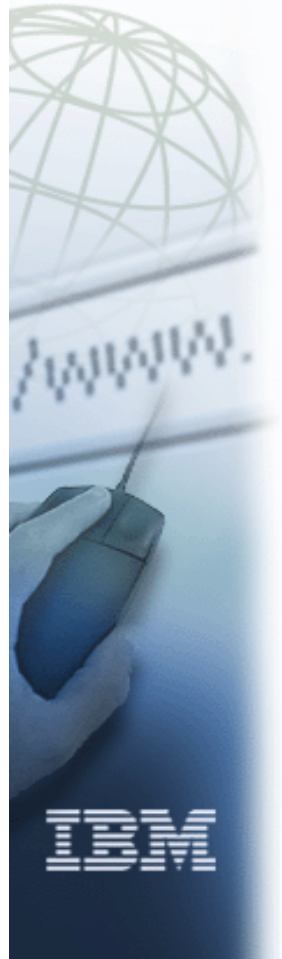


**ibm.com**



**e-business**



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

**Common , Evere 13/02/03**

**Paul Masschelein**

**paul\_masschelein@be.ibm.com**

---

© 2002 IBM Corporation

# 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

**COMPUTERWORLD**

Search



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.



(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.

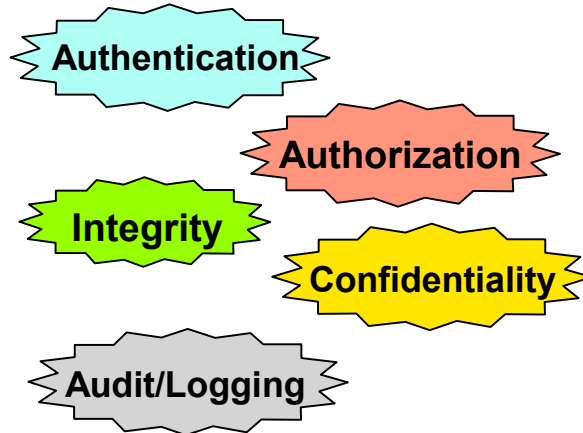
An undetermined number of employee financial records and university credit card numbers could have been exposed

© 2002 IBM Corporation

# 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/>

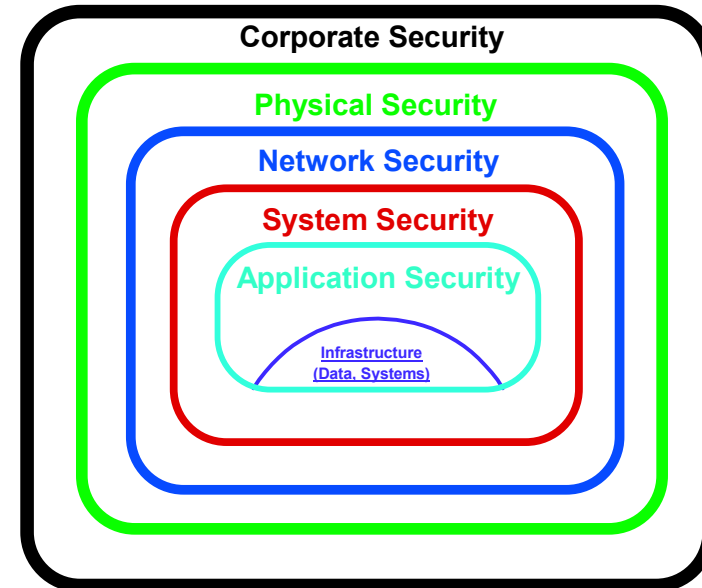
Likewise, company-developed applications or software purchased from vendors may have security weaknesses that attackers can exploit. The degree of the damage depends on the nature of the problem. The most common damage is to shut down a system. It could be more serious allowing attackers access to data 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*

# 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***

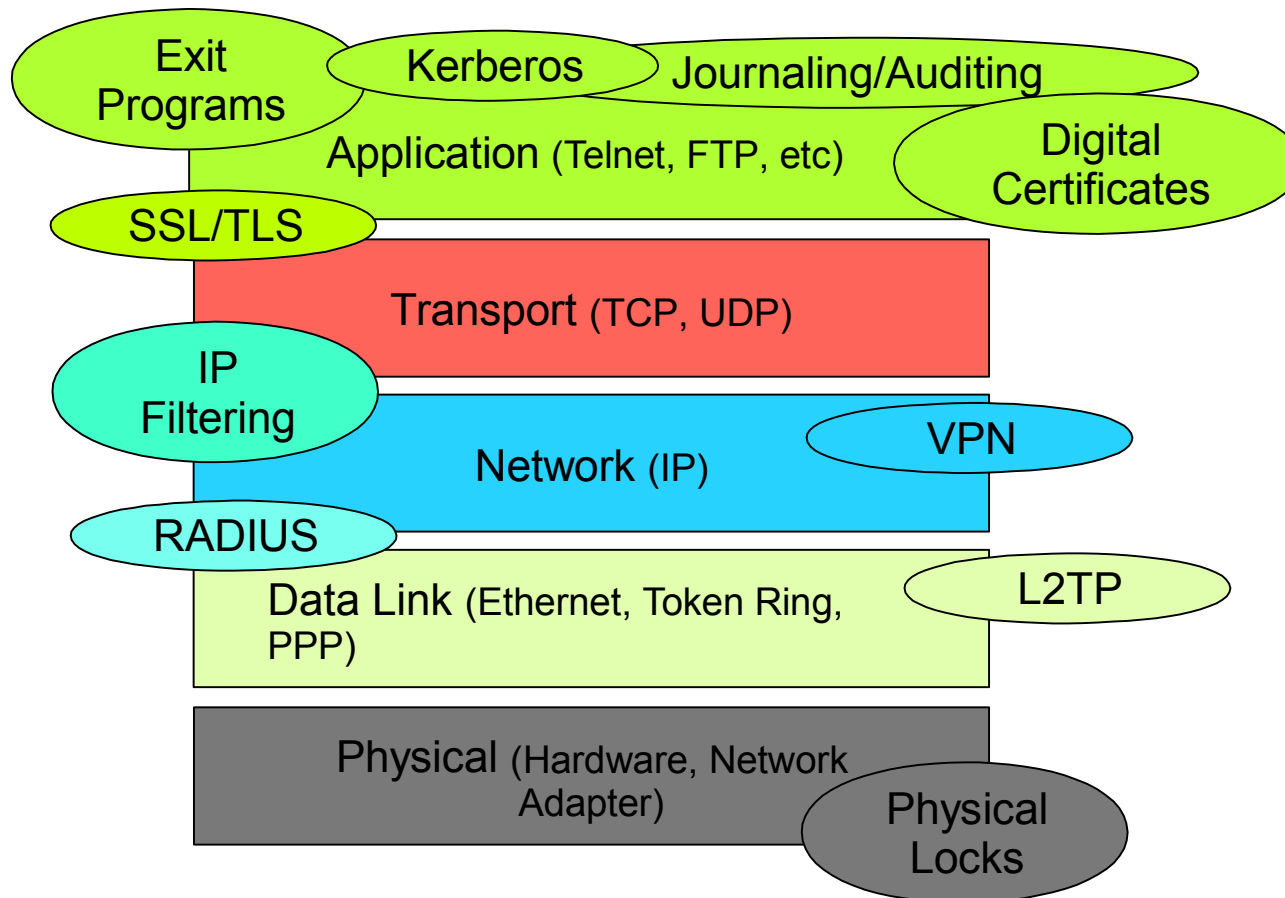


# What the iSeries Offers

# Security Services Overview



The iSeries server offers security in various layers!



© 2002 IBM Corporation

# 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 previously, 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)\*

# IP Packet Filtering



## IP Packet Filtering

- Authentication

**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

**Authorization**

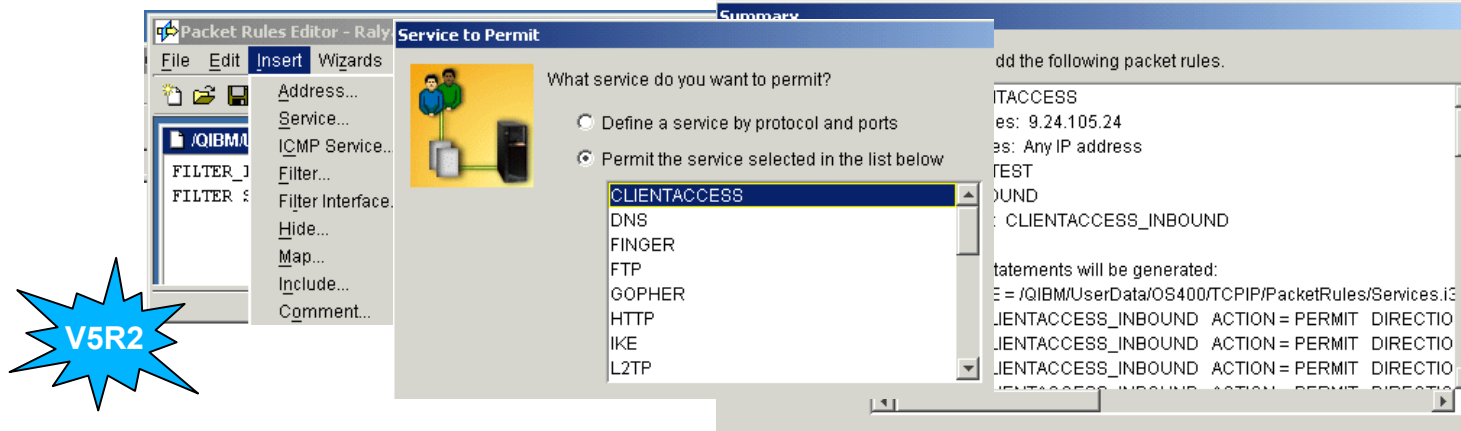
- Users are authorized only to necessary ports (applications) on the iSeries
  - For example, only allow end users to access the iSeries for Client Access (ports 8470 to 8479) and Telnet (port 23)

- Other means of security/logging

- Journaling
  - QUSRSYS/QIPFILTER journal
  - QUSRSYS/QIPNAT journal
- Auditing

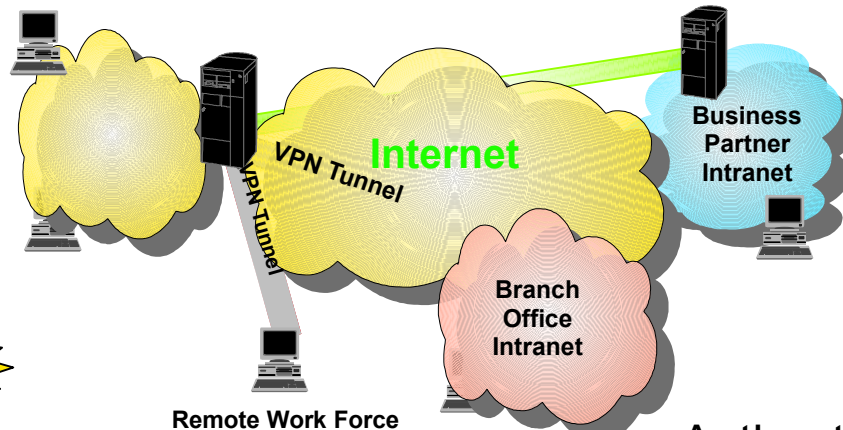
**Audit/Logging**

# 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

# 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

- 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

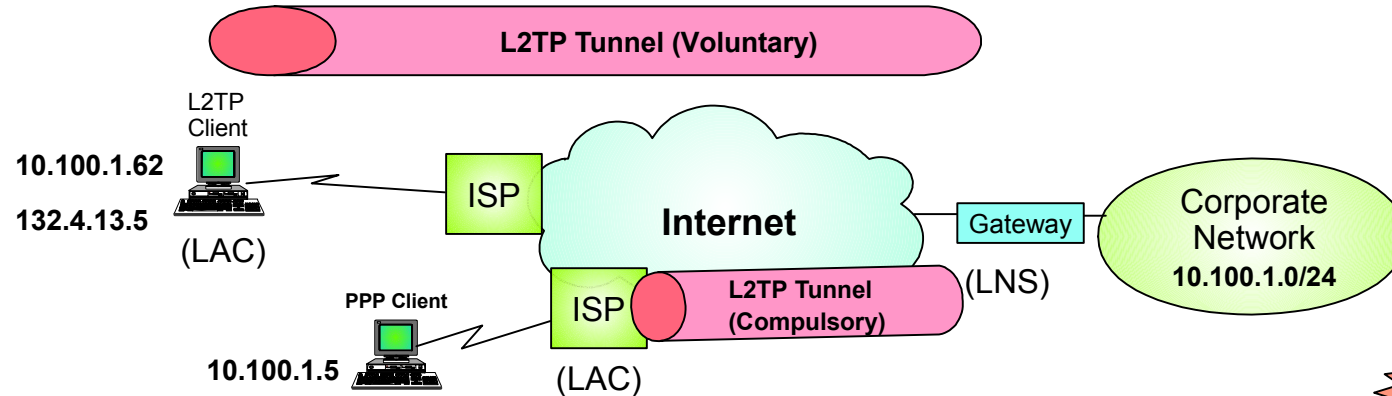
- Authorization
  - Using IP filtering within the VPN tunnel, you can authorize (permit) specific IP addresses to certain applications
  - Only communication to the defined end point in the VPN configuration will be permitted due to the IPSec anchor filter rule

## Other means of security/logging

- Journaling
  - QUSRSYS/QIPFILTER journal
  - QUSRSYS/QVPN journal
- Auditing

## Audit/Logging

# 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

**Authorization**

## • 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

**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.

# Use iSeries Navigator



**iSeries Navigator** File Edit View Help

8 minutes old

Environment: My Connections

My Connections

Name	Signed On User	Description
Another1.be.ib...		Manage this server.
Predator.hul.be...	predator	
Sf6.lahulpe.ibm...		Manage this server.
Tabasco.be.ibm...	Tabasco	
Xtreme.be.ibm....	xtreme	

Management Central (Xtreme.be.ibm.com)

- My Connections
  - Another1.be.ibm.com
  - Predator.hul.be.ibm.com
  - Sf6.lahulpe.ibm.com
  - Tabasco.be.ibm.com
  - Xtreme.be.ibm.com
    - Basic Operations
    - Work Management
    - Configuration and Service
    - Network
      - TCP/IP Configuration
      - Remote Access Services
      - Servers
      - IP Policies
        - Packet Rules
        - Virtual Private Networking
          - IP Security Policies
            - Internet Key Exchange Policies
              - Data Policies
            - Address Translation
              - Data Endpoint Pools
              - Local Service Pools
          - Secure Connections
          - Quality of Service
        - Windows Administration
        - Internet
        - IBM Network Stations
      - Security
        - Authorization Lists
        - Policies
      - Users and Groups
      - Databases
      - File Systems
      - Application Development
      - AFP Manager
      - Backup, Recovery and Media Services

My Tasks Environment tasks

1 - 5 of 5 objects

# SSL/TLS



## Secure Sockets Layer/Transport Layer Security

- Authentication

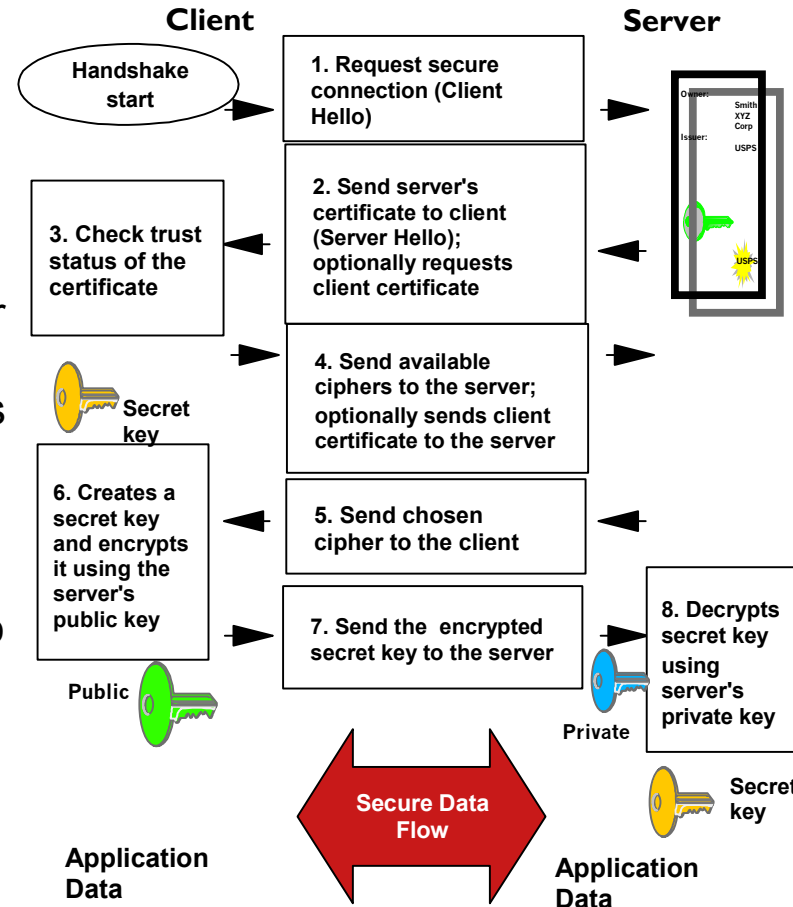


- Allows each communication partner to verify the identity of the other if required (normally the client verifies the server's identity)

- Confidentiality



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



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

**Audit/Logging**

# 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.

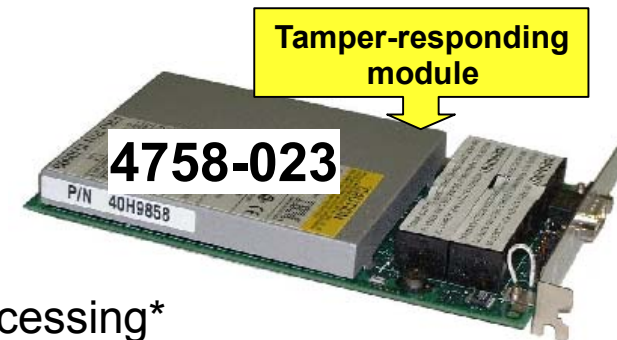
# Hardware Cryptographic Support



## 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

- 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

© 2002 IBM Corporation

# 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

© 2002 IBM Corporation

# 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 @server iSeries Security Reference*, SC41-5302.

# OS/400 Security



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

```
Work with System Security                                     System:  AS4A
Type choices, press Enter.
Allow system value security changes . . . . . 1 1=Yes, 2=No
Allow new digital certificates . . . . . 1 1=Yes, 2=No
Allow a service tools user ID with a
default and expired password to change
its own password . . . . . 2 1=Yes, 2=No
F3=Exit  F12=Cancel
```



- 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 QPDMINLEN
  - 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

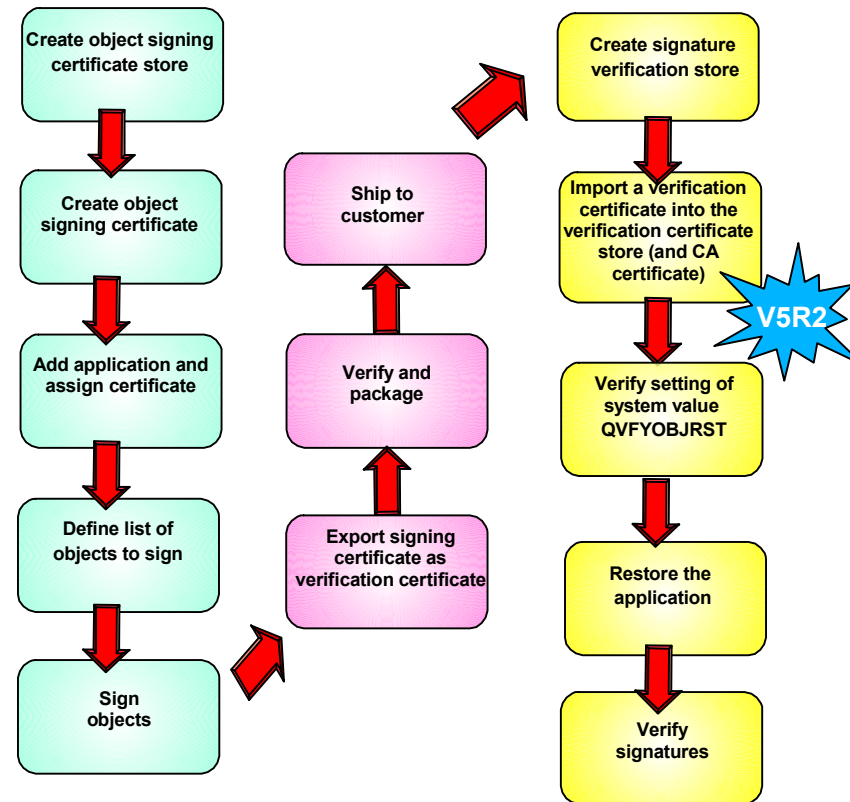


## Object signing



- Integrity

- 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



# 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 **CHKOBJTG** command can be used to check the integrity of a single object, several objects, or all objects on the system

- It not only verifies **Integrity** 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 **Audit/Logging**
  - 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.



## **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

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

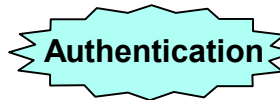
- Introduced at V5R1
- Specifies the policy for object signature verification during restore operations
- Signatures are verified when:
  - Restoring \*PGM, \*SRVPGM, \*MODULE, \*SQLPKG, \*STMF, \*CMD with attached Java programs from media or out of a save file
- Signatures are not verified when:
  - 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 encryption through 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.

# Digital certificate manager



**Digital Certificate Manager** IBM®

Select a Certificate Store

Expand All Collapse All

- ▶ [Manage User Certificates](#)
- [Create New Certificate Store](#)
- [Create a Certificate Authority \(CA\)](#)
- ▶ [Manage CRL Locations](#)
- [Manage PKIX Request Location](#)

[Return to iSeries Tasks](#)

Secure Connection

5769-NC1, 5769-NCE, 5769-SS1, 5722-SS1 (C) Copyright IBM Corporation 1997, 2002  
All rights reserved.  
US Government Users Restricted Rights -  
Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Licensed Materials - Property of IBM

Contains software from RSA Data Security, Inc.

Get Started

Local intranet

© 2002 IBM Corporation

# 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

Authentication

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

## Authorization

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

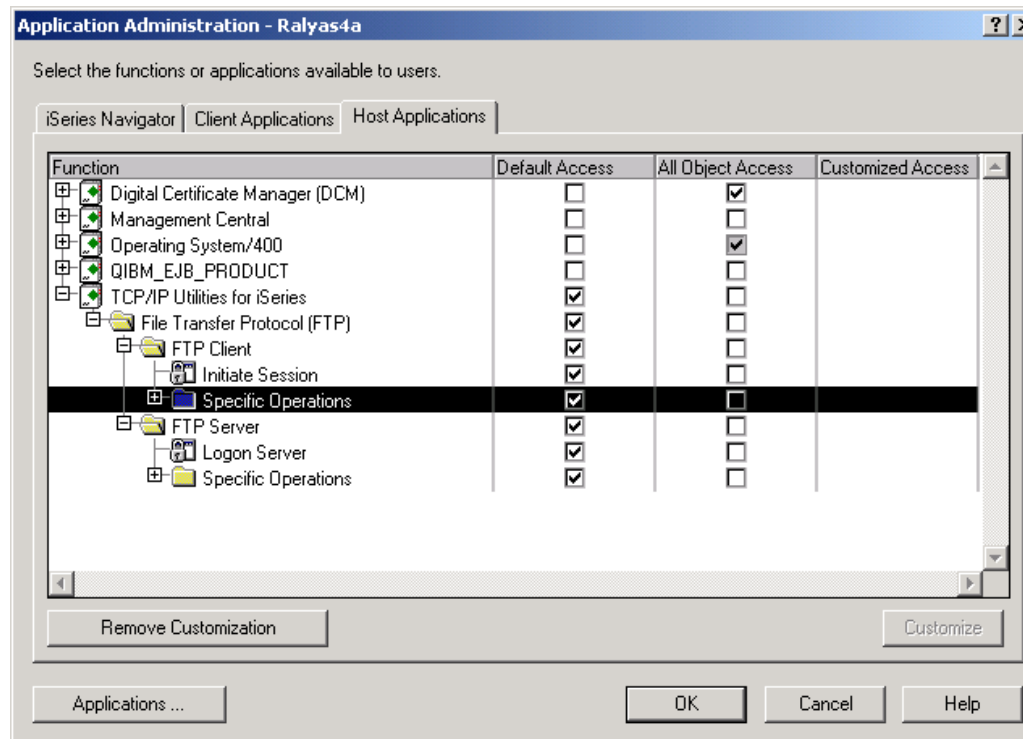
- Exit programs are excellent ways to implement custom logging facilities, for example:
  - Log all issued FTP subcommands per user
  - Keep track of signed on users and the devices/IP addresses they used

**Audit/Logging**

# 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.



© 2002 IBM Corporation

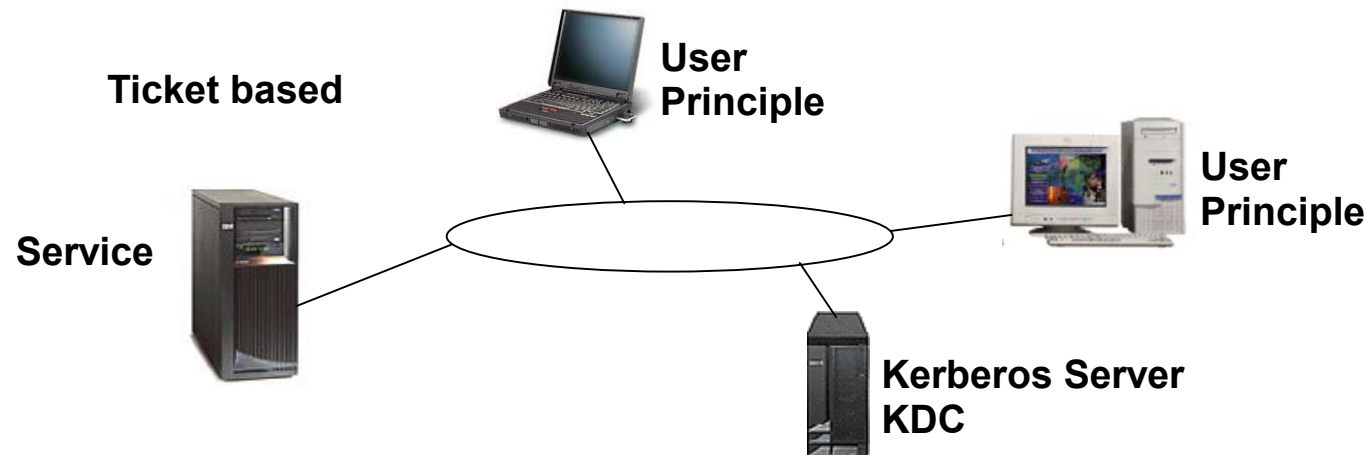
# Kerberos



## Authentication

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



© 2002 IBM Corporation

# Enterprise Identity Mapping

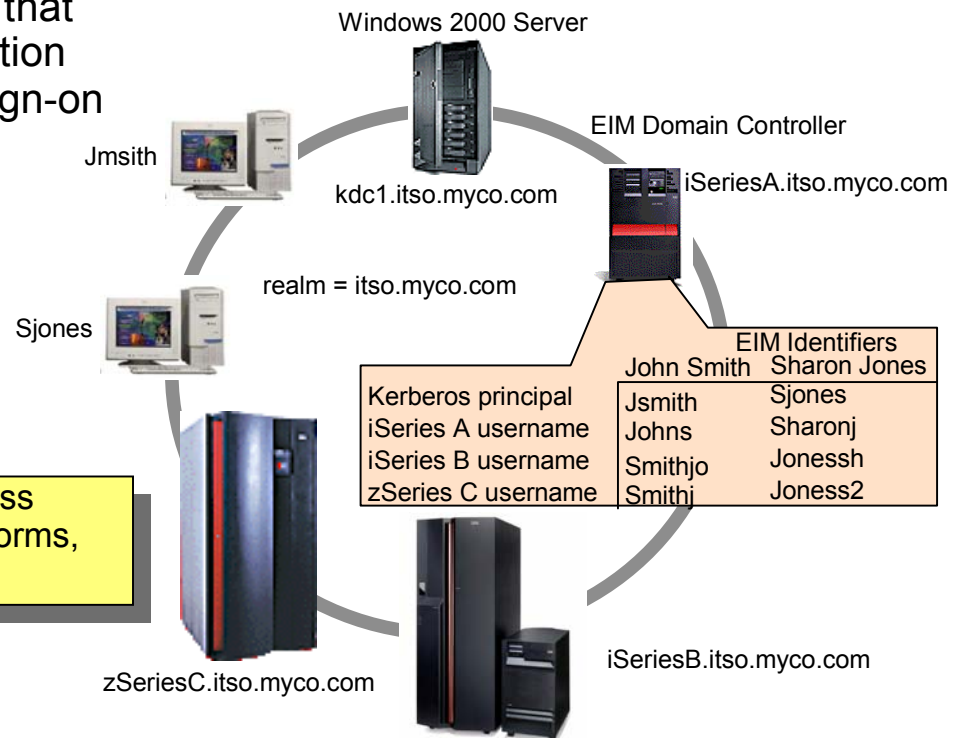
V5R2



- 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.



**Project eLiza**



# 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	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 (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

HTTP server: TESTLDAP  
Selected context: Directory /www/testldap/htdocs

Authentication name or realm: Series ITSO Corp.

User name to process requests: %%%SERVER%% or...  
(Example: QPGMR)

User authentication method to validate passwords:

- None
- Use Internet users in validation lists:
- Use user profiles
- Use user entries in LDAP server

HTTP server: TESTLDAP  
Selected context: Directory /www/testldap/htdocs

Users and groups who can access this resource:

All authenticated users (valid user name and password)

Specific users and groups:

User Name  
Example user1

Add

Group file: (path/filename) Browse

Group Name  
Example group1

Add

## 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

## SSL General Settings

HTTP server: PRODAS4A  
Selected context: /www/prodas4a/conf/httpd.conf

Enable SSL

Server certificate:

Application name:  
QIBM\_HTTP\_SERVER\_PRODAS4A or...

---

### Digital Certificate Manager

**Update Certificate Assignment**

Application type: Server

Select the application that you want to update.

	Application	Certificate Assigned
<input checked="" type="radio"/>	QIBM_HTTP_SERVER_PRODAS4A	None assigned

**Note:** Anytime you change certificate selections, you may need to end your server and start it again to have the change take effect.

# 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

© 2002 IBM Corporation

# 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)

**Authentication**

## 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

**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 cryptographic 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

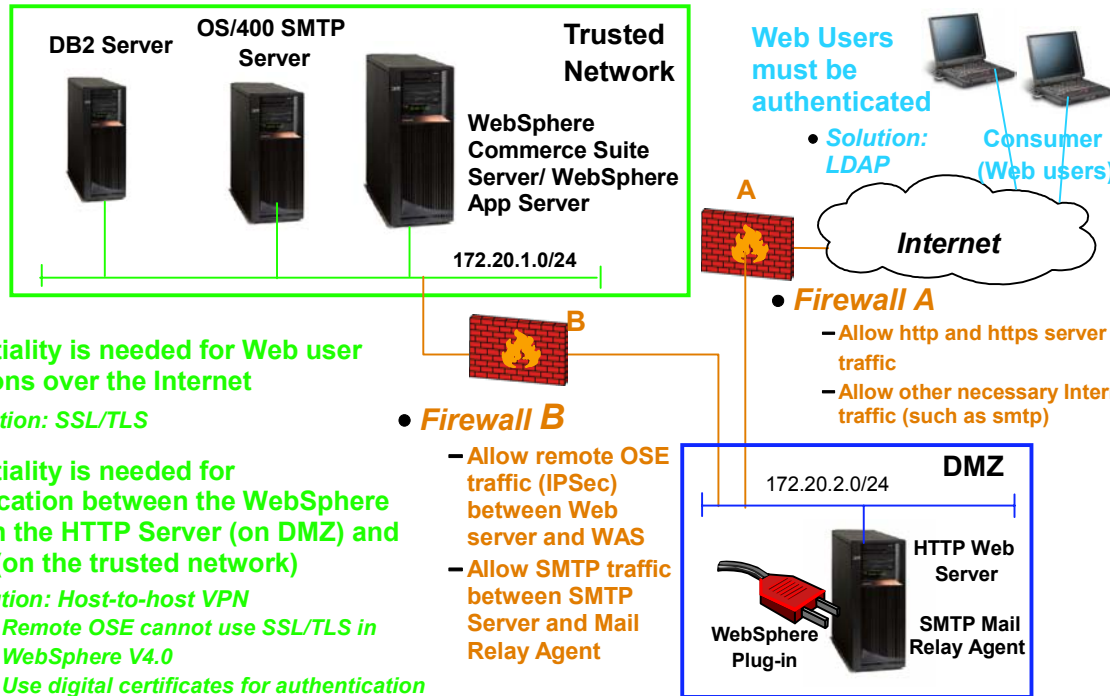


## Audit/Logging

- Custom logging can be implemented in any user application



# Security for Multi-tier Environment



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**

Web Users must be authenticated

- **Solution: LDAP**

Consumer (Web users)

• **Firewall A**

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

• **Firewall B**

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

**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

# 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

© 2002 IBM Corporation

# WebSphere Authentication



**LDAP authentication can be provided in two ways:**  Authentication

- 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)

# 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

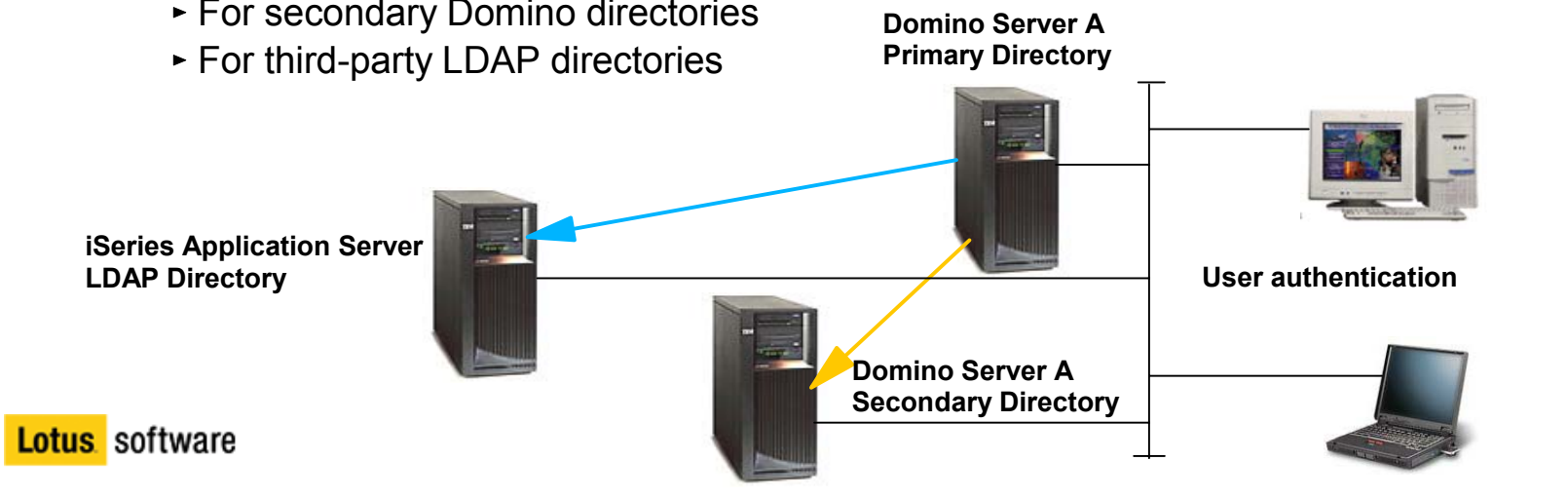
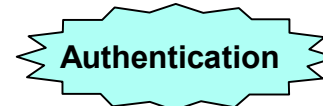
© 2002 IBM Corporation

# 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



© 2002 IBM Corporation

# 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.

