ibm.com

e-business

Security in an iSeries e-business Environment: What You Need to Know

EP05

ITSO iSeries Technical Forum

Thomas Barlen

Redbooks

International Technical Support Organization

Acknowledgments

Thanks to David Granum (IBM Rochester, MN) for his work on this presentation during an ITSO residency, led by Thomas Barlen, at the Raleigh, North Carolina center.

Objectives

Part 1. Introduction

- Discusses the role of security in an e-business environment
- Discusses the goals and threats that must be addressed
- Introduces the implementation of security in layers

Part 2. What the iSeries has to offer

- Discusses what the iSeries can provide at various layers
  - Network
  - System
  - Application

Part 3. Solution scenarios

- Explains based on solution scenarios
  - What security functions and services to use in various layers
  - How various services can be combined to achieve maximum security

Introduction

Security in an e-business Environment

Why is security needed in your e-business environment?

- Prevent unauthorized access to your data
  - Data may reside on a system or may be in transit from one system to another
- Prevent unauthorized access to your network systems
  - Servers, personal computers, firewall, routers
- Establish trust

CNN.com

Hacker exposes financial data at Georgia Tech

March 20, 2002 Posted: 8:40 a.m. EST (1340 GMT)

From... COMPUTERWORLD AN IDG.net SITE

By Brian Sullivan

(IDG) -- State and federal authorities are investigating a hack into a computer server at the Atlanta-based Georgia Institute of Technology (Georgia Tech) last week.

An undetermined number of employee financial records and university credit card numbers could have been exposed

COMPUTERWORLD

Search GO Advanced Search |

News & Features | Knowledge Centers | Careers | Communities | Subscriptions | Media Center

Headlines | Shark Tank | Emerging Technologies | QuickStudy | Columnists | This Week in Print | ROI Magazine |

NEWS
Latest Headlines
This Week in Print
Emerging Companies
QuickStudies

CAREERS
Latest Stories
Career Adviser
Surveys & Reports
Jobs

IT RESOURCES

Online billing vendor hit by network attack

BY TODD R. WEISS
(December 21, 2001)
CCBill LLC, an online transaction processing company, was hit earlier this week by a network attack that apparently allowed access to user names and passwords for customers' Web hosting servers.

Notes Security in an e-business Environment

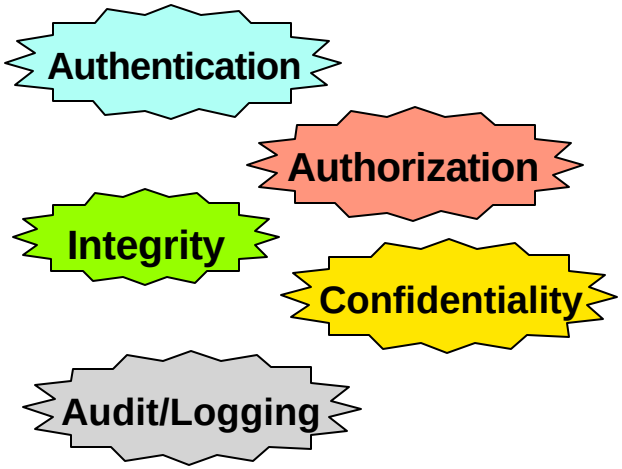
If your private network is connected to the Internet, security should be at the top of your priority list for your e-business environment. Security should be implemented to protect two entities:

- The data that is transmitted on the network
- The computers that are connected to that network.

However, security is still needed even if your private network does not connect to the Internet.

Goals and Threats of Security

Primary goals of security:



Primary threats to security

- Eavesdropping or sniffing (versus confidentiality)
- Impersonation (versus authentication)
- Decryption (versus confidentiality)
- Denial of Service (DoS)/flooding (versus availability)
- Technology and Application weaknesses (versus all)

Notes Goals and Threats of Security

GOALS

- **Authentication:** Determine that the users are who they claim to be. The most common technique to authenticate is by user ID and password.
- **Authorization:** Permit a user to access resources and perform actions on them. An example of authorization is the permissions on OS/400 objects.
- **Confidentiality:** Only authorized users can view the data. For data that is transmitted through a network, there are two ways to achieve this goal:
  - Make sure that only authorized persons can access the network
  - Encrypt the data
- **Integrity:** Only authorized users can modify the data, and they can only modify it in approved ways. The data is not changed either by accident or maliciously. For data that is transmitted over a network, there are two ways to achieve this goal: make sure that only authorized persons can access the network (not easy to achieve in public networks such as the Internet) or digitally sign the data.

THREATS

- **Sniffing:** Computers with access to the public network can record the traffic flowing through it. If data or commands are sent unencrypted, it is easy for unauthorized people to passively eavesdrop. Sniffing is a threat to confidentiality, but if user IDs and passwords are sniffed, the threat becomes more serious because the attacker could then impersonate a legitimate user.

- **Impersonation:** The attacker tricks your security system passing as an authorized user. For example, the attacker steals valid user IDs and passwords by recording network traffic while users sign on. If the communication is over a public network, and it is not digitally signed or signed with a weak technology, an attacker can modify or enter completely new data and commands. Impersonation can be a threat to all three major goals of computer security.

- **Decryption:** If data is sent over a public network, attackers can often easily obtain the encrypted data. If the encryption is weak, the attackers can decrypt the data in a fairly short time. Decryption is a threat to confidentiality.

Notes Goals and Threats of Security (Cont'd)

- **Flooding:** If an attacker sends large amounts of data, such as connection requests to a public Web server, it could fill the network bandwidth. The network resource becomes overused preventing access to other users or greatly affecting performance. Flooding is a threat to availability.

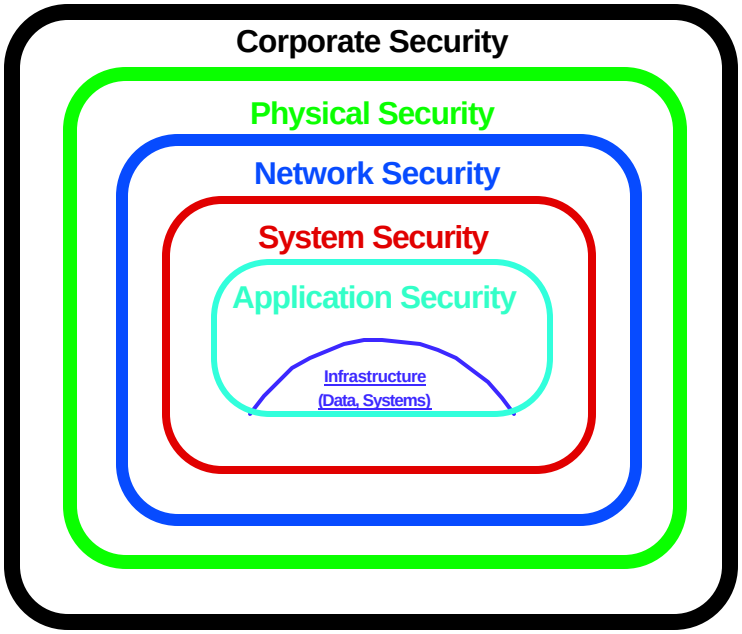
- **Technology or application weakness:** The TCP/IP protocol, some of its applications, and some operating systems have inherent security shortcomings, sometimes due to the objectives of their original design (openness, easy communication between computers and applications). For example, the UNIX sendmail application used to run e-mail is famous for a long history of security problems. Simple Network Management Protocol (SNMP), Simple Mail Transfer Protocol (SMTP), and Syn Floods all present security holes related to the insecure structure on which TCP was designed. Known security problems for UNIX, Windows, and OS/2 are documented in the Computer Emergency Response Team (CERT) Web site at http://www.cert.org/

Likewise, company-developed applications or software purchased from vendors may have security weaknesses that attackers can exploit. The degree of the damage depends on the nature of the problem. The most common damage is to shut down a system. It could be more serious allowing attackers access to data that they can alter or use to their advantage. Technology and application weaknesses exploited by malicious attackers are threats against all goals of security. To protect yourself, you must keep up to date with the vendors security updates and rely on providers with a good reputation for paying attention to security. If you develop your own applications to run on hosts that will be accessed from the network, security must always be at the top of the design goals.

Layers of Security

There are several layers within an e-business environment at which security can be implemented

- Corporate layer
  - User education, corporate security policies, etc.
- Physical layer
  - Computer room access, building and/or site access
- Network layer
  - Firewall, Security appliances, VPN gateways, etc.
- System layer
  - LAN interfaces, filtering, system values, user profiles, object access, auditing, etc.
- Application layer
  - Secure Sockets Layer (SSL), exit programs, etc.



"Security is not a product; it is a process."
Bruce Schneier

Notes Introduction

Simply implementing a firewall is not enough to prevent unwanted access to confidential data on your systems. Implementing Security in your e-business environment must begin with your corporate security plan. After you determine what that security plan entails, it should be tailored to secure your environment at all layers identified.

Security at the Corporate Layer

Security policies

- A corporate security policy is necessary to establish and implement a security plan for the entire business
- A firewall should not be your only means of security
- Continually monitor to detect any deviation from your policies and take action if needed
- Periodically review your processes and policies to update them and improve them
- You must *plan your work*, then *work your plan*

User education

- Users must know that data confidentiality and integrity are at risk when performing actions outside of the bounds specified in the corporate security policy

Security is only as strong as the weakest link in the chain

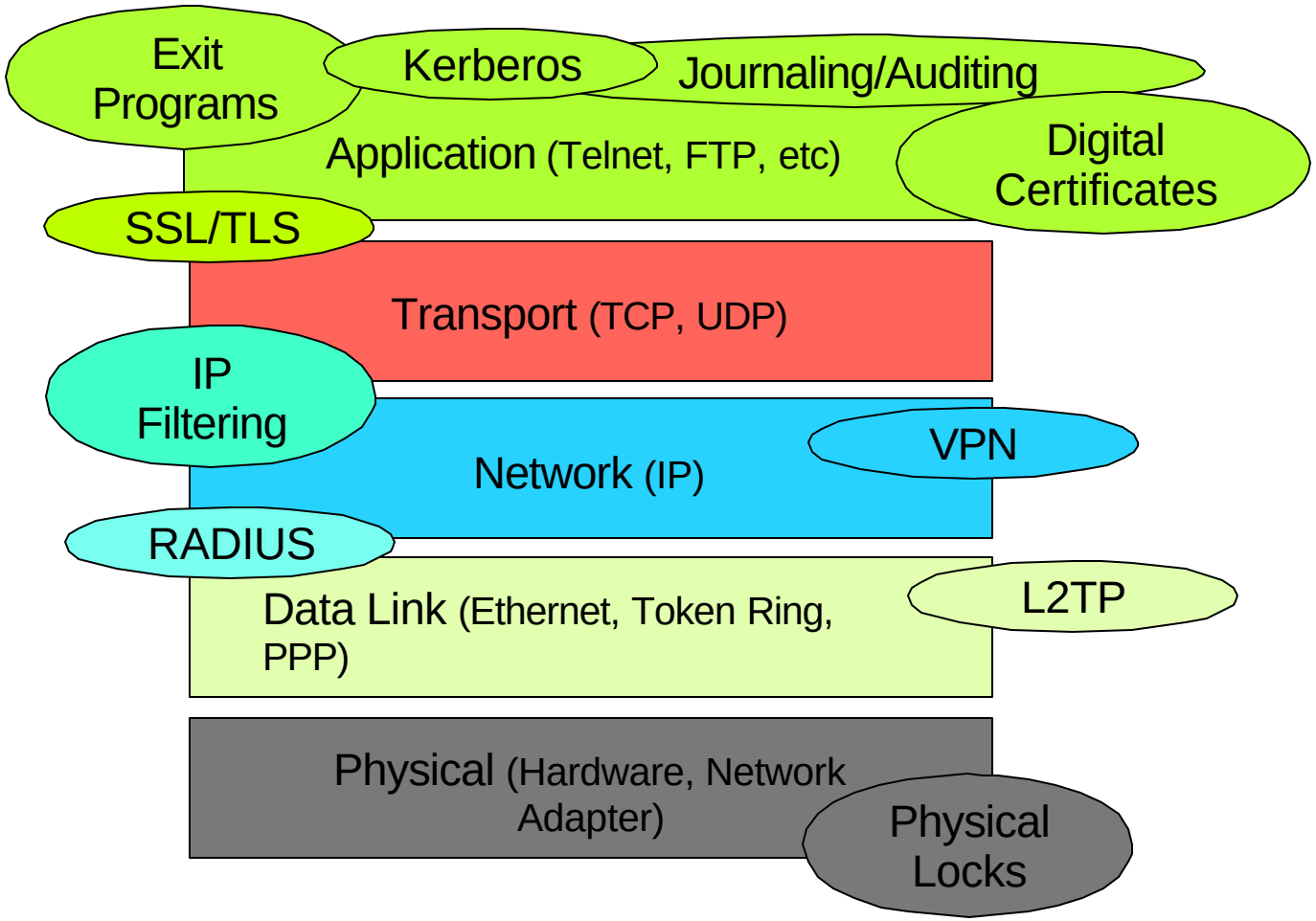
Notes Security at the Corporate Layer

When implementing a security plan, you must first determine what it is that needs to secured. Based on what was discussed previously, we know that your computer systems, the data on them, and the data being transmitted are all open to possible security breaches. This security plan must not only include implementing a firewall. Some data within your private network (salary data, other personnel data, etc.) is data that you do not want many individuals to have access to, whether these individuals are located within the corporate network or on the Internet. While you cannot lock down all of your data all of the time, you can limit access to authorized users. You can also keep data confidential by encrypting it while the data is residing on a system or in transit on the network. The security policy that is established should continuously be reviewed and updated to improve upon the existing security. Once that security policy is established, the end user must be informed of their responsibility. They must know the consequences of leaving data and systems unsecured. Action should be taken if a user deviates from the corporate security policy.

What the iSeries Offers

Security Services Overview

The iSeries server offers security in various layers!



Notes Security Services Overview

The Open System Interconnection (OSI) model is way of implementing protocols using a layering approach and is the model used by the TCP/IP Protocol Suite. We describe and provide examples of each entity and method to secure those entities at each layer.

Application (Presentation and Session included) Layer

- This layer is responsible for providing information defining and contributing to applications. This includes the interface for the end user, commands available, etc.
  - Examples: Telnet, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), HyperText Transfer Protocol (HTTP), Lightweight Directory Access Protocol (LDAP), etc.
  - Security Services: Exit Programs, Digital Certificates, Journaling, Auditing, etc.

Transport Layer

- **Note:** Sockets and Secure Sockets reside between the Transport and Application Layers.
- This layer is responsible for ensuring end-to-end data communication between two hosts on a network. It is also responsible for flow control.
  - Examples: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Sequenced Packet eXchange (SPX), etc.
  - Security Services: IP Packet Filtering (for example, on ports)

Network Layer

- This layer is responsible for routing network traffic between two hosts on different networks. Addressing is another responsibility of this layer.
  - Examples: Internet Protocol (IP), Internet Packet eXchange (IPX), etc.
  - Security Services: IP Packet Filtering, Virtual Private Networking (VPN)

Data Link Layer

- This layer is responsible for hardware addressing, defining the protocol for the architecture of the network, hardware flow control, encoding and decoding network packets into bits
  - Examples: Token Ring, Ethernet, etc.
  - Security Services: Layer 2 Tunneling Protocol (L2TP)

Physical Layer

- This layer is responsible for providing hardware that support the above protocols, physically sending a receiving the data on a given media.
  - Examples: LAN Adapter, CAT5 cabling, etc.
  - Security Services: Physical locks, logging physical access, etc.

Security at the Physical Layer

Physical locks

- Require physical key access to systems that support it
- Require a physical key or code to access rooms with systems/data
- Require a physical badge or ID to access business site

Logging

- Log access in/out of network closets, machine rooms, etc.
  - Require users to sign in and out of these rooms

Backup

- Uninterruptible Power Supply
- Air conditioning
- Alternative communication path

Security at the Network Layer

	Confidentiality	Integrity	Authentication	Authorization	Auditing/ Logging
IP Filtering			X	X	X
VPN	X	X	X	X	X
L2TP			X	X	X**
SSL/TLS*	X	X	X	X	X***

* SSL actually occurs at the application layer. However, it protects network traffic by encrypting the data.
** L2TP only when RADIUS accounting is used.
*** Logging capabilities depend on the individual application

The security goals discussed in Part 1 can be obtained by using network security tools on the iSeries

- IP packet filtering
- Virtual Private Networking (VPN)
- Layer 2 Tunneling Protocol (L2TP)
- Secure Sockets Layer (SSL)*

Notes Security at the Network Layer

In the following subsections (security at each layer), we map what each layer offers in terms of reaching our goals established in Part 1 of this presentation.

The following charts show what the iSeries offers for security from a networking standpoint. We discuss how each of these methods/tools establishes some or all of the goals specified in Part 1:

- IP Packet Filtering
- VPN
- L2TP
- SSL

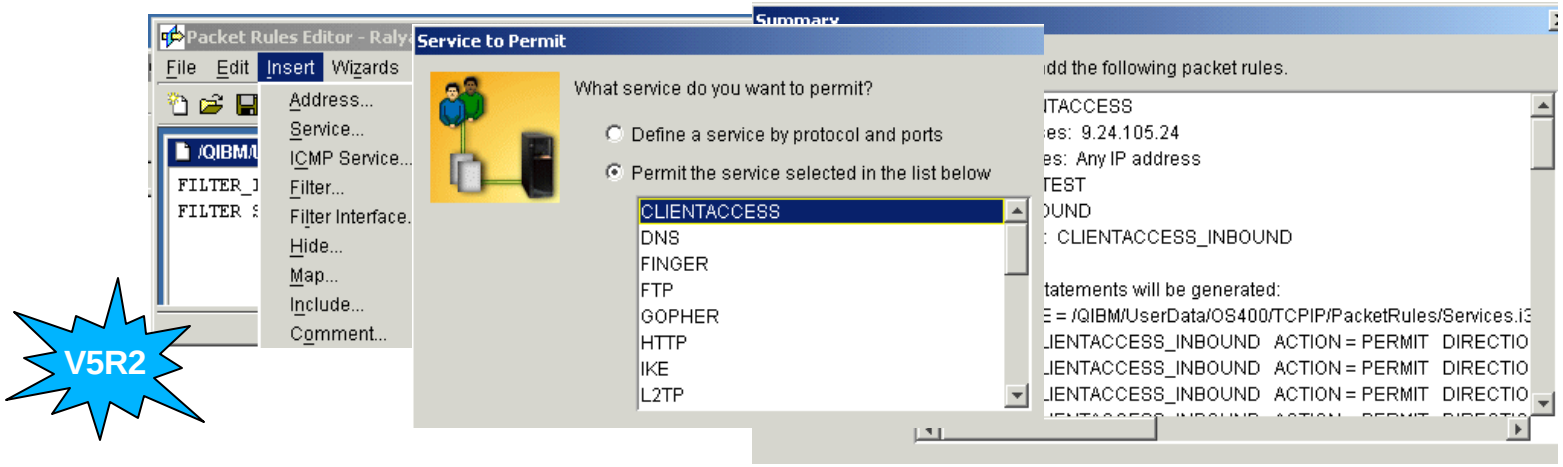
IP Packet Filtering

IP Packet Filtering

- Authentication
  - Users are authenticated in that packet rules are written to only allow access to the iSeries from specified IP addresses
    - For example, only allow IT employee, Bob, using IP address 9.1.90.28 to access the token ring port on the iSeries
    - When connecting via PPP or L2TP, you can limit access based on the authenticated user

- Authorization
  - Users are authorized only to necessary ports (applications) on the iSeries
    - For example, only allow end users to access the iSeries for Client Access (ports 8470 to 8479) and Telnet (port 23)
- Other means of security/logging
  - Journaling
    - QUSRSYS/QIPFILTER journal
    - QUSRSYS/QIPNAT journal
  - Auditing

IP Filtering Enhancements at V5R2



- Packet Rules Editor
  - New, easy to use Packet Rules Editor allows you to create and modify packet rules using wizards and property pages
- New auto-writing rules wizards
  - Permit A Service wizard
  - Address Translation wizard
  - Spoof Protection wizard
- New way to view packet rules
  - New view in iSeries Navigator allows you to easily and clearly view your filter rules file(s)
- Support for creating filter rules files
  - Support for creating packet rules files according to an XML data type definition found in the file /QIBM/XML/DTD/QtofPacketRules.dtd
- Currently, filtering does not work with IPv6

Notes IP Packet Filtering

IP packet filtering can and should be used even though you have a firewall preventing unwanted access to the iSeries. IP Packet Filtering is a second level of defense for unauthorized access into your corporate network. The IP packet filtering rules should be written so that only applications that you want users to access are opened up. A very simple example of IP packet filtering is shown below. The objective of this example is to show you the format of iSeries packet filter rules. The filter rules that you need to configure on the iSeries to allow only Telnet-SSL requests from any client (Internet or private network) to the Telnet-SSL port of the server:

```
#The following filter rules are defined on the interface:
#This filter rule file probably has no real practical use, it is used here to show the format of #iSeries packet filter rules.

#Permit inbound packets from all clients (IP address any (*)) to the SSL-Telnet server #(IP address 10.1.1.10, mask 255.255.255.255 and port 992).
FILTER SET = HOST ACTION = PERMIT DIRECTION = INBOUND
  SRCADR = * DSTADR = 10.1.1.10 PROTOCOL = TCP
  DSTPORT = 992 SRCPORT > 1023 FRAGMENTS = NONE JRN = OFF
#Permit outbound packets form the SSL-Telnet server to all clients.
FILTER SET = HOST ACTION = PERMIT DIRECTION = OUTBOUND
  SRCADR = 10.1.1.10 DSTADR = * PROTOCOL =TCP
  DSTPORT > 1023 SRCPORT = 992 FRAGMENTS = NONE JRN = OFF
#Define a filter interface associated with the AS/400 interface connected to the secure network. Add #the HOST set name to it.
FILTER_INTERFACE INTERFACE=TRNLINE SET = HOST
#All traffic that is not permitted is automatically denied.
```

The Rules Editor has been rewritten and provides a more convenient way for creating and maintaining your IP filtering environment. Several wizards are new in V5R2. They allow you to set up filter rules by answering a few questions. The Rule Editor window is now resizable and remembers the previous window size.

Notes IP Packet Filtering (Cont'd)

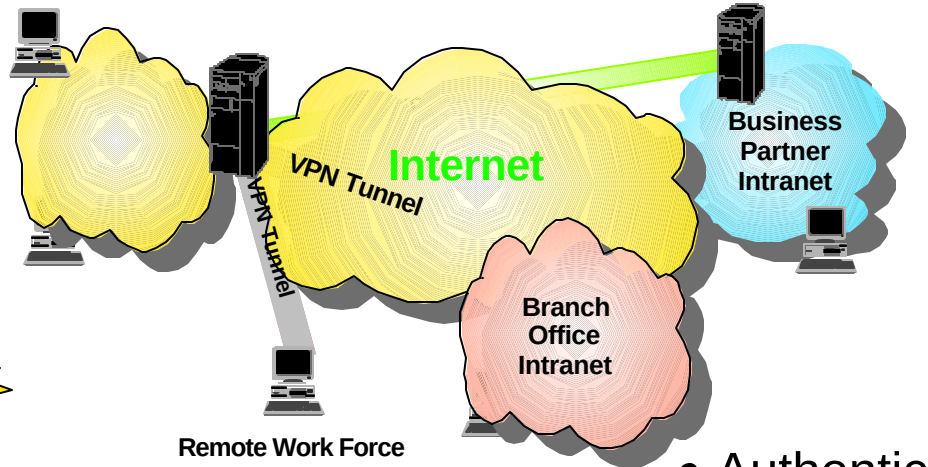
Another enhancement in V5R2 is the use of XML files for importing and exporting IP packet rules. For example, you can save existing rules into an XML file and use this file on another system or even for cross platform definitions providing the other platform supports XML. The following extract shows an XML file containing packet rules:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE QtofPacketRules SYSTEM "/QIBM/XML/DTD/QtofPacketRules.dtd">
<QtofPacketRules System="RALYAS4A.ISERIES.ITSO.RAL.IBM.COM" DTDVersion="1.0">

<Comment> ----------------------------------------------------</Comment>
<Comment> Statements to permit inbound CLIENTACCESS over ETHTEST</Comment>
<Comment> ----------------------------------------------------</Comment>
<Include Source="/QIBM/USERDATA/OS400/TCPIP/PACKETRULES/SERVICES.I3P"/>

<Filter SetName="CLIENTACCESS_INBOUND" Action="PERMIT" Direction="OUTBOUND" Journaling="OFF">
  <SourceAddress>
    <AnyIpAddress/>
  </SourceAddress>
  <DestinationAddress>
    <AnyIpAddress/>
  </DestinationAddress>
  <ServiceName>CLIENTACCESS_446_TCP_FS</ServiceName>
</Filter>
```

Virtual Private Networking (VPN)



Confidentiality

- Confidentiality
  - Data is typically encrypted in a VPN tunnel by the use of the Encapsulation Security Payload (ESP) protocol
  - Encryption algorithms that are available for the iSeries
    - Data Encryption Standard (DES)
    - Triple Data Encryption Standard (3DES)
    - RC4
    - RC5
    - Advanced Encryption Standard (AES)

V5R2

Authentication

- Authentication
  - The iSeries allows two methods to authenticate remote VPN endpoints
    - Pre-shared secret
    - Digital certificates (V5R1 and later)
- Cryptographic Access Provider 56-bit (5722-AC2) withdrawn

V5R2

Virtual Private Networking (VPN) (Cont'd)

- Integrity
  - The integrity of data is kept by the hash algorithms used by VPN that ensure no data has been changed
  - Hash algorithms that are available for the iSeries
    - HMAC MD-5
    - HMAC SHA

- Authorization
  - Using IP filtering within the VPN tunnel, you can authorize (permit) specific IP addresses to certain applications
  - Only communication to the defined end point in the VPN configuration will be permitted due to the IPSec anchor filter rule
- Other means of security/logging
  - Journaling
    - QUSRSYS/QIPFILTER journal
    - QUSRSYS/QVPN journal
  - Auditing

Notes VPN

Within the layered communications protocol stack model, the network layer (IP in the case of the TCP/IP stack) is the lowest layer that can provide end-to-end security. Network-layer security protocols provide blanket protection for all upper-layer application data carried in the payload of an IP datagram, without requiring a user to modify the applications.

VPN/Network layer security is based on the IP Security Architecture (IPSec) open framework as defined by the IPSec Working Group of the Internet Engineering Task Force (IETF). We call IPSec a framework because it provides a stable, long lasting base for providing network layer security.

IPSec was designed for interoperability. When correctly implemented, it does not affect networks and hosts that do not support it. IPSec is independent of current cryptographic algorithms. However, it supports all of the cryptographic algorithms in use today, and can also accommodate newer, more powerful, algorithms as they become available. The specific implementation of an algorithm for use by an IPSec protocol is often referred to as a transform. For example, the DES algorithm used in ESP is called the ESP DES-CBC transform.

VPN uses the following IPSec protocols:

- Authentication Header (AH), which provides data origin authentication, data integrity, and replay protection
- Encapsulating Security Payload (ESP), which provides data confidentiality, data origin authentication, data integrity, and replay protection
- Internet Key Exchange (IKE), which provides a method for automatic key management

Advanced Encryption Standard (AES) (new for V5R2) cipher algorithm was developed as a result of a contest for a follow-on standard to DES held by the National Institute for Standards and Technology (NIST). The Rijndael algorithm was selected. This is a block cipher created by Joan Daemen and Vincent Rijmen with variable block length (up to 256 bits) and variable key length (up to 256 bits). OS/400 supports a key length of 128 bits due to export regulations. The VPN implementation on the iSeries only allows AES as well as the previously supported RC4 and RC5 algorithms to be used in Phase 2 of the IKE exchange, so it is only used to protect user data, not IKE negotiations.

Beginning with V5R2, 5722-AC3 Cryptographic Access Provider 128-bit will be the only product available on iSeries. 5722-AC2 (56-bit) will not be available anymore.

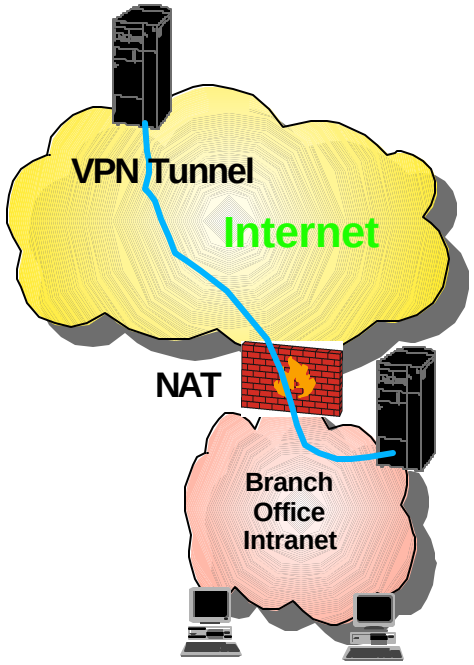
VPN Enhancements at V5R2

UDP encapsulation, a.k.a. "NAT-friendly IPSec"

- For iSeries-initiated access through a NAT system (for example, firewall)
  - Encapsulates an entire IPSec datagram into a UDP datagram, thereby allowing NAT to change the IP header in the UDP datagram rather than the hashed IP header in the original IPSec datagram
  - Currently, the iSeries can only be the initiator
  - Example of a datagram using ESP in tunnel mode

V5R2

New IP Header	UDP Header	ESP Header	Org IP Header	Payload Data	ESP Trailer	ESP Auth



Dynamic anchor filters, a.k.a. "No Policy Filters" (NPF)

- Configuring packet rules is not required to have VPN connection
- The connection can only be initiated by the local server
- The data endpoints of the connection must be single systems

Migrate Policy Filters wizard

- Migrates existing VPN policy filters from previous release
- Changes from rules file to set of GUI generated rules
- Ensures policy filters are usable when changes are made using a new interface

Notes VPN Enhancements at V5R2

Network address translation (NAT) allows you to hide your unregistered private IP addresses behind a set of registered IP addresses. This helps to protect your internal network from outside networks. NAT also helps to alleviate the IP address depletion problem, since many private addresses can be represented by a small set of registered addresses.

Unfortunately, conventional NAT does not work on IPSec packets because when the packet goes through a NAT device, the source address in the packet changes, thereby invalidating the packet. When this happens, the receiving end of the VPN connection discards the packet and the VPN connection negotiations fail. The solution is UDP encapsulation. In a nutshell, UDP encapsulation wraps an IPSec packet inside a new, but duplicate, IP/UDP header. The address in the new IP header is translated when it goes through the NAT device. Then, when the packet reaches its destination, the receiving end strips off the additional header, leaving the original IPSec packet, which should now pass all other validations. You can only apply UDP encapsulation to VPNs that will use IPSec ESP in either tunnel mode or transport mode.

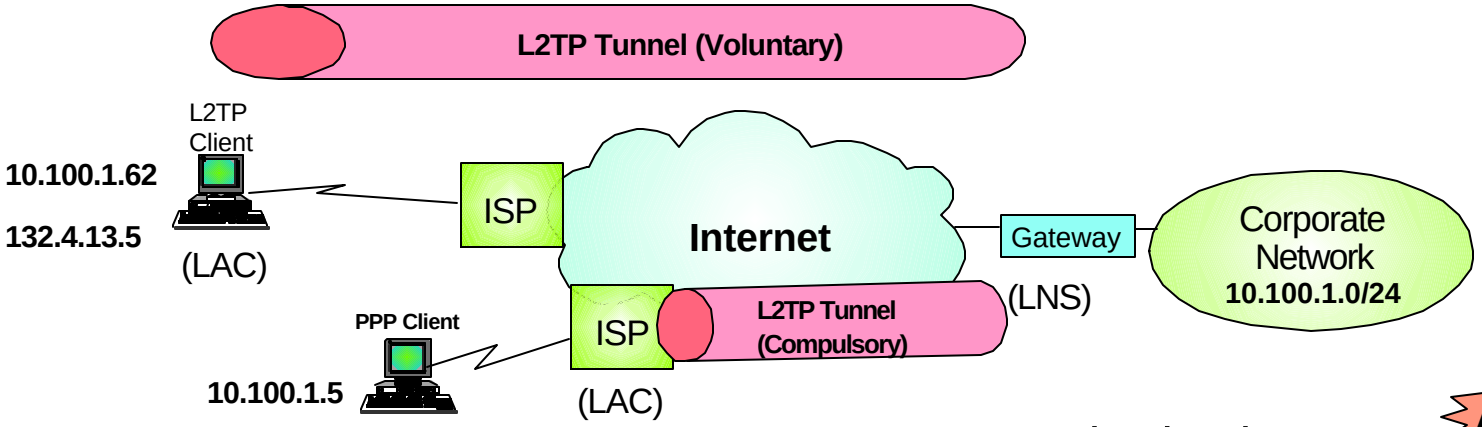
In addition, at V5R2, iSeries can only act as a client for UDP encapsulation. That is, it can only initiate UDP encapsulated traffic. Once the packet is encapsulated, the iSeries sends the packet to its VPN partner over UDP port 500. Remember, VPN partners perform IKE negotiations over UDP port 500 already. By sending UDP encapsulated traffic over the same port, the two VPN partners will not need to open additional ports through their firewalls or write any new packet rules to allow the traffic through the connection. The receiving end of the connection can determine whether the packet is an IKE packet or a UDP encapsulated packet because the first 8 bytes of the UDP payload are set to zero on a UDP encapsulated packet. Both ends of the connection must support UDP encapsulation for it to work properly

A policy filter rule defines which addresses, protocols, and ports can use a VPN and directs the appropriate traffic through the connection. In some cases, you may want to configure a connection that does not require a policy filter rule. For example, you may have non-VPN packet rules loaded on the interface that your VPN connection will use, so rather than deactivating the active rules on that interface, you decide to configure the VPN so that your system manages all filters dynamically for the connection. The policy filter for this type of connection is referred to as a "dynamic policy filter". Before you can use a dynamic policy filter for your VPN connection, all of the following must be true:

- The connection can only be initiated by the local server.
- The data endpoints of the connection must be single systems. That is, they cannot be a subnet or a range of addresses.
- No policy filter rule can be loaded for the connection.

If your connection meets this criteria, then you can configure the connection so that it does not require a policy filter. When the connection starts, traffic between the data endpoints will flow across regardless of what other packet rules are loaded on your system.

L2TP



- Authentication
  - Use PPP authentication protocols through validation lists in the L2TP connection profile
    - Password Authentication Protocol (PAP)
    - Challenge Handshake Authentication Protocol (CHAP)
    - Extensible Authentication Protocol (EAP)
  - RADIUS
    - Verify the remote system's identity by using the RADIUS Server on the network

Authentication

- Authorization
  - IP packet filtering based on filtering for the L2TP connection profile or a group of users
- Other means of security/logging
  - Journaling/Logging
    - QUSRSYS/QIPFILTER journal when IP packet filtering is being used
    - RADIUS server accounting (if auditing and accounting is activated on the network's RADIUS server
  - Auditing

Authorization

Audit/Logging

Notes L2TP

Layer Two Tunneling Protocol (L2TP) is a protocol that manages the tunneling of the link layer (for example, sync HDLC, async HDLC) of PPP. Using L2TP tunnels, it is possible to divorce the location of the initial dial-up server from the location at which the dial-up protocol connection is terminated and access to the network provided.

Virtual PPP technology extends the normal PPP session created between the client and the remote-access server to a home gateway on the Internet. The home gateway terminates the PPP session and performs all the functions of a remote-access server, including user authentication and protocol negotiation. The support of these multiprotocol virtual dial-up services (note that PPP on the iSeries system only supports the IP protocol) is of significant benefit to end users, enterprises, and Internet Service providers, because it allows the sharing of very large investments in access and core infrastructure and allows local calls to be used. It also allows existing investments in non-IP protocol applications to be supported in a secure manner while still leveraging the access infrastructure of the Internet.

L2TP provides the authentication methods of PPP. These are Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Extensible Authentication Protocol (EAP).

PAP provides a simple method for the peer to establish its identity using a two-way handshake. This is done only upon initial link establishment. After the Link Establishment phase is complete, an ID/password pair is repeatedly sent by the peer to the authenticator until authentication is acknowledged or the connection is terminated. PAP is not a strong authentication method. Passwords are sent over the link "in the clear", and there is no protection from playback or repeated trial and error attacks. The peer is in control of the frequency and timing of the attempts.

CHAP is used to periodically verify the identity of the peer using a three-way handshake. This is done upon initial link establishment, and may be repeated any time after the link has been established. CHAP provides protection against playback attack by the peer through the use of an incrementally changing identifier and a variable challenge value. The use of repeated challenges is intended to limit the time of exposure to any single attack. The authenticator is in control of the frequency and timing of the challenges. This authentication method depends on a "secret" known only to the authenticator and that peer. The secret is not sent over the link.

EAP allows third-party authentication modules to interact with the PPP implementation. EAP extends PPP by providing a standard support mechanism for authentication schemes such as token (smart) cards, Kerberos, Public Key, and S/Key. EAP responds to the increasing demand to augment RAS authentication with third-party security devices.

Notes L2TP (Cont'd)

EAP protects secure VPNs from hackers who use dictionary attacks and password guessing. However, the iSeries server currently only supports a version of EAP that is basically equivalent to CHAP-MD5.

When IPSec protocols (VPN) are used to protect the L2TP tunnel, more robust authentication transforms are in place compared to the relatively less sophisticated PPP authentication methods.

Remote Authentication Dial In User Service (RADIUS) is an open and easily integrated authentication protocol. Remote user authentication requests, initiated from an iSeries server sent to a centralized RADIUS server, are either accepted or rejected. All security information, pertaining to the authenticated user can be located in a single, central database, instead of scattered around the network in several different devices. The RADIUS server sends back to the iSeries server any services the authenticated user is authorized to use, such as an IP address.

When writing IP packet filter rules, you can associate filter rules to a given L2TP point to point connection profile. That way, those packet filter rules are only used for that (or those) L2TP user(s).

L2TP does not provide any confidentiality itself, but you can protect your L2TP tunnel with an IPSec-based VPN connection.

SSL/TLS

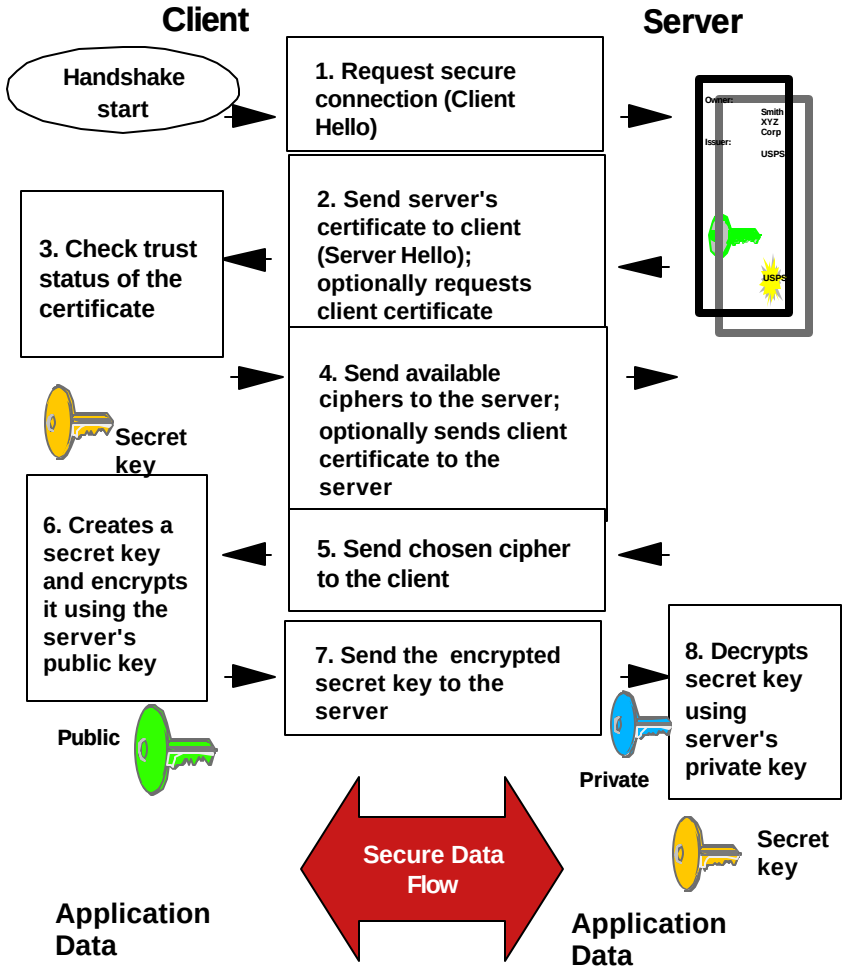
Secure Sockets Layer/Transport Layer Security

- Authentication
  - Allows each communication partner to verify the identity of the other if required (normally the client verifies the server's identity)

Authentication

- Confidentiality
  - SSL/TLS primary responsibility is to encrypt the data. This encryption is actually done at the application layer

Confidentiality



SSL/TLS (Cont'd)

- Integrity

Integrity

  - SSL/TLS ensures that data will not be changed while in transit
  - Message Authentication Codes (MACs) are used to provide this service

- Authorization

Authorization

  - At the application level based on
    - Client certificates
    - Identities provided over the secure session

- Other means of security/logging
  - Application dependent

Audit/Logging

    - For example, HTTP server logs
    - Logging via exit programs
  - Auditing

Notes SSL/TLS

The Secure Sockets Layer (SSL), originally created by Netscape, is the industry standard for session encryption between clients and servers. SSL uses asymmetric, or public key, cryptography to encrypt the session between a server and client. The client and server applications negotiate this session key during an exchange of digital certificates. The key expires automatically and the SSL process creates a different key for each server connection and each client. Consequently, even if unauthorized users intercept and decrypt a session key, they cannot use it to eavesdrop on later sessions. Certain applications provide session timeout parameters, but require a full handshake when that timeout has been reached

Based on SSL Version 3.0, Transport Layer Security (TLS) Version 1.0 is the latest industry standard SSL protocol. Its specifications are defined by the Internet Engineering Task Force (IETF) in RFC 2246, "The TLS Protocol". The major goal of TLS is to make SSL more secure and to make the specification of the protocol more precise and complete. TLS provides these enhancements over SSL Version 3.0:

- A more secure MAC algorithm
- More granular alerts
- Clearer definitions of "gray area" specifications

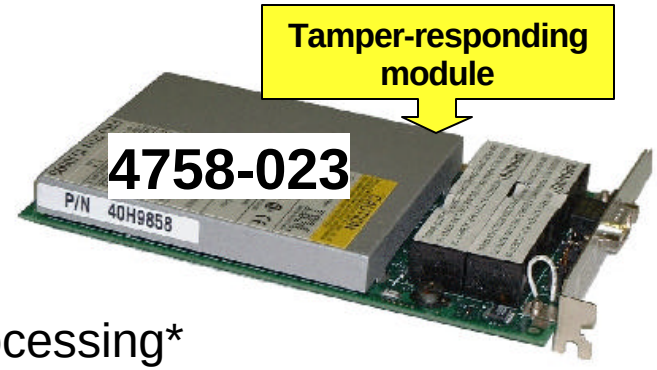
Any iSeries server applications that are enabled for SSL will automatically obtain TLS support unless the application has specifically requested to use only SSL Version 3.0 or SSL Version 2.0.

TLS provides the following security improvements over SSL Version 3.0:

- Key-Hashing for Message Authentication
  - TLS uses Key-Hashing for Message Authentication Code (HMAC), which ensures that a record cannot be altered while traveling over an open network such as the Internet. SSL Version 3.0 also provides keyed message authentication, but HMAC is considered more secure than the Message Authentication Code (MAC) function that SSL Version 3.0 uses.
- Enhanced Pseudorandom Function (PRF)
  - PRF is used for generating key data. In TLS, the PRF is defined with the HMAC. The PRF uses two hash algorithms in a way that guarantees its security. If either algorithm is exposed then the data will remain secure as long as the second algorithm is not exposed.
- Improved finished message verification
  - Both TLS Version 1.0 and SSL Version 3.0 provide a finished message to both endpoints that authenticates that the exchanged messages were not altered. However, TLS bases this finished message on the PRF and HMAC values, which again is more secure than SSL Version 3.0.
- Consistent certificate handling
  - Unlike SSL Version 3.0, TLS attempts specify the type of certificate that must be exchanged between TLS implementations.

Hardware Cryptographic Support

Functions and characteristics of the IBM 4758 PCI Cryptographic Coprocessor



- Generate random-numbers and MACs
- Clone a master key securely
- Support financial PIN-processing
- Generate and validate digital signatures
- Encrypt and decrypt data
- Improve performance for SSL handshake processing*
- Import and export encrypted DES and Triple-DES keys securely

Two models available on the iSeries server

- 4758-001 (still supported, but withdrawn from marketing)
- 4758-023

IBM 2058 e-business Cryptographic Accelerator V5R2

- Improves SSL handshake performance
- Light version of the 4758 without any key storage or generation capabilities
- Up to four adapters per system, vary on device to activate

* - Requires 4758-023

Notes Hardware Cryptographic Support

The 4758 PCI Cryptographic Coprocessor provides cryptographic processing capability and secure storage of cryptographic keys. Cryptographic functions supported include encrypt/decrypt for keeping data confidential, message digests and message authentication codes for ensuring that data has not been changed, digital signature generate/verify, and financial PIN and SET processing. You can use the coprocessor with OS/400 SSL or with custom applications written by you or an application provider.
The 4758-001 Coprocessor contains support for DES, RSA, financial PIN, and SET basic services, MD5, and SHA-1. The 4758-023 PCI Cryptographic Coprocessor supports all of the 4758-001 algorithms, plus it adds support for triple-DES and provides improved SHA-1 and RSA performance.

The main benefit of the 4758 Coprocessor is that it provides the capability to store encryption keys. It does this in a tamper-responding, battery backed-up module, which is also referred to as the "secure module". The 4758-001 PCI Cryptographic Coprocessor meets the Federal Information Processing Standard (FIPS) PUB 140-1, Level 4 requirements, and the 4758-23 PCI Cryptographic Coprocessor meets the FIPS PUB 140-1, Level 3 requirements. Another benefit of the 4758 Coprocessor is that it can be used to offload the iSeries main CPU from computationally-intensive cryptographic processing during the establishment of a SSL session. The 4758 Coprocessor provides a role-based access control facility that allows you to enable and control access to individual cryptographic operations supported by the coprocessor.

The 2058 Cryptographic Accelerator is available for customers to use with a V5R2 (or later) iSeries server. The 2058 Cryptographic Accelerator provides a competitive option to customers who do not require the high security of a 4758 Cryptographic Coprocessor, but do need the high cryptographic performance that hardware acceleration provides to offload a host processor. The 2058 Cryptographic Accelerator has been designed to improve the performance of those SSL applications that do not require secure key storage. It does not provide tamper-resistant storage for keys, like the 4758 Cryptographic Coprocessor. You can install up to four 2058 Cryptographic Accelerator cards in an iSeries server. The 2058 Cryptographic Accelerator provides special hardware that is optimized for RSA encryption (modular exponentiation) with data key lengths up to 2048 bits. The 2058 Accelerator uses multiple Rivest, Shamir and Adleman algorithm (RSA) engines.

Some features of the 2058 Cryptographic Accelerator include:

- Single card high performance cryptographic adapter (standard PCI card)
- Designed and optimized for RSA encryption
- Onboard hardware-based RNG (random number generator)
- Five mounted IBM UltraCypher Cryptographic Engines

Security at the System Layer

	Confidentiality	Integrity	Authentication	Authorization	Logging/ Auditing
User Profiles			X	X	X
Object Permissions				X	X
Object Signing and Checksum		X			X
System Values		X	X	X	X
Digital Certificates*			X	X*	
Exit Programs			X	X	X
Kerberos			X		X**

* When associated with an OS/400 user profile

** Depends on Kerberos server and services implementations

OS/400 Security

User profiles

Authentication

- Authentication
  - Simply by forcing users to sign in to an application, you *authenticate* them to the system
  - System values
    - QPWDEXPITV, QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDLVL, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, QPWDRQDDGT, QPWDRQDDIF, QPWDVLDPGM,QMAXSIGN, QMAXSGNACN

Authorization

- Authorization
  - By giving a user profile special authorities, that user will be *authorized* to various objects and can perform specific functions

Object permissions

Authorization

- Authorization
  - Specific access to an object can be given or revoked after determining if a user should have access to that object
  - System values
    - QSECURITY to enable object authorities
    - Other values to control object permissions, for example QALWUSRDMN and QUSEADPAUT

OS/400 Security (Cont'd)

Audit/Logging

- Security audit journal
- Audit journal controlled by system values
- Fine-grained options down to object level logging
- Security reports provided with OS/400 security tools (SECBATCH)
  - Authorization list authorities
  - User profile authority
  - Many different reports available
- Security tools (SECTOOLS) to control OS/400 user profile environment
  - Analyze default passwords
  - Analyze profile activity (if the user is inactive for more than xx days, then...)
  - Activation schedule
  - Expiration schedule
  - and more

User Profile Activation Schedule

User Profile	Enable Time	Disable Time	Days
BARLEN	08:00:00	17:00:00	*MON *TUE

Notes OS/400 Security

User profiles build the base for authentication and authorization on the iSeries server. Most security related settings in OS/400 are controlled by system values. The following list describes some of the values as they relate to user profiles:

QPWDEXPITV: Specifies the number of days for which passwords are valid.

- Provides password security by requiring users to change their passwords after a specified number of days. If the password is not changed within the specified number of days, the user cannot sign on until the password is changed.

QPWDLMTAJC: Specifies whether adjacent numbers are allowed in passwords.

- Makes it difficult to guess passwords by preventing the use of dates or social security numbers as passwords.

QPWDLMTCHR: Provides password security by preventing certain characters (vowels, for example) from being in a password.

- This makes it difficult to guess passwords by preventing the use of common words or names as passwords.

QPWDLMTREP: Prevents a user from using the same character more than once in the same password.

QPWDLVL: Specifies the level of password support on the system.

QPWDMAXLEN: Specifies the maximum number of characters in a password.

QPWDMINLEN: Specifies the minimum number of characters in a password.

QPWDPOSDIF: Controls the position of characters in a new password.

- Prevents the user from specifying the same character in a password corresponding to the same position in the previous password.

QPWDRQDDGT: Specifies whether a digit is required in a new password.

- Prevents the user from only using alphabetic characters.

QPWDRQDDIF: Limits how often a user can repeat the use of a password.

Notes OS/400 Security

QPWDVLDPGM: Provides the ability for a user-written program to do additional validation on passwords.

QMAXSIGN: Incorrect sign-on attempts on secured systems (security level 20 or higher, see the system value QSECURITY) occur from any of the following circumstances:

- Incorrect user ID
- Incorrect password
- The user profile does not have authority to the device from which the user ID was entered

QMAXSGNACN: Specifies how the system reacts when the maximum number of consecutive, incorrect, sign-on attempts (the system value QMAXSIGN) is reached.

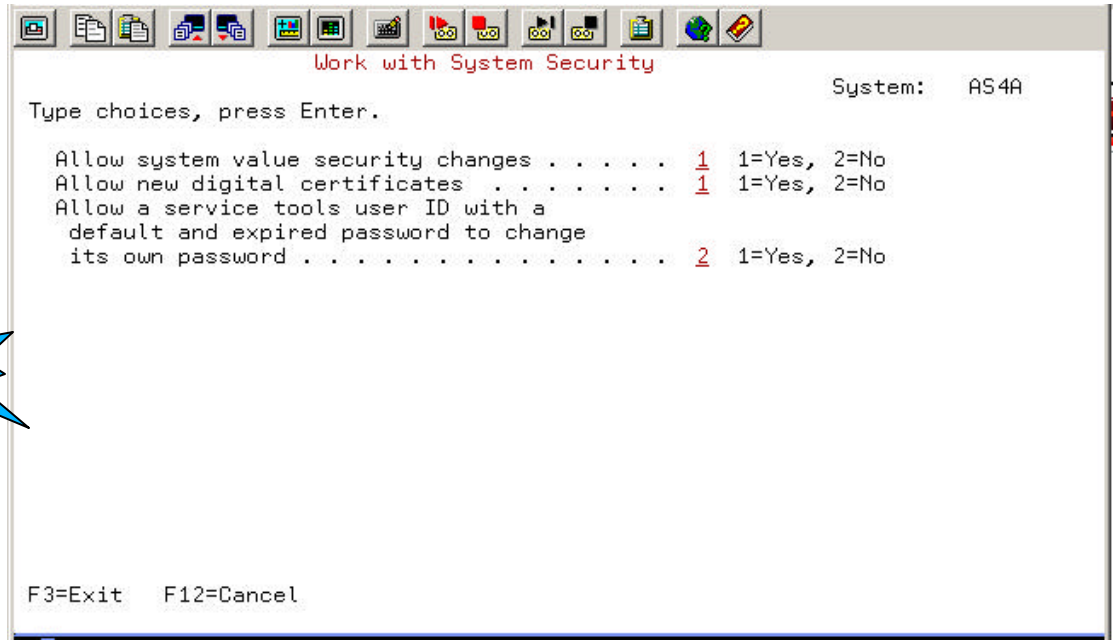
QSECURITY: Specifies the level of security on the system. (Shipped value is 40)

- 10 The system does not require a password to sign on. Users have access to all system resources. **Note:** Security level 10 is no longer supported.
- 20 The system requires a password to sign on. Users have access to all system resources.
- 30 The system requires a password to sign on and users must have authority to access objects and system resources.
- 40 The system requires a password to sign on and users must have authority to access objects and system resources. Programs fail if they try to access objects through interfaces that are not supported.
- 50 The system requires a password to sign on and users must have authority to access objects and system resources. Programs fail if they try to pass unsupported parameter values to supported interfaces or if they try to access objects through interfaces that are not supported.

For a complete list of all security related system values and their meaning refer to *IBM iSeries Security Reference*, SC41-5302.

OS/400 Security

STRSST option 7 allows you to administer general system security functions at V5R2



V5R2

- Prevents power users from changing security-related system values
- Controls whether the new Add Verifier (QYDOADDV) API can be used or certificate store passwords can be reset
- Controls the behavior of service tools users and their passwords

Notes OS/400 Security

With V5R2, you control via SST settings whether a user can change security-related system values. If set to No, a user is prevented from the changing the following values:

Lockable system values

- Auditing system values
  - Activate action auditing QAUDLVL
  - Activate object auditing QAUDCTL
  - Audit journal error action QAUDENACN
  - Default auditing for newly created objects QCRTOBJAUD
  - Maximum number of journal entries in auxiliary storage QAUDFRCLVL
- Device system values
  - Local controllers and devices QAUTOCFG
  - Pass-through devices and Telnet QAUTOVRT
  - Action to take when a device error occurs QDEVRCYACN
  - Remote controllers and devices QAUTORMT
- Jobs system values
  - Time-out interval QDSCJOBITV
  - When job reaches time-out QINACTMSGQ
  - Password system values
  - Password expiration QPWDEXPITV
  - Restrict consecutive digits QPWDLMTAJC
  - Restricted characters QPWDLMTCHR
  - Restrict repeating characters QPWDLMTREP
  - Password level QPWDLVL
  - Maximum password length QPWDMAXLEN
  - Minimum password length QPWDMINLEN
  - Require a new character in each position QPWDPOSDIF
  - Require at least one digit QPWDRQDDGT
  - Password reuse cycle QPWDRQDDIF
  - Password validation program QPWDVLDPGM
- Messages and service system values
  - Allow remote service of system QRMTSRVATR
- Restore system values
  - Verify object signatures on restore QVFYOBJRST
  - Convert objects during restore QFRCCVNRST
  - Allow restore of security sensitive objects QALWOBJRST
- Security system values
  - Security level QSECURITY
  - Allow server security information to be retained QRETSVRSEC
  - Users who can work with programs with adopted authority QUSEADPAUT
  - Default authority for newly created objects in QSYS.LIB file system QCRTAUT
  - Allow use of shared or mapped memory with write capability QSHRMEMCTL
  - Allow these objects in . . . QALWUSRDMN
- Sign-on system values
  - Use pass-through or Telnet for remote sign-on QRMTSIGN
  - Display sign-on information QDSPSGNINF
  - Restrict privileged users to specific device session QLMTSECOFR
  - Limit each user to one device session QLMTDEVSSN
  - Incorrect sign-on attempts QMAXSIGN
  - When maximum is reached QMAXSGNACN

You can also control the new Add Verifier (QYDOADDV, QydoAddVerifier) API. This API adds a certificate to a system's *SIGNATUREVERIFICATION certificate store. The system can then use the added certificate to verify signatures on objects that the certificate created. Verifying the signature allows the system to verify the integrity of the signed objects to ensure that the objects have not changed since they were signed. If the certificate store does not exist, this API creates it as it adds the certificate. When set to No, the API cannot be used to add verifies. It also prevents a user from resetting certificate store passwords.

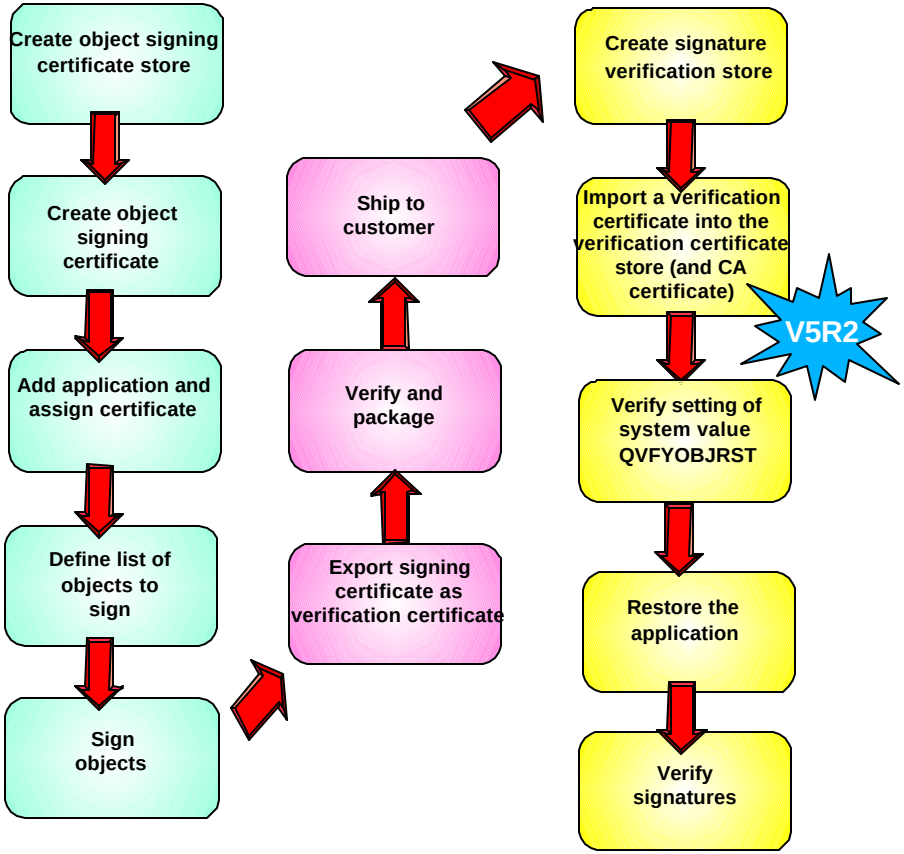
Object Signing

Object signing

Integrity

- Integrity
  - Use DCM or object signing APIs to sign objects and to verify the authenticity of digital signatures on objects. This ensures that the data in the object has not been changed since the owner of the object signed it
  - iSeries Navigator's Management Central (at V5R2) can also be used to sign objects as you package them for distribution to other iSeries systems
    - Allows a way to easily package and distribute digitally signed objects

V5R2



Notes Object Signing

Object signing and signature verification are security capabilities that you can employ to verify the integrity of a variety of iSeries objects. You use a digital certificate's private key to sign an object, and you use the certificate (which contains the corresponding public key) to verify the digital signature. A digital signature ensures the integrity of time and content of the object that you are signing. The signature is non-repudiated proof of both authenticity and authorization. It can be used to show proof of origin and detect tampering. By signing the object, you identify the source of the object and provide a means for detecting changes to the object. When you verify the signature on an object, you can determine whether there have been changes to the contents of the object since it was signed. You can also verify the source of the signature to ensure the reliability of the object's origin.

Before you can use DCM to verify signatures on objects, you must ensure that certain prerequisite conditions are met:

- The *SIGNATUREVERIFICATION store must be created to manage your signature verification certificates.
- The *SIGNATUREVERIFICATION certificate store must contain a copy of the certificate that signed the objects.
- The *SIGNATUREVERIFICATION certificate store must contain a copy of the CA certificate that issued the certificate that signed the objects.

Using Management Central to sign objects is a new function of iSeries Navigator at V5R2. Using Management Central to package and sign objects reduces the amount of time that you must spend to distribute signed objects to your company's iSeries servers. It also decreases the number of steps that you must perform to sign objects because the signing process is part of the packaging process. Signing a package of objects allows you to more easily determine whether objects have been changed after they have been signed. This may reduce some of the troubleshooting that you do in the future to track down application problems.

In V5R2, there are also a few new APIs for the object signing and signature verification environment. A particular interesting one is the Add Verifier (QYDOADDV, QydoAddVerifier) API. This API adds a certificate to a system's *SIGNATUREVERIFICATION certificate store. The system can then use the added certificate to verify signatures on objects that the certificate created. Verifying the signature allows the system to verify the integrity of the signed objects to ensure that the objects have not changed since they were signed. If the certificate store does not exist, this API creates it as it adds the certificate.

Note that for security reasons, this API does not allow you to insert a Certificate Authority (CA) certificate into the *SIGNATUREVERIFICATION certificate store. When you add a CA certificate to the certificate store, the system considers the CA to be a trusted source of certificates. Consequently, the system treats a certificate that the CA issued as having originated from a trusted source. Therefore, you cannot use the API to create an install exit program to insert a CA certificate into the certificate store. You must use Digital Certificate Manager to add a CA certificate to the certificate store to ensure that someone must specifically and manually control which CAs the system trusts. Doing so prevents the possibility that the system could import certificates from sources that an administrator did not knowingly specify as trusted.

Object Signing

The CHKOBJITG command can be used to check the integrity of a single object, several objects, or all objects on the system

Integrity

- It not only verifies object signatures, but also verifies the integrity of program objects based on checksums
- Objects that can be signed include:
  - Save files (not empty ones) in the QSYS.LIB file system
  - Programs of types *PGM, *SVRPGM, *SQLPKG, *JVAPGM, and *MODULE
  - IFS stream files in local file systems
  - *CMD objects
  - **Note:** You cannot sign objects that are compiled for a release prior to V5R1.

Audit/Logging

V5R2

QVFYOBJRST system value

- Specifies the policy to be used for object signature verification during a restore operation

Notes Object Signing

The Check Object Integrity (CHKOBJITG) command checks the objects owned by the specified user profile, the objects that match the specified path name, or all objects on the system to determine if any objects have integrity violations. An integrity violation occurs if:

- A command has been tampered with.
- An object has a digital signature that is not valid.
- An object has an incorrect domain attribute for its object type.
- A program or module object has been tampered with.
- A library's attributes have been tampered with.

If an integrity violation has occurred, the object name, library name (or pathname), object type, object owner, and type of failure are logged to a database file.

The command flags the verified files with the following flags:

- **ALTERED:** The object has been tampered with
- **BADSIG:** The object has a digital signature that is not valid
- **DMN:** The domain is not correct for the object type
- **PGMMOD**: The runnable object has been tampered with

QVFYOBJRST: Specifies the policy to be used for object signature verification during a restore operation.

- Introduced at V5R1
- Specifies the policy for object signature verification during restore operations
- Signatures are verified when:
  - Restoring *PGM, *SRVPGM, *MODULE, *SQLPKG, *STMF, *CMD with attached Java programs from media or out of a save file
- Signatures are not verified when:
  - Restoring a signed save file. Signatures on save files are verified when you attempt to restore objects from the save file.
  - Restoring stream files without attached Java programs
- The default setting (3) allows unsigned objects to be restored, but ensures that signed objects can only be restored if the objects have a valid signature. System-state objects cannot be restored without a valid signature.

Digital Certificates at System Level

Digital Certificates

- Authentication
  - Digital certificates can be used on the system level when the certificate is associated with a user profile
    - Client certificates can be used to authenticate the client user and to control access to the system or system resources

- Integrity
  - You can use DCM to create and manage certificates that you can use to digitally sign objects to ensure their integrity and provide proof of origination for objects
  - You can also create and manage the corresponding signature verification certificates that you or others can use to authenticate the signature on a signed object to ensure that the data in the object is unchanged and to verify proof of the object's origin
  - You can also use DCM to sign an object and verify the signature on a object

- Confidentiality
  - Digital certificates provides encrypti[illegible] its use of public/private keys

Notes Digital Certificates

Through DCM or the APIs Digital Certificates can be associated with user profiles. An application, such as the HTTP Server for iSeries, can authenticate users based on their client certificate. OS/400 accesses resources under the authority of the user profile the client certificate is associated with.

Beginning at V5R1, you can use Digital Certificate Manager to sign objects. Traditional object signing, as most people know, is used for signing e-mails. Usually an e-mail is signed using a person's individual certificate. The recipient, when verifying the e-mail's signature, can then determine who the person was that signed the e-mail. The object signing implementation as introduced with V5R1 does not provide a way that an individual certificate that is associated with a user profile can be used to sign objects. Instead, an object signing certificate that represents the system rather than the individual user is used to sign objects.

As part of the process of verifying digital signatures, you must decide which Certificate Authorities you trust and which certificates you trust for signing objects. When you elect to trust a CA, you can elect whether to trust signatures that someone creates by using a certificate that the trusted CA issued. When you elect not to trust a CA, you also are electing not to trust certificates that the CA issues or signatures that someone creates by using those certificates.

If you use certificates to identify users within your company, you need to consider how to store, backup, and secure them. Storing certificates on a PC ties a person to one PC. If the PC is unavailable, the person cannot access their certificate. You may want to store certificates on a local file server so that they are accessible to the people who need them, but not to everyone. When laptops are used, you need to export copies of the user’s certificates to their laptop. In all cases, you should try to make sure that users secure the certificates with a non-trivial password. You may also consider exporting copies of certificates to a secure repository in case people lose their certificates or forget the password needed to unlock it.

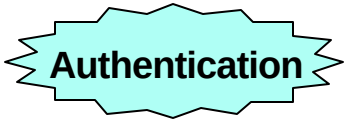
The certificate containing the public key must usually be available to the public. This can be achieved by storing the certificates in a Lightweight Directory Protocol (LDAP) directory.

Exit Programs

Exit programs can be used to:

- Add functionality to OS/400 functions or applications
- Act as an interface between user input or requests and OS/400 applications

Authentication



- Can be used to perform additional checking during authentication of users in many TCP/IP applications, including Telnet, FTP, etc.

Authorization

- Can be used to authorize users to specific objects/functions in many TCP/IP applications, including Telnet, FTP, REXEC, TFTP, etc.
- Beginning with V5R1, you can use Operations Navigator using Application Administration (AppAdmin) to grant and deny access both in and out of the system (using FTP) for individual users or for groups of users for FTP functions and commands.
  - For example, LS, CWD, PUT, GET, etc.

Exit Programs (Cont'd)

Logging

- Exit programs are excellent ways to implement custom logging facilities, for example:
  - Log all issued FTP subcommands per user
  - Keep track of signed on users and the devices/IP addresses they used

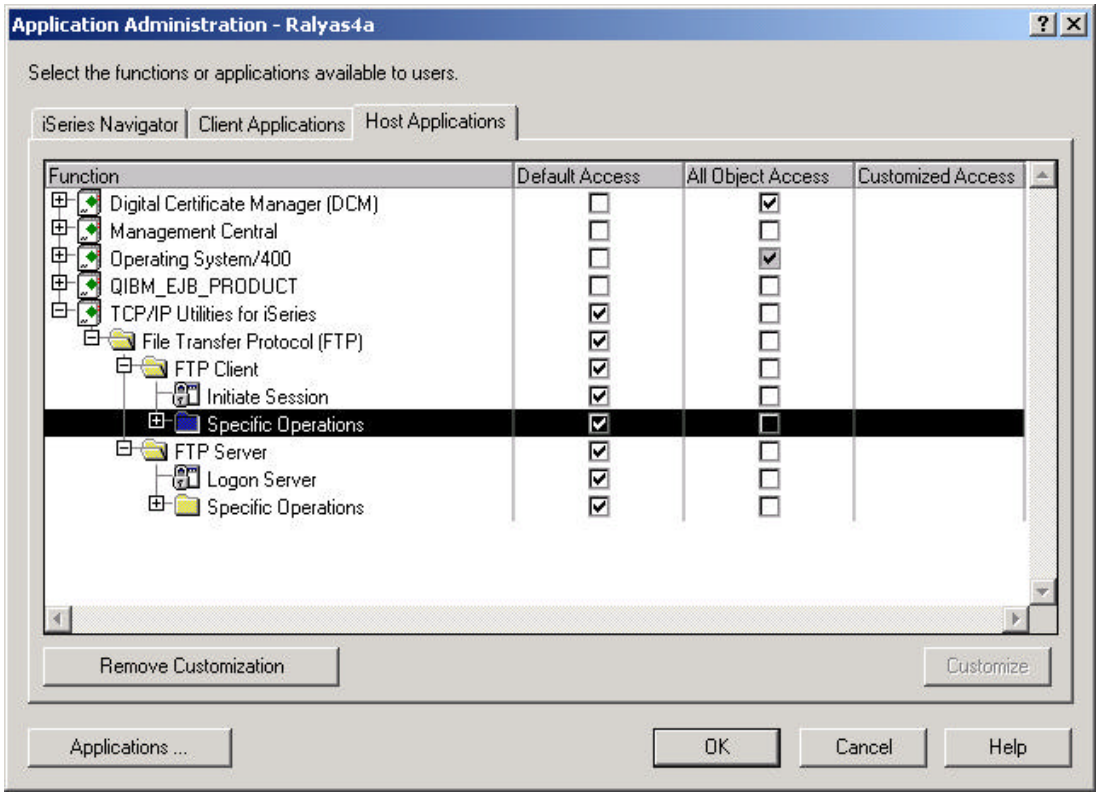
Notes Exit Programs

Exit programs exist for many OS/400 functions and applications. The purpose they serve is different for each exit program and their associated application. However, many of them, such as Telnet and FTP exit programs, can be used to perform additional checking during authentication or can be used to control what an authenticated user can do. All exit programs have to be registered. Using the Work with Registration Information (WRKREGINF) command, you can register your exit programs with exit points.

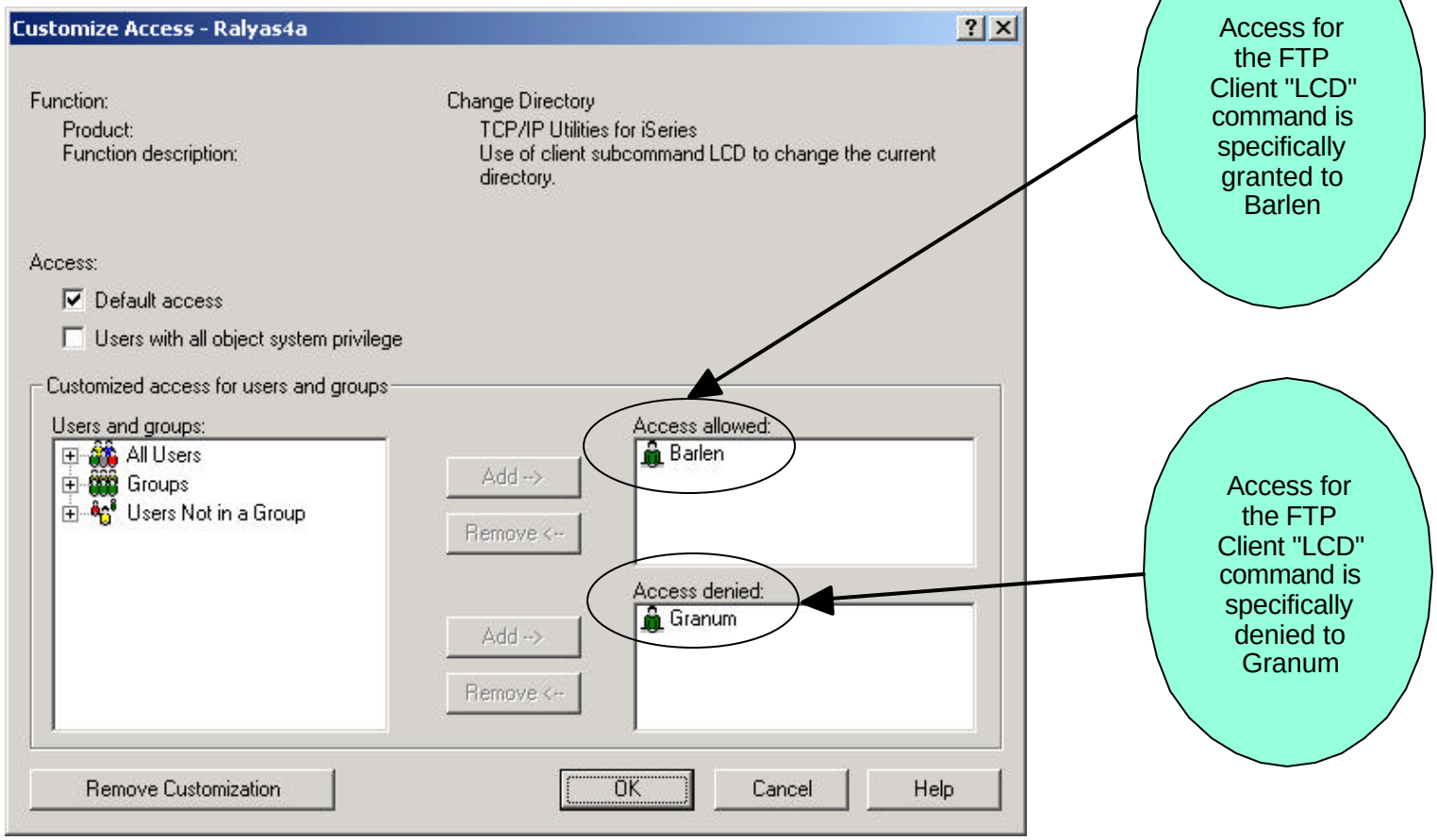
Logging and auditing is also a very important aspect when monitoring security. You can use exit programs to create your own logging mechanism for various system applications. For example, the FTP server does not provide a standard interface to enable logging of FTP subcommands performed by a signed on user. However, with the help of the Request Validation exit point, you can write your own exit program to log these commands.

Application Administration

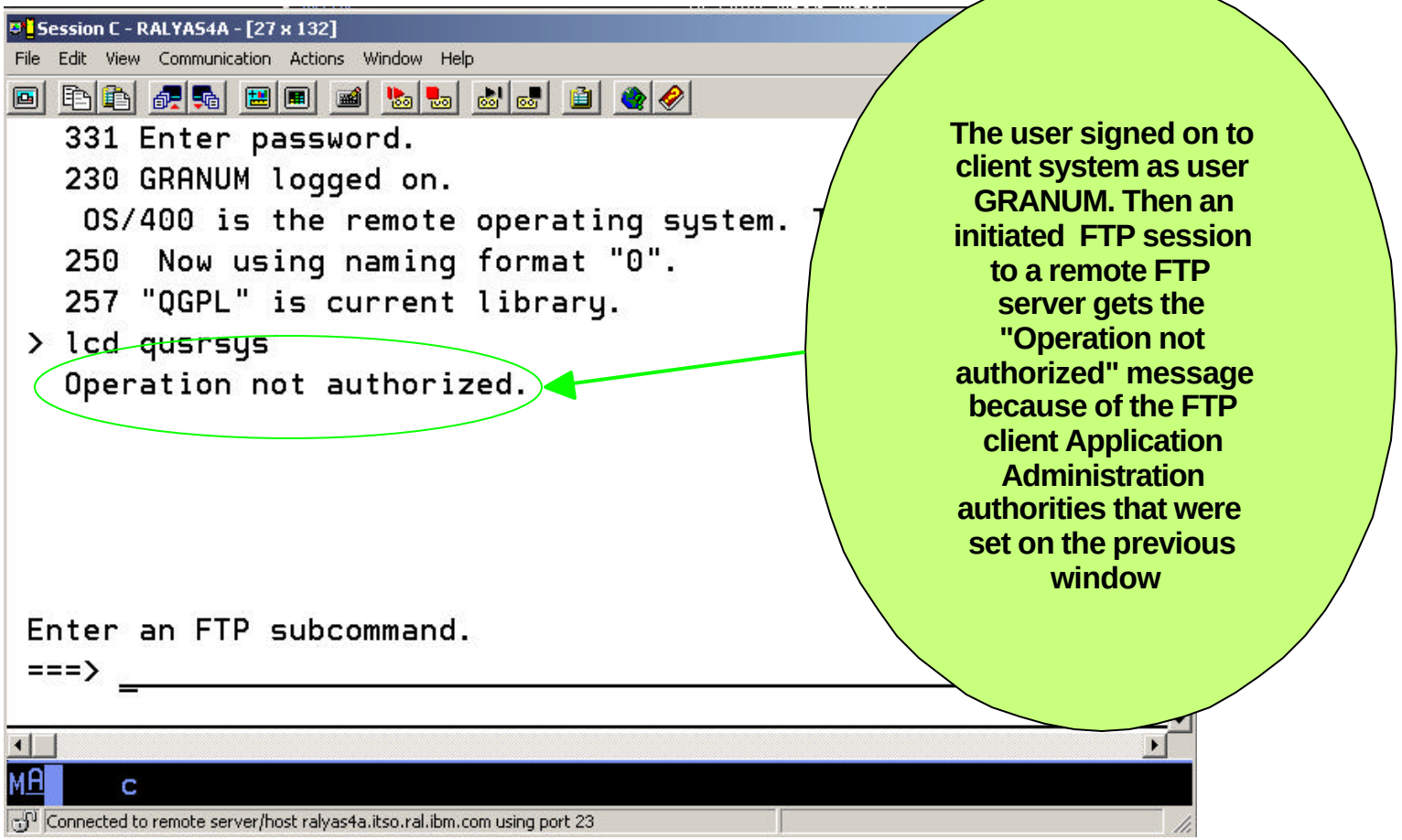
Application Administration can implement security constraints to a very fine detail and open the FTP client and server security completely or anywhere in between. This example shows you where to set authorities to limit FTP client commands for users on the iSeries.



Application Administration (Cont'd)



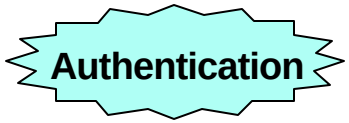
Application Administration (Cont'd)



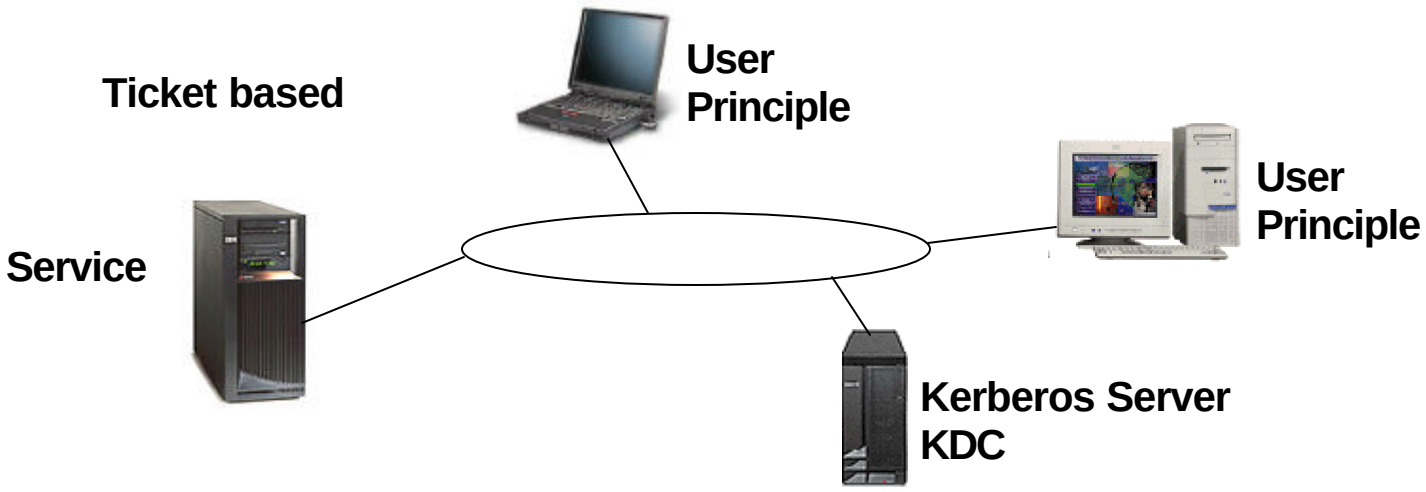
The user signed on to client system as user GRANUM. Then an initiated FTP session to a remote FTP server gets the "Operation not authorized" message because of the FTP client Application Administration authorities that were set on the previous window

Kerberos

Authentication



- Kerberos performs authentication as a trusted third-party authentication service through the use of conventional shared secret key cryptography
- Kerberos was designed with the following pretenses:
  - Does not rely on authentication by the host operating system
  - Does not base trust on host addresses
  - Does not require physical security of all the hosts on the network
  - Packets traveling along the network can be read, modified, and inserted at will



Notes Kerberos

Kerberos provides a means of verifying the identities of principals, without relying on authentication by the host operating system, without basing trust on host addresses, without requiring physical security of all the hosts on the network, and under the assumption that packets traveling along the network can be read, modified, and inserted at will.

The Kerberos protocol provides third party authentication where a user proves their identity to a centralized server, called the key distribution center (KDC), which issues tickets to the user. The user can then use these tickets to prove their identity on the network. The ticket eliminates the need for multiple sign-ons to different systems. The Kerberos APIs that the iSeries supports originated from Massachusetts Institute of Technology and have become the defacto standard for using the Kerberos protocol.

The Kerberos protocol assumes that all data exchanges occur in an environment where packets can be inserted, changed, or intercepted at will. Use Kerberos as one layer of an overall security plan. Although the Kerberos protocol allows you to authenticate users and applications across your network, you should be aware of some limitations when you define your network security objectives:

- The Kerberos protocol does not protect against denial-of-service attacks. 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 are usually best left to human administrators and users.

- Key sharing or key theft can allow impersonation attacks. If intruders somehow steal a principal's key, they will be able to masquerade as that user or service. To limit this threat, prohibit users from sharing their keys and document this policy in your security regulations for your corporate security policy.

- The Kerberos protocol does not protect against typical password vulnerabilities, such as password guessing. If a user chooses a poor password, an attacker might successfully mount an offline dictionary attack by repeatedly attempting to decrypt messages that are encrypted under a key derived from the user's password.

Notes Kerberos (Cont'd)

Network authentication service provides application program interfaces (APIs) to verify the identity of a user in a network. Application programs can use these APIs to 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 is an implementation of:

- Kerberos Version 5 protocol as defined by 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 2078
- The OS/400 implementation is designed to interoperate with authentication, delegation, and data confidentiality services compliant with these RFCs, such as Microsoft's Windows 2000 Security Service Provider Interface (SSPI) APIs

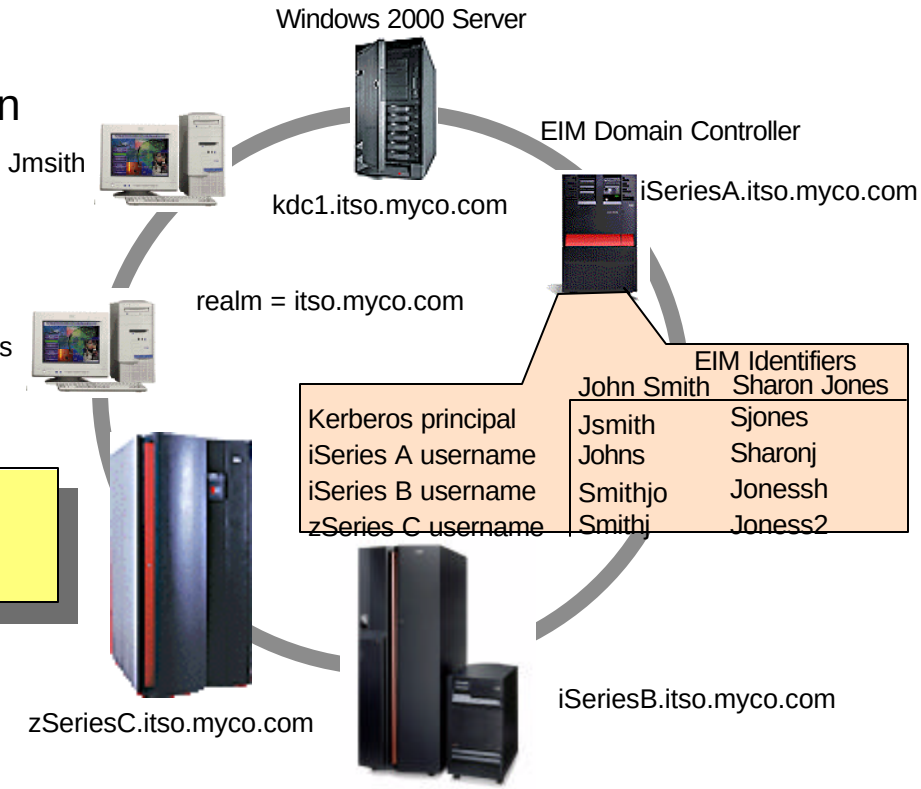
Network authentication service uses Generic Security Service (GSS) APIs to provide a framework so that programmers can write applications using the Kerberos APIs. The GSS APIs provide security services to applications that use peer-to-peer communications. Using GSS API routines, applications can perform the following operations:

- Determine another application's user identification
- Delegate access rights to another application
- Apply security services, such as confidentiality and integrity, on a per-message basis

Enterprise Identity Mapping

V5R2

- 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
- Utilizes LDAP directories and Kerberos authentication services



Authentication

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

The IBM autonomic computing initiative

Notes Enterprise Identity Mapping

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, provide users, administrators, and application developers the benefits of easier password and user identity management across multiple platforms — without changing the underlying security schema.

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 (such as 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.

EIM is a part of IBM's autonomic computing initiative (formlery known as the eLiza project). The goal of this initiative is to give businesses the ability to manage systems and technology infrastructures that are hundreds of times more complex than those in existence today.

The 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.

Security at the Application Layer

	Confidentiality	Integrity	Authentication	Authorization	Audit/ Logging
Validation Lists			X	X	X
Digital Certificates		X	X	X	
Exit Programs			X	X	X
SSL	X	X	X	X	X
Port Restrictions				X	
Kerberos			X		X

Validation Lists

Validation lists can be used to authenticate users connecting to the iSeries

Authentication

- Validation lists contain entries that consist of an identifier, data that will be encrypted when it is stored, and free-form data. Entries can be added, changed, removed, found, and validated.
- Validation lists can be used for user-written applications by using the validation list APIs
- Native applications on the iSeries that use validation lists are PPP, L2TP, and HTTP
  - Users attempting to establish a session (assuming the application is set up to perform authentication) to the iSeries need to send a user ID and password to the iSeries. This password is stored in an encrypted form on the iSeries in that validation list. When the password is received by the iSeries, it compares the two passwords to verify the password that was sent is correct.

Notes Validation Lists

Validation lists contain entries that consist of an identifier, data that will be encrypted when it is stored, and free-form data. Entries can be added, changed, removed, found, and validated. You can validate entries by providing the correct entry identifier and data that is encrypted.

One way to use validation lists is to store the user names of a Web browser. The entry identifier would be the user name, the data to encrypt would be the user's password, and the free-form data field would contain any additional data about the user that the browser wanted to store.

Validation List APIs:

- Find Validation List Entry (**QSYFDVLE**) finds an entry in a validation list object and returns it.
- Find Validation List Entry (**QsyFindValidationLstEntry()**) finds an entry in a validation list object and returns information about the validation list entry.
- Find Validation List Entry Attributes (**QsyFindValidationLstEntryAttrs()**) finds an entry in a validation list object, and the attributes associated with the entry.
- Open List of Validation List Entries (**QSYOLVLE**) returns a list of validation list entries in a validation list object.
- Remove Validation List Entry (**QsyRemoveValidationLstEntry()**) removes an entry from a validation list object.
- Remove Validation List Entry (**QSYRMVLE**) removes an entry from a validation list object.
- Verify Validation List Entry (**QsyVerifyValidationLstEntry()**) verifies an entry in a validation list object.

OS/400 TCP/IP Application Support

	Confidentiality	Integrity	Authentication	Authorization	Audit/Logging
Telnet Server	SSL/TLS	SSL/TLS	SSL/TLS (DCM), Kerberos, UserProfiles	Exit Programs	via IP Filtering Exit Programs
Telnet Client	N/A	N/A	N/A	Exit Programs	via IP Filtering Application log.
FTP Server	SSL/TLS	SSL/TLS	SSL/TLS (DCM), UserProfiles	AppAdmin, Exit Programs	via IP Filtering Exit Programs
FTP Client V5R2	SSL/TLS	SSL/TLS	SSL/TLS (CA Trust)	AppAdmin, Exit Programs	via IP Filtering
HTTP Server	SSL/TLS	SSL/TLS	SSL/TLS (DCM), UserProfiles Validation Lists, LDAP Directory	HTTP directives	via IP Filtering Server logs
LDAP Client	SSL/TLS	SSL/TLS	SSL/TLS (DCM)	N/A	via IP Filtering Appl. dependent
LDAP Server	SSL/TLS	SSL/TLS	SSL/TLS (DCM), Kerberos, UserProfiles	Access Control Lists (ACLs)	Audit journal Change log
Host Servers iSeries Access	SSL/TLS	SSL/TLS	User profiles Kerberos	AppAdmin	via IP Filtering

Note: Since VPN works at the network layer, it can provide confidentiality, integrity, authentication, and authorization for any TCP/IP application.

Notes OS/400 TCP/IP Application Support

The previous chart lists the OS/400 TCP/IP applications that have been enabled for Transport Layer Security (TLS)/Secure Sockets Layer (SSL). Depending on the application, authentication and authorization support varies.

At V5R1, you can use Transport Layer Security (TLS)/Secure Sockets Layer (SSL) connections to encrypt data transferred over FTP control and data connections as well as for Telnet server and LDAP server and client connections. For FTP, the primary reason for encryption on the control connection is to conceal the password when logging on to the FTP server. In V5R2, the OS/400 FTP client is also SSL-enabled. However, it supports only server authentication. Before using the FTP client to make secure connections to servers, you must use DCM to configure trusted certificate authorities for the FTP Client. Any certificate authorities that were used to create certificates assigned to servers that you want to connect to must be added. Exporting or importing Certificate Authority (CA) certificates may be required depending on the CAs used.

If you choose TLS/SSL encryption for the control connection, the FTP client will also encrypt the data sent on the FTP data connection by default. FTP does not allow you to have a secure data connection without a secure control connection. Encryption can have a significant performance cost and can be bypassed on the data connection. This allows you to transfer non-sensitive files without decreasing performance and still protect the system's security by not exposing passwords.

The FTP client has parameters for the STRTCPFTP CL command and subcommands that are used as part of the TLS/SSL support (SECOpen and SECData).

Notes OS/400 TCP/IP Application Support (Cont'd)

Specifying TLS/SSL protection for the iSeries FTP Client

- Control Connection
  - TLS/SSL protection can be specified on the STRTCPFTP command and the SECOPEN subcommand.
  - For the STRTCPFTP (FTP) command, specify *SSL for the SECCNN secure connection parameter to request a secure control connection. Also, you may be able to specify *IMPLICIT to obtain a secure connection on a pre-defined server port number. (See IMPLICIT SSL Connection below for more details.)
  - Within your FTP client session, the SECOPEN subcommand can be used to obtain a secure control connection.

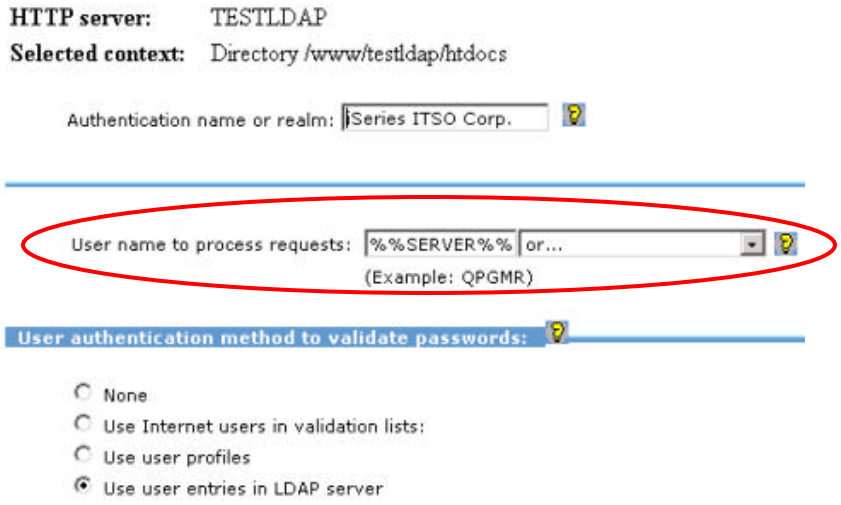
- Data Connection
  - For the STRTCPFTP (FTP) command, enter *PRIVATE for the DTAPROT data protection parameter to specify a secure data connection. Enter *CLEAR for the DTAPROT data protection parameter to specify data to be sent without encryption.
  - When you have a secure control connection, you can use the SECDATA subcommand to change the data connection protection level.

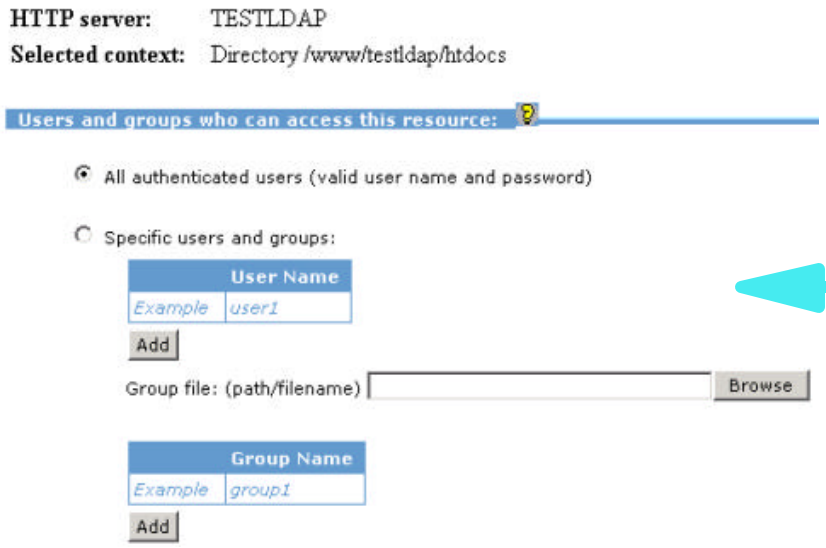
- Implicit SSL connection
  - Some FTP servers support what is called an "implicit SSL connection". This connection provides the same encryption protection as the *SSL option, but can only be done on a predetermined server port, usually 990, for which the server must be configured to expect an SSL/TLS connection negotiation.
  - This method is provided to allow secure connections to those FTP implementations that may not support the standard protocol for providing TLS/SSL protection.
  - Many early implementations of SSL support used the implicit approach, but now it is no longer recommended and has been deprecated by the IETF.

IBM HTTP Server Security

- Digital certificates
- OS/400 user profiles
- User names in validation lists
- User entries in LDAP directory

Authentication





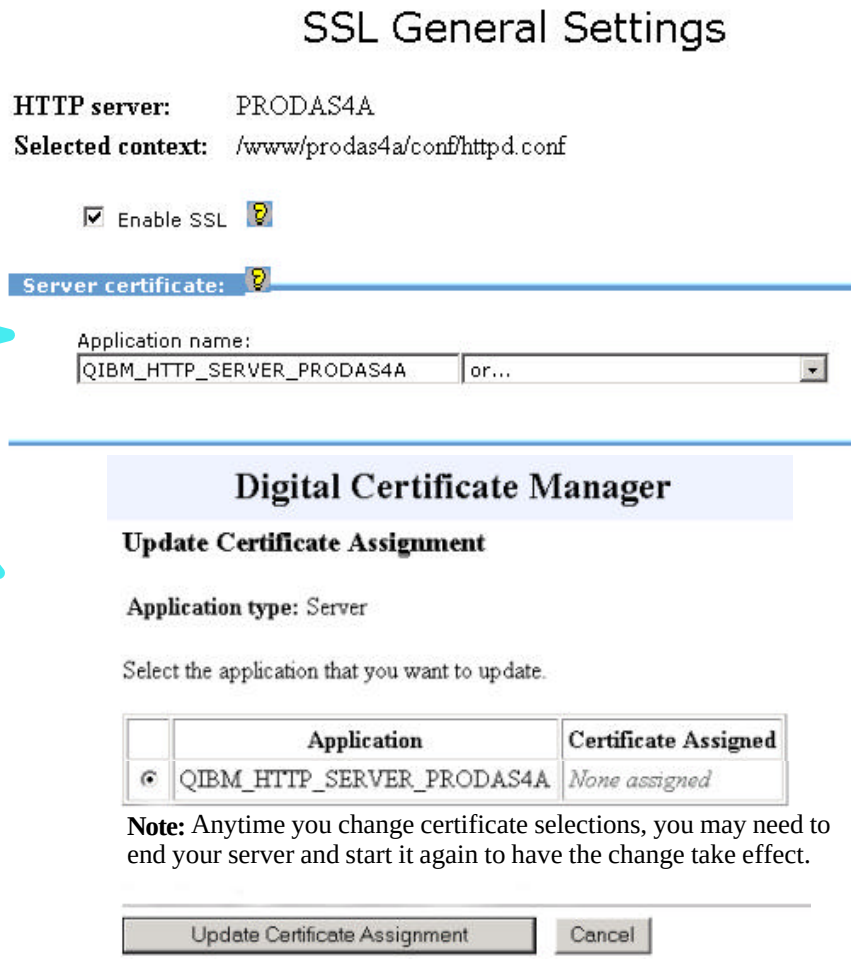
Authorization

- Configuration can allow or disallow access to resources based on:
  - Authenticated User name
  - Domain Name, IP Address, or IP Address/Subnet Mask

IBM HTTP Server Security

Confidentiality **Integrity**

- SSL/TLS with digital certificates should be used to encrypt data for transmission
  - Instance must be enabled for SSL as shown; then a certificate must be assigned to the application through DCM



Audit/Logging

Server can be configured to log:

- Access, referrals, clients
- Errors

Notes IBM HTTP Server Security

Security is always one of the main concerns on the mind of a Web server administrator. Security comes from a set of constantly updated rules and practices, specifically designed to protect the availability of your server and the integrity of your data.

Once an http request makes it to the IBM HTTP server, requests are filtered by the HTTP server security. Data is protected through:

- User authentication, which can be done through:
  - Using digital certificates
  - User ID and password verification
    - LDAP
    - OS/400 user profiles
    - User IDs existing in validation lists
- Access control: (Specifically, at this point, we are discussing access control from the IBM HTTP server's point of view – above the access control that OS/400 also enforces.)
  - This is enforced through a set of policies that define who can access your data, what kind of authority they will be granted, and what actions they will be allowed to perform on them. A server-wide access control policy is enforced on the document root and propagated upon lower level contexts unless overridden by local directives or local configuration files. In addition to that, the server never tries to access system resources for which explicit access has not been configured.

In an e-business environment, usually confidential data is transmitted across the Internet from the server to the client or vice-versa. Since these transactions are necessary, we must encrypt the data being transmitted. This can be accomplished through the use of SSL. As with any application, the IBM HTTP server must be enabled to use SSL, the application must trust the certificate granting Certificate Authority, and then the granted certificate must be associated with the HTTP Server instance. This is all done through the Digital Certificate Manager interface.

User Applications

	Confidentiality	Integrity	Authentication	Authorization	Audit/Logging
Validation Lists			X	X	X
LDAP			X	X	X
Kerberos			X	X	X
OS/400 Security			X	X	X
SSL/TLS	X	X	X	X	X
Object Signing		X			X
Self-written Functions	X	X	X	X	X

User Applications Security

Authentication for user applications can be achieved by:

- Validation lists
  - Using OS/400 validation list APIs
- LDAP
- Kerberos
  - Using OS/400 Kerberos APIs
- Self-written functions
  - Functions or programs written by you or a third party-application providing authentication (includes usage of APIs and Java packages)

Authorization for user applications can be achieved by:

- Exploiting OS/400 security as discussed previously
- Self-written functions
  - Functions or programs written by you or a third-party application providing authorization

User Applications Security (Cont'd)

Confidentiality for user applications can be achieved by:

- SSL/TLS Sockets
  - When an application is communicating over a network
- Self-written functions
  - Functions or programs written by you or a third-party application. Can use the crytographic coprocessor

Integrity for user applications can be achieved by:

- Object signing
- Self-written functions
  - Functions or programs written by you or a third party application providing integrity
  - Applications can use the object signing and signature verification APIs

Audit/Logging

- Custom logging can be implemented in any user application

Notes User Applications Security

You or a third-party vendor can write applications that meet all of the major security goals. Whether you want to use authentication or confidentiality, standard OS/400 functions or APIs can be used to build in security into your own applications.

An advantage of the integrated environment on the iSeries server is that you can choose between a rich set of security functions and services to write an application compatible with other cross platform applications. For example, if you want to authenticate users for a self-written Sockets application and this application runs on multiple servers, you can use LDAP directories as your user registry to perform the authentication.

Another example is if you want to store certain information encrypted on your disk. You can use cryptographic services available with the 4758 Cryptographic Coprocessor to encrypt and decrypt your information.

APIs are also available to sign your own programs. You can then verify the integrity of signed objects whether they are stored on the system they were signed on or shipped to another system.

Scenario 1: Business Partner and Global Workforce Scenario

Security from BP SVR to Corp SVR1

VPN tunnel endpoint from the BP SVR terminates at the Security GW

- *Corp SVR1 is unaware of VPN tunnel; receives forwarded data from Security GW in clear text*

Security GW must support IPSec UDP Encapsulation

- *iSeries currently does not support this in a responder role*

Program objects must be distributed to BP SVR securely

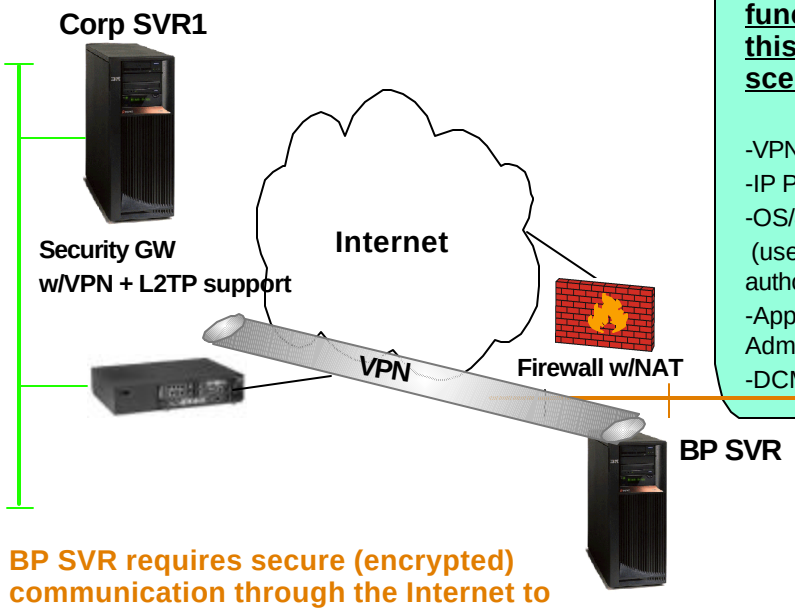
- *Solution: Objects can be signed by using DCM and then verified on the BP SVR side to guarantee the data has not been modified*

BP SVR will download objects from Corp SVR1 using FTP; access should be limited to this download only

- *Solution: Application Administration can prohibit the BP user from accessing unauthorized libraries, creating files, etc.*

Further security is required on the corporate network for Corp SVR1

- *Solution: Implement OS/400 Security and IP filtering as a second defense against unwanted internal and Internet traffic*



Security functions used in this part of the scenario

-VPN
-IP Packet Filtering
-OS/400 Security
(user profiles, object authority, etc.)
-Application Administration
-DCM (object signing)

BP SVR requires secure (encrypted) communication through the Internet to Corp SVR1 for multiple applications (ports)

- *Solution: Host to Gateway VPN*

BP SVR does not have a globally routable address

- *Solution: UDP Encapsulation (NAT-friendly IPSec)*

Further security is required by the business partner on the BP SVR

- *Solution: Implement OS/400 Security and IP filtering as a second defense against unwanted Internet traffic*

Notes Security from BP SVR to Corp SVR1

This part of scenario one, as all parts of all scenarios, must implement physical security for data and systems involved. This physical security must reflect the corporate security policies for both the Business Partner and the Corporate Office.

Another aspect of security that must be in place for every aspect of every scenario, is system-level security. In the case of the iSeries, this is OS/400 security. As discussed earlier in this presentation, there must be an aspect in your business' security policy that addresses system level security.

OS/400 Security for this part of Scenario 1 that should be implemented is:

- User profiles for business partner users should be limited in terms of special authority, if any at all
- If not necessary, BP user profiles should not have access to a command line
- Exit Programs can be used if BP users use an application that supports them
- Object level security for objects that BPusers access
- System values can be used to prohibit unauthorized access to the system
- Audit logging to detect failed login attempts, other authority failures, etc.

The primary security mechanism in this part of Scenario 1 is Virtual Private Networking (VPN). The VPN between the BP SVR system and the security gateway encrypt data before transmitting over the Internet. IPSec using UDP encapsulation will be used to protect this tunnel. UDP encapsulation is a fitting solution and a necessary one in this case because the BP SVR does not have a globally routable IP address.

In this part of the scenario, BP SVR is required to download program objects from Corp SVR1. To verify the integrity of these objects, the Corp SVR1 system signs the objects using Object Signing through Digital Certificate Manager (DCM). These objects are then saved to a SAVF on Corp SVR1. BP SVR1 is allowed FTP access into Corp SVR1 to download these objects.

Application Administration should set restrictions for the BP user (who is retrieving this file) to only have access to logon to the FTP server and to receive and list files. The user will be prohibited from changing directory, creating libraries/directories, deleting files, renaming files, sending files, and issuing CL commands through QUOTE RCMD.

Notes Security from BP SVR to Corp SVR1

IP Packet Filtering can serve as a second line of defense (since the security GW should be handling the primary line of defense) to prevent intruders from accessing the iSeries' resources. Also packet filtering should be configured so that only necessary internal systems can access the iSeries only for their specific applications. This is true for both the Corp SVR1 and BP SVR.

IP Packet Filtering configuration on the Corp SVR1 iSeries:

- For internal users, only allow necessary traffic in and out of the iSeries
- Only allow BP SVR to access necessary applications (FTP for downloading program objects, etc.)
- All other traffic will be denied

IP Packet Filtering configuration on the BP SVR iSeries:

- For internal users, only allow necessary traffic in and out of the iSeries
- Do not allow connections initiated from Corp SVR1
- All other traffic will be denied

Security Goals achieved in this part of Scenario 1:

- Authentication
  - OS/400 Security (user profiles, system values)
  - VPN using RSA Signature Mode (Digital Certificates)
- Authorization
  - Application Administration (FTP)
  - IP Filtering
  - OS/400 Security (object level security)
- Integrity
  - Object Signing
  - VPN
- Confidentiality
  - VPN

Remote Access to Corporate Network

Traveling Sales Force requires full access to the Corporate Network

- ***Solution: Use L2TP/VPN for these users***

VPN and L2TP tunnel endpoints for these users terminate at the Security GW

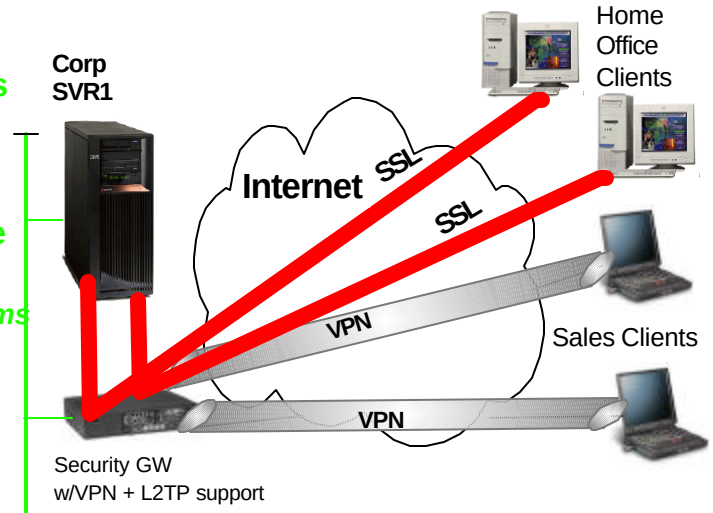
- ***These clients then appear to other systems on the network as if they are residing on the corporate network***

Home Office client requires only Host on Demand (HoD) access to Corp SVR1 iSeries

- ***Solution: Use SSL for these users (with client authentication)***

Further security is required on the corporate network for Corp SVR1 iSeries

- ***Solution: Implement OS/400 Security and IP filtering as a second defense against unwanted internal and Internet traffic***



Security functions used in this part of the scenario

-VPN
-L2TP
- SSL/TLS
-IP Packet Filtering
-OS/400 Security
(user profiles, object authority, etc.)
- Telnet server SSL with client authentication

Notes Remote Access to Corporate Network

This part of Scenario 1 shows the types of security that can be used for users working for your company who require remote access to the corporate network.

Both the traveling sales people and the home office user require secure (data confidentiality) access to the corporate network. Logically, VPN and SSL are the solutions. In the case of the traveling sales people, we use an L2TP tunnel protected by IPSec to extend the corporate network addressing scheme to these users. L2TP allows these users to have a private IP address that makes them appear to the rest of the network as if they are locally connected to the corporate network. IPSec provides the confidentiality needed for the data that is being transmitted over the Internet.

The home office remote user will access confidential information. Some form of encryption is needed in this case to hide the data while it is being transmitted over the Internet. Telnet should be enabled for SSL to use Host on Demand for these users. Since these users only need access to one application (HoD), VPN is not necessary.

As in any security scenario, system-level security should be used on the iSeries. The same OS/400 security mechanisms mentioned in the first part of this scenario should be used in this part of the scenario as well.

IP Packet Filtering configuration on the iSeries:

- For internal users, only allow necessary traffic in and out of the iSeries
- For external Home Office Clients, only allow traffic in and out of the iSeries over port 992 (SSL Telnet for HoD) and port 8999 (HOD Service Manager Port). **Note:** At V5.0 of WebSphere HOD, this port can be changed to port 80 or 443 so that additional ports do not need to opened on the firewall.
- Traveling sales clients will appear to be local to the iSeries and will have the same access to the iSeries as the rest of the internal traffic
- All other traffic will be denied

Notes Remote Access to Corporate Network

Security goals achieved in this part of Scenario 1:

- Authentication
  - OS/400 Security (user profiles, system values)
  - User ID/password for L2TP
  - Digital Certificates for VPN
- Authorization
  - IP Filtering
  - OS/400 Security (object level security)
- Integrity
  - VPN
  - SSL/TLS
- Confidentiality
  - VPN
  - SSL/TLS

Scenario 2: HTTP Server/Sofware Vendor Scenario

Apache Web Server Security

Require high availability for Web servers

- *Solution: Configure WebSphere Edge Server for load balancing and high availability for Web servers A & B*

Web servers should have some defense against attacks from the Internet

- *Solution: Configure Apache servers with Denial of Service (DoS)*

Web servers must be able to send pages confidentially over the Internet

- *Solution: SSL for Apache*

Web users must be authenticated to the Web servers

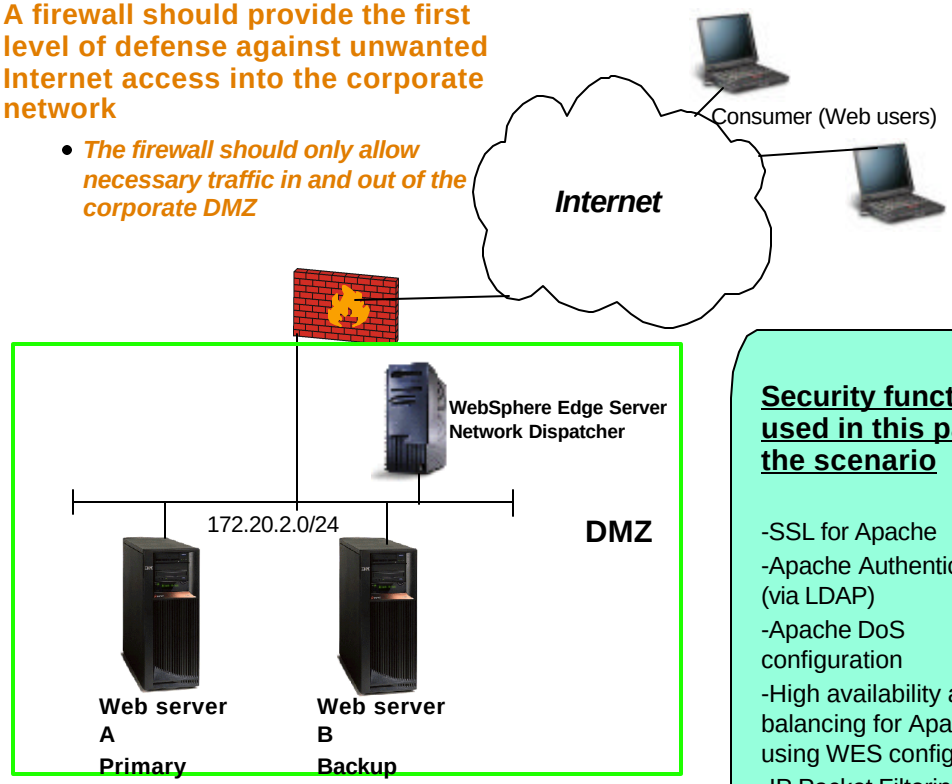
- *Solution: Authentication via LDAP*

Further security is required on the corporate network for the Web Servers

- *Solution: Implement OS/400 Security and IP filtering as a second defense against unwanted internal and Internet traffic*

A firewall should provide the first level of defense against unwanted Internet access into the corporate network

- *The firewall should only allow necessary traffic in and out of the corporate DMZ*

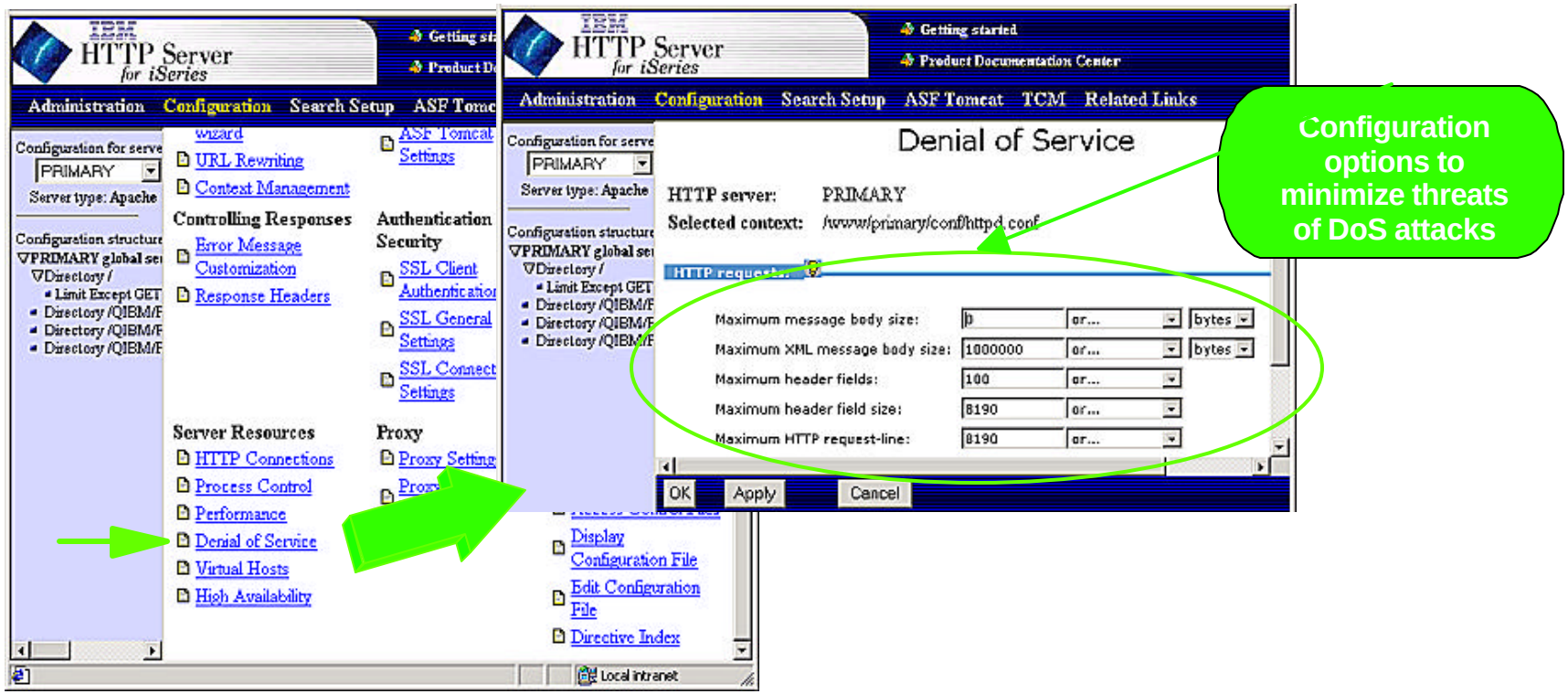


Security functions used in this part of the scenario

-SSL for Apache
-Apache Authentication (via LDAP)
-Apache DoS configuration
-High availability and load balancing for Apache using WES configuration
-IP Packet Filtering
-OS/400 Security
(user profiles, object authority, etc.)

Notes DoS Prevention for Apache

A Denial of Service attack is a type of attack that may slow down or even completely paralyze your server. DoS prevention is a necessary component of network security for Web servers. For the Apache server, there are options that can be configured to minimize the possibilities of DoS happening to your Web servers. These same options should be configured on both Apache servers.



Notes DoS Prevention for Apache

The denial of service attribute is equally a performance setting as well as a security setting. This setting allows you to identify the possibility of an attack based on the data frame size. The HTTP server may identify an attack because the frame size differs from the one it expects. Although this setting impacts the server performance as each request is tracked, it allows you to prevent a more dangerous performance degradation when dealing with a type of attack that may intentionally slow down or even completely paralyze your server.

The HTTP Server (powered by Apache) includes the following attributes to prevent a denial of service attack:

- **Maximum message body size**: Allows you to limit the size of an HTTP request message body within the context the directive is given (server, per-directory, per-file or per-location). The default value is zero (0), which indicates there is no maximum size specified. The directive is LimitRequestBody.
- **Maximum XML message body size**: Allows you to limit the size of an XML-based request body. The default value is 1000000 bytes. The directive is LimitXMLRequestBody.
- **Maximum header fields**: Allows you to modify the limit on the number of request header fields allowed in an HTTP request. The default value is 100. The directive is LimitRequestFields.
- **Maximum header field size**: Allows you to limit the size for an HTTP request header field below the default size compiled with the server. The default value is 8190. The directive is LimitRequestFieldSize.

The CERT Coordination Center defines a denial of service attack as an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include:

- Attempts to "flood" a network, thereby preventing legitimate network traffic
- Attempts to disrupt connections between two machines, thereby preventing access to a service
- Attempts to prevent a particular individual from accessing a service
- Attempts to disrupt service to a specific system or person

Source: http://www.cert.org/tech_tips/denial_of_service.html

Notes Web User Authentication-LDAP

The HTTP Server for iSeries allows you to minimize user and configuration administration and management by leveraging LDAP directory services. You can use the LDAP server for authenticating Web users who want to access protected resources on your HTTP Web server. The advantage of using a centralized storage for user information is that many different applications, such as WebSphere Application Server, Lotus Domino, and HTTP Server, can use authentication information that is kept in a single directory. For example, a user needs to change their password only once and all applications that use LDAP for authenticating users will perform authentication with the changed user password. Once users are authenticated, it must be determined if they have the authorization or permission to perform the requested operation on the specific object.

Another feature of the HTTP Server for iSeries allows you to store server configuration directives in an LDAP directory. This is especially useful when operating a cluster of servers that are used for load sharing or backup purposes as in this scenario. In this case, the IT department has to maintain only one set of configuration directives that are shared by all servers.

The characteristics of the scenario include:

- The company operates an HTTP Web server (powered by Apache) on both of their iSeries servers to improve availability and load balancing. Since both Web servers serve the same information, the Apache Web server configuration is the same for both servers. So the company wants to maintain the server configuration only in a single place. This approach minimizes the administration effort and allows for easy expansion in case they want to add additional servers to the cluster. To achieve this goal, the company exploits the LDAP configuration support included with the HTTP Server for iSeries product.
- The company's technical support department wants to offer special information to their premium customers over the Web. To ensure that only premium customers have access to the information, the content is protected by the Web server. Customers would then have to authenticate to get access. That means, each customer is registered and needs a user ID and password to sign on. The operation of multiple Web servers raises another question: How can the company make sure that all Web servers have access to the user authentication data without replicating or copying the information to all Web servers? Well, the answer is easy. The IT department registers all customers in the iSeries LDAP directory. Then they modify the centrally stored Web server configuration to authenticate Internet users via user information stored in the LDAP directory.

Notes Apache Web Server Security Goals

Security goals achieved in this part of Scenario 2:

- Authentication
  - Web clients are authenticated to the Apache Servers via entries in an LDAP directory
- Authorization
  - IP Filtering
  - OS/400 Security (object level security)
  - Web clients' requests are performed under a specific user profile. This profile must be authorized to access the requested objects.
- Integrity
  - SSL/TLS
- Confidentiality
  - SSL/TLS

Software Vendor Security Configuration

Remote iSeries shop users must be authenticated to iSeries Web Server on the software vendor's DMZ

- *Use OS/400 user profiles*

Remote iSeries shop users must be authorized to directories on an iSeries Web server

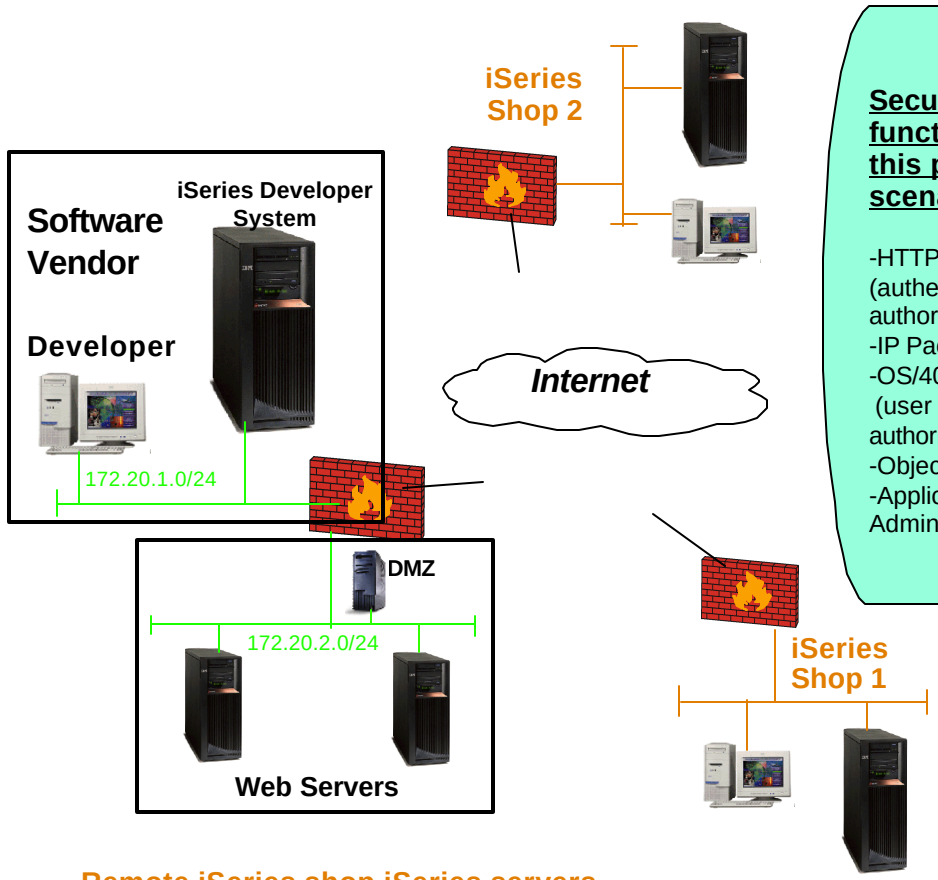
- *Use protection methods in http configuration*
- *Use OS/400 user profiles in conjunction with object level security*

Object signing should be used for the software being distributed (for integrity)

- Two methods of distribution
  - Transfer program objects via e-mail
  - Transfer program objects via FTP (use iSeries Navigator's Application Administration for command authorization)

Further security is required on the software vendor network for the iSeries

- *Solution: Implement OS/400 Security and IP filtering as a second defense against unwanted internal and Internet traffic*



Remote iSeries shop iSeries servers should also implement security

- *Solution: Implement OS/400 Security and IP filtering*

Security functions used in this part of the scenario

-HTTP security (authentication and authorization)
-IP Packet Filtering
-OS/400 Security (user profiles, object authority, etc.)
-Object Signing
-Application Administration

Notes Software Vendor Security Configuration

In this scenario, the remote iSeries shops require two functions from the software vendor:

- That program objects be distributed to them
- That the appropriate users in these shops can access data from the software vendor's Web server for information relating to the vendor's product

The software is distributed to the remote iSeries shops via e-mail. This can be done while maintaining the integrity of the program objects being sent by using object signing on the Software Vendor's developer system.

- The objects are signed by the iSeries developer system and placed into a save file on the iSeries.
- This save file is downloaded to the developer's PC via FTP.
- The save file is then attached to an e-mail and sent to the remote iSeries shops.
- The remote iSeries shops then detach the save file and FTP it up to their local iSeries.
- The objects are restored on the iSeries shop local system.
- The restored objects' integrity is checked at that time as long as the QVFYOBJRST system value is set at 2 or higher.

Note that the remote iSeries shops could also retrieve the software (save files) via FTP. The save files need to be transferred from the iSeries Developer system to the iSeries Web server so that the remote iSeries shops could access the files through the Internet via FTP (as in Scenario 1). The iSeries Web server should use iSeries Navigator's Application Administration to allow only these users to have access to login to the FTP server, and even then, limit the amount of access to the system (for example, the GET subcommand is allowed for this group of users, the PUT subcommand is not allowed, etc.) Also, if the addresses of these remote iSeries shops is static, IP filtering could be used to only allow FTP access from these IP addresses. This would further secure the iSeries Web server.

When the signed object are sent, a copy of the certificate that signed the object must be included. The iSeries developer system does this by using DCM to export the object signing certificate (without the certificate's private key) as a signature verification certificate. Before the signature can be validated on the objects by the remote iSeries shops, a copy of this certificate that signed the object must be received. The remote iSeries shops must also obtain a copy of the Certificate Authority certificate for the CA that issued the certificate that signed the object.

In this scenario, the remote users should be authenticated via user profiles when attempting to access the HTTP server. This provides security by:

- Authenticating the users (users must sign in with an iSeries user profile and password)
- Authorizing these authenticated users to only necessary resources

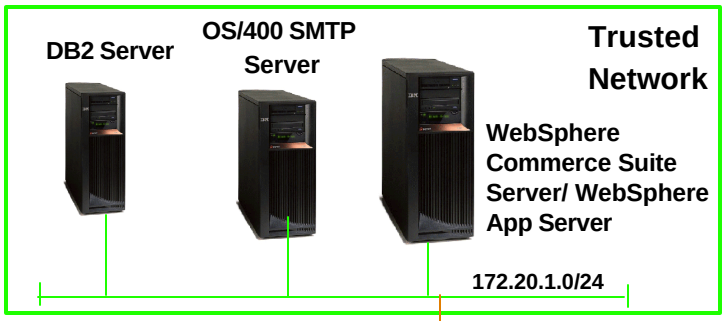
Notes Software Vendor Security Configuration

Security goals achieved in this part of Scenario 2:

- Authentication
  - Web clients are authenticated to the HTTP Server via OS/400 user profiles
- Authorization
  - IP Filtering
  - Protect directives in HTTP configuration
  - OS/400 Security (object level security)
  - Application Administration (for FTP)
- Integrity
  - Object signing
- Confidentiality
  - None

Scenario 3: E-commerce in a Multi-tier Environment Scenario

Security for Multi-tier Environment



Web Users must be authenticated

- *Solution: LDAP*

Consumer (Web users)

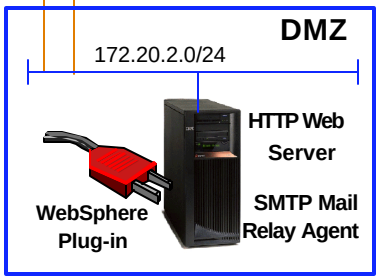
A

Internet

- *Firewall A*
  - Allow http and https server traffic
  - Allow other necessary Internet traffic (such as smtp)

B

- *Firewall B*
  - Allow remote OSE traffic (IPSec) between Web server and WAS
  - Allow SMTP traffic between SMTP Server and Mail Relay Agent



Confidentiality is needed for Web user transactions over the Internet

- *Solution: SSL/TLS*

Confidentiality is needed for communication between the WebSphere plug-in on the HTTP Server (on DMZ) and the WAS (on the trusted network)

- *Solution: Host-to-host VPN*
  - *Remote OSE cannot use SSL/TLS in WebSphere V4.0*
  - *Use digital certificates for authentication of the two hosts*

Further security is required for both iSeries' on the DMZ as well as the one on the trusted network

- *Solution: Implement OS/400 Security and IP filtering as a second defense against unwanted internal and Internet traffic*

Security functions used in this part of the scenario

-SSL/TLS
-VPN
-LDAP Authentication
-SMTP Security
-IP Packet Filtering
-Digital Certificates
-OS/400 Security (user profiles, object authority, etc.)
-- WebSphere Security Center/Roles

E-mail must be secured

- *OS/400 Mail Relay Agent will only relay mail from OS/400 SMTP Server on the trusted network*
- *OS/400 Mail Relay Agent is registered mail server for company's mail*
- *OS/400 SMTP Server will only accept mail from the Mail Relay (using IP filtering and SMTP Relay restrictions)*
- *SMTP filters and Real-time Blackhole List Servers will filter unwanted mail and prohibit unwanted SMTP connections to the Mail Relay Agent*

Notes Security for Multi-tier Environment

Open Servlet Engine (OSE) is a lightweight communication protocol developed by IBM for interprocess communication. *Remote* OSE uses this proprietary transport to route requests from the Web server plug-in to application servers on remote machines.

Usually, a WebSphere administrative server on a Web server machine generates Web server plug-in configuration files to tell the Web server how to route requests. However, the Remote OSE configuration does not place an administrative server on the Web server machine. Instead, a Remote OSE script runs on the Web server machine, communicating with an administrative server on the remote application server machine. The script gathers the necessary information about the application server configuration and generates the plug-in configuration files.

Remote OSE requires the following firewall ports to be opened:

- One port for each application server or clone process
- A port if WebSphere security is used on the machine that hosts the Web server
- A port to run the remote OSE configuration script, OSERemoteConfig *(for our case, this won't be necessary due to the use of VPN)*

For HTTP transactions that are made across the Internet, SSL/TLS is the logical choice for encryption of this data, since almost all Web browsers are already setup to do SSL/TLS.

An intermediate firewall between the HTTP server and the WCS/WAS/DB2 server provides an additional layer of security from the Internet. When transactions need to be sent from the HTTP Server's WebSphere Plug-in to the back end servers for processing, the data should still be encrypted. Since remote OSE does not support SSL (at V4.0), we must use VPN to do this encryption. The VPN configuration would be a simple host to host scenario. We chose to use digital certificates for authentication in this scenario (for additional security). However, pre-shared secrets could also be used. Using VPN requires you to open up only ports UDP 500 to 500 (and the protocols for IPSec) on Firewall B, eliminating any opening of ports on the firewall for Remote OSE.

SMTP security is necessary so that other Internet systems do not use your SMTP mail router to relay e-mail. This can be prevented by specifying that only authorized systems (IP addresses) can use the Mail Router. This configuration can be done via CHGSMTPA and ADDSMTPLE or through the "Relay Restrictions" found in the SMTP server Properties in Operations Navigator. In this scenario, the OS/400 on the DMZ is the registered mail server for this company's domain. Through IP packet filtering, the SMTP Server on the trusted network can be configured so that it only accepts mail from the Mail Router on the DMZ. Also, the SMTP Server on the trusted network should be configured to only relay mail from its internal network(s) (172.20.1.0/24 in this scenario.) SMTP filters can also discard mail based on originator's address, subject, etc. This is configured in the "Filters" tab in the SMTP server Properties in Operations Navigator. SMTP Blackhole Lists can allow you to reject connections from known "spammers" by querying real-time blacklist (RBL) servers or by specifying individual IP addresses/subnets. This is configured in the "Connection Restrictions" tab in Operations Navigator. You can view a list of known spammers or report spammers by going to http://mail-abuse.org/rbl

Notes Security for Multi-tier Environment

Security goals achieved in Scenario 3:

- Authentication
  - Web users authenticated via LDAP
  - Digital Certificates for host to host VPN connection between HTTP Server and WCS/WAS/DB2 Server
  - WebSphere Security Center - authentication via LDAP
  - WebSphere Commerce Suite - authentication via LDAP
- Authorization
  - IP Filtering
  - OS/400 Security (object level security and user profiles)
  - SMTP mail relay agent configuration
    - SMTP Filters
    - SMTP Blackhole Lists
  - WebSphere Security Center/ application deployment (roles, resources protection)
- Integrity
  - SSL/TLS for traffic from Internet to HTTP Server
  - VPN for traffic from HTTP Server on DMZ to WCS/WAS/DB2 Server on Trusted Network
- Confidentiality
  - SSL/TLS for traffic from Internet to HTTP Server
  - VPN for traffic from HTTP Server on DMZ to WCS/WAS/DB2 Server on Trusted Network

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
AS/400 Internet Security: Implementing AS/400 Virtual Private Networks	SG24-5404-00
OS/400 V5R1 Virtual Private Networks: Remote Access to the IBM iSeries Server with Windows 2000 VPN Clients	REDP0153
IBM iSeries Wired Network Security: OS/400 V5R1 DCM and Cryptographic Enhancements	SG24-6168-00
AS/400 Internet Security: Developing a Digital Certificate Infrastructure	SG24-5659-00
Tips and Tools for Securing Your iSeries	SC41-5300-05
AS/400 Internet Security Scenarios: A Practical Approach	SG24-5954-00
Lotus Notes and Domino R5.0 Security Infrastructure Revealed	SG24-5341-00
Implementation and Practical Use of LDAP on the IBM iSeries Server	SG24-6193-00
IBM WebSphere V4.0 Advanced Edition Handbook	SG24-6176-00
IBM WebSphere V4.0 Advanced Edition Security	SG24-6520-00

Related Publications - Continued

Other Publications

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

Title	Publication Number
HTTP Server (powered by Apache): An Integrated Solution for IBM iSeries Servers	SG24-6716-00
WebSphere Commerce Suite V5.1 Handbook	SG24-6167-00
WebSphere Commerce Suite V5.1 for iSeries Implementation and Deployment Guide	REDP0159
WebSphere Edge Server: Working with Web Traffic Express and Network Dispatcher	SG24-6172-00
WebSphere Edge Server *online*	http://www-4.ibm.com/software/webservers/edgeserver/
iSeries Information Center *online*	http://publib.boulder.ibm.com/html/as400/infocenter.html
iSeries Security Advisor *online*	http://www.redbooks.ibm.com/tstudio/secure1/advisor/secwiz.htm

Additional Material

IBM WebSphere Application Server

	Confidentiality	Integrity	Authentication	Authorization	Audit/Logging
User Profiles			X		X*
LDAP			X		X*
Digital Certificates			X		X*
SSL/TLS	X	X	X		X*
Security Center			X	X	
WSAD				X	X*

* Written into the application

WSAD = WebSphere Studio Application Developer

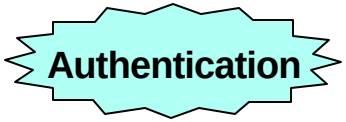
The IBM WebSphere Application Server is the premier Java technology-based Web application server

It integrates enterprise data and transactions with the e-business world

WebSphere. software

WebSphere Authentication

LDAP authentication can be provided in two ways:



- Password based
  - WebSphere uses the user and password sent by the client to attempt to find a match in the LDAP directory
- Certificate based
  - WebSphere works with the LDAP server to perform a credential mapping of the client's certificate to the contents of the LDAP directory

User Profiles

- WebSphere can use native OS users/passwords to authenticate connecting clients

Notes WebSphere Authentication

Operating systems support Basic and Form-based authentication, whereas LDAP and Custom user registries support both password-(basic, form) and certificate-based authentication. LTPA is not supported by OS registries.

WebSphere supports the LDAP LTPA authentication mechanism. Note the following important facts about LDAP authentication:

- This is not available for WebSphere Single Server Edition.
- The user should not be a root DN or administrator DN because it is unnecessary to expose the root password.
- You may want to secure the connection between the application server and LDAP using SSL.
- LDAP authentication can be set to use one of the following authentication mechanisms:
  - Password based Authentication
  - Client Certificate Authentication

The native OS authentication mechanism uses native OS/400 routines (OS/400 user profiles) to authenticate the user. Native OS is easier to configure than LTPA, but can be used for only the simplest topologies. Note that authenticating through the native OS mechanism does not log the user onto the iSeries server. Even though user profiles and passwords are used for authentication, no jobs or threads are executed under the users' profiles.

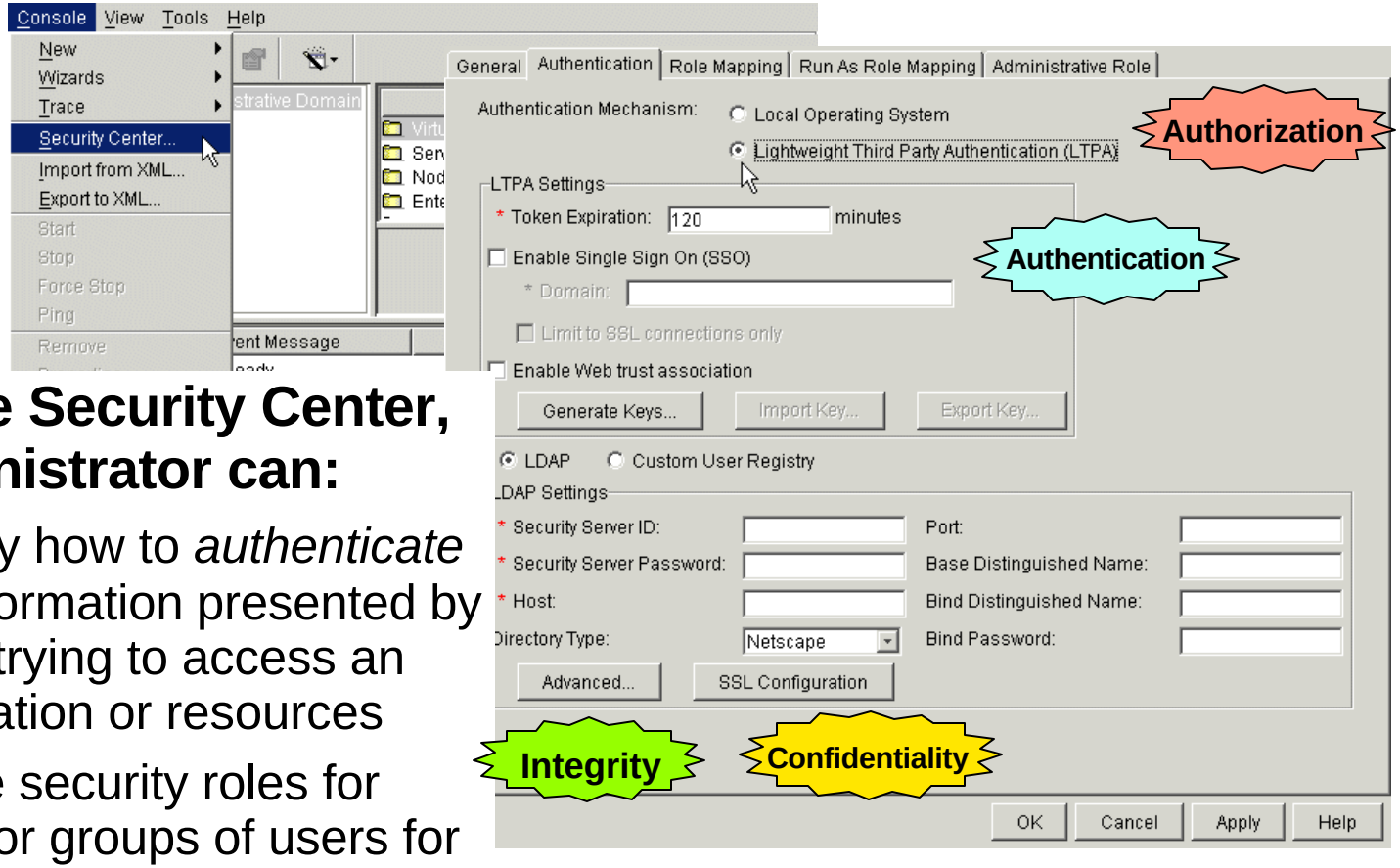
A challenge type specifies how a server will challenge and retrieve authentication data from a user. The choices for challenge type are:

- **None:** The user is not challenged for authentication data. If the requested resource is protected, then the user will not be served the resource.
- **Basic:** The user is challenged for a user ID and password.
- **Custom:** Applicable only to Web clients. The custom challenge type is used when one wants to configure the server to use a customized HTML form to retrieve the user ID and password.
- **Certificate:** Applicable only to Web clients. The user is required to present a digital certificate (X.509) to establish the connection. With the certificate challenge type, the Web server is trusted to authenticate the user through the SSL exchange. Then for authorization purposes, the WebSphere security infrastructure identifies the principal by extracting information from the certificate and mapping it to an entry in the user registry.

A user registry is where the user and group information is stored. It contains a mapping of principals to authentication information and privilege attributes such as access ID, password and group IDs. A "principal" is a representation of a human user or system entity such as a server process. The choices for user registry are:

- Native OS - OS/400 profiles
- Lightweight Directory Access Protocol (LDAP)

Security Center



Using the Security Center, the administrator can:

- Specify how to *authenticate* the information presented by users trying to access an application or resources
- Create security roles for users or groups of users for *authorization* purposes

WebSphere. software

Lotus Domino

	Confidentiality	Integrity	Authentication	Authorization	Audit/Logging
User IDs within Domino Directory			X		X*
LDAP			X		X
Client Certificates			X		
ACLs				X	X
Admin Tools				X	
DB Encryption	X				X*
SSL/TLS	X	X	X	X	
Digital Signatures		X			

* Domino application dependent

Domino is the premier platform for collaborative Web applications

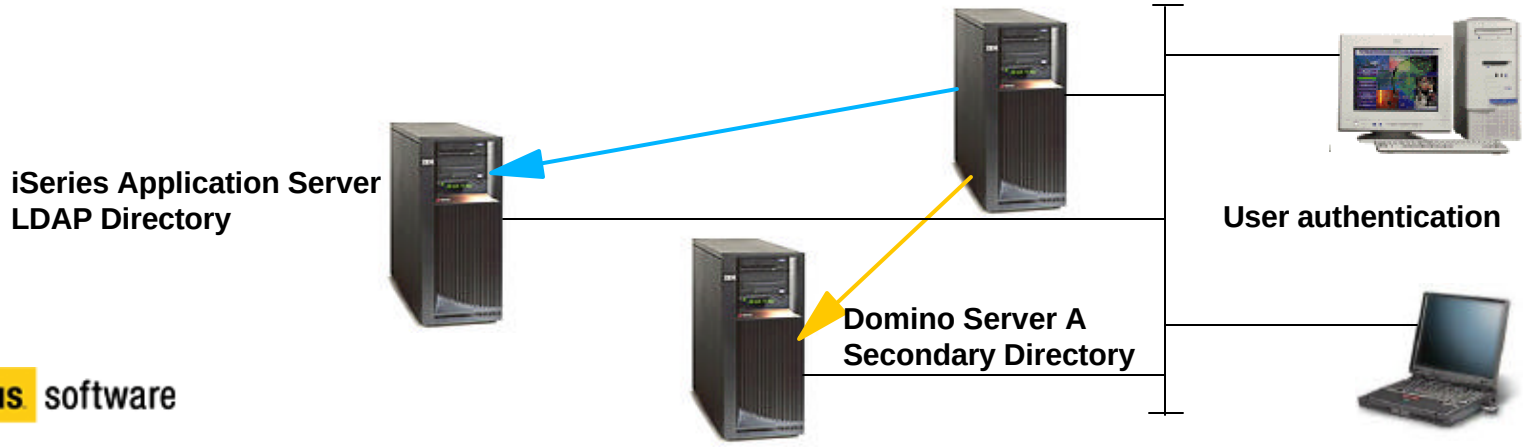
Domino's integrated application services-such as security, workflow and content management

Lotus software

Domino Authentication

Domino authentication is based on certificates or user profiles

- Domino Directory
  - A directory of information about users, servers, and groups
  - Contains documents that control directory services, manage server tasks, and define server-to-server communication
- Directory Assistance
  - Enables users to locate client information in a directory that is not the server's primary Domino Directory
    - For secondary Domino directories
    - For third-party LDAP directories



Lotus software

Notes Domino Authentication

Information about each ID is maintained on the Domino server. The Domino Directory contains a Person document for each user with a lot of information about the user including:

- The user's name and domain
  - The user ID in the Domino Directory is typically stored in the shortname attribute
- The user's public key/certificate
- An attachment with the user's ID file to be distributed the first time the user contacts the server, if it was chosen to store it here at user registration time
  - If a server has been registered, a Server document is put in the Domain Directory with similar information about the server. Certifiers are also represented in the Domino Directory by Server Certificate documents.

Directory Assistance is a feature that enables users to locate information in a directory that is not the server's primary Domino Directory. Directory Assistance also enables you to authenticate Web clients by using a directory that is not the primary Domino Directory on the server to which the clients connect. You can configure Directory Assistance for secondary Domino directories and for LDAP directories — for example, third-party LDAP directories.

To configure Directory Assistance, you create a Directory Assistance database from the template DA50.NTF. For each directory that you want to use in directory assistance, you create a Directory Assistance document that describes the entries in the directory and its use.

Include secondary Domino directories in directory assistance to:

- Use the directories to authenticate Web clients
- Allow Notes users to easily address mail to users registered in the directories
- Extend LDAP client searches to secondary Domino directories

Include LDAP directories in Directory Assistance to:

- Use the directories to authenticate Web clients that use the Domino Web service
- Use one directory to verify Web clients' membership in groups in the directory
- Refer LDAP clients that connect to a Domino LDAP service to the directories
- Allow Notes users to verify mail addresses of users in the LDAP directories

Domino Access Control Lists

Protect critical resources by limiting access to authorized and authenticated users:

- The Domino Administrator or the resource owner of the database can specify
  - Who can access the information
  - How it can be accessed
  - Under what conditions it can be accessed

- Domino provides the capability to define an access control list for every Domino database. Access control lists provide authorities that are similar to iSeries object authorities.
  - For example, editor authority lets a user change any document in a database. However, an editor cannot delete a database or give other users authority to the database.

Lotus software

Notes Domino Access Control Lists

Every database includes an Access Control List (ACL), which Domino uses to determine the level of access that users and servers have to that database. When a user opens a database, Domino classifies the user according to an access level that determines privileges. The access level for a user may vary in different databases. The access level assigned to a user determines the tasks that the user can perform in the database. The access level assigned to a server determines what information the server can replicate within a particular database. Only someone with Manager access can create or modify the ACL of a database located on a server.

Data Access Security

After users or other servers gain access to a server, they want some level of access to the data held by that server. The database layer of the Domino security model and the layers below that deal with data access, each layer providing more granular control than its predecessor. In fact, the access controls mirror the way that Domino stores and presents data:

- Database access
  - At the heart of the system lie Domino databases. The records in a database are actually documents that have usually been entered by a user or administrator. Database access control facilities, therefore, provide the broadest control over who can do what to data on a Domino server.
- Form access
  - Documents are not free-form text, but are in fact filled-in forms. When designing forms, you can use form access controls to specify who has access to the contents of a database in more detail than you can by using the database access controls.
- Document access
  - Once a form has been filled in to create a document, the owner of the document can further restrict access to it. To what degree this is allowed depends on controls within the form from which the document is created.
- Section access
  - Many documents contain data of varying sensitivity. In practice, this means that you want to prevent certain users from updating or possibly even from reading parts of the document, but you want other users to have full access. One way to achieve this is to divide the form into sections and apply section access controls to it.
- Field access
  - This is the most granular form of data access control. It allows you to control access to individual fields on a form or document. In addition to specifying user access, field access controls can limit the treatment that data receives when it is transmitted or stored.

DB Encryption & Digital Signatures

Database encryption

- Entire databases can be encrypted
- Individual fields within a database can be encrypted
  - Fields can be protected during form design from update by authors after the initial document is created. Field property security options include an option specifying that a user must have at least editor access to use the field.

Digital signatures

- Individual users can choose to sign mail messages
  - When a user adds a digital signature to a mail message, Domino uses a secure hash algorithm to generate a message digest of the data being signed and encrypts the digest with the author's private key.
  - When the receiver accesses the signed data, Domino authenticates the sender's identity and decrypts the signed data using the public key in the user's certificate.

Lotus software

Notes DB Encryption & Digital Signatures

Database designers can design fields that can be encrypted with an encryption key. To decrypt and read the document, users must have the same key. Fields may also be protected during form design from update by authors after the initial document is created. Field property security options include an option specifying that a user must have at least editor access to use the field.

A database designer controls whether fields and sections of a database are signable; individual users can choose to sign mail messages.

When databases are replicated or mail messages are routed through the network, there is the risk they could be modified. We must be able to tell if the data that was received is the same as the data that was sent. In order to detect any changes, we use digital signatures. Data integrity implies the current condition of the data is equal to the original “pure” condition. It guarantees that information is not changed in transit. A digital signature can verify that the person who originated the data is the author and that no one has tampered with the data. Originators can add their digital signature to mail messages or fields, and to sections of Notes documents.

Digital signatures use the same RSA key pair that are used in the verification and authentication process. When a user adds a digital signature to a mail message, Domino uses a secure hash algorithm to generate a message digest (or “fingerprint”) of the data being signed and encrypts the digest with the author’s private key. This signature is attached to the data.

When the receiver accesses the signed data, Domino authenticates the sender’s identity and decrypts the signed data using the public key in the user’s certificate.

Domino indicates who signed the message if decryption of the signature is successful. Otherwise, Domino indicates that it cannot verify the signature. Two things are guaranteed by this signature process: the sender is authenticated (because the digest must have been encrypted in the sender’s private key), and the message arrived unmodified (because the digests are identical). Otherwise the receiver knows the data has been tampered with or that the sender does not have a certificate trusted by the reader.