations



- Kerberos requires the client to be "secure"
 - Does not protect against Trojans or other password sniffing techniques
 - Putting all of your trust on the client security solution
- Vulnerable to offline brute-force and dictionary attacks
- Not available "everywhere" on any platform
- Clients must use same time (skew of 5 minutes - default)
- Not trivial to set up and understand



Notes Limitations



Kerberos imposes a few assumptions on the environment in which it can properly function:

- "Denial of service" attacks are not solved with Kerberos. There are places in these protocols where an intruder can prevent an application from participating in the proper authentication steps. Detection and solution of such attacks (some of which can appear to be common "normal" failure modes for the system) are usually best left to the human administrators and users.
- Principals must keep their secret keys secret. If an intruder somehow steals a principal's key, it will be able to masquerade as that principal or impersonate any server to the legitimate principal.
- Data encrypted with info in the Ticket by the GSS APIs is vulnerable if someone can sniff both pieces (ticket and encrypted data), but the info in the service ticket or TGT is only valuable for a relatively short period of time and therefore not nearly as susceptible to brute-force attacks.
- "Password guessing" attacks are not solved by Kerberos. If a user chooses a poor password, it is possible for an attacker to successfully mount an offline dictionary attack by repeatedly attempting to decrypt, with successive entries from a dictionary, messages obtained which are encrypted under a key derived from the user's password.
- Each host on the network must have a clock which is "loosely synchronized" to the time of the other hosts; this synchronization is used to reduce the bookkeeping needs of application servers when they do replay detection. The degree of "looseness" can be configured on a per-server basis. If the clocks are synchronized over the network, the clock synchronization protocol must itself be secured from network attackers.
- Principal identifiers are not recycled on a short-term basis. A typical mode of access control will use access control lists (ACLs) to grant permissions to particular principals. If a stale ACL entry remains for a deleted principal and the principal identifier is reused, the new principal will inherit rights specified in the stale ACL entry. By not reusing principal identifiers, the danger of inadvertent access is removed.

Kerberos Limitations (cont)



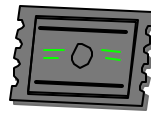
Kerberos addresses authentication only

Client application says
"I am 'patriciaboats@MYCOM.WIN2KDOMAIN1'
and here's proof. "



Windows
2000

Kerberos Ticket



OS/400

How does OS/400
know what
OS/400 resources
the Windows user
is allowed to
access?



OS/400 says:

"I know who you are over there; but I need to know who you are over here to determine what you can access."

Notes Limitations (cont)



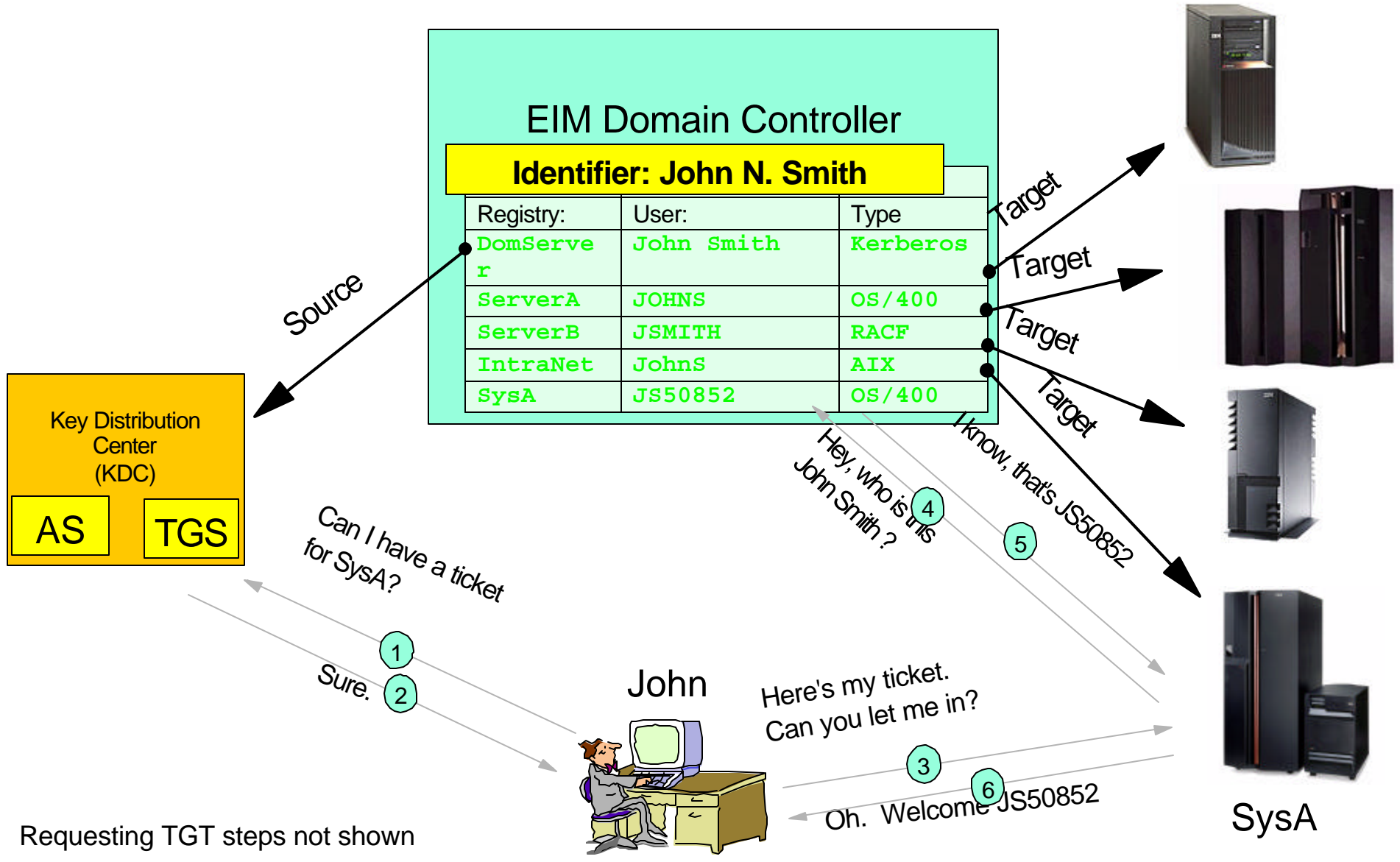
This is a classic problem of authentication (which is what Kerberos does very well) and authorization (which is the security realm of the operating system and applications).

Kerberos alone does not solve the entire problem of multiple user registries for all classes of users!

- Addresses authentication
- Administrators still have to worry about AUTHORIZATION
- Application providers still have to worry about building applications that are
 - easy to deploy
 - easy to manage
 - easy to secure
 - easy to use

Another piece of the technology puzzle is needed! Hint: EIM and Kerberos Together is the next foil.

EIM and Kerberos Together



Requesting TGT steps not shown

Notes EIM and Kerberos Together



The following steps summarize how EIM and Kerberos are used for single sign-on assuming the client already has a TGT:

- 1.) Credentials for a service are requested from the TGS.
- 2.) A service ticket is returned for *Sys_A*.
- 3.) The client requests access to the service on *SysA* using the service ticket.
- 4.) *Sys_A*, which is capable of handling EIM requests, uses EIM APIs to forward the user identity to the EIM domain controller. The EIM controller looks at the "**source**" user and registry to find an identifier in the EIM database.
- 5.) The EIM server returns the user ID for which that identifier has a "**target**" registry entry.
- 6.) *Sys_A* opens the connection for *John* and lets him in as the OS/400 user JS50852, with appropriate authorizations.



Implementing EIM

Prerequisites



iSeries

- OS/400 V5R2 (5722-SS1)
 - Including Qshell interpreter (Option 30) and Host Servers (opt.12)
- Cryptographic Access Provider 128-bit (5722-AC3)
- iSeries Access for Windows (5722-XE1)
- The "latest" PTF package installation

Client

- Windows 2000/XP
- iSeries Access (Version 5 Release 2 or higher)
 - iSeries Navigator including the "Network" and "Security" components (for administration)
- Other clients that support Kerberos authentication

KDC

- Supporting Kerberos Version 5
- iSeries does not support KDC functionality

Our Environment



Client1

Windows 2000 Professional
iSeries Navigator V5R2 installed
Windows user: John
Ralyas4a: JOHNS
Ralyas4b: JSMITH



Domain:
ISERIES
Kerberos Realm:
ISERIES.ITSO.RAL.IBM.COM
EIM Domain:
ITSO_EIM



Ral400kdc

Windows 2000 Advanced Server, Active Directory
Host name: ral400kdc.iseries.itso.ral.ibm.com
IP: 9.25.105.57
Domain Server for: ISERIES



RALYAS4A

iSeries Model 270, OS/400 V5R2
Host name: ralyas4a.iseries.itso.ral.ibm.com
IP: 9.25.105.24
Principal name:
krbsvr400/ralyas4a.iseries.itso.ral.ibm.com
Operates as EIM Domain Controller



RALYAS4B

iSeries Model 170, OS/400 V5R2
Host name: ralyas4b.iseries.itso.ral.ibm.com
IP: 9.25.105.25
Principal name:
krbsvr400/ralyas4b.iseries.itso.ral.ibm.com

Implementation Overview



1. Create a Windows user and principal
2. Configure NAS on the iSeries server
3. Configure EIM
4. Add a domain and identifier to EIM
5. Set up iSeries Navigator to use Kerberos and EIM
6. Add another iSeries registry

Step 1: Windows: Create Principal



Create a Windows user account for the iSeries Principal.

Active Directory Users and Computers

Active Directory Users and Computers

New Object - User

Create in: ISERIES.ITSO.RAL.IBM.com/Users

First name: RALYAS4A Initials:

Last name:

Full name: RALYAS4A

User logon name: RALYAS4A @ISERIES.ITSO.RAL.IBM.COM

User logon name (pre-Windows 2000): ISERIES\RALYAS4A

New Object - User

Create in: ISERIES.ITSO.RAL.IBM.com/Users

Password:

Confirm password:

User must change password at next logon

User cannot change password

Password never expires

Account is disabled

< Back Next > Cancel

Notes Windows Setup - Create Principal



This setup assumes your Windows 2000 Server is already using Active Directory. If not, the easiest way to set it up is just to start the Windows configuration wizard. Click **Start -> Settings -> Control Panel -> Administrative Tools -> Configure Your Server**. If you are unfamiliar with Active Directory, do not attempt to do this in a production environment without first reading about the implications. Microsoft's Active Directory home page on the Web provides more information about the Active Directory:

<http://www.microsoft.com/windows2000/technologies/directory/ad/default.asp>

The principal* for the iSeries will be tied to a Windows user account. This is required for the Kerberos functionality to work. We recommend (but do not require) that the Windows User name is the same as the intended iSeries server name.

* Principal

The name of a user or service in a Kerberos network. A user is considered a person where a service is used to identify a specific application or set of operating system services. On iSeries, the krbsvr400 service principal is used to identify the service used by iSeries Access for Windows, QFileSrv.400 and Telnet servers when authenticating from the client to the iSeries.

Note: You could very well configure Kerberos and EIM on the OS/400 side first. By doing this you will clearly know what the name of the principal they should create on the Windows side. The principal name is "krbsvr400/<hostname>". By running the wizards on OS/400 first will tell you the exact name you need to create.

Step 1: Windows: Create Principal (Cont'd)



- Map the Windows user account to the iSeries principal name.
- Making the account trusted to use delegatable tickets

> ktpass..

Principal name

Realm

```
C:\>ktpass -princ krb5vr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM
-mapuser RALYAS4A -pass password
Successfully mapped krb5vr400/ralyas4a.iseries.itso.ral.ibm.com to RALYAS4A.
Key created.
Account has been set for DES-only encryption.
C:\>
```

Windows User

Password
(Secret key)

RALYAS4A Properties

Published Certificates | Member Of | Dial-in | Object | Security

Environment | Sessions | Remote control | Terminal Services Profile

General | Address | Account | Profile | Telephones | Organization

User logon name: krb5vr400/ralyas4a.iseries.itso.ral.ibm.com @ISERIES.ITSO.RAL.IBM.COM

User logon name (pre-Windows 2000): ISERIES\RALYAS4A

Logon Hours... Log On To...

Account is locked out

Options:

- Account is disabled
- Smart card is required for interactive logon
- Account is trusted for delegation
- Account is sensitive and cannot be delegated
- Use DES encryption

Select the **Account is trusted for delegation** box.

Active Directory Users and Computers

Console Window Help

Action View

Tree

- Active Directory Users
- ISERIES.ITSO.RAL
 - Builtin
 - Computers
 - Domain Control
 - ForeignSecurity
 - LostAndFound
 - System
 - Users

Users 17 objects

Name	Type
Domain Users	Security Group - Domain Local
Enterprise Admins	Security Group - Domain Local
Group Policy Creator Owners	Security Group - Domain Local
Guest	User - Domain Local
krbtgt	User - Domain Local
RALYAS4A	User - Domain Local
RALYAS4B	User - Domain Local
RAS and IAS Servers	Security Group - Domain Local
Schema Admins	Security Group - Global



Notes Windows: Create Principal



The ktpass command is a part of the Support Tools that are included on the Windows 2000 Server installation CD. You must run this command to tie the principal name to a Windows User ID. krbsrv400 is the service name that iSeries Access will ask for when attempting to get authorization to a service.

Note: This is done in the Active Directory on the Windows server.

The ktpass options include:

- princ** The principal name <user@Realm>
- mapuser** Map principal to user account
- pass** The password or secret key that will be shared

Include the following option to override existing principal mapping (useful if you think you made a mistake the first time).

-mapOp set

The correct syntax would look like this.

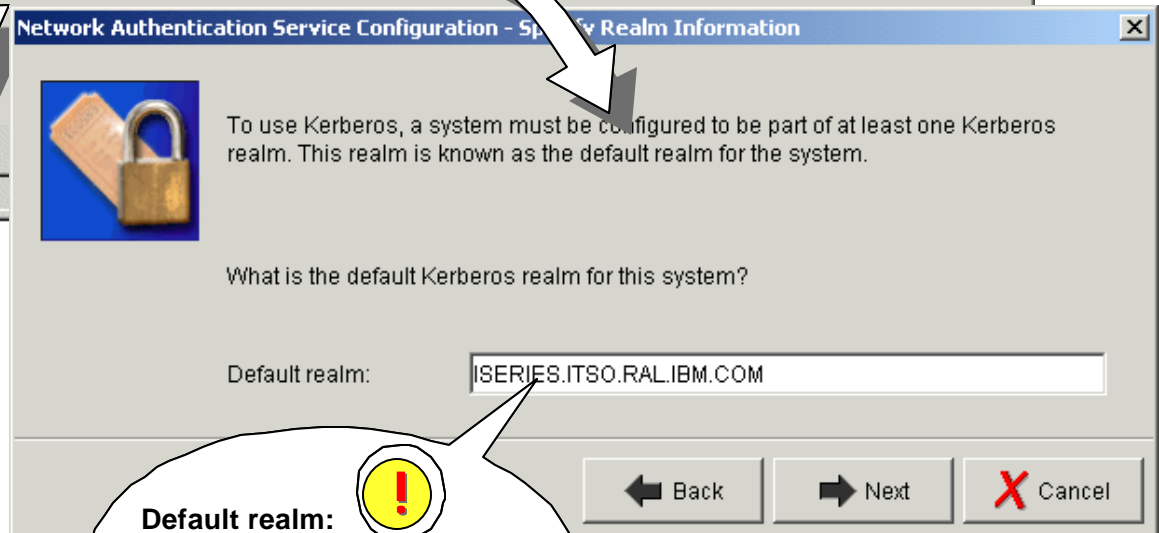
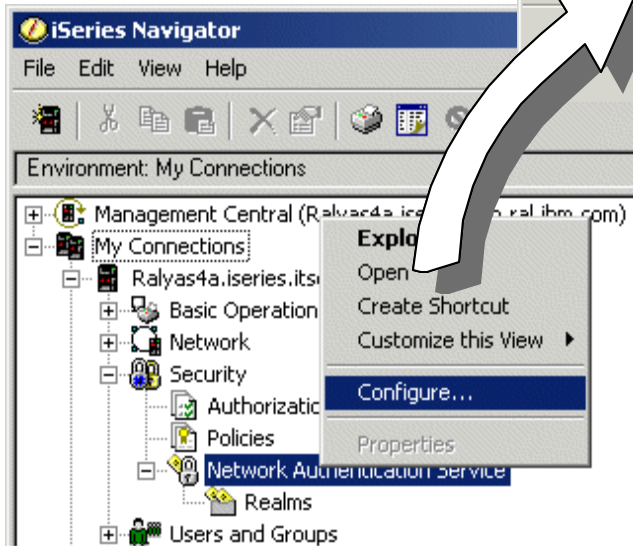
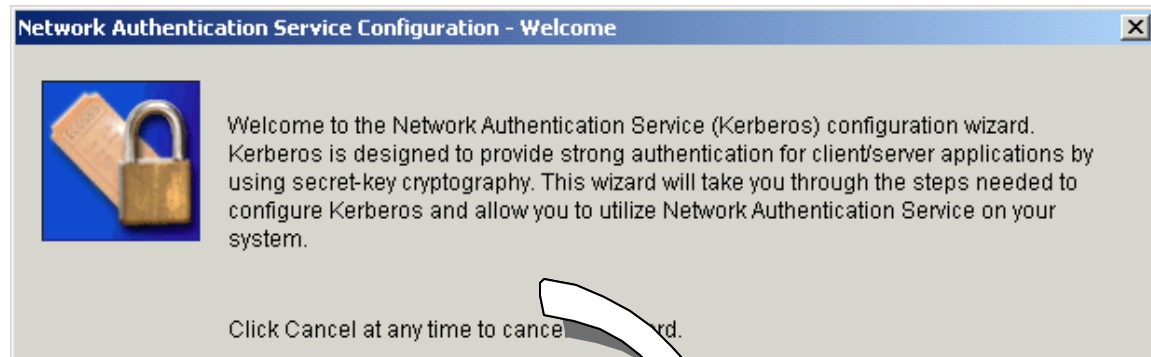
```
ktpass -princ krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM -mapuser RALYAS4A  
-pass password -mapOp set
```


Select **Account is trusted for delegation** to allow the iSeries to forward its ticket to other services, like QFileSrv.400, DRDA, and PC5250. This flag generates "Forwardable" tickets. Service principals must be trusted for delegation for single signon.

Step 2: OS/400: Configure NAS



**Configure NAS
Enter Kerberos Realm.**



Default realm: 
In our case, the Windows Domain name.
Note: Use all uppercase

Step 2: OS/400: Configure NAS (Cont'd)



Add an entry for the KDC and password server (the same server in our case).

Network Authentication Service Configuration - Specify KDC Information

A Kerberos Key Distribution Center (KDC) has two functions. It authenticates principals in the realm and provides service tickets which clients use to access enabled services.

What is the name of your KDC for the default realm?

KDC:

Port:

Next

KDC: Enter the name of the KDC.
Port: 88 (default)

Network Authentication Service Configuration - Specify Password Server Information

A Kerberos password server allows clients to change their password on the KDC remotely. The password server typically runs on the same machine as the KDC.

Do you want to configure this system to use a password server for the default realm?

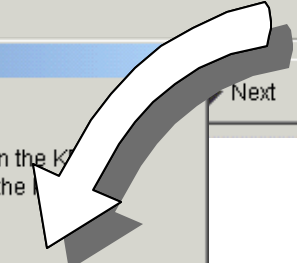
Yes

Password server:

Port:

No

Back



Password Server:
The same as the KDC
Port: 464 (default used by Windows)

Notes OS/400: Configure NAS



Port 88 is the default port used by the Kerberos authentication mechanism. Although the standard says use UDP, Microsoft's solution uses TCP to allow for more data to pass in the Authorization data field, including the Security Identification (SID), all the groups in which the user has membership).

Since the Windows Server should only provide this information if the requesting host is also a Windows platform, and that it also responds to requests on UDP port 88, the KDC should still be compatible with other Kerberos systems.

Port 464 is the port for password administration. This enables the password to be changed remotely from a Kerberos client using the **kpasswd** command. The password server typically runs on the KDC.

Step 2: OS/400: Configure NAS (Cont'd)



**Specify the key table.
Enter the secret key for
the iSeries Principal.**

Network Authentication Service Configuration - Create iSeries Keytab Entry

Kerberos enabled services require a keytab file is used to securely store a term key.

For which of the following services would you like to create a keytab entry?

- iSeries Kerberos Authentication
- LDAP
- iSeries

Keytab: /QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab

Principal: krbsvr400/ralyas4a.iseries.itso.ral.ibm.com

Password: [masked]

Confirm password: [masked]

Back Next Cancel

Principal: This entry is picked up from the iSeries Access client.

Key table entry:
We only create key table entries for iSeries Kerberos Authentication

The password used is for the iSeries principal on the KDC (the secret key that was previously entered in ktpass, step 1).

Notes OS/400: Configure NAS



Network authentication can also be used to set up authentication against the LDAP Server (Directory) or the NetServer. In the NetServer case, the EIM environment can include that service as a target, but currently not the LDAP Server. You can, however, still use Kerberos for authentication against the LDAP server outside of EIM.

The full Principal name is picked up from the iSeries connection that is currently being used to run the wizard. If this does not match the exact, fully qualified name of the iSeries server, you have to check the client setup, for instance, the hosts file or DNS configuration.

The *krbsvr400* principal name is the service name for iSeries Access.

The password entered here is for the secret key between the iSeries and the KDC.

If you select LDAP, an entry for LDAP is created in the key table:

LDAP/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM

Selecting NetServer creates a long list of the possible NetServer names that all can be used to access the NetServer as a Kerberos service:

HOST/ralyas4a...

cifs/ralyas4a...

HOST/QRALYAS4a...

HOST/9.25.105.24@ISERIES.ITSO..

etc..

Step 2: OS/400: Configure NAS (Cont'd)



Finalize the configuration.



krb5.conf

/QIBM/UserData/OS400/NetworkAuthentication/
The Kerberos configuration file:

```

??(libdefaults??)
default_keytab_name =
/QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab
default_realm = ISERIES.ITSO.RAL.IBM.COM
??(realms??)
ISERIES.ITSO.RAL.IBM.COM = ??<
kdc = ral400kdc.iseries.itso.ral.ibm.com:88
kpasswd_server = ral400kdc.iseries.itso.ral.ibm.com:464
??>
??(domain_realm??)
ralyas4a.iseries.itso.ral.ibm.com = ISERIES.ITSO.RAL.IBM.COM
??(capath??)
    
```

Network Authentication Service Configuration - Summary

You have completed all the steps necessary to configure Kerberos on your system.

Click Finish to configure Kerberos with the following settings:

Setting	Value
Default realm:	ISERIES.ITSO.RAL.IBM.COM
KDC:	ral400kdc.iseries.itso.ral.ibm.com:88
Password server:	ral400kdc.iseries.itso.ral.ibm.com:464
Set iSeries keytab entry:	Yes
Set LDAP keytab entry:	No
Set NetServer keytab entries:	No

Note: The configuration file can be viewed at
/QIBM/UserData/OS400/NetworkAuthentication/krb5.conf upon

Back Finish Cancel

Click **Finish**. Then you quickly return to iSeries Navigator.



krb5.keytab

/QIBM/UserData/OS400/NetworkAuthentication/keytab/
The Kerberos key table.

Notes OS/400: Configure NAS



The configuration file `/QIBM/UserData/OS400/NetworkAuthentication/krb5.conf` contains the following fields after the initial configuration

libdefaults	Sets the Kerberos defaults for your system
realm	States were to find the KDC for each realm
domain_realm	Maps domain names to realms
capath	Cross realm authentication path; would contain paths for direct (nonhierarchical) authentication

`/QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab`

The key table* is created and stored in this directory. The iSeries server uses the keytab to open tickets received from clients. It is, of course, very important that this file is protected.

***key table:**

A key table is a file on the service's host system. Each entry in the file contains the service principal's name and secret key. On the iSeries, a key table file is created during configuration of network authentication service. When a service requests authentication to an iSeries with Network Authentication Service configured, that iSeries checks the key table file for that service's credentials. To ensure that users and services are authenticated properly, you must have services enrolled on the iSeries server.

Since the server cannot enter its secret key (password) manually, it reads it from the keytab file.

Note: The easiest way to remove a NAS configuration is to delete the `krb5.keytab` and replace `krb5.conf` with the one found in the `ProdData` directory (`/QIBM/ProdData/OS400/NetworkAuthentication/krb5.conf`).

Step 2: OS/400 QSH: Verify NAS Setup



Start Qshell (QSH), and run the following commands to verify the Kerberos configuration.

1

Run the command **keytab list**. This lists the keys existing in the default key table.

2

Use **kinit -k <principal>** to initiate a ticket exchange between the iSeries and the KDC.

```
QSH Command Entry
$
> keytab list
Key table: /QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.ke
Principal: krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM
Key version: 1
Key type: 56-bit DES
Entry timestamp: 2002/04/29-11:28:38

Principal: krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM
Key version: 1
Key type: 56-bit DES using key derivation
Entry timestamp: 2002/04/29-11:28:38

Principal: krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM
Key version: 1
Key type: 168-bit DES using key derivation
Entry timestamp: 2002/04/29-11:28:38

F3=Exit F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top F18=Bottom F21=CL command entry

> kinit -k krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM
$
> klist
Ticket cache:
FILE:/QIBM/USERDATA/OS400/NETWORKAUTHENTICATION/creds/krbcred_ccfff7e0
Default principal:
krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM

Server: krbtgt/ISERIES.ITSO.RAL.IBM.COM@ISERIES.ITSO.RAL.IBM.COM
Valid 2002/05/01-14:45:20 to 2002/05/02-00:45:20
$
====>
F3=Exit F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top F18=Bottom F21=CL command entry
```

Note: The user performing these steps must have a home directory.



3

Use **klist** to display the default ticket cache.

Notes OS/400 QSH: Verify NAS Setup



These steps are not required for the Network authentication to work. However, by performing these steps, you confirm that the Kerberos environment is working correctly.

Note: The user performing these steps must have a home directory in the IFS. The home directory stores the krb5ccname file, containing the link to the credential cache.

> keytab list

This lists the current keys in the Kerberos key table. If the wizard completed correctly and made contact with the KDC, it should now contain three entries for the krbsvr400 principal (at different encryption levels). If the principal name of the krbsvr400 service displays a wrong host name, verify that the host table on the PC you are performing the configuration on has the correct entries.

> kinit -k krbsvr400/ralyas4a.iseries.itso.ral.ibm.com@ISERIES.ITSO.RAL.IBM.COM

This requests a TGT from the KDC. This should complete with out error and return the prompt.*

> klist

This lists the tickets in the ticket cache and should display the newly received ticket from the KDC.

* Some errors that could occur at the kinit stage:

Unable to obtain name of default credentials cache

- While in QSH create a home directory for your user profile with the command mkdir /home/<userprofile>

Unable to obtain initial credentials.

Status 0x96c73a06 - Client principal is not found in security registry.

The krbsvr400 principal had been misspelled.

Status 0x96c73a25 - Time differential exceeds maximum clock skew.

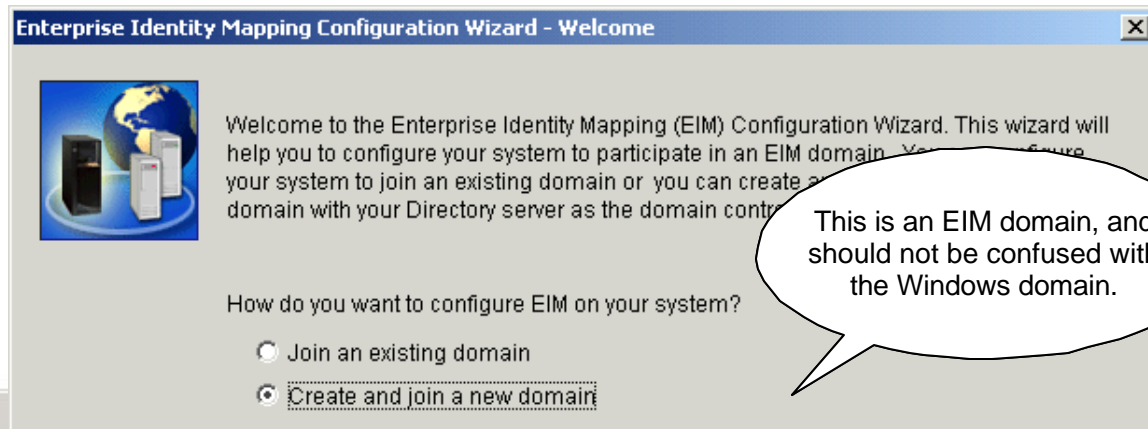
The KDC was using daylight savings time.

Status 0x96c73a9a - Unable to locate security server © 2003 IBM Corporation Realm name resolving incorrectly. Check case sensitivity.

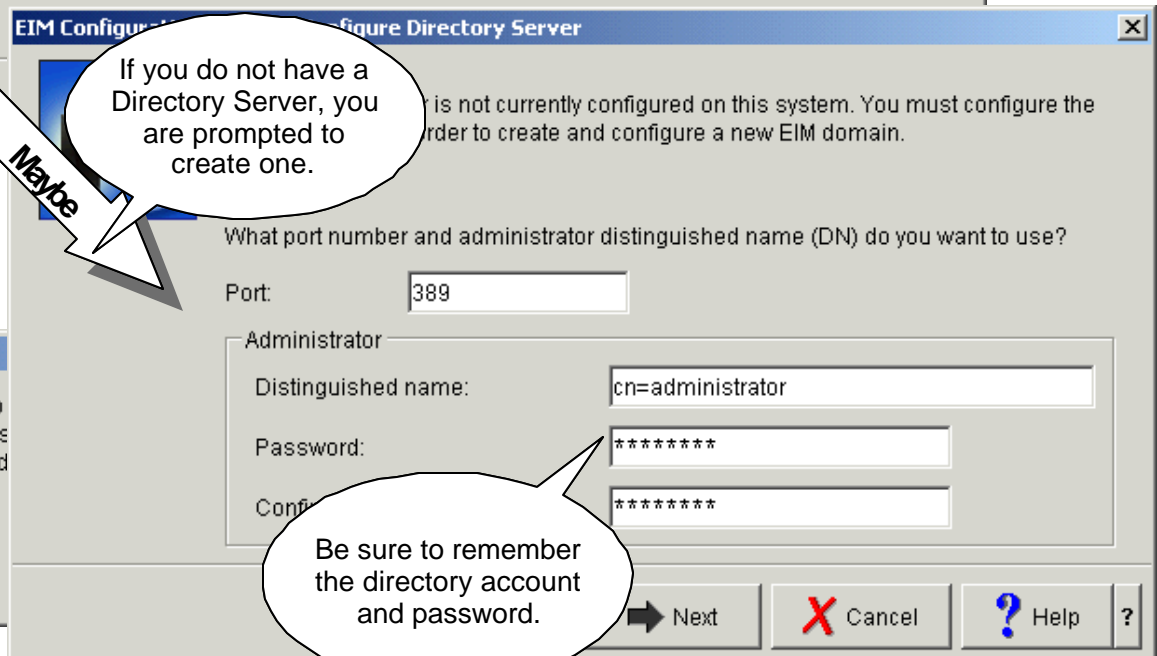
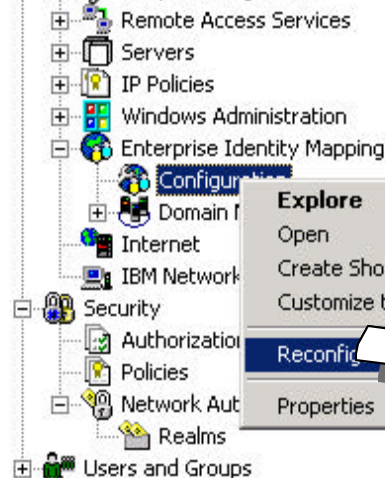
Step 3: OS/400: Configure EIM



Create a new domain using the EIM wizard. If no Directory exists, one can be created.

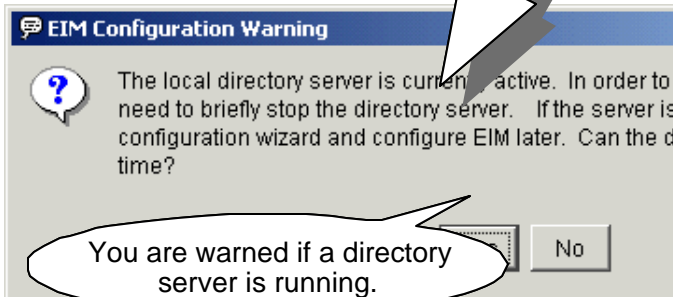


This is an EIM domain, and should not be confused with the Windows domain.



If you do not have a Directory Server, you are prompted to create one.

Be sure to remember the directory account and password.



You are warned if a directory server is running.

Notes OS/400: Configure EIM



When creating a new EIM Domain, the wizard looks for an existing Directory server configuration on the system. If one is not configured, the wizard prompts you with the option to create a basic configured Directory Service. If one is found and active, you are warned that it will be temporarily stopped. You will need an LDAP Directory user (distinguished name) and password with authority to create the objects for EIM.

If you setup a new Directory server, be sure to remember the administrator password.

Step 3: OS/400: Configure EIM (Cont'd)



Name the EIM domain and specify the DN to be used.

EIM Configuration Wizard - Specify Domain

The EIM domain consists of a domain controller and participating user registries in the network. Your local Directory server will be the domain controller for the new EIM domain.

What is the name of the domain you want to create?

Domain:

Description:

EIM Configuration Wizard - Specify Parent DN for Domain

The parent distinguished name (DN) for the EIM domain further defines the location of the EIM data in the directory.

Would you like to specify a parent DN for the EIM domain?

Yes

Parent DN:

No

Back Next Cancel Help ?

In this case, we choose to put the EIM domain under the root in the directory.

Notes OS/400: Configure EIM



If this is an existing Directory, you have the option to insert the EIM Domain under a parent DN in the Directory Information Tree (DIT).

At this stage, you must enter a Directory user with authorization to create the 'branch' for the EIM Domain. Commonly this would be the Directory Administrator. If a Directory was created during the previous step, this would be the same user and password.

If you are by chance using Kerberos for authentication against an existing directory, enter those credentials instead.

EIM Configuration Wizard - Specify User For Connection

In order for the wizard to complete EIM configuration, the wizard must connect to the domain controller with an authorized user.

What user do you want the EIM Configuration Wizard to use?

User type:

User

Distinguished name:

Password:

Confirm password:

If an LDAP server is running, you are prompted for a user to create the EIM object (typically an administrator).

Step 3: OS/400: Configure EIM (Cont'd)



Add the iSeries and Kerberos registries to the EIM database

EIM Configuration Wizard - Registry Information

User registries are a collection of user definitions for an application. Only those user registries that participate in EIM.

Which user registries do you want to add to the EIM database?

Local OS/400

RALYAS4A.ISERIES.ITSO.RAL.IBM.COM

Kerberos

ISERIES.ITSO.RAL.IBM.COM

Kerberos user identities are case sensitive

Cancel Help

be sure to select the **..case sensitive** box.

EIM Configuration Wizard - Specify EIM System User

Various operating system functions use EIM. The operating system connects to the domain controller as this user when performing these various functions. What user do you want the operating system to use for performing EIM functions?

Note: This user also has authority to EIM identifiers and to the local EIM registry.

User type: Distinguished name and password

User

Distinguished name: cn=adminstrator

Password: *****

Confirm password: *****

Verify Connection

Back Next Cancel Help

Enter user for EIM operations.

Notes OS/400: Configure EIM



You can (recommended) add this iSeries and the Kerberos registry to the EIM Domain at configuration time. This can also be performed after the EIM Domain is configured. If the Kerberos registry is a Windows 2000 Server, be sure to select the **case sensitive** box or unexpected errors can occur.

In this step, you also enter the user that will administer the Directory on behalf of the system. This is not necessarily the administrator user, but it must have proper authorization to perform tasks on the EIM branch of the DIT. This could be, for example, if you were placing the EIM branch into an already managed Directory Server with multiple DNs.

Step 3: OS/400: Configure EIM (Finished)



Finish the EIM wizard.

The EIM configuration process take from a few seconds up to a couple of minutes. Once complete, you return to the iSeries Navigator.

EIM Configuration Wizard - Summary

You have completed the steps necessary to create and configure a... Your Directory server has also been configured to be the domain controller for the EIM domain.

Click Finish to configure EIM and to join the EIM domain.

Setting	Value
Port:	389
Administrator DN:	cn=admin
Domain:	ITSO_EIM
Domain description:	Created by some ITSO guys
Wizard user for configuration:	cn=admin
Local OS/400 registry:	RALYAS4A.ISERIES.ITSO.RAL.IBM.COM
Kerberos registry:	ISERIES.ITSO.RAL.IBM.COM
OS/400 EIM system user:	cn=admin

Buttons:

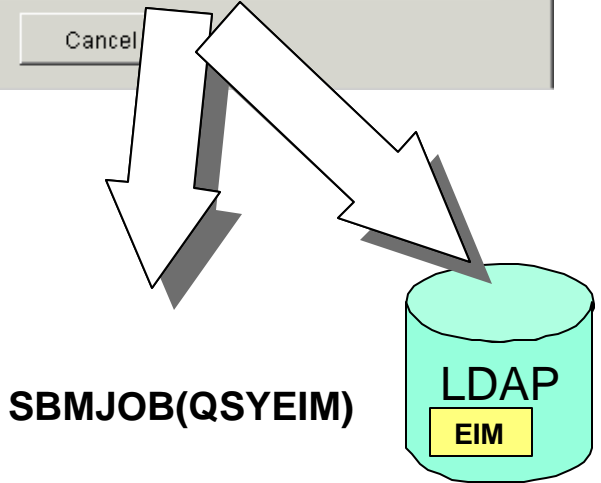
Finish:
Verify that all information is correct.

EIM Configuration Wizard

EIM configuration in progress...

- ✓ Configuring domain controller...
- ✓ Starting domain controller...
- Configuring EIM registries...
- Configuring RALYAS4A EIM system user...
- Updating system RALYAS4A EIM configuration...

Buttons:

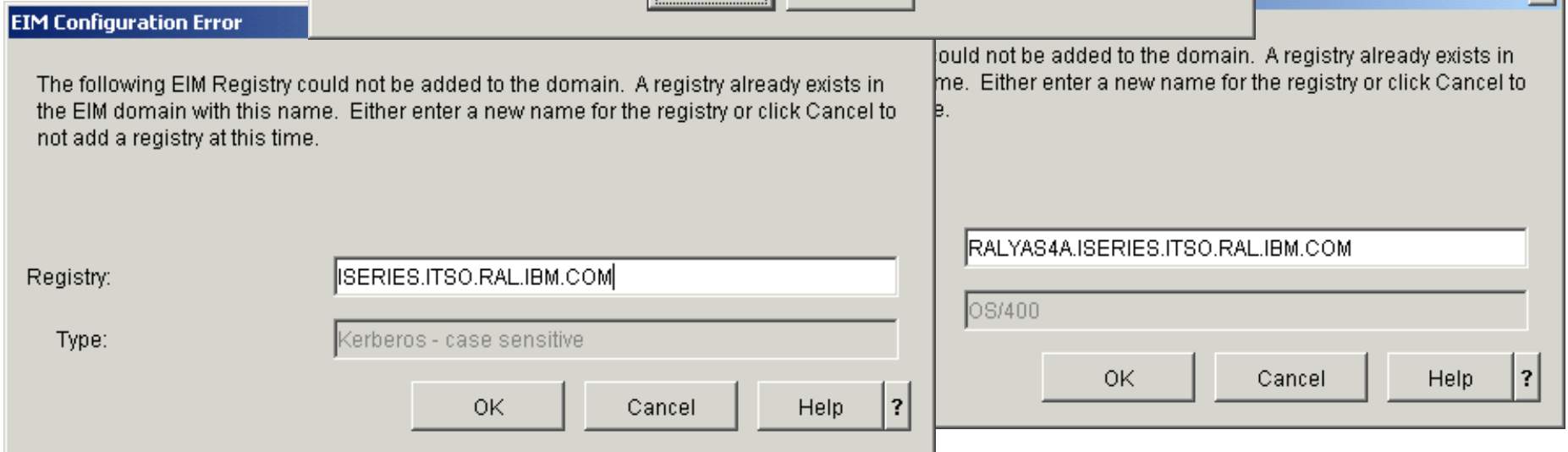
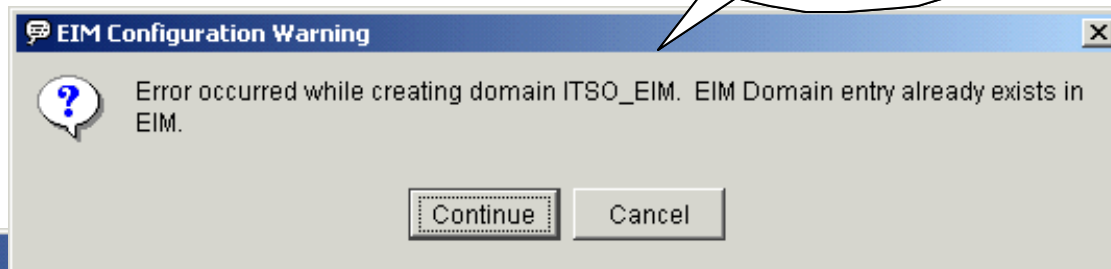


Notes OS/400: Configure EIM



The wizard creates the EIM entries needed for storing EIM objects and adds an auto-start entry for QSYEIM, a job that handles EIM requests on behalf of the system.

If the Domain already exists in the Directory, you receive these error messages.



Step 4: OS/400: Add the Domain



Add the EIM domain for management from your client.

Domain information

Domain: ITSO_EIM Browse

Parent DN: None

Domain controller: RALYAS4A.ISERIES.ITSO.RAL.IBM.COM

Connection

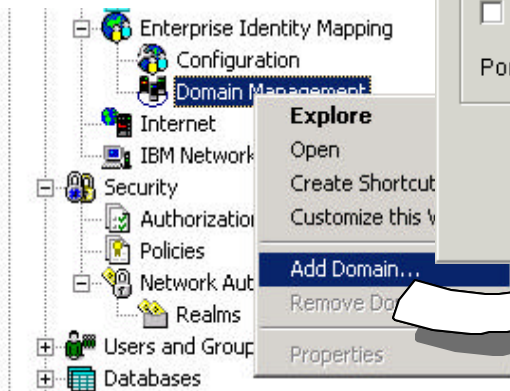
Use secure connection (SSL or TLS)

Port: 389

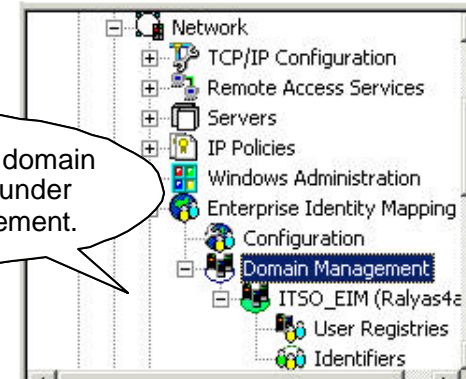
Verify Connection

OK Cancel Help ?

Optionally, you can browse a directory for domains.



Once added, the domain should appear under Domain Management.



Notes OS/400: Add the Domain



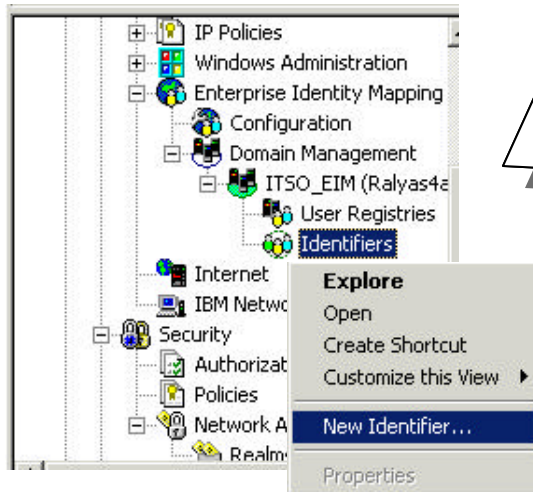
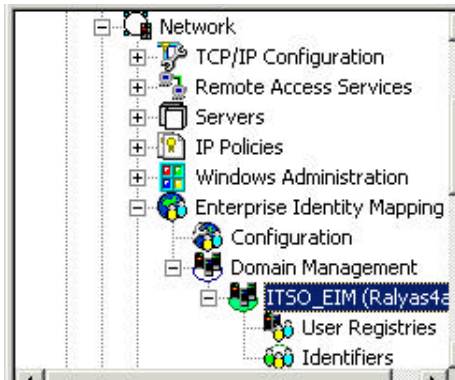
The purpose of adding a domain is for your client to manage it. The EIM Domain Management in the iSeries Navigator uses the EIM APIs to access the EIM configuration in the Directory. You can have several domains configured, but not necessarily all in the same Directory Server.

The Domain Management environment is stored locally on your iSeries Navigator client. Managed domains can be added and removed as one sees fit. The authentication against the EIM domain is first performed when you attempt to browse a domain.

Step 4: OS/400: Create an Identifier



Sign on to your domain and create a unique identifier for an enterprise user.



Domain controller: RALYAS4A.ISERIES.ITSO.RAL.IBM.COM

User type: Distinguished name

Distinguished name: cn=administrator

Password

Specify password: *****

Use system password

OK Cancel Help ?

Identified person: John Smith

Create unique identifier: John Smith, Sales rep. for Dumm...

Aliases

Alias: [] Add

Alias

OK Cancel Help ?

This is the password used previously for the directory administrator account.

Create an identifier using the full name of the intended person.

An alias could be added for example, if you have several "John Smiths" in the enterprise.

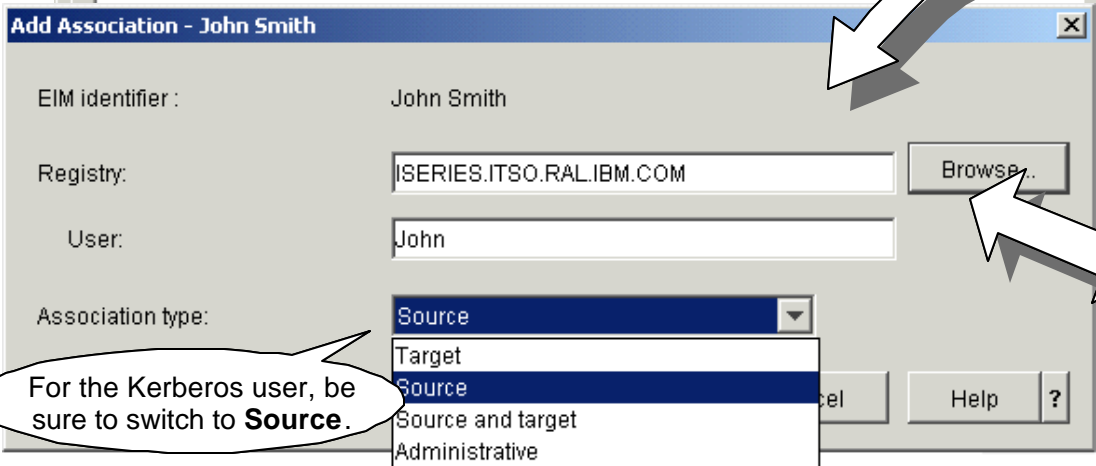
Step 4: OS/400: Identifier Association



Select Properties, and add a Kerberos user association to the identifier.



...and Add.



Step 4: OS/400: Identifier Association (Cont'd)



Add an association for the iSeries user profile.

Add Association - John Smith

EIM identifier : John Smith

Registry: RALYAS4A.ISERIES.ITSO.RAL.IBM.COM

User: JOHNS

Association type: Target

?

John Smith Properties - Ralyas4a.iseries.itso.ral.ibm.com

General Associations

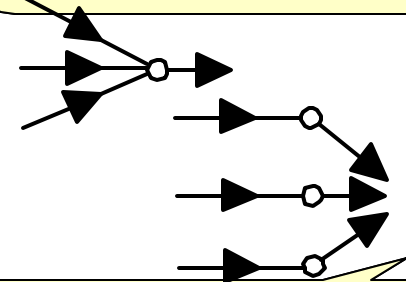
Associations for EIM identifier: John Smith

User	Assoc...	Registry	Registry Type
John	Source	ISERIES.ITSO.RAL.IBM.COM	Kerberos - case sensitive
JOHNS	Target	RALYAS4A.ISERIES.ITSO.RAL.IBM.COM	OS/400

?

Each iSeries user needs a home directory, for example, /home/JOHNS

Note: Using multiple sources from the same registry allows one person to have several user IDs with different authority levels.



Note: Having multiple identifiers point to the same user ID would enable the functionality of a group user.

Notes OS/400: Identifier Association



In these three charts, you create one identifier with two associations.

The first, a **source** association, is for finding identifiers with EIM lookup operations. A source is not returned as a user ID to a system. A user could have multiple sources from the same registry. This allows a Windows user that has multiple Windows user IDs (admin user and normal user) to use different user levels on the Windows Domain, but still map to the same OS/400 user ID.

A **Target** association is the user ID that is to be used on an intended system by a user (identifier). Having multiple targets to the same system would most likely result in an error. In the iSeries case, an ambiguous error would occur, and the user would not be signed on at all.

Source and Target is useful when a user both signs directly on to a system and accesses it remotely from another system.

An **administrative** (not to be confused with "administrator") association is only used to indicate that the person (identifier) has a user on that system. But unlike a target, it will not be returned on lookup operations. This is useful if the target system is considered so sensitive that the systems registry will not be included in the EIM Domain, but you want to keep track of all the person's users.

On an iSeries server, all users have to have a directory under the IFS */home* directory. Use **WRKLNK '/home'** to verify if one exists and create additional directories if needed. The user's home directory is for storing the user's credential cache (or the link to the cache to be more precise). This makes the authentication process quicker, because the iSeries does not need to wait for the KDC or the user to supply these credentials.

Note: If one identifier has multiple "targets" or one "source" points to multiple identifiers, it is up to the requesting registry to handle the returned multiple entries.



Step 5: Use EIM with iSeries Navigator



In your iSeries Navigator connection properties, change the sign-on information to Use Kerberos principal...

...select **Connection...**

...and select the **Kerberos** radio button.

Right-click... **Properties...**

Restart iSeries Navigator and click your connection. If all works, you should not be prompted for a user/password and should be signed on as the EIM-mapped user... **Johns**.

Name	Signed On User	Description
Ralyas4a.iseries.itso.ral.ibm.com	Johns	Manage this server.

Notes Use EIM with iSeries Navigator



You should now be able to use the Kerberos authentication method with iSeries Navigator. The OS/400 host servers, in turn, uses the EIM functionality to map the incoming (source) identifier to a target OS/400 user profile.

On the properties field for the iSeries connection, select **Use Kerberos System name, no prompting**. Restart the Navigator. The sign-on process should be quick and seamless.

Note: The Kerberos environment allows for caching of tickets and session keys. If the client still has a valid ticket/session key in its cache, it attempts to reconnect without requesting a new service ticket from the TGS. To renew the actual Kerberos information for the Windows User, you have to log off the computer.

Notes What if It Didn't Work?



CWBSY1012 - Kerberos principal not found on server...

Either the KDC did not have the principal entry for the iSeries, or we are asking incorrectly. In this case, the client was resolving the hostname for the iSeries incorrectly.

CWBSY1017 - rc=608 Kerberos credentials not valid on server...

The iSeries apparently did not think the ticket received was intended for its service. In this case, the iSeries was not resolving its own host name correctly. The fully qualified host name had to be first in the host table entry.

(CFGTCP opt.10)

```
_ 9.25.105.24 RALYAS4A.ISERIES.ITSO.RAL.IBM.COM
RALYAS4A
```

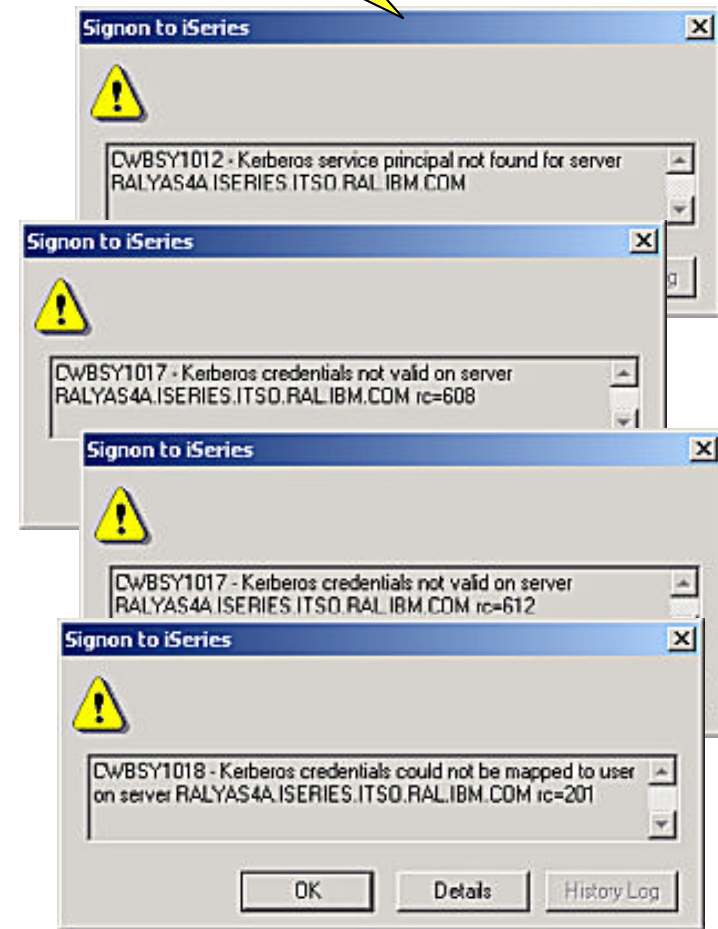
CWBSY1017 - rc=612 Kerberos credentials not valid on server...

Again the iSeries did not like the ticket received. In this case, the problem was that the password for the kerberos principal on OS/400 and the one in Windows do not match.

CWBSY1018 - Kerberos credentials could not be mapped to user...

The Kerberos authentication was successful, but the Windows user was mapped incorrectly in EIM.

Note: These are some of the problems that can be encountered, but definitely not all of them.



Step 6: Add Another OS/400 Registry



Sign on to System B:
Start the EIM configuration wizard and select Join an existing domain.

But...first you must complete steps 1 (User/principal) and 2 (NAS setup) for System B.

The screenshot displays two overlapping windows from the EIM Configuration Wizard. The top window, titled "EIM Configuration Wizard - Specify Domain Controller", contains the following text and fields:

- Icon: A globe with server icons.
- Text: "The EIM domain controller controls access to all of the EIM data in the domain."
- Text: "What is the name of the domain controller for the EIM domain you want your system to join?"
- Field: "Domain controller name:" with the value "RALYAS4A.ISERIES.ITSO.RAL.IBM.COM".
- Section: "Connection"
- Field: "Use secure connection (SSL or TLS)" with an unchecked checkbox.
- Field: "Port:" with the value "389".
- Button: "Verify Connection".

The bottom window, titled "EIM Configuration Wizard - Specify User For Connection", contains the following text and fields:

- Icon: A globe with server icons.
- Text: "In order for the wizard to complete EIM configuration, the wizard must connect to the domain controller with an authorized user."
- Text: "What user do you want the EIM Configuration Wizard to use?"
- Field: "User type:" with a dropdown menu showing "Distinguished name and password".
- Section: "User"
- Field: "Distinguished name:" with the value "cn=administrator".
- Field: "Password:" with masked characters "*****".
- Field: "Confirm password:" with masked characters "*****".
- Button: "Verify Connection".

At the bottom of the wizard interface, there are navigation buttons: "Back", "Next", "Cancel", and "Help".

Notes Add Another OS/400 Registry



We already completed the creation of a principal entry on our Windows KDC, configured NAS and EIM, and added the domain to EIM management. So adding a second iSeries to our EIM Domain should prove a simple task.

But before we add the second iSeries registry (RALYAS4B) to the EIM domain, you must first repeat steps 1 and 2. That is create a RALYAS4B Windows service principal entry representing system RALYAS4B and set up Network Authentication Services for this iSeries.

You should then sign-on to RALYAS4B with iSeries Navigator. Click **Network-> Enterprise Identity Mapping**. Then right-click **Configuration** and select **Configure**. On this panel, select **Join an existing Domain**. This prompts you for the name of the EIM Controller, which is actually the Directory Server on RALYAS4A.

Enter a directory user that has sufficient authority to search the directory and create objects.

Step 6: Add Another OS/400 Registry (Cont'd)



Mark the existing EIM domain and add the local registry.

EIM Configuration Wizard - Specify Domain

The EIM domain consists of the domain controller and user registries. Once this system has joined the domain, you can create mappings from users on this system to users in the domain.

What domain do you want this system to join?

Domains:

Domain	Parent
ITSO_EIM	

← Back Next →

EIM Configuration Wizard - Registry Information

User registries are a collection of user definitions for a particular operating system or application. Only those user registries that have been added to the EIM domain can participate in EIM.

Which user registries do you want to add to your domain?

Local OS/400

Kerberos

RALYAS4B.ISERIES.ITSO.RAL.IBM.COM

ISERIES.ITSO.RAL.IBM.COM

EIM Configuration Wizard - Specify EIM System User

Various operating system functions use EIM. The operating system connects to the domain controller as this user when performing these various functions. What user do you want the operating system to use for performing EIM functions?

Note: This user also has authority to EIM identifiers and to the local EIM registry.

User type: Distinguished name and password

User

Distinguished name: cn=administrator

Password: *****

Confirm password: *****

Verify Connection

Use the directory administrator/password for 'RALYAS4A'.

Notes Add Another OS/400 Registry



The wizard should locate existing EIM Domains within the directory. Mark the preferred EIM domain and click **Next**. You now have the option to add this system's user registry and the Kerberos registry.

Finally add the Directory user that will be used for performing EIM operations from this system. Again, this does not have to be the directory administrator but could, for example, be a user dedicated for this system. This would be useful if you want to give each server its own directory user, preventing them from accessing EIM information of other systems.

Step 6: Add Another OS/400 Registry (Finished)



Finish the EIM configuration.

EIM Configuration Wizard - Summary

You have completed all the steps necessary to configure your system to participate in an EIM domain on your network.

Click Finish to join the EIM domain.

Setting	Value
Domain controller name:	rallyas4a.iseries.ibm.com
Use secure connection (SSL or TLS):	No
Port:	389
Wizard user for configuration:	cn=admin
Domain:	EIM
Domain parent DN:	
Local OS/400 registry:	RALYAS4B.ISERIES.ITSO.RAL.IBM.COM
OS/400 EIM system user:	cn=admin

Back Finish Cancel Help ?

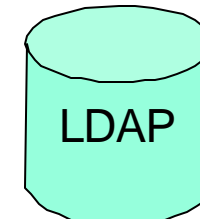
EIM Configuration Wizard - Configuration In Progress

EIM configuration in progress...

- ✓ Configuring EIM registries...
- Configuring RALYAS4B EIM system user...
- Updating system RALYAS4B EIM configuration...

Cancel

SBMJOB
(QSYEIM)



(on Remote system
RALYAS4A)

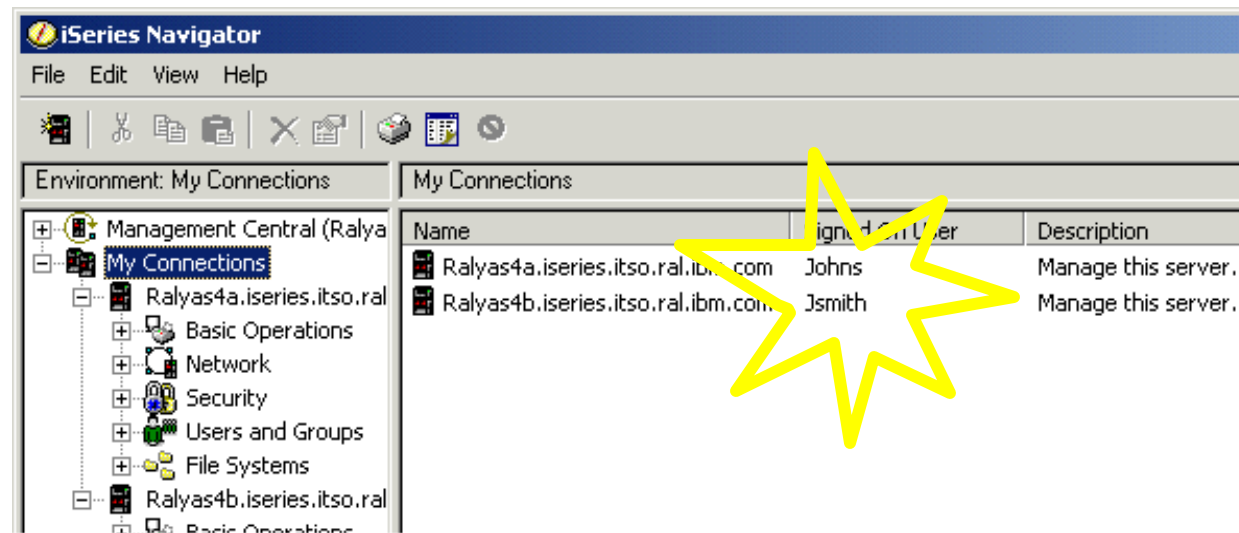
Notes Add Another OS/400 Registry



Finalizing the setup activates the EIM functionality on RALYAS4B and updates the EIM database on RALYAS4A.

You should now be able to use the Kerberos authentication method with the connection against RALYAS4B.

Repeat step 4 (adding an identifier for RALYAS4B) and step 5 (enabling Kerberos authentication for iSeries Navigator) to complete the setup for system RALYAS4B. After you restart iSeries Navigator, you should see that the signed on user profiles in iSeries Navigator represent the EIM-mapped user identifiers.



Notes Problem Determination



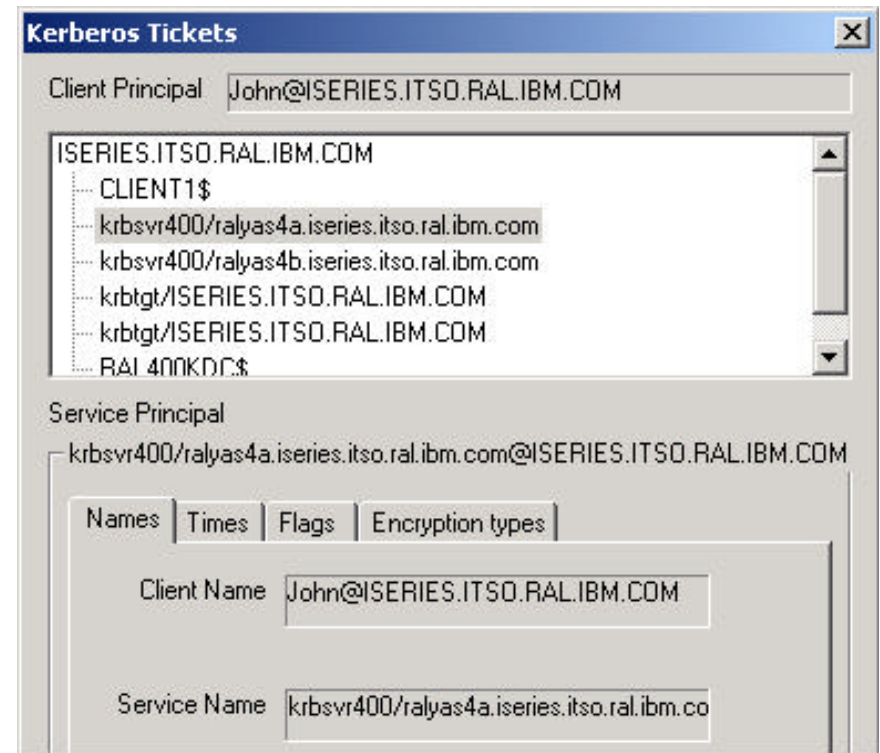
We found more information in the QZSOSIGN job log (the sign-on server) and QDIRSRV job log (directory server).

If you add (export) the following variables to the .profile in the home directory for the user, you can get Kerberos trace information output to a file:

```
_EUV_SVC_MSG_LOGGING=STDOUT_LOGGING
_EUV_SVC_MSG_LEVEL=VERBOSE
_EUV_SVC_STDOUT_FILENAME=/home/<USERDIR>/trace.txt
_EUV_SVC_DBG_MSG_LOGGING=1
_EUV_SVC_DBG_TRACE=1
_EUV_SVC_DBG=*9
```

KerbTray is a useful utility to view the tickets residing on a Windows 2000 client. You can download it from Microsoft's Web site by searching for "kerbtray". The site is located at:

<http://www.microsoft.com>



Summary



When setting up EIM for single sign-on, we:

- Added principal to KDC
- Configured NAS on RALYAS4A
- Configured EIM services on RALYAS4A
- Created an EIM identifier and mapped user
- Changed authentication method for the iSeries Navigator
- Configured RALYAS4B to use EIM

Some possible next steps:

- Add more identifiers
- Enable iSeries Access to use "Kerberized" Telnet
- Enable NetServer usage using Kerberos
- SSL-enable the directory

Notes Summary



The steps that have now been performed should have given you the beginning of an EIM-enabled environment. As applications are added that use other registries, or as operating systems are added that exploit EIM, they can be added to the EIM domain. It should not impact your existing authentication method (which is assumed to be user ID/password). It is possible to introduce users at a controlled pace. Once a user is mapped in EIM and is using Kerberos authentication, the iSeries user profile value **PASSWORD** can be set to ***NONE**, preventing the user from sign-on using other methods.

Repeat step 4 to add additional users to the EIM Domain. If you are using iSeries Access for terminal emulation, you can use the same authentication method as iSeries Navigator. (*IBM Personal Communications* currently does not support Kerberos authentication.) Make sure the system value **QRMTSIGN** is set to something other than ***FRCSIGNON** when you want to bypass sign on with EIM and Kerberos.

By default, LDAP sends the user name and password in clear text when connecting to the directory. Therefore, we strongly recommend that systems in an EIM domain use SSL and optionally Kerberos to authenticate themselves to the EIM Controller.

The screenshot shows the IBM iSeries Administration Center interface. On the left, a tree view displays the environment structure: Environment: My Connections > Enterprise Identity Mapping > Configuration > Domain Management > ITSO_EIM (Ralyas4a) > User Registries > Identifiers. The 'Identifiers' folder is selected. On the right, a table displays the list of identifiers for the selected folder.

Identifier	Description
Brian Krings	That developer guy who knows everything about Kerberos
Erik Larsson	Erik, That dude from Sweden
John Smith	John Smith, Sales rep. for DummyCorp.
Pat Botz	Lead eServer Architect for EIM
Scott McCreddie	Cool EIM developer
Thomas Barlen	ITSO iSeries and Security pro.

Related Publications



- *The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this workshop.*

International Technical Support Organization Publications

- For information on ordering ITSO publications, visit us at <http://www.redbooks.ibm.com> (Internet Web site) or
- <http://w3.itso.ibm.com> (intranet Web site)

For Technical Support see <http://www.ibm.com/support> and <http://w3.ibm.com/support>

Redbooks on CD-ROMs

- Redbooks are available on CD-ROMs.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbook	SK2T-8038
AS/400 Redbooks Collection	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SK2T-8041
Application Development Redbooks Collection	SK2T-8037
Personal Systems Redbooks Collection	SK2T-8042

Related Publications - Continued



Other Publications

- *These publications are also relevant as further information sources:*

Title	Publication Number
The Kerberos Network Authentication Service (V5), RFC1510	http://www.ietf.org/rfc/rfc1510.txt
Microsoft's Active Directory home page	http://www.microsoft.com/activedirectory
V5R2 iSeries Information Center, Security topics	http://www.iseries.ibm.com/infocenter
Kerberos, A Network Authentication System	ISBN 0-201-37924-4
<i>Implementation and Practical Use of LDAP on the IBM iSeries Server</i>	SG24-6193