This presentation was compiled by Doug Worden, 21st Century Computer Specialists.

Agenda

- **Part 1: Introduction To Security Issues**
  - **Security Threats that need to be addressed**
  - **Layered approach to addressing security issues**

- **Part 2: iSeries Security Implementation**
  - **iSeries security basics**
  - **New and advanced security features**
  - **Detection & auditing**

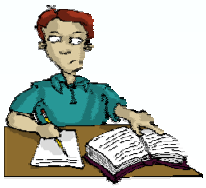
Hopefully we all learn something and maybe have some fun, too!

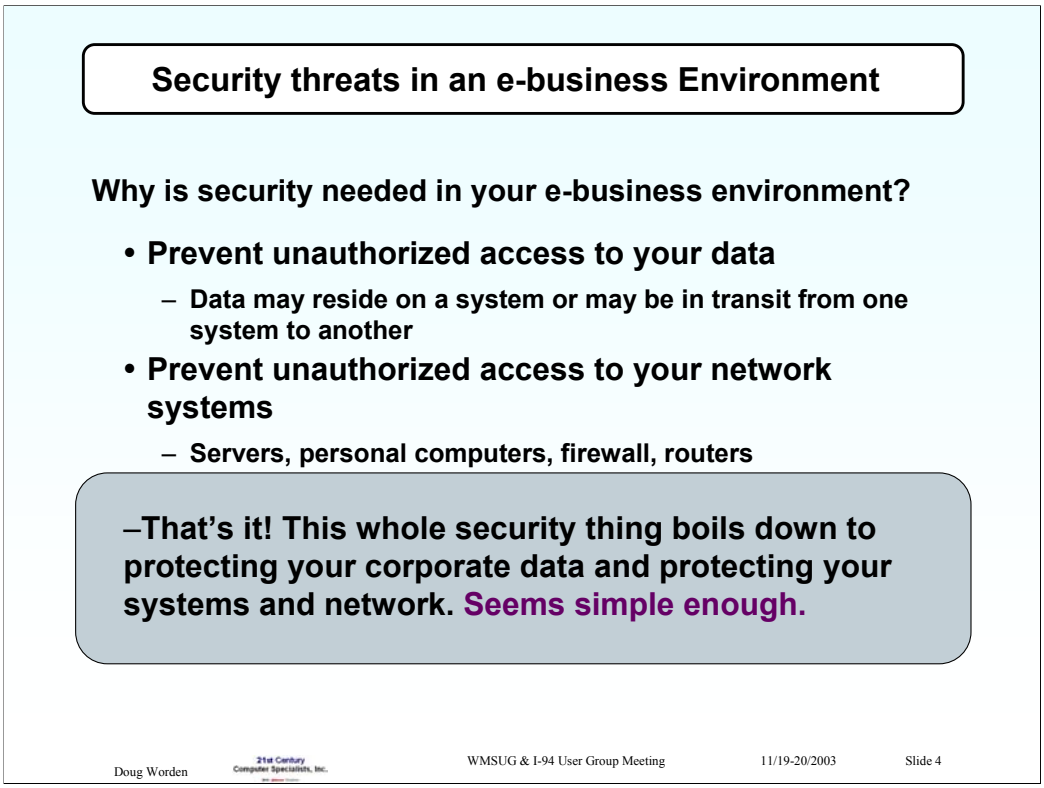
Big Disclaimer!

I have used material from whatever source I could find. This includes presentations from IBM Tech Conferences, Internet web sites, lots of iSeries manuals, etc.

There is even some of my own stuff in here.

A special thanks to everyone who contributed!

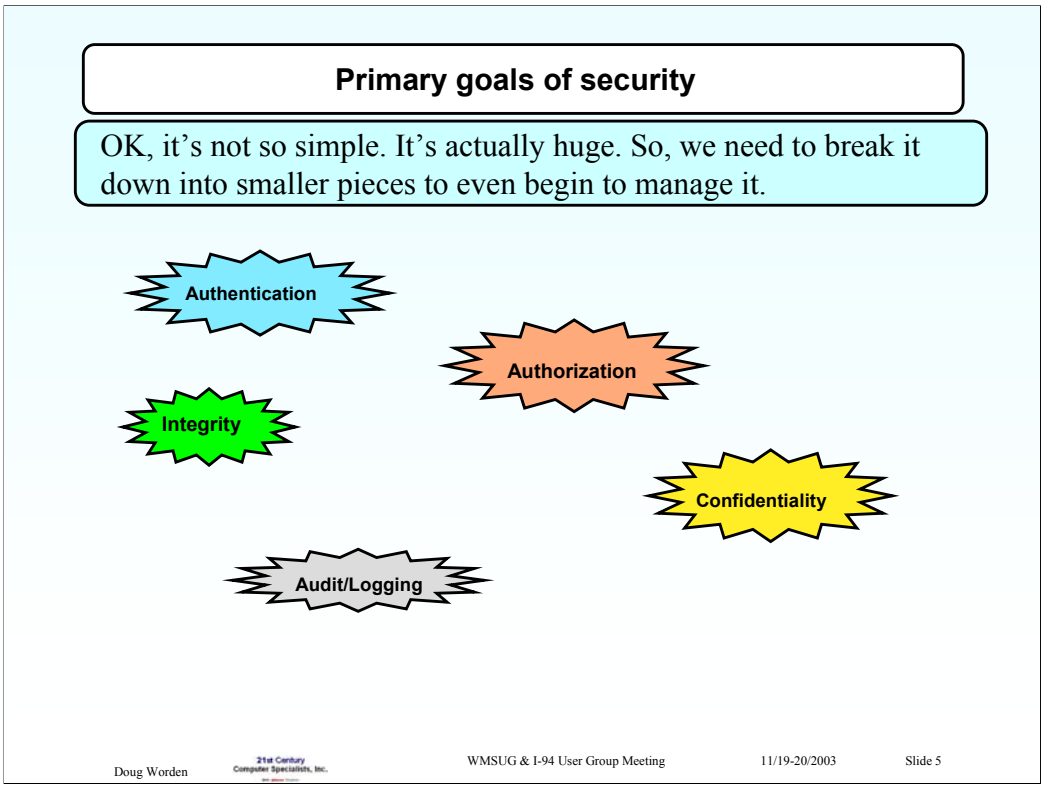




If your private network is connected to the Internet, security should be at the top of your priority list for your e-business environment. Security should be implemented to protect two entities:

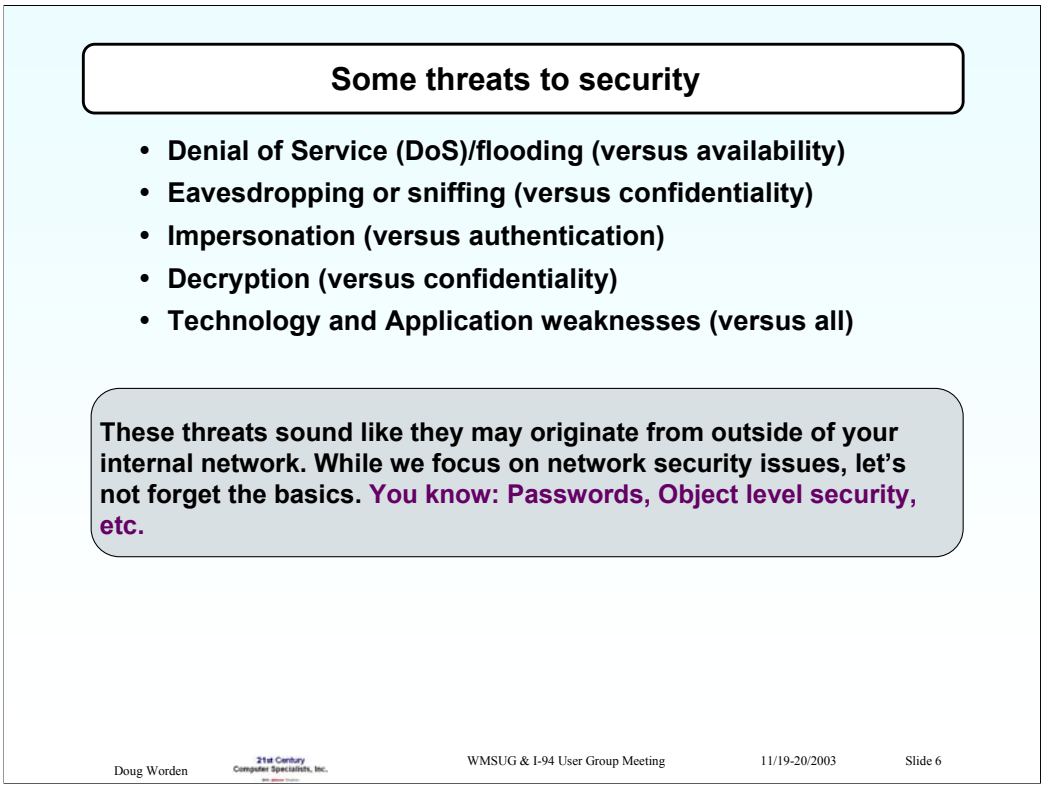
- The data that is transmitted on the network
- The computers that are connected to that network.

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



GOALS

- **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.
- Auditing/Logging: Log security violations/attacks for analysis.



THREATS

•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.

•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.

•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.

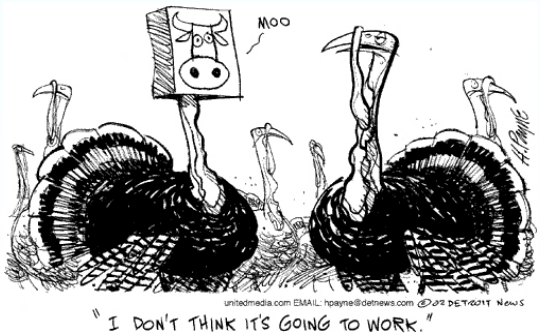
Network-based Attacks: Denial of Service attacks (DoS attacks)

- Ping of death attack
- Ping flood attack
- SYN Flood attack
- Smurf attack
- Land/Latierra attack
- Teardrop, Teardrop2/Bonk/Boink attack
- Distributed Denial of Service attack (DDoS attack)

The Ping of Death attack uses a TCP protocol stack bug in some operating systems. The size of IP datagram for ICMP Echo (ping) is less than 65536 bytes.
There are TCP protocol stack bugs in some operating systems that they cannot handle IP datagram for ICMP Echo which size is over 65535 bytes. If intruders sends IP datagram for ICMP Echo which size is over 65535 bytes, TCP protocol stack hung or total system hung happens in the target system.
To issue oversized ICMP Echo, type the following command:
Ping Flood attack is to send a large amount of ICMP echo requests to the target system. It causes operating system to slow down due to the lack of system resources because TCP stack need to handle each incoming ICMP Echo request packet.
In some operating systems, PING command has a option to specify the number of ICMP echo requests to be sent to the target system.
SYN Flood attack is to send a large amount of TCP SYN packets to the target system. Every time the target system receives a TCP SYN packet, the TCP stack
spends a system resource to create a work area for TCP connection. The target system is waiting for the ACK packet after it sent SYN ACK packet. If the target system won't receive ACK packet, it keeps reserving work areas for TCP connection. This situation causes a system slow down, a system crash,
or an inoperative service due to the lack of system resources.
Smurf attack uses a router vulnerability that the IP-Directed broadcast address relays ICMP Echo Request packet to each client under the same subnet.
Below shows a diagram of Smurf attack. Attacker creates a invalid ICMP Echo Request packet which includes a fake source IP address 172.21.0.1, then sends it to the IP-Directed broadcast address 192.168.1.255. Notice that the fake source IP address 172.21.0.1 is target's IP address.
The IP-Directed broadcast address relays it to each client; for example,192.168.1.10 and 192.168.1.20. Each client sends ICMP Echo Reply packet back to
the source IP address 172.21.0.1. This situation causes a Target system slow down, system crash, or an inoperative service due to the lack of system resources
by receiving a lot of ICMP Echo Reply packets from clients.
Land attack uses a TCP protocol stack bug in some operating systems. If an attacker sends a SYN packet which source IP address and destination IP address are
the same as the target's IP address, TCP protocol stack in target system falls into a dead loop trying to complete a TCP initial connection.
An attacker sends SYN packet which source IP address and destination IP address are same as target's IP address. Target system sends SYN ACK packet,
but it will be received by itself. Target system sends RST packet to notify to the other system to reset the TCP connection. Then target system sends SYN packet
to destination to start a TCP connection from target's side. But this packet will be received by itself. This dead loop condition causes a TCP protocol stack to freeze.
Latierra attack is similar with Land attack. Latierra attack uses the same port numbers in SYN packet for attacking.
Teardrop attack sends two IP packets which fragments are over wrapping in the data area. If the operating system cannot reassemble those IP packets, it may freeze.
Teardrop2/Bonk/Boink attack sends a malformed UDP header to the target computer. If the operating system cannot handle a malformed UDP header, it may crash.
Distributed Denial Service attack(DDoS attack) is the method to create many Daemon computers to attack the target. DDoS also makes it difficult to trace attacker back from target because Daemon computers are attacking the target computer with SYN flood, Smurf, or other Denial of Service attacks.

Spoofing attacks

- **IP spoofing attack**
- **DNS spoofing attack**
- **Web spoofing attack**



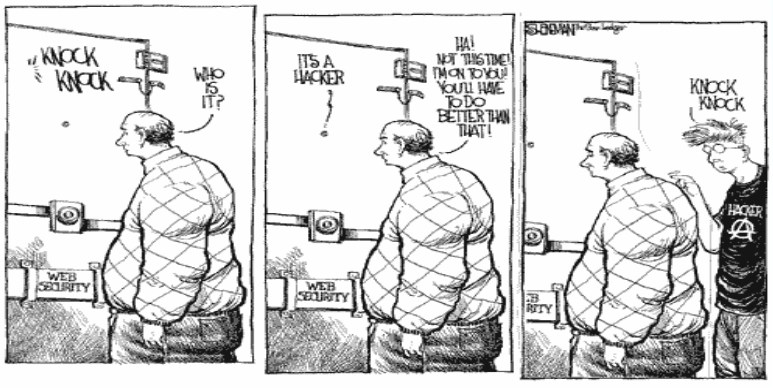
IP Spoofing attack is to hijack TCP session between server and its trusted host.

Every system in a TCP/IP network has an IP address. Someone who uses IP spoofing sets up a system (usually a PC) to pretend to be an existing IP address or a trusted IP address. Thus, the imposter can establish a connection with your system by pretending to be a system that you normally connect with.

To protect your network from spoofing, you should configure packet filters on your security gateway to the Internet.

More Network-based Attacks

- **Buffer Overflow Attack**
- **Port Scanning**
- **Password Cracking**



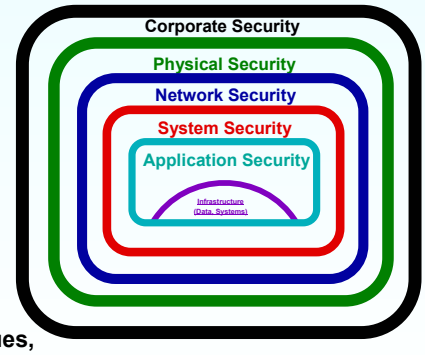
Recently, we often see Buffer overflow vulnerability case in CERT Advisory. The Buffer overflow vulnerability is caused by network program bug. If the network program accepts oversized data and stores it onto the buffer area, it exceeds the received buffer area border in the memory. If there is a program area just neighbor of the received buffer area in the memory, it corrupts the program area and it causes a program crash or operating system crash.

Intruders can also run their program with privileged authority with buffer overflow attack. Below shows the case that intruders can run their program with buffer overflow attack. A network program is handling a received buffer in the subroutine. A network program is going to fetch received buffer contents and is going to back to received buffer handling program at address 200000. Intruders try to create 1006bytes data and send it to the server. The server receives data and stores it onto the received buffer area. Because the network program doesn't check the data length, it allows buffer overflow condition on the received buffer area. Now the return address area is overwritten with 300000 due to the buffer overflow. A network program tries to continue the program at address 300000 where the intruder's program is ready to run. If the network program has a privileged authority, intruders can run their program with privileged authority.

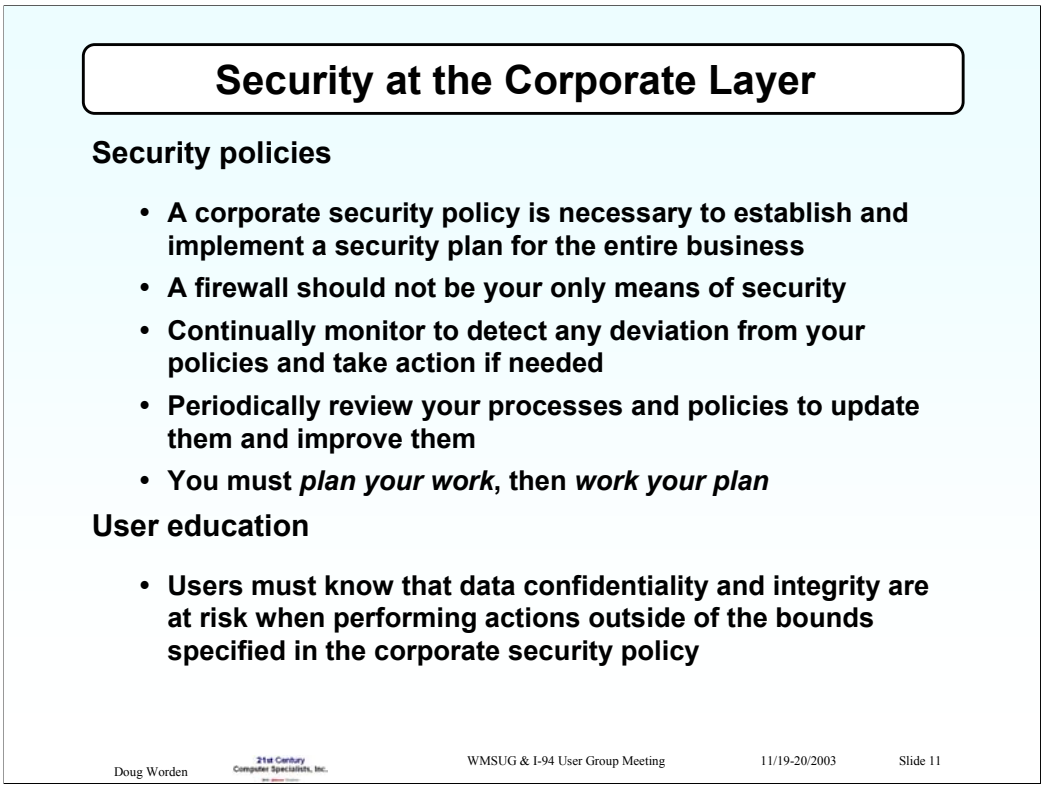
Layers of Security

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.**



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.



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.

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

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

iSeries Security Implementation

iSeries Security at the Physical Layer

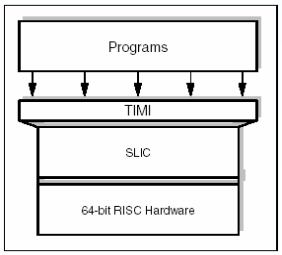
Physical locks

- The iSeries contains a coded key which is required to:
  - Power down the system via power switch
  - Enter Dedicated Service Tools Mode

iSeries Security at the System Layer: Let's Start with the Basics

Architecture

- **Single-level storage**
  - **application does not deal with storage device specifics.**
- **Technology Independent Machine Interface and System Licensed Internal Code**
  - **insulates application programs and their users from changing hardware characteristics.**
  - **Supervisory resource management functions in SLIC include validity and authorization checks.**
  - **Performed at the hardware level.**
  - **Integrated into the system.**
  - **Fast Execution!**



I am continually amazed at how versatile and flexible the iSeries is. How many years ago did we switch to 64bit technology? Did we have to purchase new applications to make that switch, or even recompile our existing applications?

The developers of this architecture, where the applications are insulated from the hardware, were truly amazing, futuristic thinkers. Do you remember that this architecture was developed in 1978? (that's 25 years ago!) It is this separation of the application and the hardware, the integration into the operating system of the relational database, the communications, the user interface, the SECURIY functions, etc, and the object-oriented operating system that make this iSeries still one of the most advanced systems on the planet TODAY.

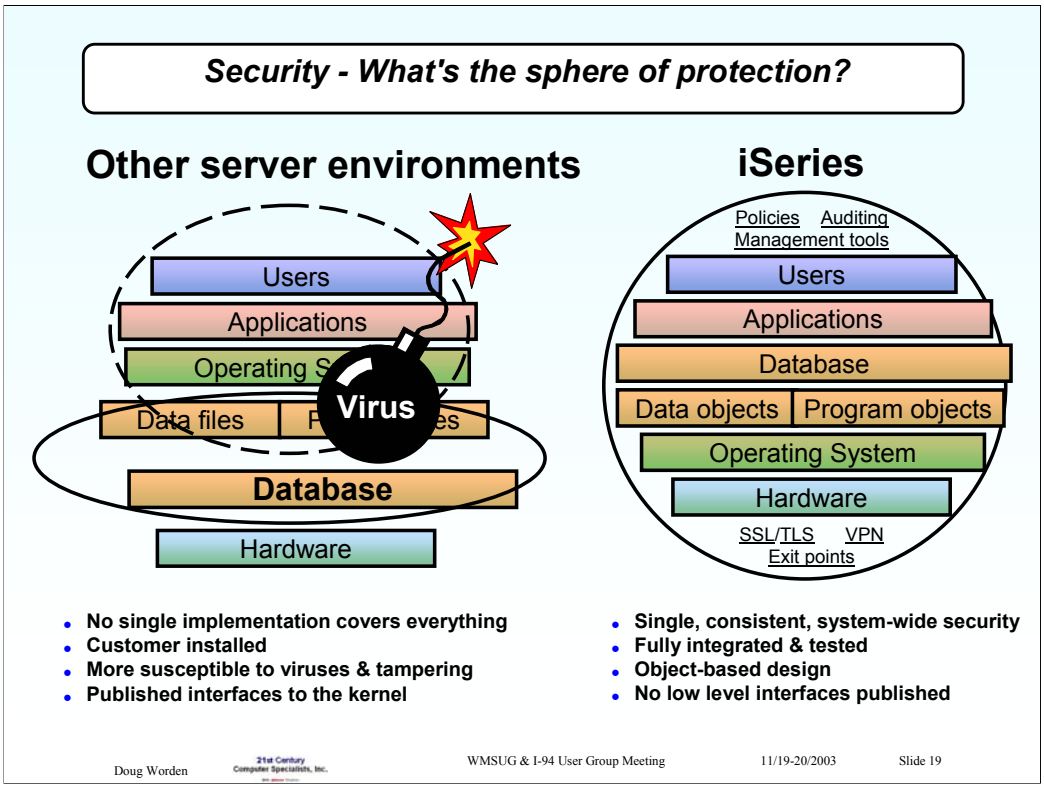
iSeries – an Integrated Approach

Operating System/400 (OS/400) integration design concepts.

- **Relational Database (DB2-UDB)**
- **Workload management**
- **Storage Management**
- **Communication Management**
- **Security Management**

All integrated into a pre-tested operating environment for business applications.

You can't run an iSeries system without also running these integrated functions!



Middleware (layered software) often comes from multiple sources in other server environments. Security is no exception. The mere addition of a relational database, such as DB2 or Oracle, introduces special security requirements. In that regard, security can neither be described as consistent nor system-wide.

You notice that the sphere around some of these layers is represented with a dashed line. This is because interfaces are published to the security kernel, making it possible for developers to bypass security or compromise data integrity. Typically these developers have good intentions, but more and more frequently we are finding that uninvited guests (hackers) are continually seeking new ways to deliver a harmful payloads to vulnerable servers.

By contrast, with iSeries you get a single, consistent, system-wide security implementation. The sphere of protection is comprehensive. You have the opportunity to implement multiple layers of defense without having to add any additional security packages.

You can secure users to identify who is authorized to the system and what power they have, if any. Application programs and data, representing your information assets, can be secured resource by resource, object by object, to define what operations can be performed and by whom. Because of the complete integration of hardware and software designed into iSeries, even the hardware is included in the security implementation.

In addition to user and resource security, system-wide policies can be established and globally enforced. Plus you have a high degree of granularity in defining the auditing and reporting that you want to use in monitoring your system security.

Furthermore, recognizing that as businesses transform to e-businesses the need for security extends to the network, security standards such as Virtual Private Networks (VPN), Secure Sockets Layer (SSL) and more recently Transport Layer Security (TLS) have also been built in. For added flexibility, controlled "exit points" have been provided that can be used by customers to uniquely extend their protection. There are no low level interfaces published to iSeries' SLIC (System Licensed Internal Code). Simply stated, iSeries is among the most securable servers in the industry.

Finally, you notice that with the iSeries we use the term "objects". Unlike other platforms, iSeries is an object-based system.

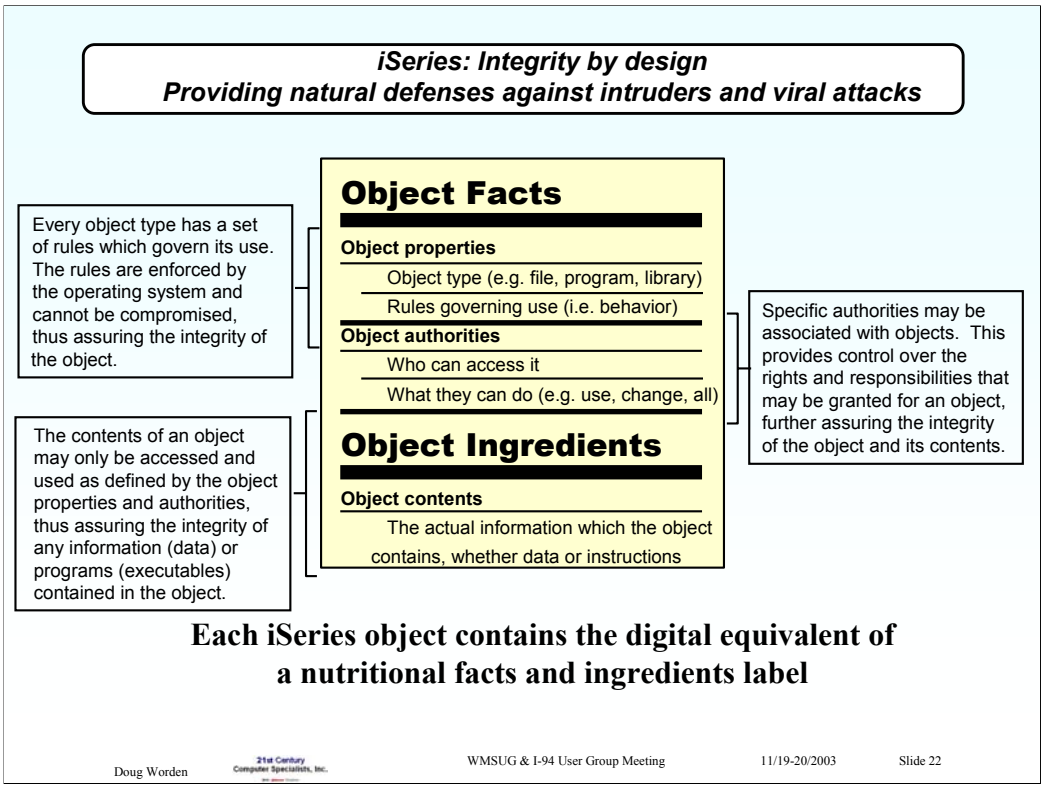
OS/400 – an Object Based Architecture

OS/400 supports an object-based design that makes it highly virus resistant

- **Everything the Operating System uses is packaged in an Object**
- **All Objects are Encapsulated (can't see inside them)**
- **The valid ways in which that object can be used is inseparable from the object.**
- **Data cannot be treated as executable code**
- **Executable code cannot be treated as data**

iSeries – Types of Objects

*ALRTBL
*AUTL
*BNDDIR
*CFGL
*CHTFMT
*CLD
*CLS
*CMD
*CNNL
*COSD
*CSI
*CSPMAP
*CSPTBL
*CTLD
*CRG
*CRQD
*DEVD
*DOC
*DTAARA
*DTADCT
*DTAQ
*EDTD
*EXITRG
*FCT
*FILE
*FLR
*FNTTBL
*FNTRSC
*FORMDF
*FTR
*GSS
*IMGCLG
*IPXD
*JOBD
*JOBQ
*JOBSCD
*JRN
*JRNRCV
*LIB
*LIND
*LOCALE
*MEDDFN
*MENU
*MODD
*MODULE
*MSGF
*MSGQ
*MGTCOL
*M36
*M36CFG
*NODL
*NODGRP
*NWID
*OUTQ
*NWSD
*NTBD
*OVL
*PAGDFN
*PAGSEG
*PDG
*PGM
*PNLGRP
*PRDDFN
*PRDLOD
*PSFCFG
*QMFORM
*QMQRY
*QRYDFN
*RCT
*SBSD
*SCHIDX
*SPADCT
*SQLPKG
*SQLUDT
*SRVPGM
*SVRSTG
*SSND
*S36
*TBL
*USRIDX
*USRPRF
*USRQ
*USRSPC
*VLDL
*WSCST



As an object based system, each iSeries object contains the digital equivalent of a nutritional facts and ingredients label.
Everything within the system is an "object". Each object has two inseparable parts. The first is the descriptive part -- seen here as "Object Facts" -- which defines the object type and valid ways of using that object (object properties) along with any specific authorities that are associated with the object (object authorities). The second is the data part -- seen here as "Object Ingredients" -- which serves as the functional aspect of the object.
The **only** operations allowed on an object are those which are defined in the descriptive part, so long as the requester has authority to the object. This insures data integrity for all objects in the system.
Why is this important? Because one of the common ways that viruses are introduced into a system is to have a program masquerade as data. A file is introduced into a system as data, then, when opened, morphs into something executable -- with potentially damaging consequences. Such a change of characteristics is not possible on iSeries. If an object is allowed to enter the system as data, it retains the characteristics of data forever. It cannot pretend to be something that it is not. The object cannot be compromised.
As such, iSeries provides natural defenses against intruders and viral attacks. Other servers have no equivalent. If iSeries provides the digital equivalent of nutritional facts and ingredients label, then other servers provide the digital equivalent of a brown paper bag. For that reason, it costs more in other environments to achieve a security policy with the same level of integrity as provided with iSeries. Other safety measures or perimeter defenses become necessary, and these activities carry an added cost.

Authority Information Stored with Object

- **Public authority**
- **Owner name**
- **Owner's authority to object**
- **Primary group name**
- **Primary group's authority to object**
- **Authorization list name**
- **Object auditing value**
- **Whether any private authority exists**
- **Whether any private authority is less than public**

Ever wonder why saves and restores take so long? There's a lot more stuff being saved than just the records in a file!

Authority Information Stored with User Profile

Heading Information:

- The user profile attributes shown on the Create User Profile display.
- The uid and gid.

Private Authority Information:

- Private authority to objects. This includes private authority to authorization lists.

Ownership Information:

- List of owned objects
- For each owned object, a list of users with private authority to the object.

Primary Group Information:

- List of objects for which the profile is the primary group.

Auditing Information:

- Action auditing value
- Object auditing value

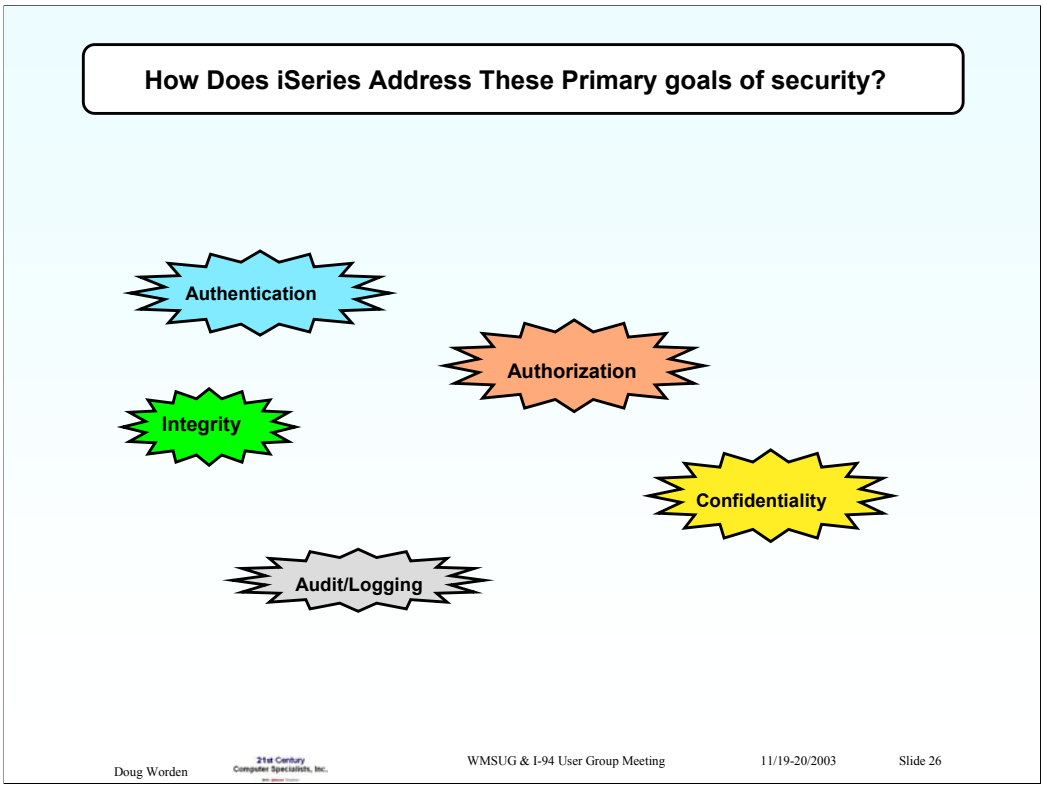
Function Usage Information:

- Usage settings for registered functions.

Authority Information Stored with Authorization Lists

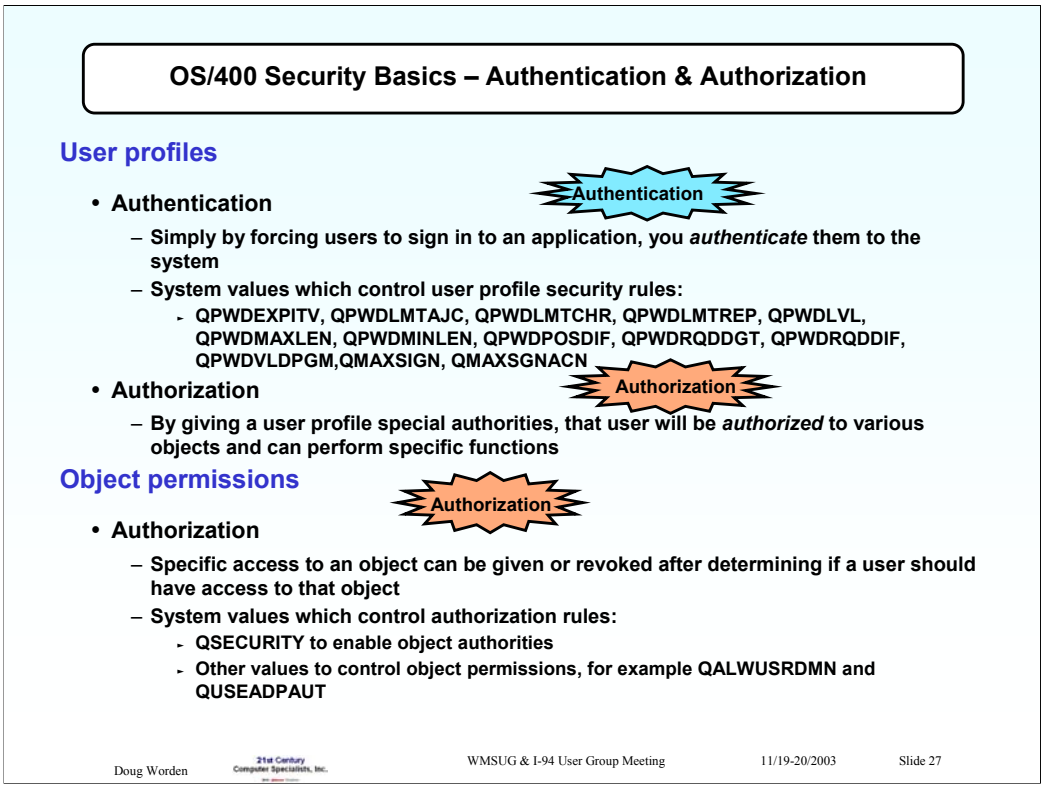
Definition: A list of related objects used to quickly authorize users to a large number of objects

- Normal authority information stored with any object, such as the public authority and owner.
- List of all objects secured by the authorization list.



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.



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:

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.

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.

What Object Authorities can a User Profile be given to an Object (IE: Permissions to the object itself)

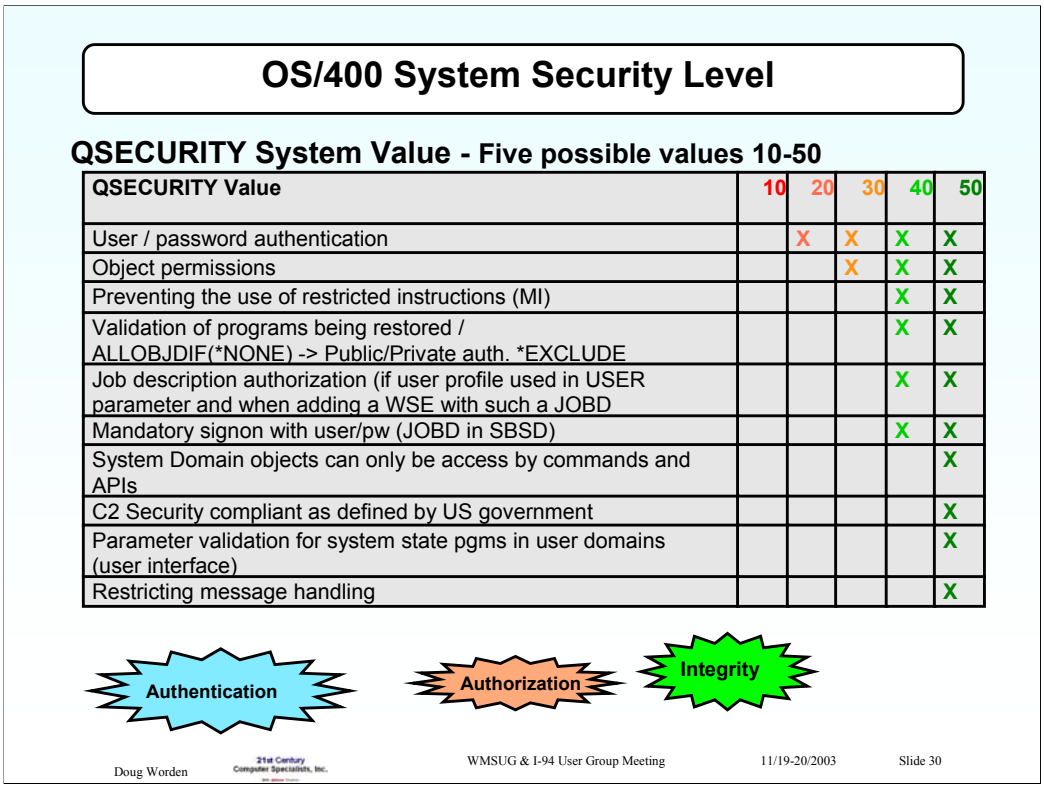
Authority	Name	Functions Allowed
Object Authorities:		
*OBJOPR	Object Operational	Look at the description of an object. Use the object as determined by the user's data authorities.
*OBJMGT	Object Management	Specify the security for the object. Move or rename the object. All functions defined for *OBJALTER and *OBJREF.
*OBJEXIST	Object Existence	Delete the object. Free storage of the object. Perform save and restore operations for the object [1]. Transfer ownership of the object.
*OBJALTER	Object Alter	Add, clear, initialize and reorganize members of the database files. Alter and add attributes of database files: add and remove triggers. Change the attributes of SQL packages.
*OBJREF	Object Reference	Specify a database file as the parent in a referential constraint. For example, you want to define a rule that a customer record must exist in the CUSMAS file before an order for the customer can be added to the CUSORD file. You need *OBJREF authority to the CUSMAS file to define this rule.
*AUTLMGT	Authorization List Management	Add and remove users and their authorities from the authorization list [2].

What Data Authorities can a User Profile be given to an Object(IE: Permissions to the data portion of the object)

Authority	Name	Functions Allowed
Data Authorities:		
*READ	Read	Display the contents of the object, such as viewing records in a file.
*ADD	Add	Add entries to an object, such as adding messages to a message queue or adding records to a file.
*UPD	Update	Change the entries in an object, such as changing records in a file.
*DLT	Delete	Remove entries from an object, such as removing messages from a message queue or deleting records from a file.
*EXECUTE	Execute	Run a program, service program, or SQL package. Locate an object in a library or a directory.
Field Authorities:		
*Mgt	Management	Specify the security for the field.
*Alter	Alter	Change the attributes of the field.
*Ref	Reference	Specify the field as part of the parent key in a referential constraint.
*Read	Read	Access the contents of the field. For example, display the contents of the field.
*Add	Add	Add entries to data, such as adding information to a specific field.
*Update	Update	Change the content of existing entries in the field.

[1] If a user has save system (*SAVSYS) special authority, object existence authority is not required to perform save and restore operations on the object.

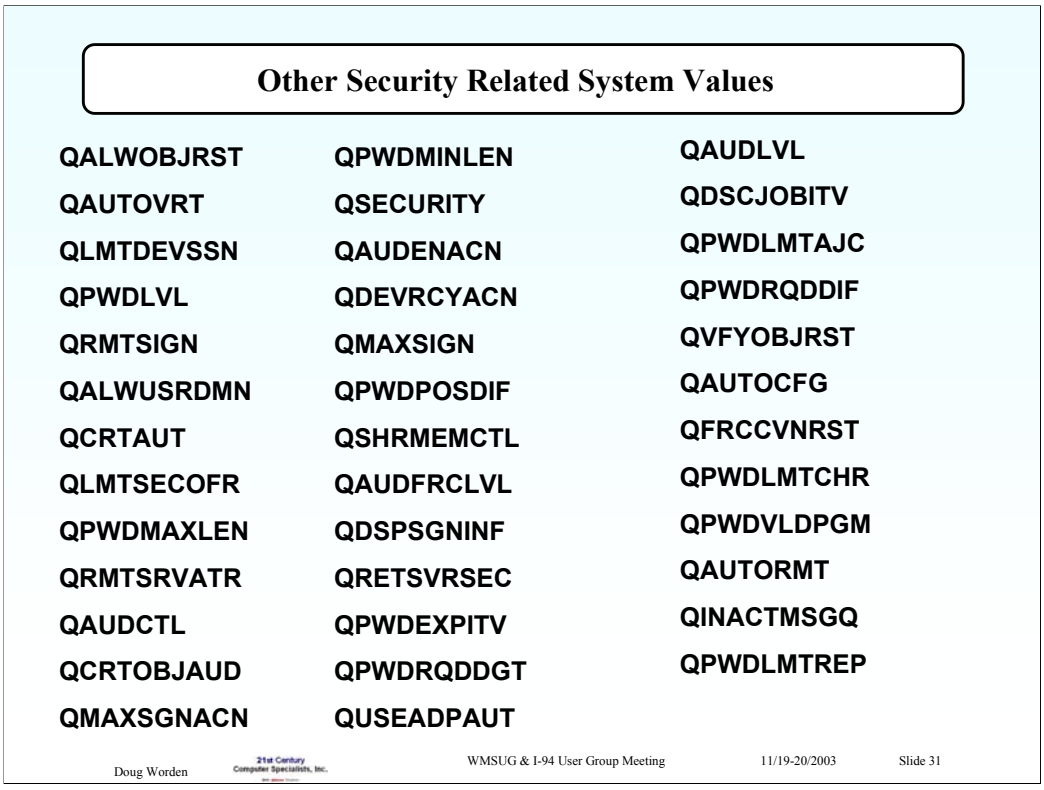
[2] See the topic "Authorization List Management" on page 127 for more information.



Setting the System Security Level System Value is the first step in securing your iSeries.

QSECURITY level 10 is no longer an option.

See chapter 2 of iSeries Security Reference manual for detailed descriptions of the different levels of WSECURITY system value.



See Chapter 3 ‘Security System Values’ of the iSeries Security Reference manual for detailed descriptions of these values.

iSeries Advanced Security Functions

DST User IDs & Passwords

- **New at V5R1, enhanced at V5R2**
  - **At V5R2, IDs can be created & changed using DST & SST**
  - **If still at V5R1, can enter DST by entering '21' in the control panel**
- **IBM provided User IDs**
  - **QSECOFR**
  - **QSRV**
  - **11111111**
  - **22222222**
  - **Create your own User ID**
  - **These Ids are totally separate from OS/400 Ids**
  - **Case-sensitive, 128 digit passwords (using new SHA encryption)**
- **OS/400 QSECOFR password can be reset via DST, by the DST QSECOFR user ID**
- **DST QSECOFR password can be reset via CHGDSTPWD command, by the OS/400 QSECOFR user profile**

Service tools user IDs are separate from OS/400 user profiles. Passwords for service tools user IDs are

encrypted at different levels for security. The default password level uses Data Encryption Standard (DES)

encryption. You should use DES encryption if you have pre-V5R1 clients using iSeries Navigator to

connect to service functions such as logical partitions and disk unit management.

You can change the password level to use Secure Hash Algorithm (SHA) encryption, which is

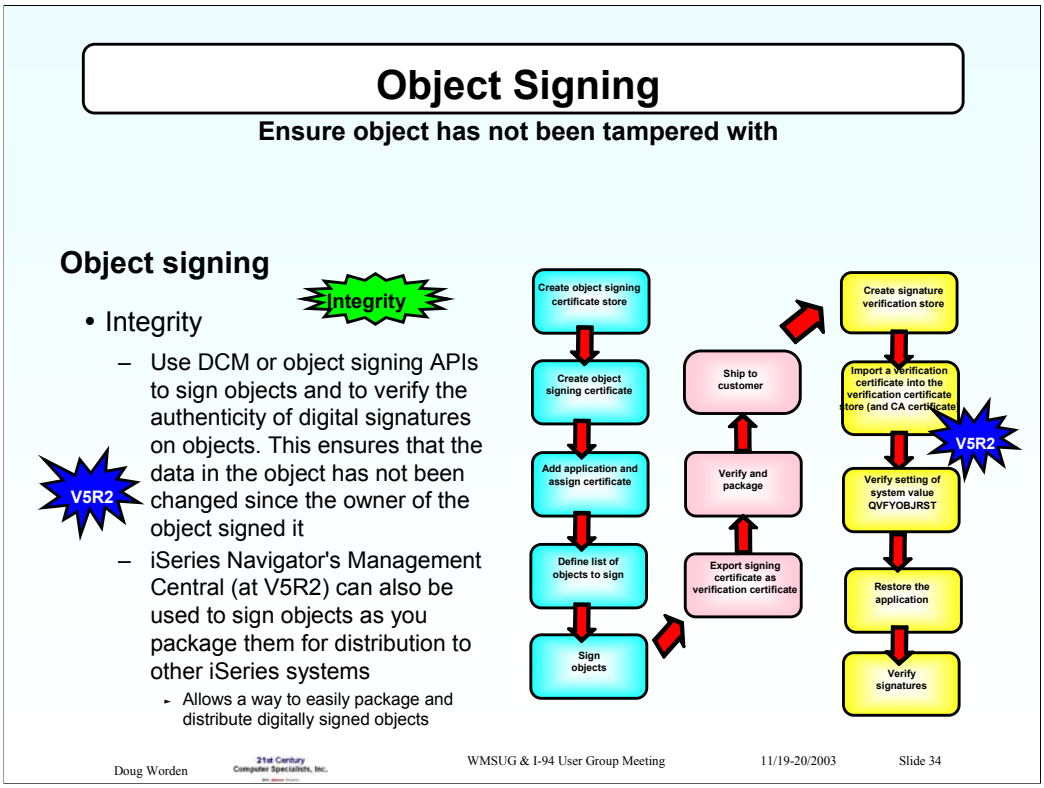
mathematically impossible to reverse and provides stronger encryption and a higher level of security. Once

you change to SHA encryption, however, you cannot change back to DES encryption. If you change to

SHA encryption, you will no longer be able to connect to the service tools server with pre-V5R1 clients

such as Operations Console. You will need to upgrade any clients that will be using these functions when

you upgrade your password level to SHA.



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

- 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.

QVFYOBJRST system value

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

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.

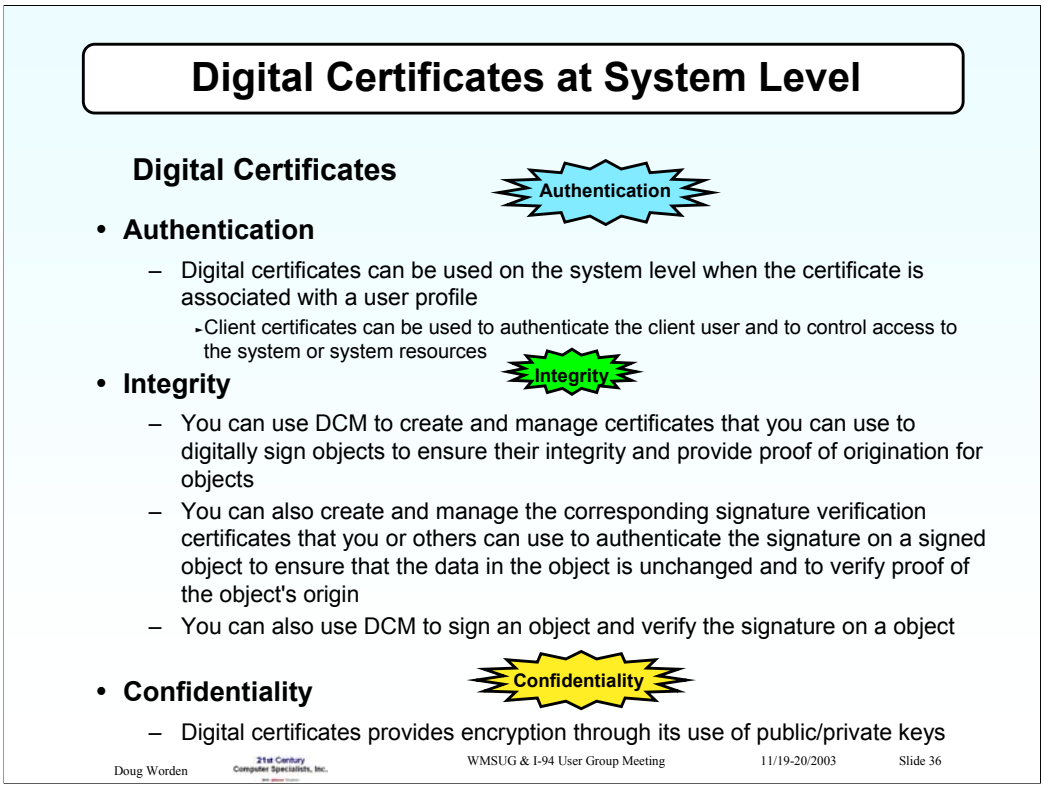
If an integrity violation has occurred, the object name, library name (or pathname), object type, object owner, and type of failure are logged to a database file.

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.



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.

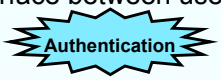
The certificate containing the public key must usually be available to the public. This can be achieved by storing the certificates in a Lightweight Directory Protocol (LDAP) directory.

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

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.

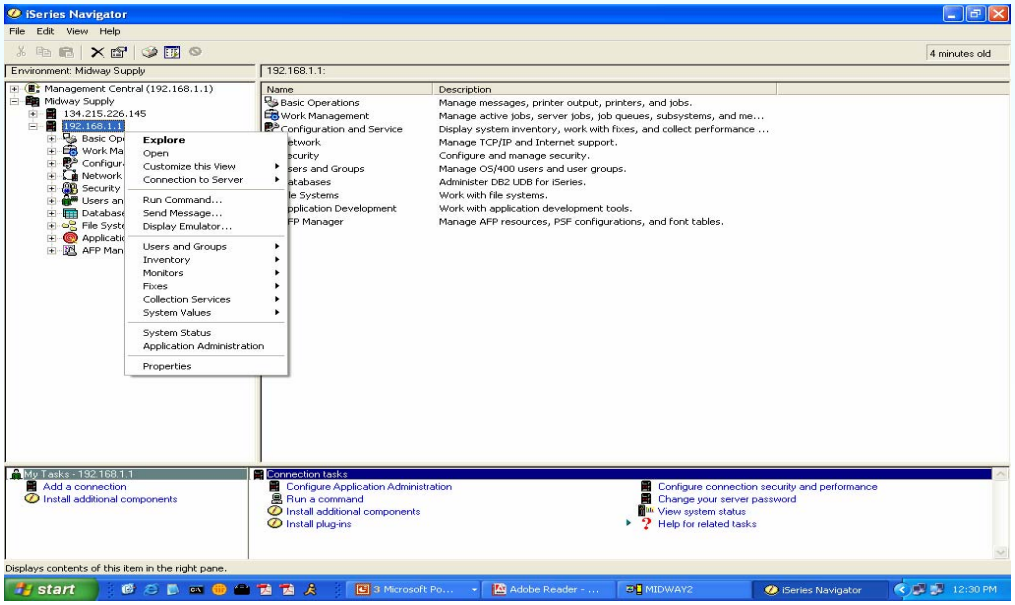
Logging and auditing is also a very important aspect when monitoring security. You can use exit programs to create your own logging mechanism for various system applications. For example, the FTP server does not provide a standard interface to enable logging of FTP subcommands performed by a signed on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

Telnet Exit Program Example

Following are examples of what you can do when you start the exit program

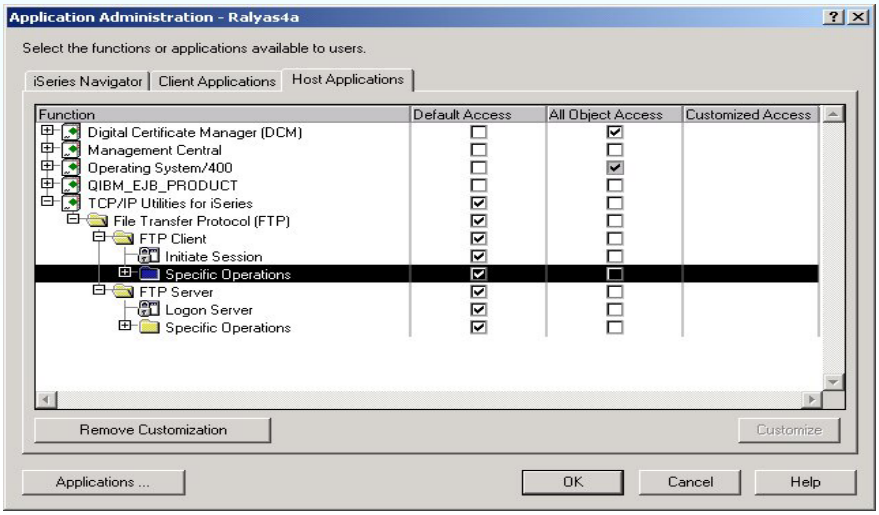
- you can use the Server (local) IP address on multi-homed iSeries 400 servers to route connections to different subsystems based on the network interface (IP address).
- Allow or deny the session, based on any known criteria, such as the user's IP address, the time of day, and the requested user profile.
- Assign a specific iSeries device description for the session. This allows routing of the interactive job to any sub-system set up to receive those devices.
- Assign specific National Language values for the session, such as keyboard and character set.
- Assign a specific user profile for the session.
- Automatically sign on the requestor (without displaying a Sign On display).
- Set up audit logging for the session.

Application Administration from iSeries Access

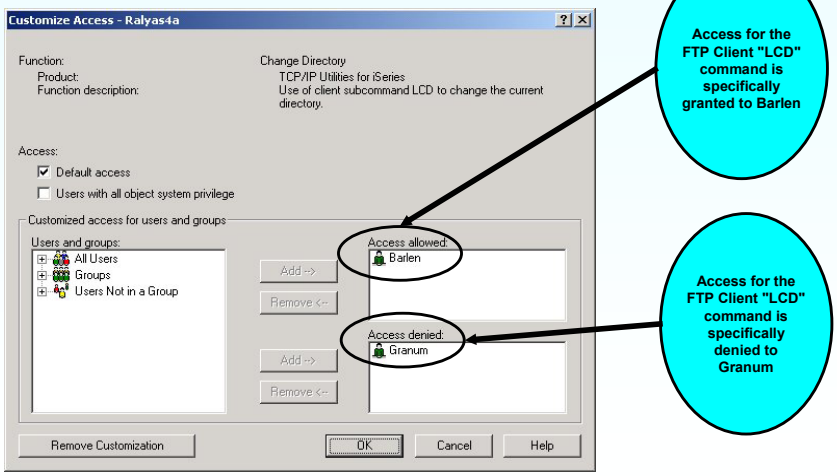


Application Administration

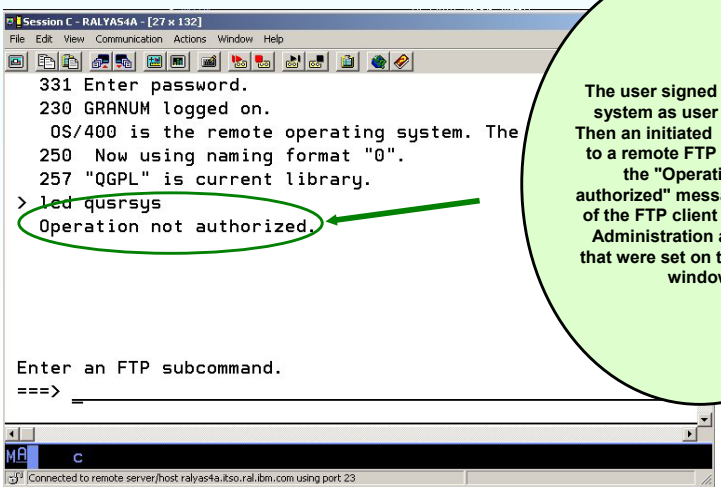
Application Administration can implement security constraints to a very fine detail and open the FTP client and server security completely or anywhere in between. This example shows you where to set authorities to limit FTP client commands for users on the iSeries.



Application Administration (Cont'd)



Application Administration (Cont'd)

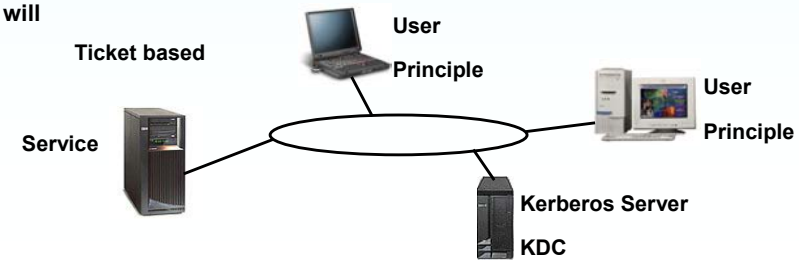


The user signed on to client system as user GRANUM. Then an initiated FTP session to a remote FTP server gets the "Operation not authorized" message because of the FTP client Application Administration authorities that were set on the previous window

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**



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.

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

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.

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.

This 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).

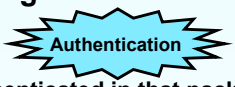
Specifying TLS/SSL protection for the iSeries FTP Client

- 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.

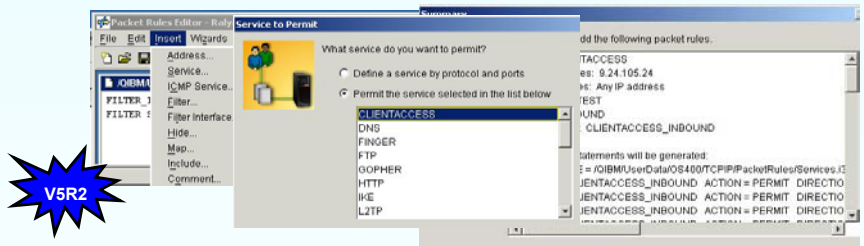
IP Packet Filtering

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**

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:

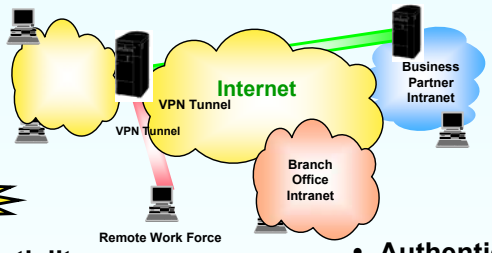
#The following filter rules are defined on the interface:
#This filter rule file probably has no real practical use, it is used here to show the format of #iSeries packet filter rules.
#Permit inbound packets from all clients (IP address any (*)) to the SSL-Telnet server #(IP address 10.1.1.10, mask 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.

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

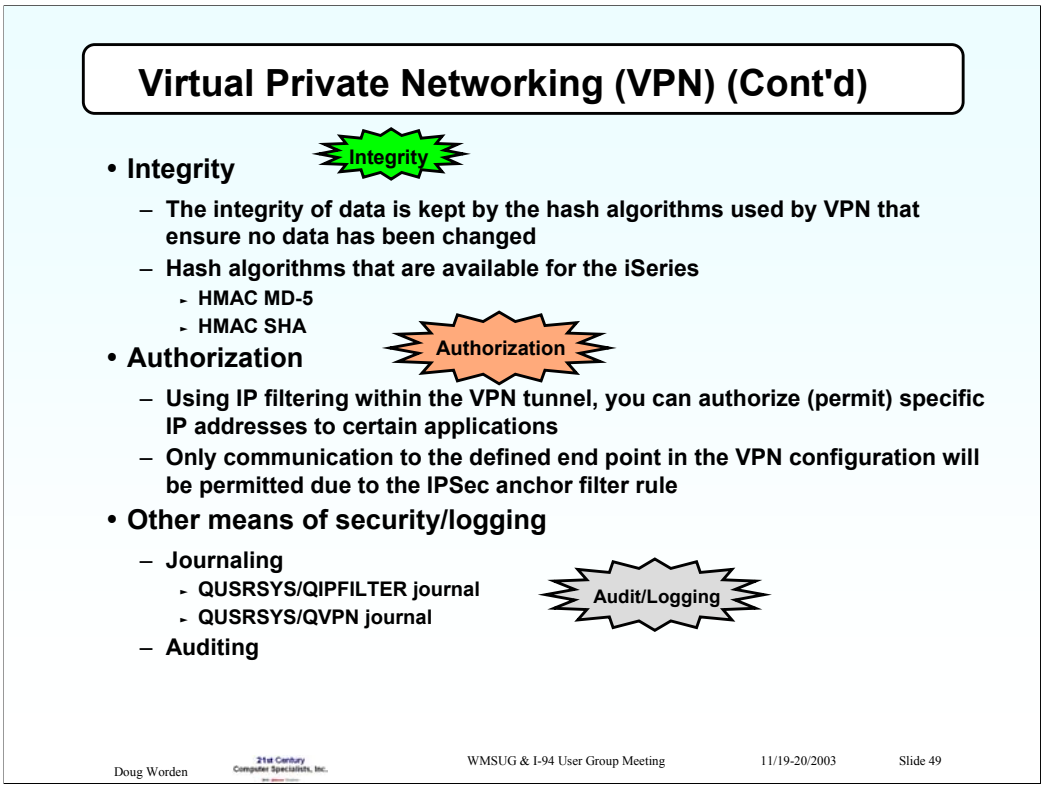
- Confidentiality
  - Data is typically encrypted in a VPN tunnel by the use of the Encapsulation Security Payload (ESP) protocol
  - Encryption algorithms that are available for the iSeries
    - Data Encryption Standard (DES)
    - Triple Data Encryption Standard (3DES)
    - RC4
    - RC5
    - Advanced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

V5R2

- Authentication

Authentication

  - The iSeries allows two methods to authenticate remote VPN endpoints
    - Pre-shared secret
    - Digital certificates (V5R1 and later)



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.

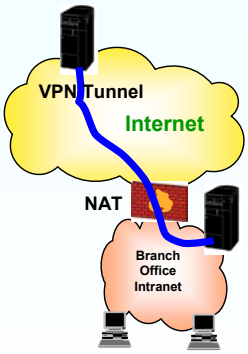
VPN Enhancements at V5R2

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

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.

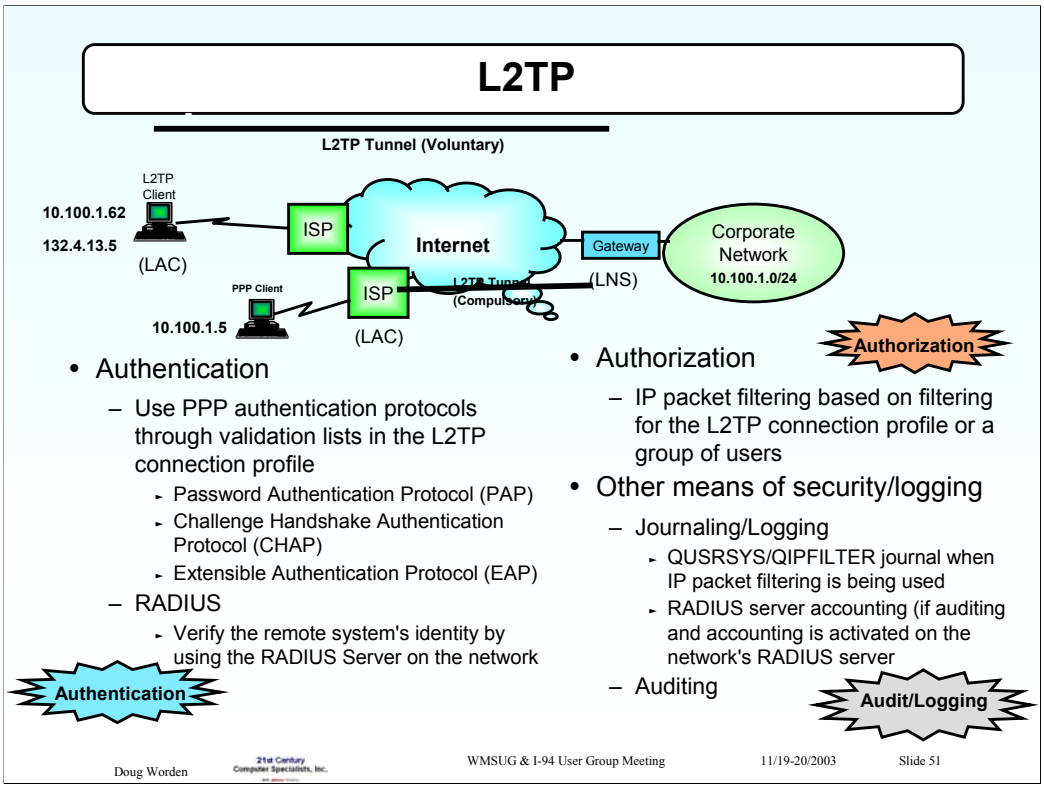
Unfortunately, conventional NAT does not work on IPSec packets because when the packet goes through a NAT device, the source address in the packet changes, thereby invalidating the packet. When this happens, the receiving end of the VPN connection discards the packet and the VPN connection negotiations 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.



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.

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

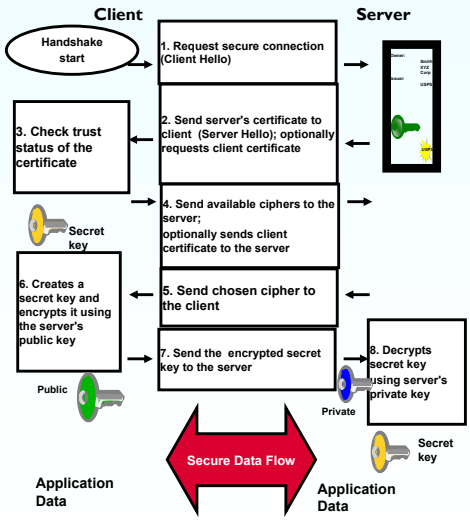
Secure Sockets Layer/Transport Layer Security

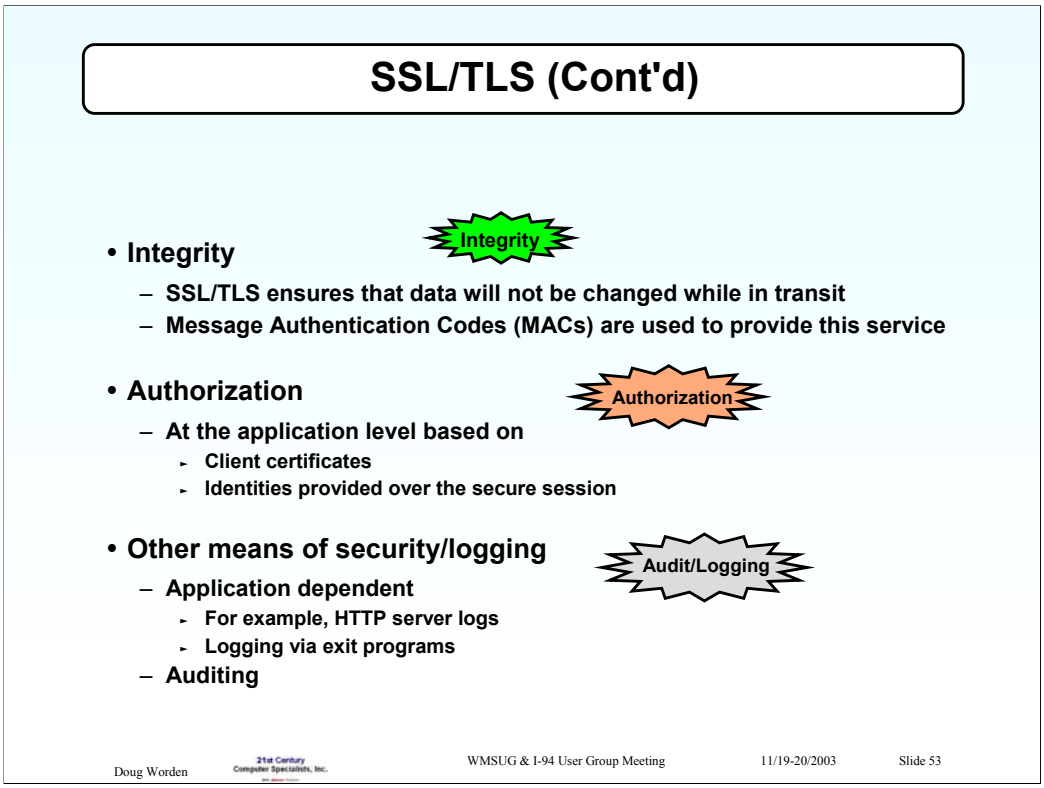
Authentication

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

Confidentiality

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





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

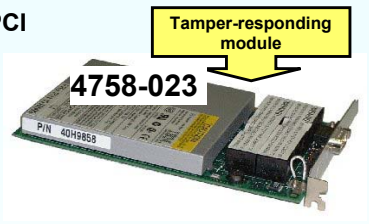
Any iSeries server applications that are enabled for SSL will automatically obtain TLS support unless the application has specifically requested to use only SSL Version 3.0 or SSL Version 2.0.

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**

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

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

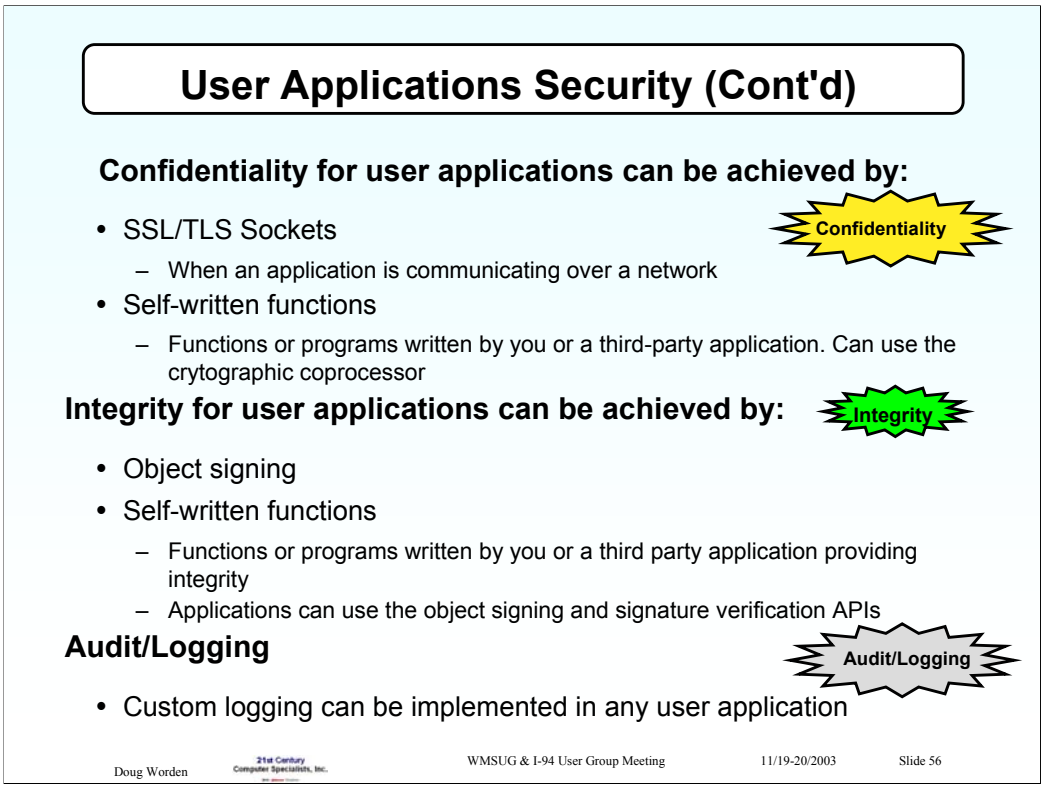
User Applications Security

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**

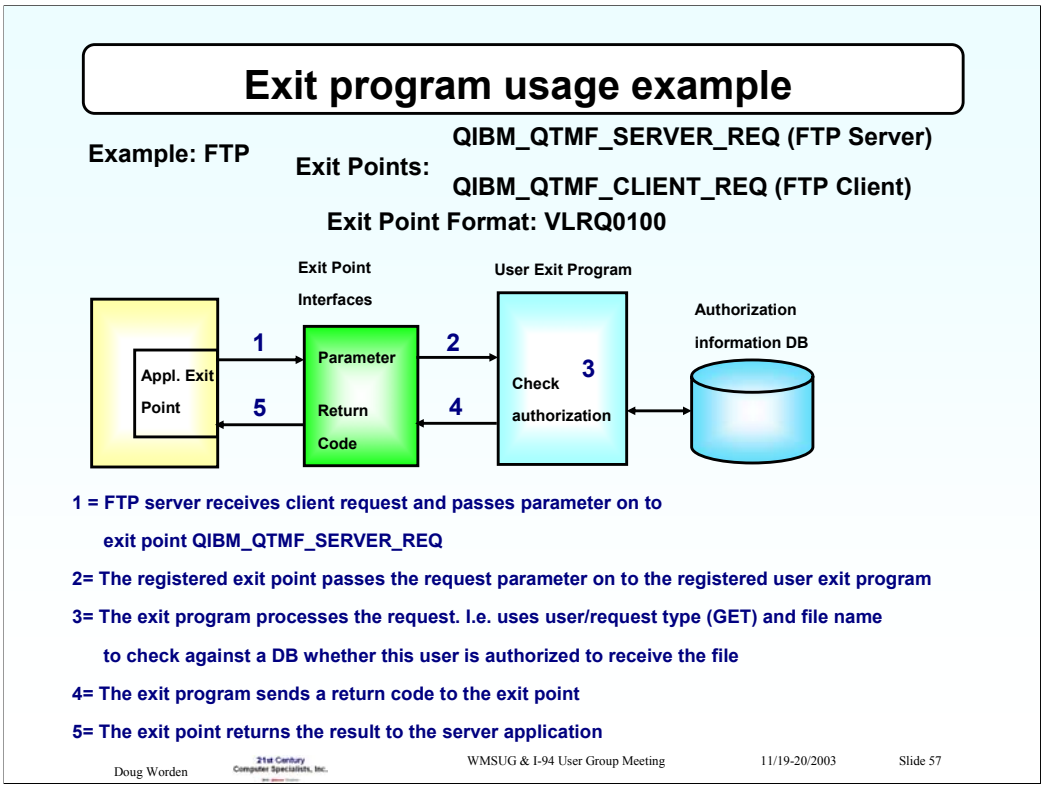


You or a third-party vendor can write applications that meet all of the major security goals. Whether you want to use authentication or confidentiality, standard OS/400 functions or APIs can be used to build in security into your own applications.

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.

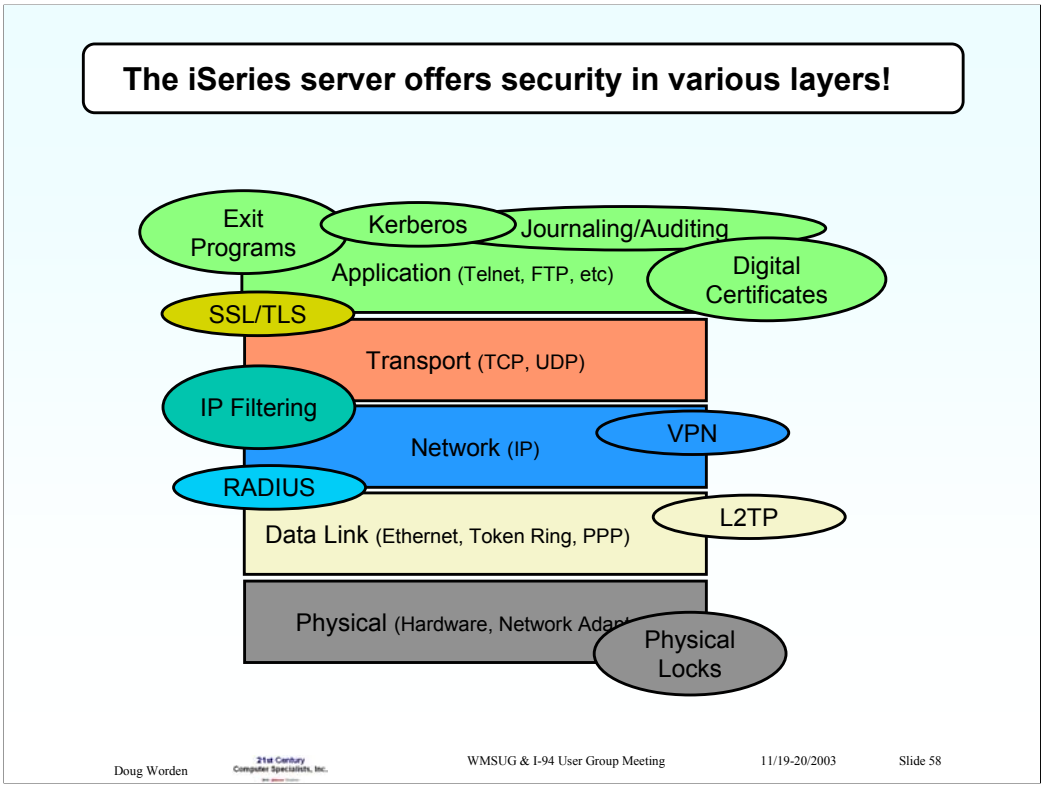


The previous chart shows an example of how exit programs work. In this case, a user exit program was written to handle requests received by the FTP server. Since the exit point format VLRQ0100 has the same format for the FTP server and the client, a single program can be used to handle server and client requests.

The exit point format VLRQ0100 is as follows:

Parameter	Description	Input or Output
Type and length		
1	Application identifier	Input
Binary (4)		
2	Operation identifier	Input
Binary (4)		
3	User profile	Input
Char (10)		
4	Remote IP address	Input
Char (10)		
5	Length of remote IP address	Input
Binary (4)		
6	Operation-specific information	Input
Char (*)		
7	Length of operation-specific information	Input
Binary (4)		
8	Allow operation	Output
Binary (4)		

Using the information provided by the exit point, the exit program can, for example, determine based on the user, the requested operation and the file the request is for, whether the request should be accepted or denied. The permission or denial response is returned in parameter 8.



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.

Where to Start with iSeries Security

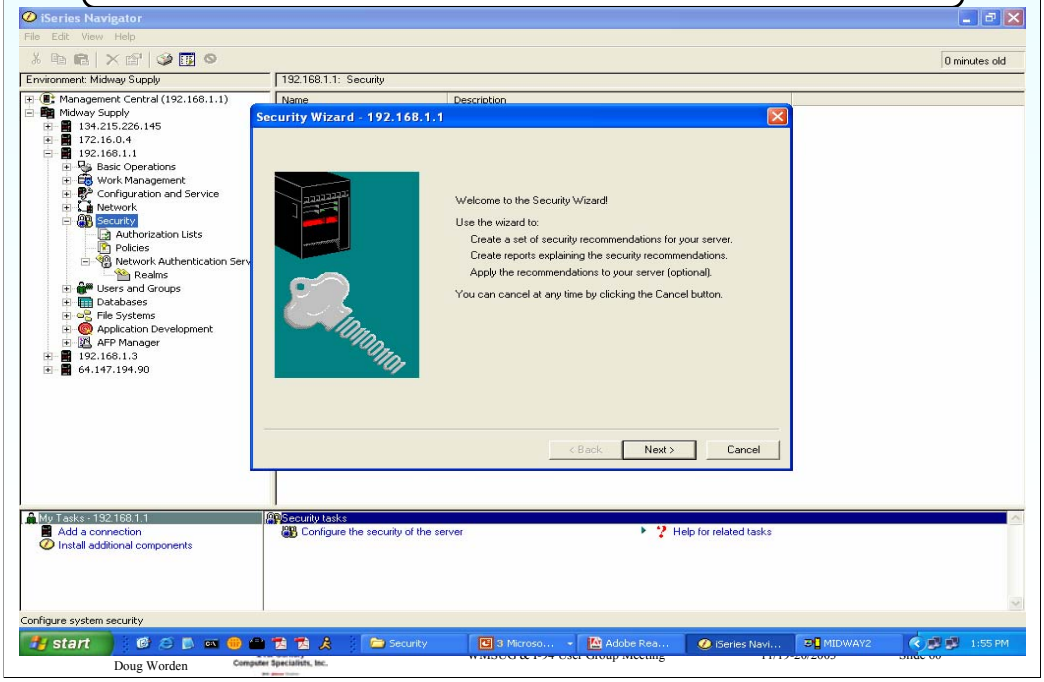
iSeries 400 Security Advisor

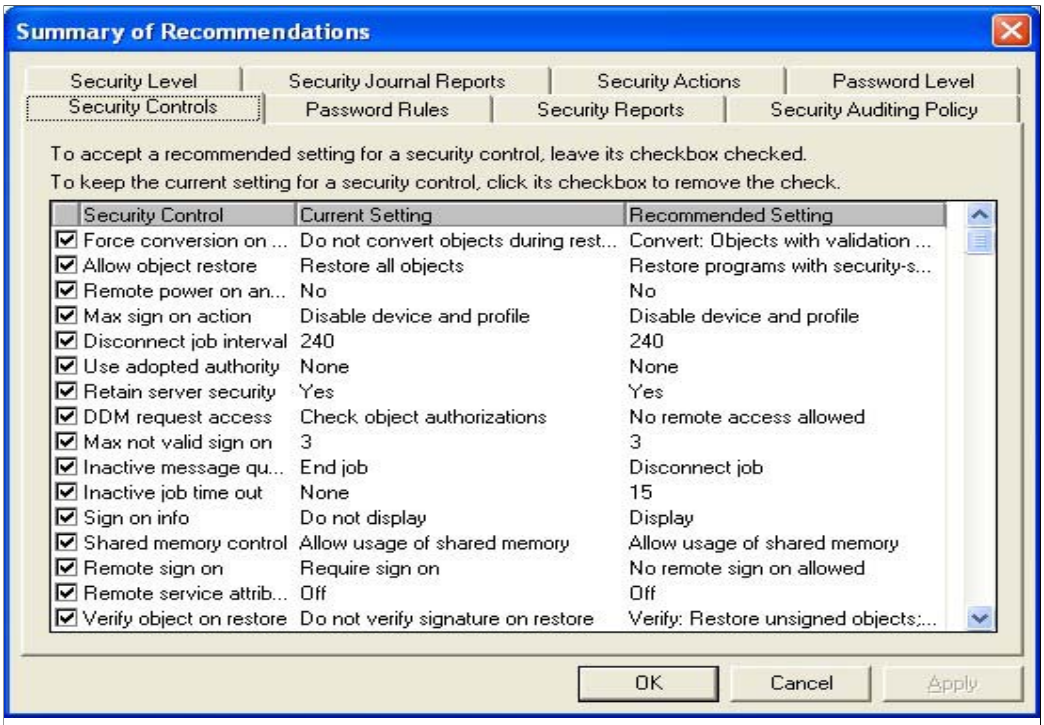
Available at IBM Tech Studio
Available as a Wizard in Operations Navigator
Websitehttp://www.as400.ibm.com/tstudio/secure1/index_av.htm

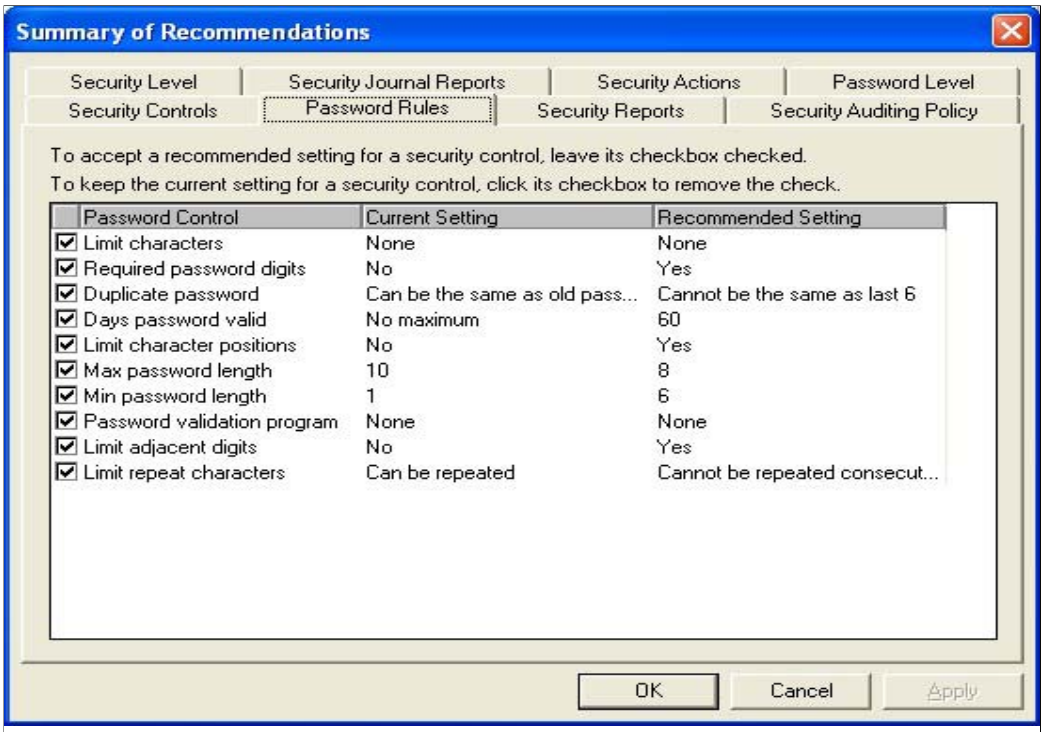
iSeries Publications

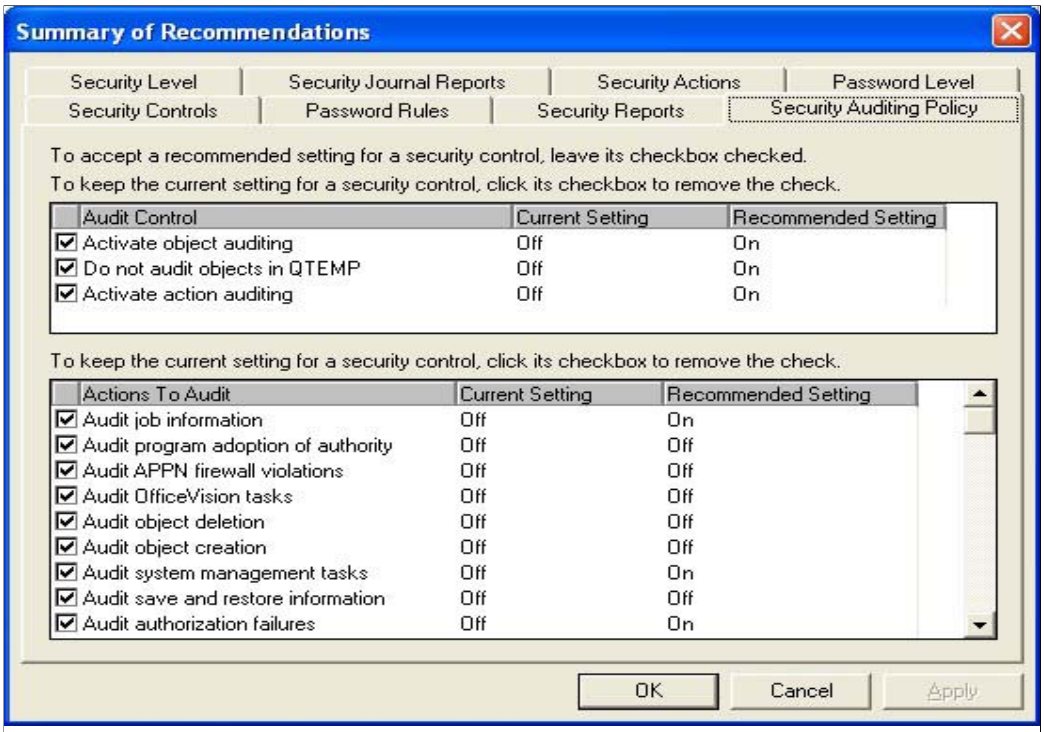
iSeries Security Reference: SC41-5302
Tips and Tools for Securing Your iSeries: SC41-5300

Starting the iSeries Navigator Security Wizard




Summary of Recommendations
Security Level
Security Journal Reports
Security Actions
Password Level
Security Controls
Password Rules
Security Reports
Security Auditing Policy
To accept a recommended setting for a security control, leave its checkbox checked.
To keep the current setting for a security control, click its checkbox to remove the check.
Security Control
Current Setting
Recommended Setting
Force conversion on ...
Do not convert objects during rest...
Convert: Objects with validation ...
Allow object restore
Restore all objects
Restore programs with security-s...
Remote power on an...
No
No
Max sign on action
Disable device and profile
Disable device and profile
Disconnect job interval
240
240
Use adopted authority
None
None
Retain server security
Yes
Yes
DDM request access
Check object authorizations
No remote access allowed
Max not valid sign on
3
3
Inactive message qu...
End job
Disconnect job
Inactive job time out
None
15
Sign on info
Do not display
Display
Shared memory control
Allow usage of shared memory
Allow usage of shared memory
Remote sign on
Require sign on
No remote sign on allowed
Remote service attrib...
Off
Off
Verify object on restore
Do not verify signature on restore
Verify: Restore unsigned objects;...
OK
Cancel
Apply


Summary of Recommendations
Security Level
Security Journal Reports
Security Actions
Password Level
Security Controls
Password Rules
Security Reports
Security Auditing Policy
To accept a recommended setting for a security control, leave its checkbox checked.
To keep the current setting for a security control, click its checkbox to remove the check.
Password Control
Current Setting
Recommended Setting
Limit characters
None
None
Required password digits
No
Yes
Duplicate password
Can be the same as old pass...
Cannot be the same as last 6
Days password valid
No maximum
60
Limit character positions
No
Yes
Max password length
10
8
Min password length
1
6
Password validation program
None
None
Limit adjacent digits
No
Yes
Limit repeat characters
Can be repeated
Cannot be repeated consecut...
OK
Cancel
Apply


Summary of Recommendations
Security Level
Security Journal Reports
Security Actions
Password Level
Security Controls
Password Rules
Security Reports
Security Auditing Policy
To accept a recommended setting for a security control, leave its checkbox checked.
To keep the current setting for a security control, click its checkbox to remove the check.
Audit Control
Current Setting
Recommended Setting
Activate object auditing
Off
On
Do not audit objects in QTEMP
Off
On
Activate action auditing
Off
On
To keep the current setting for a security control, click its checkbox to remove the check.
Actions To Audit
Current Setting
Recommended Setting
Audit job information
Off
On
Audit program adoption of authority
Off
Off
Audit APPN firewall violations
Off
Off
Audit OfficeVision tasks
Off
Off
Audit object deletion
Off
Off
Audit object creation
Off
Off
Audit system management tasks
Off
On
Audit save and restore information
Off
Off
Audit authorization failures
Off
On
OK
Cancel
Apply

iSeries Navigator Security Wizard

- **Security wizard will create 2 reports**
  - **Reports describe recommended changes and how changes will affect the system**
- **Wizard will offer to implement the recommended changes**
  - **Use this option with caution… unexpected results may occur!**
- **Try it!**
  - **It's easy to run**
  - **A good way to 'verify' your current security settings**

Limit Users Ability to Change System Values

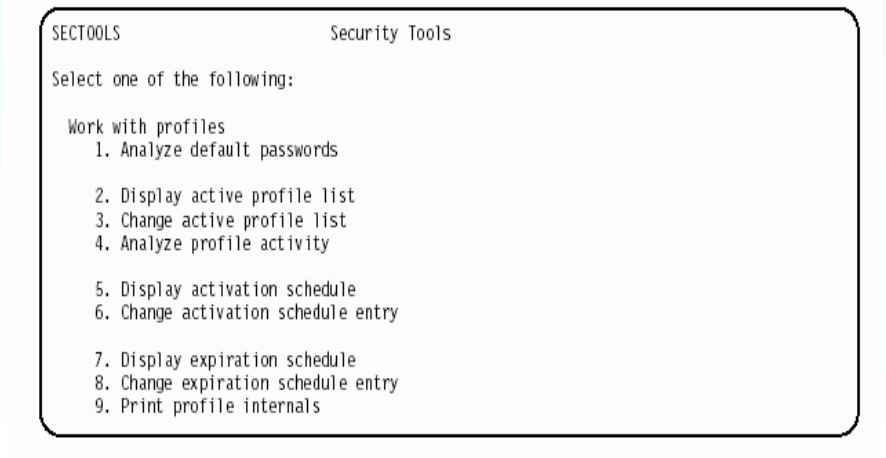
- **New for V5R2**
- **All of the system values on the previous slide can be restricted**
  - **Even users with *SECADM & *ALLOBJ authority can be restricted**
- **To restrict access:**
  - **STRSST – sign on as 'QSECOFR'**
  - **Option 7 'work with system security'**
  - **Select 'NO' for each System Value you want to restrict**

All/Any of these Security Related System Values can be Restricted

QALWOBJRST	QPWDMINLEN	QAUDLVL
QAUTOVRT	QSECURITY	QDSCJOBITV
QLMTDEVSSN	QAUDENACN	QPWDLMTAJC
QPWDLVL	QDEVRCYACN	QPWDRQDDIF
QRMTSIGN	QMAXSIGN	QVFYOBJRST
QALWUSRDMN	QPWDPOSDIF	QAUTOCFG
QCRTAUT	QSHRMEMCTL	QFRCCVNRST
QLMTSECOFR	QAUDFRCLVL	QPWDLMTCHR
QPWDMAXLEN	QDSPSGNINF	QPWDVLDPGM
QRMTSRVATR	QRETSVRSEC	QAUTORMT
QAUDCTL	QPWDEXPITV	QINACTMSGQ
QCRTOBJAUD	QPWDRQDDGT	QPWDLMTREP
QMAXSGNACN	QUSEADPAUT	

Use the Security Tools Menu

GO SECTOOLS



Security Tools Menu – all options

1. Analyze default passwords
2. Display active profile list
3. Change active profile list
4. Analyze profile activity
5. Display activation schedule
6. Change activation schedule entry
7. Display expiration schedule
8. Change expiration schedule entry
9. Print profile internals
10. Change security auditing
11. Display security auditing
20. Submit or schedule security reports to batch
21. Adopting objects
22. Audit journal entries
23. Authorization list authorities
24. Command authority
25. Command private authority
26. Communications security
27. Directory authority
28. Directory private authority
29. Document authority
30. Document private authority
31. File authority
32. File private authority
33. Folder authority
34. Folder private authority
35. Job description authority
36. Library authority
37. Library private authority
38. Object authority
39. Private authority
40. Program authority
41. Program private authority
42. User profile authority
43. User profile private authority
44. Job and output queue authority
45. Subsystem authority
46. System security attributes
47. Trigger programs
48. User objects
49. User profile information
60. Configure system security
61. Revoke public authority to objects
62. Check object integrity

Table 6. Tool Commands for User Profiles

Menu[1] Option	Command Name	Description	Database File Used
1	ANZDFTPWD	Use the Analyze Default Passwords command to report on and take action on user profiles that have a password equal to the user profile name.	QASECPWD[2]
2	DSPACTPRFL	Use the Display Active Profile List command to display or print the list of user profiles that are exempt from ANZPRFACT processing.	QASECIDL[2]
3	CHGACTPRFL	Use the Change Active Profile List command to add and remove user profiles from the exemption list for the ANZPRFACT command. A user profile that is on the active profile list is permanently active (until you remove the profile from the list). The ANZPRFACT command does not disable a profile that is on the active profile list, no matter how long the profile has been inactive.	QASECIDL[2]
4	ANZPRFACT	Use the Analyze Profile Activity command to disable user profiles that have not been used for a specified number of days. After you use the ANZPRFACT command to specify the number of days, the system runs the ANZPRFACT job nightly. You can use the CHGACTPRFL command to exempt user profiles from being disabled.	QASECIDL[2]

5	DSPACTSCD	Use the Display Profile Activation Schedule command to display or print information about the schedule for enabling and disabling specific user profiles. You create the schedule with the CHGACTSCDE command.	QASECACT[2]
6	CHGACTSCDE	Use the Change Activation Schedule Entry command to make a user profile available for sign on only at certain times of the day or week. For each user profile that you schedule, the system creates job schedule entries for the enable and disable times.	QASECACT[2]
7	DSPEXPSCD	Use the Display Expiration Schedule command to display or print the list of user profiles that are scheduled to be disabled or removed from the system in the future. You use the CHGEXPSCDE command to set up user profiles to expire.	QASECEXP[2]
8	CHGEXPSCDE	Use the Change Expiration Schedule Entry command to schedule a user profile for removal. You can remove it temporarily (by disabling it) or you can delete it from the system. This command uses a job schedule entry that runs every day at 00:01 (1 minute after midnight). The job looks at the QASECEXP file to determine whether any user profiles are set up to expire on that day. Use the DSPEXPSCD command to display the user profiles that are scheduled to expire.	QASECEXP[2]
9	PRTPRFINT	Use the Print Profile Internals command to print a report containing information on the number of entries contained in a user profile. The number of entries determines the size of the user profile.	

Table 7. Tool Commands for Security Auditing

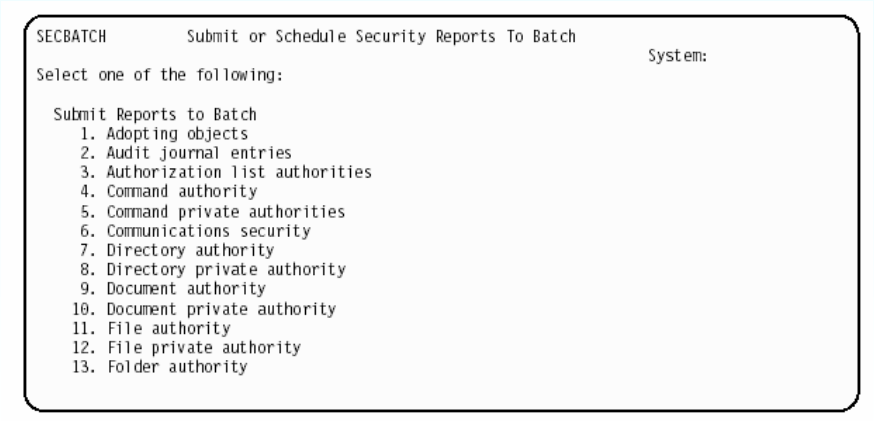
Menu[1] Option	Command Name	Description	Database File Used
10	CHGSECAUD	Use the Change Security Auditing command to set up security auditing and to change the system values that control security auditing. When you run the CHGSECAUD command, the system creates the security audit (QAUDJRN) journal if it does not exist. The CHGSECAUD command provides options that make it simpler to set the QAUDLVL (audit level) system value. You can specify *ALL to activate all of the possible audit level settings. Or, you can specify *DFTSET to activate the most commonly used settings (*AUTFAIL, *CREATE, *DELETE, *SECURITY, and *SAVRST). **Note:** If you use the security tools to set up auditing, be sure to plan for management of your audit journal receivers. Otherwise, you might quickly encounter problems with disk utilization.	
11	DSPSECAUD	Use the Display Security Auditing command to display information about the security audit journal and the system values that control security auditing.	

Notes:

1. Options are from the SECTOOLS menu.

Use the Batch Security Tools Menu

GO SECBATCH



Batch Security Tools Menu- all options

1. Adopting objects
2. Audit journal entries
3. Authorization list authoritie
4. Command authority
5. Command private authority
6. Communications security
7. Directory authority
8. Directory private authority
9. Document authority
10. Document private authority
11. File authority
12. File private authority
13. Folder authority
14. Folder private authority
15. Job description authority
16. Library authority
17. Library private authority
18. Object authority
19. Private authority
20. Program authority
21. Program private authority
22. User profile authority
23. User profile private authority
24. Job and output queue authority
25. Subsystem authority
26. System security attributes
27. Trigger programs
28. User objects
29. User profile information
30. User profile internals
31. Check object integrity

Menu[1] Option	Command Name	Description	Database File Used
1, 40	PRTADPOBJ	Use the Print Adopting Objects command to print a list of objects that adopt the authority of the specified user profile. You can specify a single profile, a generic profile name (such as all profiles that begin with Q), or all user profiles on the system. This report has two versions. The full report lists all adopted objects that meet the selection criteria. The changed report lists differences between adopted objects that are currently on the system and adopted objects that were on the system the last time that you ran the report.	QSECADPOLD[2]
2, 41	DSPAUDJRNE	Use the Display Audit Journal Entries command to display or print information about entries in the security audit journal. You can select specific entry types, specific users, and a time period.	QASYxxJ4[3]
3, 42	PRTPVTAUT *AUTL	When you use the Print Private Authorities command for *AUTL objects, you receive a list of all the authorization lists on the system. The report includes the users who are authorized to each list and what authority the users have to the list. Use this information to help you analyze sources of object authority on your system. This report has three versions. The full report lists all authorization lists on the system. The changed report lists additions and changes to authorization since you last ran the report. The deleted report lists users whose authority to the authorization list has been deleted since you last ran the report. When you print the full report, you have the option to print a list of objects that each authorization list secures. The system will create a separate report for each authorization list.	QSECATLOLD[2]
6, 45	PRTCMNSEC	Use the Print Communications Security command to print the security-relevant settings for objects that affect communications on your system. These settings affect how users and jobs can enter your system. This command produces two reports: a report that displays the settings for configuration lists on the system and a report that lists security-relevant parameters for line descriptions, controllers, and device descriptions. Each of these reports has a full version and a changed version.	QSECCMNOLD[2]

Menu[1] Option	Command Name	Description	Database File Used
15, 54	PRTJOBDAUT	Use the Print Job Description Authority command to print a list of job descriptions that specify a user profile and have public authority that is not *EXCLUDE. The report shows the special authorities for the user profile that is specified in the job description. This report has two versions. The full report lists all job description objects that meet the selection criteria. The changed report lists differences between job description objects that are currently on the system and job description objects that were on the system the last time that you ran the report.	QSECJBDOLD[2]
See note 4	PRTPUBAUT	Use the Print Publicly Authorized Objects command to print a list of objects whose public authority is not *EXCLUDE. When you run the command, you specify the type of object and the library or libraries for the report. Use the PRTPUBAUT command to print information about objects that every user on the system can access. This report has two versions. The full report lists all objects that meet the selection criteria. The changed report lists differences between the specified objects that are currently on the system and objects (of the same type in the same library) that were on the system the last time that you ran the report.	QPBxxxxxx[3]
See note 5.	PRTPVTAUT	Use the Print Private Authorities command to print a list of the private authorities to objects of the specified type in the specified library. Use this report to help you determine the sources of authority to objects. This report has three versions. The full report lists all objects that meet the selection criteria. The changed report lists differences between the specified objects that are currently on the system and objects (of the same type in the same library) that were on the system the last time that you ran the report. The deleted report lists users whose authority to an object has been deleted since you last printed the report.	QPVxxxxxx[3]

Menu[1] Option	Command Name	Description	Database File Used
24, 63	PRTQAUT	Use the Print Queue Report to print the security settings for output queues and job queues on your system. These settings control who can view and change entries in the output queue or job queue. This report has two versions. The full report lists all output queue and job queue objects that meet the selection criteria. The changed report lists differences between output queue and job queue objects that are currently on the system and output queue and job queue objects that were on the system the last time that you ran the report.	QSECQOLD[2]
25, 64	PRTSBSDAUT	Use the Print Subsystem Description command to print the security-relevant communications entries for subsystem descriptions on your system. These settings control how work can enter your system and how jobs run. The report prints a subsystem description only if it has communications entries that specify a user profile name. This report has two versions. The full report lists all subsystem description objects that meet the selection criteria. The changed report lists differences between subsystem description objects that are currently on the system and subsystem description objects that were on the system the last time that you ran the report.	QSECSBDOLD[2]
26, 65	PRTSYSSECA	Use the Print System Security Attributes command to print a list of security-relevant system values and network attributes. The report shows the current value and the recommended value.	
27, 66	PRTTRGPGM	Use the Print Trigger Programs command to print a list of trigger programs that are associated with database files on your system. This report has two versions. The full report lists every trigger program that is assigned and meets your selection criteria. The changed report lists trigger programs that have been assigned since the last time that you ran the report.	QSECTRGOLD[2]

Menu[1] Option	Command Name	Description	Database File Used
28, 67	PRTUSROBJ	Use the Print User Objects command to print a list of the user objects (objects not supplied by IBM) that are in a library. You might use this report to print a list of user objects that are in a library (such as QSYS) that is in the system portion of the library list. This report has two versions. The full report lists all user objects that meet the selection criteria. The changed report lists differences between user objects that are currently on the system and user objects that were on the system the last time that you ran the report.	QSECPUOLD[2]
29, 68	PRTUSRPRF	Use the Print User Profile command to analyze user profiles that meet specified criteria. You can select user profiles based on special authorities, user class, or a mismatch between special authorities and user class. You can print authority information, environment information, password information, or password level information.	
30, 69	PRTPRFINT	Use the Print Profile Internals command to print a report of internal information on the number of entries.	
31, 70	CHKOBJITG	Use the Check Object Integrity command to determine whether operable objects (such as programs) have been changed without using a compiler. This command can help you to detect attempts to introduce a virus program on your system or to change a program to perform unauthorized instructions. The *iSeries Security Reference* book provides more information about the CHKOBJITG command.	

Notes:

1. Options are from the SECBATCH menu.
2. This file is in the QUSRSYS library.
3. xx is the two-character journal entry type. For example, the model output file for AE journal entries is QSYS/QASYAEJ4. The model output files are described in Appendix F of the *iSeries Security Reference* book.
4. The SECBATCH menu contains options for the object types that are typically of concern to security administrators. For example, use options 11 or 50 to run the PRTPUBAUT command against *FILE objects. Use the general options (18 and 57) to specify the object type.
5. The SECBATCH menu contains options for the object types that are typically of concern to security administrators. For example, options 12 or 51 run the PRTPVTAUT command against *FILE objects. Use the general options (19 and 58) to specify the object type.
6. The xxxxxx in the name of the file is the object type. For example, the file for program objects is called QPBPGM for public authorities and QPVPGM for private authorities. The files are in the QUSRSYS library.

   The file contains a member for each library for which you have printed the report. The member name is the same as the library name.

Integrated File System (IFS) Tips

- **Limit access to /QSYS.LIB file system**
  - **Modify a user's authority to the PWFSERVER Authorization List**
    - ***EXCLUDE, user cannot enter /QSYS.LIB**
    - ***USE, user can enter /QSYS.LIB – then object authority takes over**
    - **IBM Shipped value is *USE**
    - **Affects iSeries Access File Systems, Netserver**
    - **Does not affect FTP, ODBC**
- **PC type Viruses can be stored in the IFS!**
  - **Using NETSERVER, directories can be 'mapped' to a network drive**
  - **Some of the latest viruses spread to all 'mapped' drives**
  - **Scan the mapped IFS directories using virus detection software, just as you would scan PC directories**
    - **Through a PC program**
    - **Via a 3rd party OS/400 native application**

Intrusion Detection & Auditing

- **How can you effectively detect security violations?**
  - **Turn on system's auditing capabilities**
    - **Regularly review the audit journals**
    - **Develop programs to help automate the review process or**
    - **Use third party programs**
  - **Regularly create and review system's security reports**
    - **SECTOOLS Menu**
    - **SECBATCH Menu**

Activating Security Auditing

Security auditing is activated using these 5 System Values

- **QAUDCTL**
  - **Auditing control – turns auditing on or off**
- **QAUDENDACN**
  - **Auditing end action if audit journal cannot be written**
    - ***NOTIFY**
    - ***PWRDWNSYS – use with caution!**
- **QAUDFRCLVL**
  - **Auditing force level – How often are journal entries written to disk**
- **QAUDLVL**
  - **Auditing level – What events to audit**
- **QCRTOBJAUD**
  - **Create default auditing – What should be audited for newly created objects**

QAUDLVL System Value

- **This is the system value which determines what events are going to be audited.**
  - **Lots of options**

*AUTFAIL	*NETCMN	*PRTDTA	*SYSMGT
*CMD	*OPTICAL	*SAVRST	
*CREATE	*OBJMGT	*SECURITY	
*DELETE	*PGMADP	*SERVICE	
*JOBDTA	*PGMFAIL	*SPLFDTA	

  - **The Security Wizard will set this up for you**
  - **Many of these options can also be set in the User Profile**
    - **Use the CHGUSRAUD command**

See chapter 9 of the iSeries Security Reference Manual for descriptions of what the auditing options do.

Activating Security Auditing

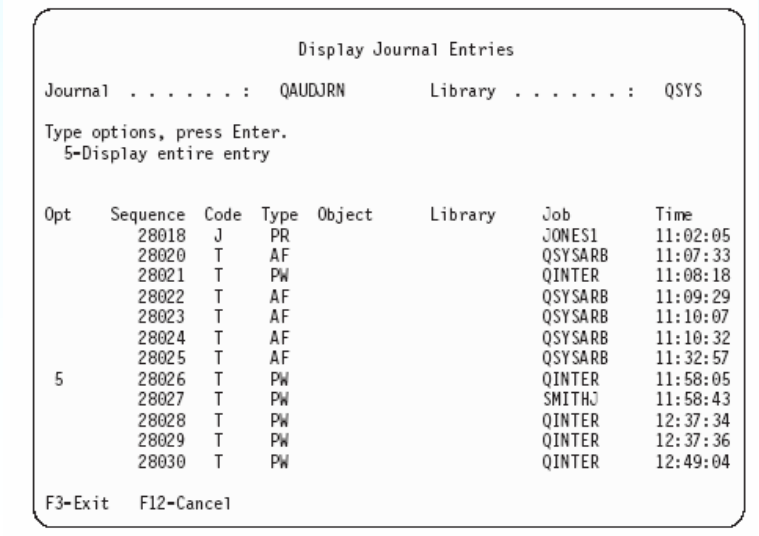
- **CRTJRNRCV**
  - **Name it like 'audjrn0001'**
- **CRTJRN QSYS/QAUDJRN**
  - **Name must be QAUDJRN**
- **WRKSYSVAL *SEC**
  - **Modify the previously referenced System Values**
- **CHGOBJAUD**
  - **Set auditing for individual objects**
- **CHGDLOAUD**
  - **Set auditing for individual document objects**
- **CHGUSRAUD**
  - **Sets auditing for individual users**


```
CRTJRNRCV JRNRCV(JRNLIB/AUDRCV0001) +
THRESHOLD(100000) AUT(*EXCLUDE) +
TEXT(' Auditing Journal Receiver' )
CRTJRN JRN(QSYS/QAUDJRN) +
JRNRCV(JRNLIB/AUDRCV0001) +
MNGRCV(*SYSTEM) DLTRCV(*NO) +
AUT(*EXCLUDE) TEXT(' Auditing Journal' )
```

Analyzing the Audit Journal

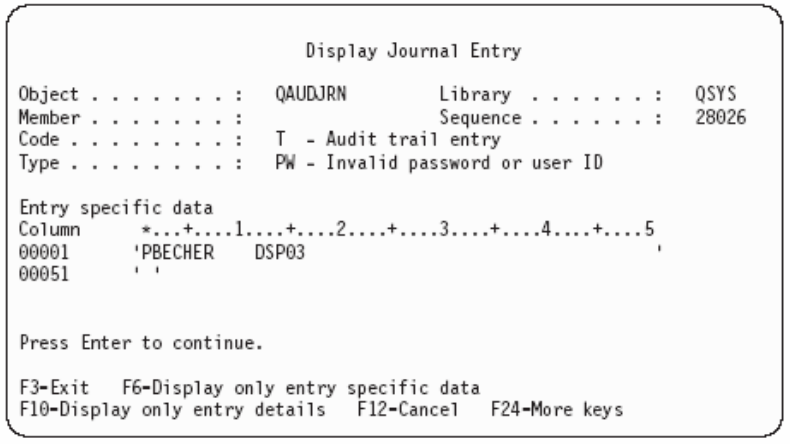
- **DSPJRN QAUDJRN command**
- **DSPAUDJRNE command from the SECTOOLS menu**
- **Use a query tool or program to analyze entries**
  - **DSPJRN to an outfile**
    - **File layout is different for different journal entry types**
    - **Use Appendix F of the iSeries Security Reference manual for the name of the model database outfiles**
    - **Create a duplicate object of the appropriate database model and use it as your outfile**
    - **Analyzing journals can be complex – be prepared to do research**
    - **Maybe use third party programs which have already done the work for you**

DSPJRN QAUDJRN



```
                          Display Journal Entries

Journal  . . . . . :   QAUDJRN         Library  . . . . . :   QSYS

Type options, press Enter.
  5-Display entire entry


Opt    Sequence  Code  Type  Object      Library     Job         Time
          28018   J     PR                          JONES1      11:02:05
          28020   T     AF                          QSYSARB     11:07:33
          28021   T     PW                          QINTER      11:08:18
          28022   T     AF                          QSYSARB     11:09:29
          28023   T     AF                          QSYSARB     11:10:07
          28024   T     AF                          QSYSARB     11:10:32
          28025   T     AF                          QSYSARB     11:32:57
 5        28026   T     PW                          QINTER      11:58:05
          28027   T     PW                          SMITHJ      11:58:43
          28028   T     PW                          QINTER      12:37:34
          28029   T     PW                          QINTER      12:37:36
          28030   T     PW                          QINTER      12:49:04

F3-Exit   F12-Cancel
```

DSPJRN QAUDJRN Continued



```
                          Display Journal Entry

Object . . . . . :   QAUDJRN         Library  . . . . . :   QSYS
Member . . . . . :                   Sequence . . . . . :   28026
Code . . . . . :   T  - Audit trail entry
Type . . . . . :   PW - Invalid password or user ID

Entry specific data
Column      *...+....1....+....2....+....3....+....4....+....5
00001      'PBECHER    DSP03                                  '
00051      ' '


Press Enter to continue.

F3-Exit   F6-Display only entry specific data
F10-Display only entry details   F12-Cancel   F24-More keys
```

Sifting Through the Audit Journal

Detailed info on layout of the Audit Journal Entries is in Appendix 'F' of the iSeries Security Reference manual, SC41-5302

iSeries Auditing: Review

Audit/Logging

- Use 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

Misc. Internet based Security Information

- **CERT Coordination Center**
  - **http://www.cert.org/advisories**
    - **Contains Internet Security Information**

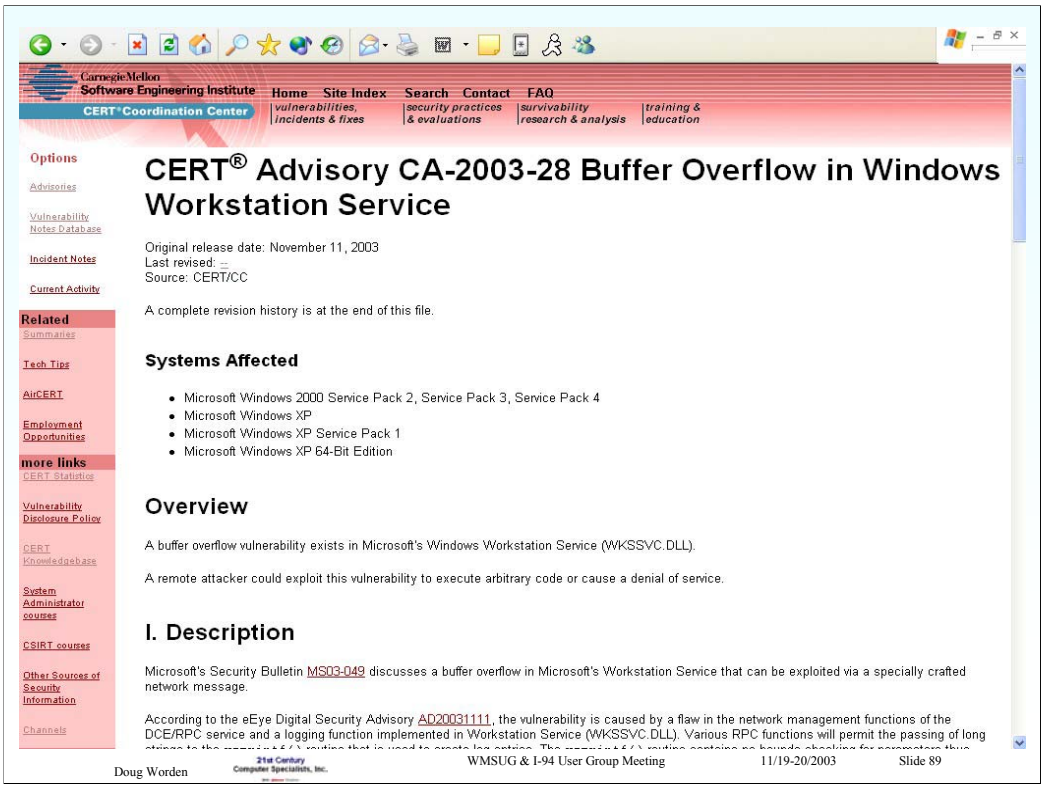
Example of an Advisory:

CA-2003-28 :Buffer Overflow in Windows Workstation Service

November 11, 2003

A buffer overflow vulnerability exists in Microsoft's Windows Workstation Service (WKSSVC.DLL).

A remote attacker could exploit this vulnerability to execute arbitrary code or cause a denial of service.



Carnegie Mellon Software Engineering Institute
CERT® Coordination Center
Home Site Index Search Contact FAQ
vulnerabilities, incidents & fixes | security practices & evaluations | survivability research & analysis | training & education

Options
Advisories
Vulnerability Notes Database
Incident Notes
Current Activity

Related
Summaries
Tech Tips
AirCERT
Employment Opportunities

more links
CERT Statistics
Vulnerability Disclosure Policy
CERT Knowledgebase
System Administrator courses
CSIRT courses
Other Sources of Security Information
Channels

CERT® Advisory CA-2003-28 Buffer Overflow in Windows Workstation Service

Original release date: November 11, 2003
Last revised: --
Source: CERT/CC

A complete revision history is at the end of this file.

Systems Affected

- Microsoft Windows 2000 Service Pack 2, Service Pack 3, Service Pack 4
- Microsoft Windows XP
- Microsoft Windows XP Service Pack 1
- Microsoft Windows XP 64-Bit Edition

Overview

A buffer overflow vulnerability exists in Microsoft's Windows Workstation Service (WKSSVC.DLL).

A remote attacker could exploit this vulnerability to execute arbitrary code or cause a denial of service.

I. Description

Microsoft's Security Bulletin MS03-049 discusses a buffer overflow in Microsoft's Workstation Service that can be exploited via a specially crafted network message.

According to the eEye Digital Security Advisory AD20031111, the vulnerability is caused by a flaw in the network management functions of the DCE/RPC service and a logging function implemented in Workstation Service (WKSSVC.DLL). Various RPC functions will permit the passing of long

More Examples of CERT Advisories

CA-2003-20 :W32/Blaster worm

August 11, 2003

The CERT/CC is receiving reports of widespread activity related to a new piece of malicious code known as W32/Blaster. This worm appears to exploit known vulnerabilities in the Microsoft Remote Procedure Call (RPC) Interface.

CA-2003-15 :Cisco IOS Interface Blocked by IPv4 Packet

July 16, 2003

A vulnerability in many versions of Cisco IOS could allow an intruder to execute a denial-of-service attack against a vulnerable device.

CA-2003-11 :Multiple Vulnerabilities in Lotus Notes and Domino

March 26, 2003

Multiple vulnerabilities have been reported to affect Lotus Notes clients and Domino servers. Multiple reporters, the close timing, and some ambiguity caused confusion about what releases are vulnerable. We are issuing this advisory to help clarify the details of the vulnerabilities, the versions affected, and the patches that resolve these issues.

My iSeries is Secure:
Why do I care about these CERT Advisories?

- **Your iSeries is so versatile: it can be running vulnerable applications/servers like:**
- **Netserver**
  - **Scan for viruses in mapped network drives**
- **Lotus Notes or native SMTP Server**
  - **SMTP servers are often targeted (SPAM, DoS)**
- **APACHE Web Server**
  - **Web servers are always a target for attacks**
- **IXS or IXA**
  - **If you are the administrator for these Windows servers, you know you have your hands full keeping them secure!**

My iSeries is Secure:
Why do I care about these CERT Advisories?

- DNS, DHCP servers (yes, the iSeries does this too)
- Telnet, FTP servers, etc, etc
  - All of theses TCP servers have the potential to be compromised
- LINUX, AIX

- Are you running LINUX or (future) AIX as a guest operating system?
  - These are 'open' systems, and may be running servers of their own (SMTP, TELNET, HTTP, FTP, Etc)
  - All of these open system implementations will be vulnerable, just as they would be on a stand-alone system
- It looks like this security stuff really is a full-time job!

SPAM OUTPACES LEGITIMATE E-MAIL

How bad is the spam problem? According to META Group, between 60 percent and 70 percent of all inbound

corporate e-mail is spam, and about 50 percent of SMTP server processing capacity is devoted to spam.

In addition to distracting users from legitimate e-mails -- and how many of us haven't deleted an important e-mail

while trashing our spam -- spam gobbles up about 40 percent of users' storage capacity, META Group reports.

SPAM & DNS Info at these Web Sites

- **www.spamcop.net**
- **www.ordb.org**
- **Many other sites**
  - **List of blacklisted (spamming) mail servers**
  - **Allows you to test a (your) mailserver via IP address**
- **www.dnsreport.com**
  - **Allows you to test for connectivity to a mailserver and to a particular mail userID**
  - **Will present a very thorough report on DNS issues**
    - **Some email issues are caused by DNS problems**
    - **For example, having no reverse DNS lookup for your MX record**

iSeries security is an on-going Process

- **You must allocate some time each week into implementing your security policies**
  - **iSeries administrators need to be more proactive**
  - **If you are also responsible for Network security administration and Windows administration, iSeries probably gets lower priority because of it's reliability, but….**
  - **Don't become complacent! Some of your greatest corporate assets are on that iSeries!**

"Hey…. That was funny!"

The End

Related Publications

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 @server iSeries Server with Windows 2000 VPN Clients	REDP0153
IBM @server 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 @server 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 @server 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