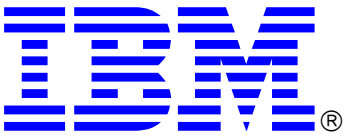
Securing an SNA Environment

for the 21st century



Written by the IBM z/Center of Excellence:

Thomas Cosenza: z/Security Consultant, CISSP

Joe Welsh: z/Networking Consultant

Jerzy Buczak: z/Networking Consultant

Acknowledgments:

Sam Reynolds: IBM Enterprise Networking Solutions Architect

Linwood Overby: IBM Enterprise Networking Solutions Architect

Chris Meyer: IBM Enterprise Networking Solutions Architect

Executive Summary:

In today's ever expanding business environment, securing an IT Enterprise takes center stage in an overall business strategy. However, over the past 10 years as the IT industry has enhanced security defenses and best practices for IP networks, configuring secure SNA environments have not been pursued with the same zeal for some organizations.

In the past, SNA system programmers were able to rely on a hierarchical architecture, strong physical controls, and a limited amount of access to protect their critical business services. As the IT industry developed more available and larger networks these assumptions no longer held true. The introduction of new technologies that enhanced the availability of SNA with more dynamic network recovery and the use of the faster IP infrastructure has "opened" the SNA networking environment. This loss of physical security, however, can be replaced by a mix of other strong security options that are integrated into the different layers of SNA.

SNA at its core was designed with the ability to wrap different layers of connections with a blanket of security. In order to communicate within an SNA environment you would first have to connect to a node and establish and maintain a link connection into the network. You then have to correctly negotiate a proper session and then handle the flows within the session itself. At each level there are different security controls that can govern the connections and protect the different session information.

From network border searching controls, tuning options and the use of encryption ciphers you can harden the walls of your SNA network from the inside out. Even the data transactions themselves are wrapped within several layers of connections that can contain multiple security checks.

SNA has even been enhanced to take advantage of IP networking technology. In order to protect the transactions that flow over an IP network you can deploy a mix of SNA and IP security. Using these controls, you can authenticate the traffic flow and protect the data from prying eyes using advanced key management and cipher algorithms.

This paper will educate the SNA-skilled reader on network security concepts and includes recommendations for evaluating and configuring their APPN and Subarea SNA environment to enhance security.

This paper is broken up into 7 sections as follows:

1. Basic IT Security
2. Policy Security Recommendations
3. VTAM Controls
4. Enterprise Extender

5. Other Connections
6. Dealing with Searches
7. Application Security

Who this paper is written for

This paper is written for those who need to evaluate and implement security on an SNA network. The reader should be knowledgeable in z/OS concepts and have at the very least a basic understanding of Encryption, IPSec, and the System Authorization Facility on z/OS.

TABLE OF CONTENTS

Chapter 1: Introduction to Security Concepts

Introduction

SNA continues to be a key network technology of enterprise business infrastructures around the world. SNA has evolved from subarea environments that used dedicated leased lines to connect special communication hardware into a complex mesh architecture using peer to peer networking. This new environment has allowed for higher availability using dynamic session routing techniques, new transmission media types, and the ability to tunnel SNA traffic through IP networks. Information that traverses SNA networks today varies from financial records, personal medical records, as well as other data, and will continue to do so for many years to come.

Over the past 20 years, corporations have been moving toward the IP network business model for their new applications. As part of the growth into IP networking, properly securing the networking connections has become a major focus of the IT industry. While this is a positive step, many enterprises have not necessarily taken the same steps to secure their SNA data flow. There are several reasons for this. Historically SNA has not experienced many of the same exposures as IP. This paper was written with the intent to educate the reader on the importance of reviewing their SNA environment and the options that they have to provide a better, more secure environment.

IT Security at a Glance

Understanding what the IT security requirements are and what types of controls are available has become vital in creating a cohesive security strategy. While this paper does not intend to give the reader a deep understanding of all security concepts, the writers do want to convey the basics of IT security to the novice reader.

Breakdown of Security Controls

In general, all IT security falls into one of five categories; physical, policy, platform, application, and network.

Physical Security – Physical security consists of the actual controls that exist to protect the computer hardware system. They can range from low tech, such as human security guards, to the latest technologies, such as retinal scanners. Within the scope of System z, the physical controls deployed should make sure that the production level System z hardware systems are protected in a high security area with controlled access.

In the past, SNA relied on very strong physical controls. High end server systems were in locked computer rooms with controlled access. Network cables were in locked wiring closets to prevent unauthorized tapping of transmissions. As networks have grown and demand for 24 by 7 services has increased, the SNA environment also had to change. SNA networks have become more open and the more open access that is allowed by the

physical infrastructure the more the other four security layers will have to come together to ensure protection of the SNA resources.

Policy Security – Policies can be used to restrict resources or data to certain individuals. Security polices allows an enterprise to set limits to the level of access to different resources within the network. This allows an enterprise to protect its resources in a way that relates to the goals of the organization. Within the SNA realm, this could be related to defining rules for which peer nodes are allowed to connect within the peer network or how users will be able to interact within the system. These policies will encompass every type of access available within an SNA environment to the way other businesses will be allowed to access resource within your network.

Platform Hardening - Application platforms are required to deliver data in a secure, reliable fashion, with assurances that data integrity, confidentiality and availability are maintained. One way to attain this assurance is to ensure the platforms are installed and maintained in a manner that prevents unauthorized access, unauthorized use, and disruptions in service. The process of assuring data integrity, availability, accountability and preventing unauthorized access to resources is called platform hardening. Within the scope of this paper, it could be applied to the use of EE Models as points of entry or access security to the VTAMLIB dataset.

Network Security – Network security controls protect system resources from external security threats and the data while it is being transmitted. Some examples of network security controls for the IP world include IP Filter tables, Virtual Private Networks (VPN), and Proxy servers. When discussing SNA network security, this could include but is not limited to border searching controls, IPSec VPNs for Enterprise Extender (EE) links, and TN3270 SSL security.

Application Security – Application security is the step application programmers must take to protect data that is stored or sent by that application. This security is independent of what the operating system is doing. While application security is usually beyond the reach of the infrastructure pieces, within SNA one can secure transmissions between logical unit (LU) sessions.

CIA^2 of Security

The types of security discussed in the previous section are used to satisfy an enterprise need for the CIA^2 of security. CIA^2 stands for Confidentiality, Integrity, Authentication and Authorization which are the main requirements one aims at achieving through security.

- Confidentiality is the assurance that data in any state cannot be understood by those for whom the data is not intended.

- Integrity is the assurance the data has not been tampered with between the sender and the receiver or while the data is at rest.

- Authentication is the process of providing assigned or personal credentials to allow a subject access to computer resources.

- Authorization is what the authenticated user is allowed to access on a system.

These four concepts working in concert give an enterprise a solid foundation to do secure business in today's day and age.

SNA vs IP security

In the context of this paper, the difference between SNA and IP is that SNA was originally designed with security functions in mind, whereas IP has had to add security aspects into the protocol over time. The result of this is SNA has many basic security features built in. This ground up approach to the SNA design makes it difficult for an attacker to steal information or to mount a Denial of Service (DoS) attack. On the other hand, IP has recently been augmented with some very clever and sophisticated techniques which, when implemented properly, will provide a very high level of security. What this means is that:

- To make SNA **moderately** secure, very little effort is needed. To make SNA **very** secure, you need to make an extra effort.

- To make TCP/IP **moderately** secure, you need to make an extra effort. However you can get to be **very** secure for very little extra effort.

However, in the current business environment a system programmer should do their due diligence to ensure their system is secure to the level that their business requirements dictate.

There is no SNA Internet

Most, if not all, SNA networks in the world are interconnected in some way. SNA began as internal communications within a corporation and then grew to allow cross-corporation communication. (This was true before TCP/IP and the Internet became popular.) There are configurable SNA controls to allow and disallow connections for these cross-corporation communications.

The points of interconnection between SNA networks can be considered control points for security. For example, there are no "SNA ISPs" to which a private individual may connect for a fee and gain access to the world wide SNA infrastructure. SNA networks can be thought of as a group of interconnected islands. This is a direct opposite of IP connections where free wireless access points and other free internet access points are commonly available and provide access to the whole Internet. With SNA, a hacker must be already connected to some organization's SNA network to access the SNA topology.

The advent of Enterprise Extender[1] has somewhat blurred the distinction between the IP Internet and the set of connected SNA networks. Now it is easier for a hacker to use an IP ISP and attempt to connect across the Internet to a given SNA/EE node. However, SNA is still a peer to peer network and is designed to require definitions for any terminal already to connect to the SNA network.

In summary, it is not impossible for a hacker to connect to a suitable SNA "gateway"; however, due to the peer to peer nature of SNA it is rather more difficult than doing it for IP. Most hackers will attack an organization through its IP interfaces simply because it is easier.

SNA is connection oriented

In SNA, it is not sufficient to send lots of masqueraded or malformed packets into the network and hope that some of them find their mark. A hacker must:

- Establish a layer 2 connection to an SNA node
- Establish a layer 4 connection (an SNA session) to that same node, in order to make use of SNA routing functions.
- Establish an SNA session to the target application, in order to send it some bad data.

All these connections mean not only is it more difficult to hack into SNA, but also that there are more opportunities for organizations to build in defenses against intrusion. The local ("SNA gateway") node must be able to accept connections at two different layers:

- The remote (application owning) node must permit a session request from the hacker's node
- The application itself must allow a session from the hacker's machine.

Only after these steps are completed would a hacker reach the user authentication step that is present across almost all SNA applications.

SNA Encryption vs. IP Encryption

Within a SNA environment, there are currently two types of transmission links that can be used. Those that run over native SNA links (XCF, AHHC, Ethernet, Token-ring) and those that tunnel in some way over IP links (TN3270, DLSw, Enterprise Extender). IP encryption and key management techniques have advanced at a much greater pace then those within the native SNA environment. However, SNA encryption can still be used to secure the whole path of an LU to LU session regardless of the number of hops in between the LU partners, whereas IP security can only protect portions of the path that

[1] Enterprise Extender allows for SNA data to flow over IP networks.

traverse IP. So the question comes, “Which type of encryption should I use, SNA or IP?”

SNA certainly supports encryption and authentication and can be used to protect the entire LU-LU session path. However everything is done using manually defined symmetric keys. The passing of the key, refreshing and distribution takes a large amount of manual effort and must be guarded from hackers.

IP security such as IPSec is limited by the length of the LU path that it can protect however an administrator can take advantage of the dynamic key distribution, refreshing, or standard cipher key protections through the use of Certificates and protocols such as the Internet Key Exchange (IKE). IP security can also take advantage of AES encryption that is quickly becoming the new standard for encryption ciphers compared to SNA which can only support up to Triple Des encryption when this paper was written.

Using encryption from the IP or SNA side or a mix of both will be up to a corporation’s governance on risk and what the requirements are for securing the SNA data.

SNA Hacker Profiles

The number of people within the industry that know about SNA network and flows has diminished over time. This same trend affects the hacking community as well since many “script kiddie”[2] and amateur hackers do not have the knowledge to attack a SNA network. For these groups, it is more desirable to attack the simpler IP infrastructure of a network than to attack the more sophisticated SNA network.

However, unorthodox governments and crime syndicates have the money and the resources to find career IT criminals that have the sophistication to attack a SNA network in order to find a big prize. This is why it is always important to protect your SNA network with the technology provided within the protocol itself.

Other Platforms

While this paper is very System z centric in its approach, many of the options here are also available in some form or another on other platforms. Using this white paper as a guide, you can adopt many of the techniques described here to the platform you are using.

[2] Script Kiddie are those hackers that have a limited IT knowledge and use tools found on the internet to create attack scripts.

Chapter 2: Basic IT Policies

Basic IT Policies as they relate to System z

An IT security policy is the organizational statement of how an enterprise is going to protect the data and communications used for the day to day operations. The creation and enforcement of any governing policies that secure your systems fall to the directors and CIOs of your organization. Independent of any technical controls, a business security policy should state a specific set of intentions and conditions that will govern how a company's assets are accessed and used.

The security policies deployed are the essential foundation for any comprehensive strategy whether it be SNA, IPv4 or IPv6 networks. In most cases, it is unclear policies and not technology failures that lead to successful IT attacks. This section will discuss some of the policy security topics that directly relate to SNA networking and have been shown as needing attention after an audit in several customer environments.

Separating Different Environments

One of the most basic principles in IT security is to keep unrelated systems detached from one another. In most datacenters, there are three types of systems deployed for every day IT use.

- There are the standard production systems for the core business
- Development LPARs used as a test bed for developers to create new application

Test systems made up of one or two LPARs to allow system programmers to test changes. These three sets of systems all have very different uses and usually different security levels. Many times the test systems users have much more access to the operations of System z than what would be given in the production or development systems.

Enterprise environment systems should be separated from each other, ensuring there are no SNA connections between any of these systems. By not allowing these systems to be connected via SNA, you do not expose the enterprise to unknown security risks due to systems that do not have the same level of security.

Separation of Duties

Separation of duty, as a security principle, has as its primary objective the prevention of fraud and errors. This objective is achieved by disseminating the tasks and associated privileges for a specific business process among multiple users. This principle is demonstrated in the traditional example of separation of duty found in the requirement of two signatures on a check. This principle also applies in the System z environment and there are different roles that are played out every day within a System z datacenter. There are two main roles that should not be performed by any one person. These are the roles of system programmer and system operator.

- The system programmer in an enterprise is the person or a group of people responsible for configuration of the z/OS operating system.

- The system operators are in charge of running the systems, starting and stopping procedures, and making sure the health of the systems are preserved.

Need to Know Concepts

One of the standard IT security concepts is "Need to Know". The idea is to ensure the personnel accessing any type of object (data, process, etc) have a particular and defined business need. Restricting the operation of the SNA resources to only those operators and systems programmers is a big part of the security solution. Without proper policies in place to secure your systems, any security devices can be circumvented. No one entity should have access to more business processes than is required. This concept can also be expanded to include the nodes that are connecting within an SNA network. Making sure only authorized nodes can connect, search, and create sessions to other resources is critical to any proper security policy.

Proper Cleanup of Resources

One type security flaw that could be found within SNA networks are old atrophied definitions for resources still being started within the SNA network but that no longer physically exist. Although these resources were decommissioned for one reason or another, all the changes were not made to remove the definitions from the system. This could open a security hole where a hacker could exploit these resources and gain access into the peer network. It is vital to the security of an organization that proper procedures are in place for decommissioning of resources in your network and also that a constant audit is conducted of what resources are currently active on all of the systems.

Chapter 3: SNA Network Controls

Start Options

VTAM has a number of start options relevant to security. They can be divided roughly into two areas:

- Cryptographic security techniques
- Access to SNA resources

Crypto-based Start Options

The start options that fall under this heading are ENCRYPTN, ENCRPREF, VERIFYCP and SECLVLCP. The latter two, VERIFYCP and SECLVLCP, are described in the "Application Security" section, since they involve session partner authentication, albeit for CP-CP sessions. Use these options if there is a requirement to verify the authenticity of any partner CP that is currently connected to the node and the link is not an Enterprise Extender connection where the stronger IPSec authentication can be used.

ENCRYPTN determines whether VTAM supports cryptography and which cryptographic API it expects to use. ENCRYPTN should be set to ENCRYPTN=CCA in the ATCSTRXX member. This will allow VTAM to use Triple DES encryption[3], which is the only cipher suite the writers recommend. The value of CCA also means that, upon initialization, VTAM will attempt to detect what crypto products are available, and decide the level of encryption that will be supported. Prior to the activation of the LPAR you must have installed the cryptographic hardware and had your service group define it properly on the System z hardware. In turn, ensure ICSF is properly configured and active on each LPAR where encryption is to be used. It is recommended, though not necessary on any supported z/OS release[4], that you start ICSF first before any other subsystems are started.

ENCRPREF merely defines a prefix used for the labels of the master symmetric keys used to generate the session keys. Its default is "no prefix" and there is no reason to change this.

Access Control Start Options

These start options establish default behavior for permitting or denying access to this system from remote SNA resources. They can be regarded loosely as the SNA equivalent of an IP packet filter, bearing in mind the difference between connectionless

---

[3] At the time this paper was created Triple DES is the strongest cipher available.

[4] Note that in earlier releases of z/OS ICSF had to come up before VTAM. However any of the supported z/OS releases (V1R7 and above) do not have this requirement

IP and connection-oriented SNA. Many of these options have the characters "DYN" within their names. They can be overridden on individual connections by means of VTAM definitions, but the start options allow you to define the SNA equivalent of that "deny all" filter that protects against anything you forgot to think about.

There are seven options to be considered, roughly falling into three categories:

- APPN-related option are DYNADJCP and CDRDYN
- Subarea-related options are CDRDYN, DYNASSCP and SSCPDYN
- LEN-related options are DYNLU, ALSREQ and CPCDRSC

CDRDYN controls whether this VTAM is allowed to create dynamic CDRSCs that represent resources owned by other nodes in the network. CDRDYN can be specified as a start option or on the host CDRM definition; if the CDRDYN start option is specified, then it overrides the CDRDYN value specified on the host CDRM definition. CDRDYN is set to YES in virtually all VTAM installations because of the huge amount of work that would be involved in predefining all remote session partners as CDRSCs.

DYNADJCP (defaulting to YES) allows VTAM to define adjacent control points (CPs) dynamically. The behavior of this option can be overridden on individual link station definitions. If DYNADJCP is set to NO, each adjacent CP to which VTAM connects must be defined in an ADJCP major node. You need to balance the security requirements against the effort in coding definitions. Code DYNADJCP=NO as a start option, override it on those link stations (for example, Switched major node PU) where you are sure of the identity of the nodes that are native to VTAM, and code an ADJCP major node for any remaining valid partners.

DYNLU controls whether VTAM allows resources represented by dynamically created CDRSCs to use peripheral (type 2.1) links. If DYNLU for a link is coded to NO, either at the start option level or on the link definition, then you will have to code Cross Domain Resource (CDRSCs) definitions on that host for every cross-domain resource that is allowed to use peripheral (type 2.1) links for sessions. While this may sound like a good way to control the use of external links, this method will work only when ISR routing is used on the link. If you allow HPR, which is recommended for APPN links, the value coded for DYNLU is not an effective security control.

SSCPDYN allows VTAM to add a known partner CDRM to any adjacent SSCP table if that partner sends in a session request. DYNASSCP lets VTAM create adjacent SSCP tables dynamically.

Unlike the APPN case, there is no way that VTAM will search a partner CDRM unless that CDRM has been predefined. To minimize the likelihood that one of those partners has been hacked and now contains a bad LU, set both of these start option to NO to ensure each resource is searched ONLY in the partner networks where it is expected to be found.

ALSREQ and CPCDRSC are only useful when you have LEN connections to this VTAM, which should be a rarity these days. ALSREQ (default NO) allows VTAM to accept session requests from a remote LU that has not been predefined with the correct link station name. CPCDRSC (default NO) allows VTAM to establish outbound sessions to an adjacent LEN CP that has not been predefined. Because of the lack of authentication methods for LEN resources (there are no CP-CP sessions), optimum security demands that everything is predefined (ALSREQ=YES and CPCDRSC=NO) unless there is business justification for doing otherwise.

DYNPU and Connection Networks

One other definition in this category, although not implemented as a start option, is DYNPU. DYNPU (coded as a keyword on line group definitions, and usually defaulting to NO) allows a remote link station to be created dynamically without matching a switched major node PU definition. For EE connections, where most inbound switched PUs may be found these days, set DYNPU as NO and predefine all connection partners. A Connection Network is exempt from the DYNPU rules, by design, however a Connection Network also cannot be used to carry CP-CP sessions and thus a Locate for a session would never flow over this type of link. This requires that any node participating in the Connection Network will always have been predefined by another switched major node to its NN server and thus can be authenticated by that server. Also due to the nature of Connection Networks, described in the next section, you can control the network flow using IPSecurity controls such as encrypted tunnels or IP packet filters.

Chapter 4: Enterprise Extender

Introduction

Enterprise Extender is the latest technology available to enable companies to transport SNA traffic over an IP network. Enterprise Extender is not a product but an extension to the Advanced Peer to Peer Network (APPN) and High Performance Routing (HPR) protocol. Enterprise Extender technology provides an encapsulation of SNA application traffic within UDP datagrams by HPR-capable devices at the edges of an IP network. To the IP network, the SNA traffic is broken down into UDP datagrams that are transmitted through the IP backbone. To the end user, it is a normal SNA session with the same Class of Service (COS) as in a traditional SNA network. By wrapping the SNA application traffic in this way, Enterprise Extender enables SNA data to be carried over an IP backbone without changing either the SNA application or the IP infrastructure.

Types of Enterprise Extender Connections

There are 2 types of Enterprise Extender Connections:

1. Simple Enterprise Extender connection (static or dynamic PUs)

2. Connection Network

Regardless of the type of Enterprise Extender connection defined on z/OS, each requires the creation of a VTAM External Communication Adapter (XCA) major node to identify at a minimum:

- UDP Ports for Enterprise Extender traffic are located at port numbers 12000-12004 in the stack and should not be changed

- Type of Service (TOS) settings in the IP Header (defaults are 20, 40, 80, C0) that are used to define the SNA Class-of-Service defined by the application as the SNA traffic crosses the IP network. The IP routed network must honor the IP TOS settings (as does the OSA Express in QDIO mode) to preserve the priority of the SNA traffic; otherwise, there is no priority.

- Logical Data Link Control Timers

Simple Enterprise Extender Connection

A simple Enterprise Extender connection on z/OS is defined with a VTAM switched major node using either static (predefined) PU definitions or dynamic PU definitions. When comparing the two strategies, static (predefined) PU definitions are the most secure but do require more initial effort by the system programmer.

Static (Predefined) definitions	Dynamic definitions
Individual definitions to each remote Enterprise Extender endpoint which include remote CPNAME	No individual PU definitions are required
More secure	Less Secure
Ability to specify unique naming convention for SNA resources	Limited ability to specify naming convention for SNA resources
Flexibility to define different link characteristics per remote peer	Less flexibility to define link characteristics
Ability to use TG numbering to identify remote partner	No control over TG numbering prior to Z/OS V1.10
May require large number of definitions to configure and maintain	Minimal definition

By using the static (predefined) method, the system programmer can define the expected CPNAME, TG Number, and link characteristics for each remote Enterprise Extender endpoint. Only those connections calling in matching those traits will be successful and thus provide some level of security. Static definition is the most common Enterprise Extender implementation method.

While the dynamic method requires the least amount of configuration, it also provides the least amount of security to validate the remote Enterprise Extender endpoint.

Enterprise Extender Model major node

In the event the dynamic method is selected for defining Enterprise Extender PUs, one should consider deploying a model major node. A model major node provides the ability to define specific TG characteristics of the dynamic PU (such as assigning the TG number), DISCNT=NO to preserve the EE physical link after the last session terminates, and automatic redial capabilities in the event the of link failures.

Also, due to the dynamic nature of the connection from the SNA perspective, it may be prudent to add some IP controls. These could be in the form of IPSec Policy Filters or Virtual Private Networks (VPN) to ensure that the proper IP address of the incoming Enterprise Extender connections are allowed to connect to this node.

Connection Network

A Connection Network is an Enterprise Extender implementation across a shared access transport facility (SATF) (i.e. IP network) involving a common virtual routing node (VRN) for communication. Each participant in the Connection Network defines the same VRN name. When a session request is initiated, an Enterprise Extender dynamic link is created as needed between the endpoints if they share the same SATF. The benefit an organization receives by using a Connection Network is the linear growth of definitions in a fully meshed network compared to exponential growth. In other words, Connection Network requires 2n definitions while the alternative method requires n(n-1), with n nodes. However, this dynamic environment can be exploited since there are no controls on who can connect to a Virtual Routing Node. As mentioned in the section titled "

Start Options", there are several methods in both IP and SNA to control access to the host through a Connection Network. These controls are discussed in detail below.

UDP/IP Considerations

As described, Enterprise Extender encapsulates the SNA data into UDP datagrams to route across the IP network. For each IP stack acting as an Enterprise Extender end-point, configuring the following changes are recommended for UDP.

- Reservation of UDP ports 12000-12004 for the Enterprise Extender traffic
- Sufficient UDPRCVBUFRSIZE and UDPSENDBFRSIZE sizes. (Recommended to use 65535 for both parameters.)

Network Address Translation Considerations

Network Address Translation (NAT) introduces additional complexity to an Enterprise Extender solution in that each endpoint must establish connectivity to a static IP address; therefore, only static NAT is supported. Dynamic NAT is not appropriate because an IP address needs to be determined at configuration time. If a NAT boundary is associated with an Enterprise Extender Connection Network path, hostname-based Enterprise Extender definitions are required to establish network connectivity because the EE Connection Network protocol carries IP identities within the payload.

Enterprise Extender IP Security

Enterprise Extender allows for SNA data to be encapsulated within UDP datagrams and transferred over an IP network. As a result, Enterprise Extender traffic is exposed to the same threats as all other IP traffic. There are two methods that can be used to control the UDP/IP traffic: Policy Filters or IPSec.

The simplest way is by using a packet filtering firewall to only allow EE UDP datagrams in from particular IP addresses for UDP ports 12000-12004. This can be done from the router or at the host using the IPSec Policy Filters. These filters use information in the IP and UDP packet headers to either permit or deny the traffic coming into the enterprise. Usually this is deployed when an encrypted data link is already deployed; usually through

a third party ISP, to connect two organizations. Since there is no need to encrypt or authenticate the traffic coming across this link, packet filtering is sufficient. However, since the security end point is moving closer and closer to the end points of a connection, simple policy filters are no longer sufficient.

To truly authenticate and encrypt the Enterprise Extender traffic from both endpoints, IP Security via IPSec is the recommended solution. IPSec is an industry standard protocol that provides end-to-end authentication and encryption. It is available on multiple platforms, including but not limited too, z/OS, ISeries, Linux for System z[5], Cisco Routers with the right IOS, and Windows. Using IPSec, the Enterprise Extender traffic can be protected as it flows over the non-secure portions of the network or the complete EE connection path.

To enable IPSec on z/OS Communications Server, a Policy Agent daemon (PAGENT) started task must be configured with a set of policies. The IPSec policies define the IP addresses of the Virtual Private Network (VPN) endpoints, UDP ports for application data transmission, cipher suites used for encryption and authentication, the location of the server and certificate authority (CA) certificates, and when encryption keys should be refreshed. The server and CA certificates enable the VPN endpoints to authenticate their identity and to thus negotiate a dynamic tunnel connection for encrypting the Enterprise Extender traffic. Once the dynamic tunnel is established, all Enterprise Extender traffic transmitted within the VPN will be protected from unauthorized access. This is the recommended way to protect Enterprise Extender links.

[5] A third party firewall product should be used in this case

Chapter 5: Other Access Controls

TN3270 Security

This section will deal with the most common type of user connection into an environment, the TN3270 connection. This section will discuss the ways a system programmer can modify the TN3270 server to create an environment that meets their security requirements.

TN3270 Background

The most common way to connect from workstations to SNA applications in a traditional enterprise is to use a 3270 session. This connection in the past was made via hardware devices like a 3290 through a front end processor such as a 3745 and into the MVS host. This type of connection, by its very nature, provided a great deal of physical security since these links were protected within the infrastructure of a building. However, as time went on and the requirement for greater availability to applications grew, the TN3270 emulation was created. This is a session that runs over a TCP/IP network connection and can take the form of anything from a basic green screen to a complex set of web pages created by Host Application Transformation Services (HATS). While this has done wonders in modernizing how enterprise networks can scale their business, it has also opened up security issues within an enterprise.

Since the 3270 emulation mimics the same data flows as the older hardware, it also took on many of the assumptions that the hardware did. The 3270 emulation passes many critical pieces of data in the clear, including user id and password information. Also this emulation at its base has no way of authenticating where the session originated from. These were reasonable assumptions since prior to TN3270 all connections were hardware based, the origins of these sessions were well known, and the transmissions from these terminals could not be easily captured. In today's environment of shared Ethernet LANs and wireless networks the assumptions made back in the 1960's and 70's no longer hold up.

Securing TN3270 IP Flows

It was critical that security technology be deployed within a TN3270 session flow to replace the security that was lost in this new environment. The Secure Socket Layer (SSL) protocol was enabled within the TN3270 server to provide the ability to add encryption and authentication for a session. The SSL-enabled TN3270 server protects all data between the server and TN3270 SSL-enabled Clients such as IBM Host On Demand and PCOMM.

SSL/TLS Support

The SSL/TLS protocol can provide data encryption, data origin authentication, and message integrity for TCP applications in a network. This is accomplished using X.509 certificates which can be used to trade an encrypted session key using Asymmetric encryption Figure 1 shows a high level view of how SSL/TLS negotiates a session

between an enabled host and client. Properly negotiated SSL sessions will, by default, authenticate the server to the connecting client. Optionally the server can be configured to request a certificate of the client to authenticate itself to the server. Also during the handshake, security session parameters, such as cryptographic algorithms, are negotiated and session keys created. After the handshake, the data is protected during transmission with data origin authentication and optional encryption using the session keys.

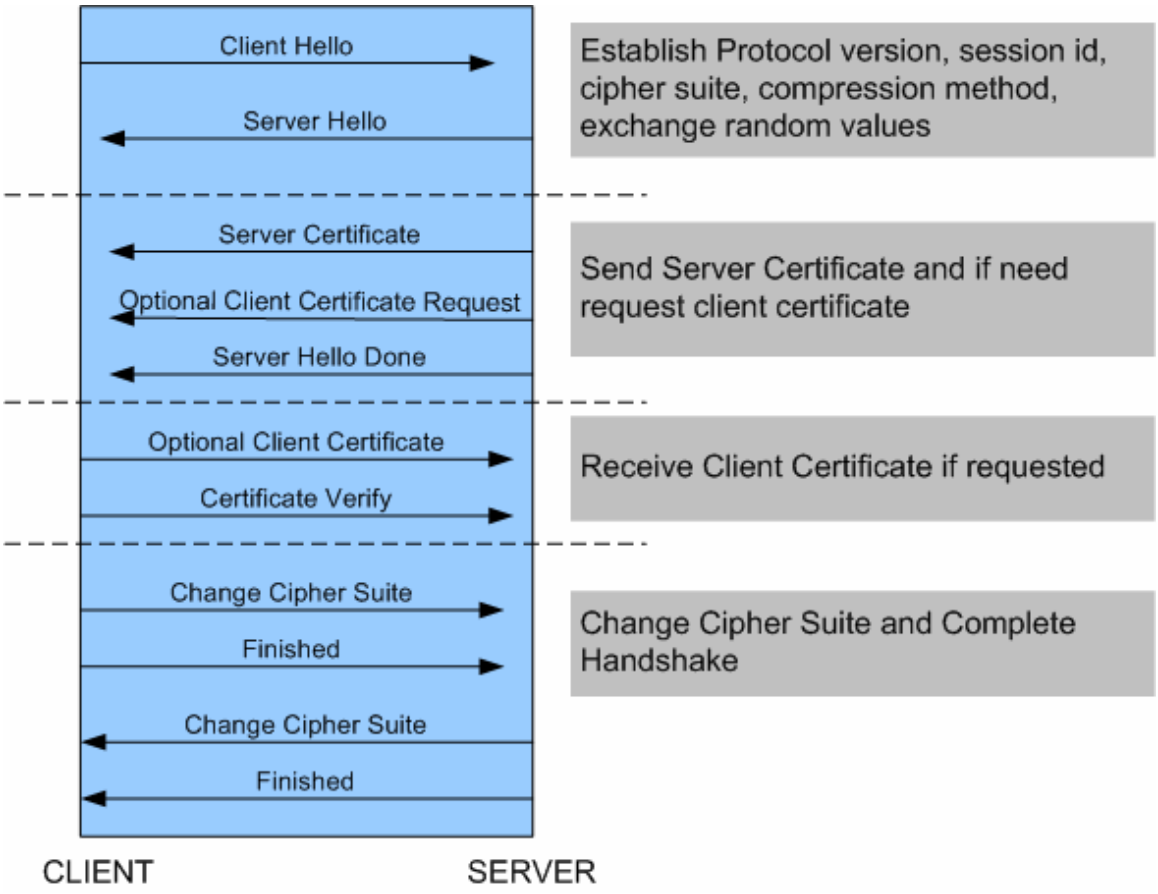


Figure 1 Typical SSL Negotiation

The cryptographic algorithms that are used for an SSL session are based on the algorithms the server and client negotiate during setup time. During the SSL handshake, the client and server exchange a list of algorithms. The algorithm selected is based on the best match between the client's list and the server's list. The selectable algorithms can be limited by configuring a subset of allowable algorithms at the server. Servers can support encryption using Triple DES as well as other encryption algorithms (RC2, RC4, DES, and AES). A hardware crypto coprocessor, if available, is used for DES, Triple DES, and AES encryption.

TN3270 SSL Support

SSL/TLS support for the z/OS TN3270 server can be implemented either within the server itself, or (recommended from Release 9 onwards) using an AT-TLS Policy with the Policy Agent. In each case, the function is much the same but the definitions are coded differently.

Each flavor of SSL/TLS support provides two different forms of the SSL/TLS negotiation.

- Server side SSL/TLS
- Client Authenticated SSL/TLS

The major difference between them is whether a certificate is required to authenticate the client to the server. In most cases, the certificate distribution for client side authentication is a very administrative-heavy process and the normal userid logon for applications is enough; however, it is a really strong authentication technique and goes a long way towards satisfying security standards such as SOX, PCI DSS, or HIPPA. The server can also dictate the level of encryption and in the case that the server should not allow any encryption below the Triple DES encryption.

Currently there are 3 versions of the SSL/TLS negotiation. TLS, SSLv3 and SSLv2 are different application layer security negotiations that can be used. While TLS and SSLv3 are extremely similar the SSLv2 protocol is much older and not considered as secure. Within the server, there are controls to limit what the client can request relating to the version of SSL negotiation should be. Do not allow SSLv2 negotiation by using the NOSSLV2 option.

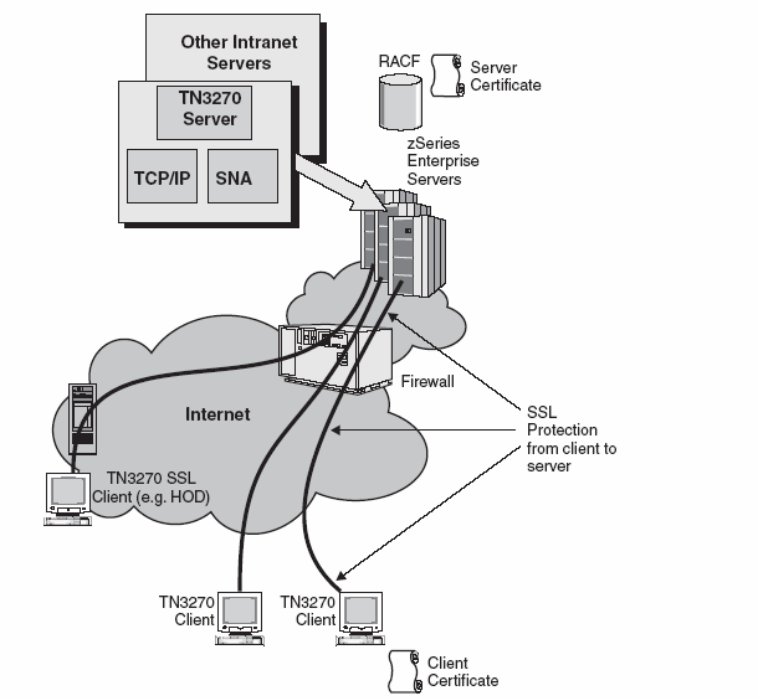


Figure 2 TN3270 SSL negotiation

Client Access Controls

The TN3270 server on z/OS has extra controls to secure access across the SNA network. Using a mixture of LUNAME, IPGROUP, and PORT statements, a system administrator can control which applications a user can access. For example, if you wanted to ensure only a particular set of users can access a sensitive application, you can design your TN3270 server to allow access only to the application via a specific TCP port on the stack. You can add security to the port using the SSL/TLS support mentioned above to ensure the confidentiality and security of the application.

Securing DLSw Connections

Enterprises still using SNI (FID4 connections) or boundary (peripheral) links to devices may use DLSw to communicate between organizations. While this is a good way to communicate between two different NetIDs, SNI traffic does flow across a TCP connection. To protect the data as it traverses this link, use either IPSec VPNs or a hardware encrypted channel.

Another possible security exposure can be introduced if a DLSw router is enabled into passive mode; this type of configuration would allow a hacker to connect to the DLSw router as a peer. The hacker could gain access into the SNA network without having to authenticate their system to the router. Either all DLSw peers should be pre-coded or IPSec AH or ESP Null encryption with authentication VPNs should be used to authenticate the peers at the IP level.

Chapter 6: Dealing with Searches

While connecting different business ventures is critical in the current business environment, it also has added significant obstacles in dealing with securing those connections. Within SNA, these obstacles involve controlling searches from going out and coming into the corporate network. This section will discuss techniques that one can use to control the searching behavior for applications within an SNA network.

Basics of Searching

Since most SNA networks are a mix of subarea and APPN environments it is important to understand how VTAM conducts searches. Where does a search originate from within the network? Does it come from the subarea portion of a network or does it start from an APPN node? The answer to these questions will determine how the search will be conducted and what options will come into play.

Subarea Searches

When a search for a resource comes from a subarea environment VTAM will use two start options to determine how the search will be conducted; SORDER and SSCPORD. The table in Figure 3 describes the relationship of how these two options effect the overall searching for a resource within the subarea environment.

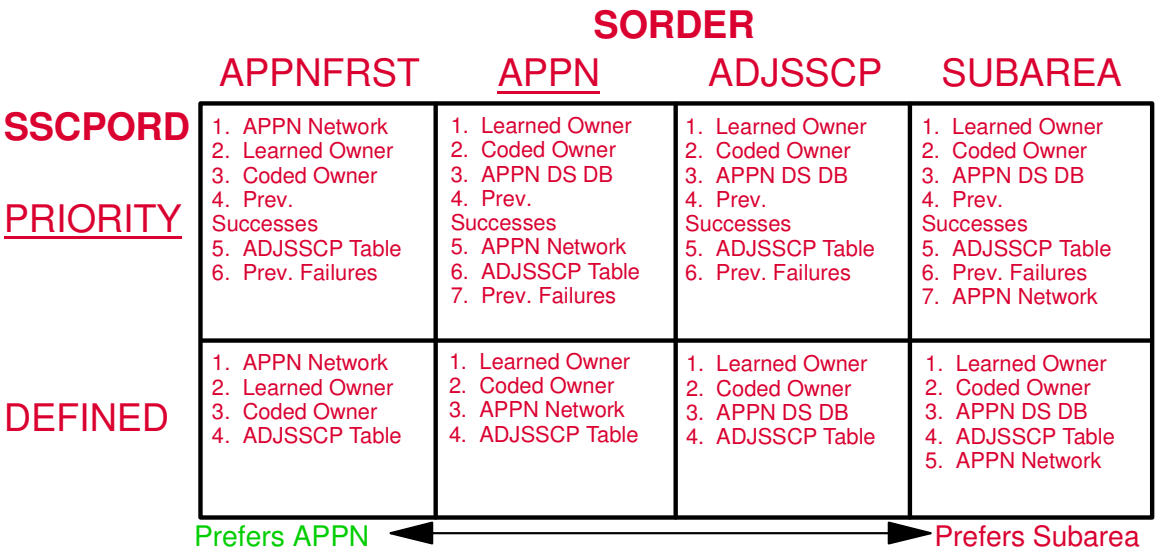


Figure 3 SSCPORD & SORDER table

While SORDER and SSCPORD control the **order** in which certain searches are performed, they do not (for the most part) control the **contents** of the search list. The contents of the search list are controlled by the adjacent SSCP tables that are defined to

VTAM and by the SSCPDYN and DYNASSCP start options. The two start options are discussed in detail in the "VTAM Controls" section above. For optimum security, you should define exactly what nodes are to be searched in the adjacent SSCP tables, and code the above two start options as NO to prevent VTAM adding any unwanted nodes into the search order.

Note that if the search is started in APPN and is now being preformed within the subarea environment all of the APPN searches are ignored in the table above.

Searching an APPN network

If a resource needs to be discovered within an APPN environment, a locate is sent into the network and is ether answered by the owning network node of the resource, by a specialized network node called a central directory server (CDS), or by a network node that is connected to another NetID called a border node. .

Within the section labeled "Start Options" the importance of using the DYNADJCP option in controlling which nodes will be allowed to create CPCP sessions with VTAM is discussed. However this does not address how you control the searches within an APPN network. Consider writing a Directory Services Management exit (DSME) for any type of APPN search that can occur in a SNA network. However, it is rare that one is concerned with the searching that occurs within the native environment. More important is the ability to control the searches that come in from and go out to other business' networks.

Controlling Searches of Other APPN networks

When searching other networks, a special type of network node called an Extended Border Node (EBN) is used. The EBN can make connections and initiate searches into other NetIDs. In most practical applications of SNA, EBNs are used to connect different enterprises together for some shared business purpose. Ensuring that an enterprise sends appropriate searches to the correct customer networks is an important issue.

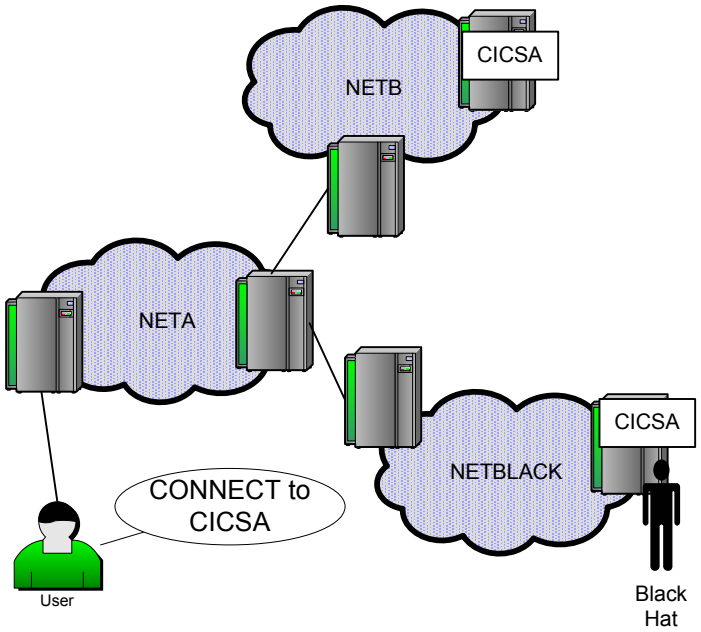


Figure 4 Searching Example

You can have a situation as depicted in Figure 4 where a company is connected to two different business partners. If searches are not sent carefully, a user within the enterprise could think that they are connecting to an application on a particular business' network; however, the user is really connecting to a hacker's application that is used to gather data such as account numbers and passwords. This section will discuss how to prevent this and other attacks by tuning your VTAM settings.

Tuning options for searching other APPN networks

There are two APPN specific start options that are used to control cross-network searches in an APPN network; BNDYN and BNORD. BNORD in an APPN network is similar to SSCPORD for SNI connections. Depending on what was coded, VTAM could take previous successful searches and uses this to then reorder the searching sequence. While this option has a minimal effect when it comes to the overall security of the SNA network it is recommended that BNORD=DEFINED is coded.

The BNDYN option, however, has a tremendous effect on how your searching will occur and thus must be looked at in a security context as well. When APPN searches for resources in different NetIDs, VTAM creates a table called an Adjacent Cluster Routing table for each NetID to control the searching. BNDYN defines how and if VTAM will add nodes dynamically to the adjacent cluster routing table. There are three settings for the BNDYN option.

- When BNDYN=NONE is specified, there will not be any entries in the routing table except those explicitly coded within a predefined adjacent cluster routing list.

- When BNDYN=LIMITED is specified, the Border Node automatically adds routing table entries if one of the two conditions have been met.

    - BNs and nonnative network nodes that VTAM learns about whose NetID matches the NetID of the desired resource.

    - Any node which performed a search into the network that originated from another NetID that matches the DLU's NetID of the current search.

- When BNDYN=FULL is specified, all active border nodes in the native subnetwork as well as active border nodes and peripheral network nodes in nonnative subnetworks attached to this node are automatically added to the routing list.

Your BNDYN setting can have a dramatic effect on where your searches go. For example, having BNDYN=FULL would allow APPN searches to go to every known Border Node that is connected to the corporate NetID. For example, looking at the Figure 4, if the option is set to BNDYN=FULL there is a chance that NETBLACK will be searched prior to NETB. This could enable a session between the black hats' CICS application and user session to allow a masquerade attack. Set BNDYN to NONE to control searching using an ADJCLUST table that will be described below.

ADJCLUST Tables

Use an adjacent cluster routing definition list (ADJCLUST table) to customize routing between different APPN NetIDs. When a border node can not satisfy a search request within its own domain, it will use the ADJCLUST table to determine which cross network domains to search.

Separate adjacent cluster tables can be coded for each individual NetID that an enterprise is connected to. This can include setting up searches to go through intermediate networks which allow two businesses to connect without having a direct connection. Below is a simple example of an enterprise that uses NETA as its NetID with a border node connected to two different NetIDs (NETB, NETC).

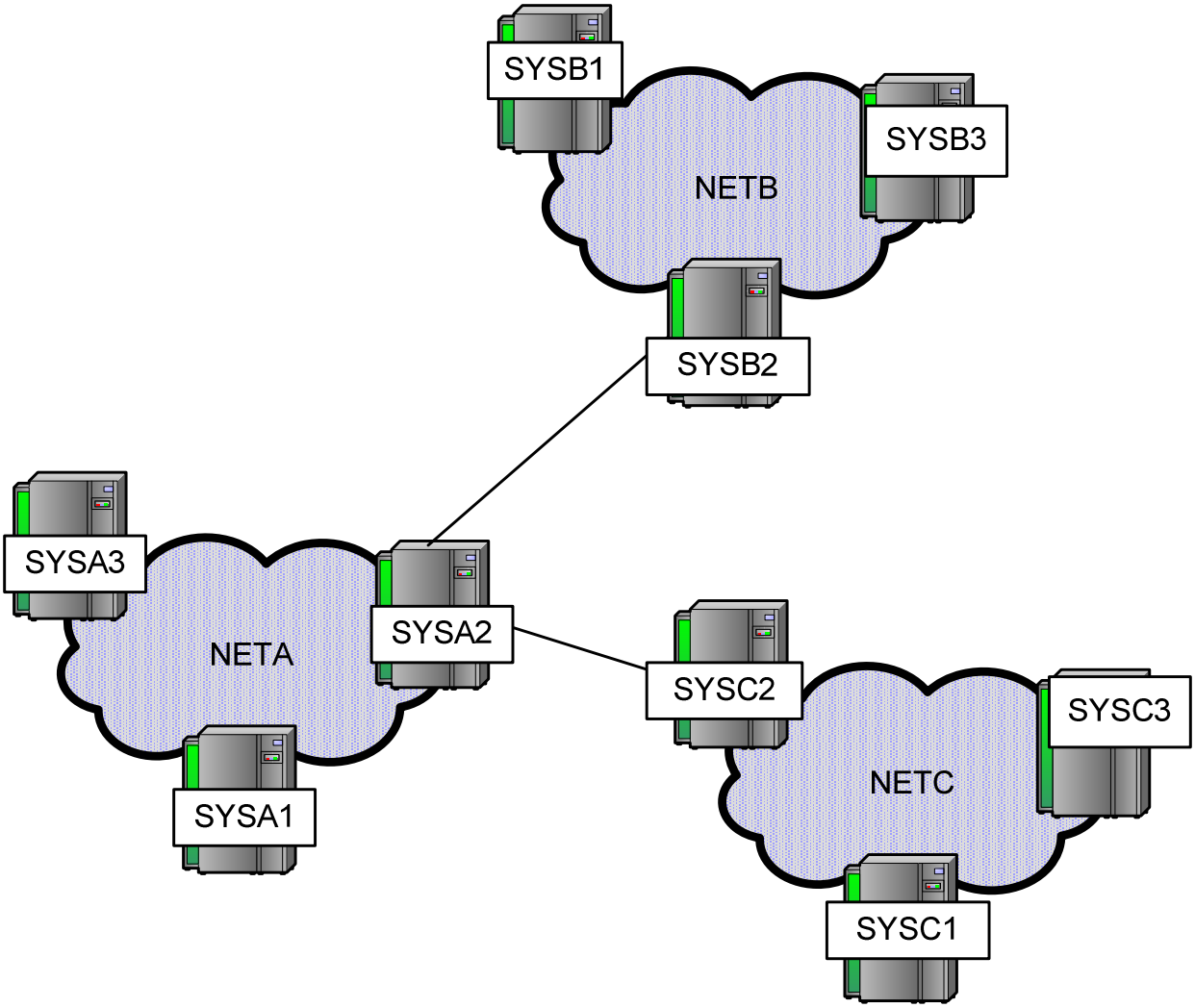


Figure 5 Basic Network Design

Below is an example of an ADJCLUST table that could be coded on SYSA2.

```
*********************************************************************
SAMADJCL   VBUILD TYPE=ADJCLUST
*********************************************************************
* DEFAULT NETWORK ID
*********************************************************************
NONET    NETWORK   SNVC=4,                ALLOW DEPTH OF 4 NETWORKS
                   BNDYN=LIMITED          ALLOW LIMITED DYNAMICS
ASYS2    NEXTCP    CPNAME=NETA.SYSA2
BSYS2    NEXTCP    CPNAME=NETB.SYSB2
CSYS2    NEXTCP    CPNAME=NETC.CSYC2
*********************************************************************
* ROUTING FOR NETID=NETB
*********************************************************************
NETB     NETWORK  NETID=NETB,
                  BNDYN=LIMITED,
                  SNVC=4                     ALLOW DEPTH OF 4 SUBNETS
BSYS2    NEXTCP   CPNAME=NETB.SYSB2
*********************************************************************
* ROUTING FOR NETID=NETB
*********************************************************************
NETC    NETWORK   NETID=NETC,
                  BNDYN=LIMITED,
                  SNVC=4                     ALLOW DEPTH OF 4 SUBNETS
CSYS2   NEXTCP    CPNAME=NETC.SYSC2
```

Figure 6 ADJCLUST table example

In the ADJCLUST table, you can override the BNORD, BNDYN, and SNVC parameters for each NetID the APPN network is connected. For searches originating in SYSA1 without a qualified NetID, all the nodes attached to SYSA2 will be searched. This could allow a black hat to insert their SNA application into a network and have users unwittingly connect to that application in a network beyond the reach of the business.

Example of a well designed ADJCLUST table

```
*****************************************************************
SAMADJCL   VBUILD TYPE=ADJCLUST
*****************************************************************
* DEFAULT NETWORK ID
*****************************************************************
NONET   NETWORK   SNVC=1,              ALLOW DEPTH OF 1 NETWORKS
                  BNDYN=NONE
ASYS2  NEXTCP    CPNAME=NETA.ASYS2
*****************************************************************
* ROUTING FOR NETID=NETB
*****************************************************************
NETB   NETWORK    NETID=NETB,
                  BNDYN=NONE,
                  SNVC=2                       ALLOW DEPTH OF 1 SUBNETS
BSYS2   NEXTCP    CPNAME=NETB.BSYS2
*****************************************************************
* ROUTING FOR NETID=NETB
*****************************************************************
NETC  NETWORK   NETID=NETC,
                BNDYN=NONE,
                SNVC=2                      ALLOW DEPTH OF 1 SUBNETS
CSYS2 NEXTCP    CPNAME=NETC.CSYS2
```

Figure 7 Better ADJCLUST Table

The table in the figure above shows a more secure adjacent cluster table. This table will not allow searches that are not NetID qualified properly to go out of the enterprise NetID. This design would also control fully qualified searches (those containing NetID) to other networks only searching NETC resources for NETC and NETB resources for NETB requests.

Network Qualifying Searches

Creating an Adjacent Cluster table for a Boarder Node that prevents unqualified searches from leaving the native network, carries with it a responsibility to ensure that valid searches are qualified with the correct NetID. This has the additional benefit of optimizing the search patterns. There are two ways to achieve it:

1. On **every** node, define a CDRSC with a NetID for every cross-net target (same-net CDRSCs need not be so defined). This ensures that all cross-net searches originating in this node carry a NetID, and thus fall under the control of the ADJCLUST table for that NetID instead of a default table. For example:

```
         VBUILD TYPE=CDRSC
       NETWORK NETID=USIBMN92
LU925    CDRSC
LU926    CDRSC
LU996    CDRSC
```

2. On the **network** nodes only, define a CDRSC with a NetID and CP name. This creates an APPN directory entry. The value of NQNMODE also needs to be NAME rather than NQNAME, but this is usually so because of the default start option value which is NAME. This technique ensures that alias (unqualified) searches from served end nodes get their NetID added by the NN server before any further searching is performed, and therefore use the correct ADJCLUST table when they reach a border node. For example:

```
          VBUILD TYPE=CDRSC
       NETWORK NETID=USIBMN92
LU925    CDRSC CPNAME=EN587
LU926    CDRSC CPNAME=EN594
LU996    CDRSC CPNAME=NN556
```

The CPNAME value does not need to be correct (although you will save an extra search by getting them right). What matters is the creation of an APPN directory entry, because that is the first place the NN server looks when it receives a search request from a served end node. If the coded values are not correct, the search will fail, but the subsequent broadcast will succeed and will update the values in the directory entry.

Which of these two methods you use depends on your network topology and the relative amounts of effort needed to implement each. The first method works for searches originating from this node or received over a subarea connection. The second method works for all searches, including those received over APPN.

Controlling Searches entering a network

While controlling searches originating in your network is important, it is even more important to control searches destined to your network. Each enterprise needs to protect their network from unwanted searches. In the past, this was manually done using one of two exits (SME or DSME) to control the searching; however, in recent releases of z/OS, some of the functions in these exits have been incorporated into z/OS Communications Server .

Session Management Exit

The session management exit (SME) is a multi-function exit routine you can use to control and manage LU-LU session-related functions. You can use the exit to authorize session establishments, obtain session accounting data, and better manage SSCP and GWPATH selection. In addition, VTAM can invoke the SME exit for the following functions:

- For each session establishment prior to any cross-domain flows
- For each session establishment after the destination resource for the session has been determined

- Determine ordering for the adjacent SSCP tables to try for each cross-network session establishment

- To perform name translation, owning SSCP determination, or CoS and logon mode translation (some of these functions are for cross-domain session establishments, others are for cross-network session establishments)

- To determine the appropriate ALS to use as the connection for an independent LU that is the destination LU (DLU) for the session

- To modify the virtual routes (VRs) and the associated transmission priorities (TPs) that are to be used in session establishment or RTP pipe set-up

This exit will be called only during subarea-side function where the session awareness is maintained and only when a session between the OLU and DLU will take place. VTAM can invoke the session management exit at certain points in processing:

1. When VTAM initialization has completed

2. When normal VTAM termination occurs

3. During XRF session switch

4. SSCP takeover

5. MNPS recovery

Directory Services Management Exit

The Directory Services Management Exit (DSME) is similar to the SME exit however it is used to authenticate and control APPN searches where the SME exit is used for Subarea searches. It has many of the same base functions the SME exit has but for APPN searches such as:

1. Initial Authorization for an LU search

2. Border Node Selection

3. Re-drive of Authorizations for LULU search

In addition, the DSME exit can control which session searches are allowed depending on the following criteria:

1. OLU name/NetID

2. If the search is fully qualified or not

3. The adjacent CP name/NetID in the OLU direction of the node performing the search

4. Many others

Non-Network-Qualified Searches

Use the Adjacent control point (ADJCP) major node to control how adjacent CPs can connect to this VTAM. One of the parameters is the ALIASRCH key word. This will control whether an adjacent node from another NetID is allowed to send non-fully qualified locates into a border node on your network. If ALIASRCH is set to NO, the adjacent node will not be allowed to send non-qualified searches into the enterprise network. This does require you to create an ADJCP table entry for each of the cross-network connections; however, as a general rule it can be useful as an aid to keep track of the nodes the enterprise should be connected to.

Authorized Cross Net Searches

In z/OS V1.10, you have a new option in the adjacent control point table called AUTHNET. The AUTHNET option allows you to limit what NetIDs can be searched through your network. This ability is available in releases prior to V1.10; however you must write a DSME exit with the equivalent function.

In V1.10 you will only have to add the AUTHNET parameter to an ADJCP definition with a list of NetIDs that can be searched for by the adjacent node being defined. This would block the ability of a blackhat to use an intermediate network to perform any searches into another network unless they are explicitly allowed. For example:

In Figure 8 there are three networks connected to NETX for one reason or another. NETBLACK has a hacker that is attempting to access NETA or NETC resources even though there is no reason that a search should be allowed through NETX to ether of these networks.

On NETX.SYSX1 the following ADJCP list has been predefined:

```
*********************************************************
SAMADJCP   VBUILD  TYPE=ADJCP
*********************************************************
*  Allow NETA to search NETC C                          *
*********************************************************
SYSA2 ADJCP NETID=NETA,NN=YES,AUTHNETS=(NETC)        X
            ALIASRHC=NO
*********************************************************
*  Allow NETC to search NETC A                          *
*********************************************************
SYSC2 ADJCP NETID=NETC,NN=YES,AUTHNETS=(NETA),       X
            ALIASRHC=NO
*********************************************************
*  Do not allow NETBLACK to search any networks         *
```

```
************************************************************
SYSB2 ADJCP NETID=NETBLACK,NN=YES,AUTHNETS=         X
            ALIASRHC=NO
```

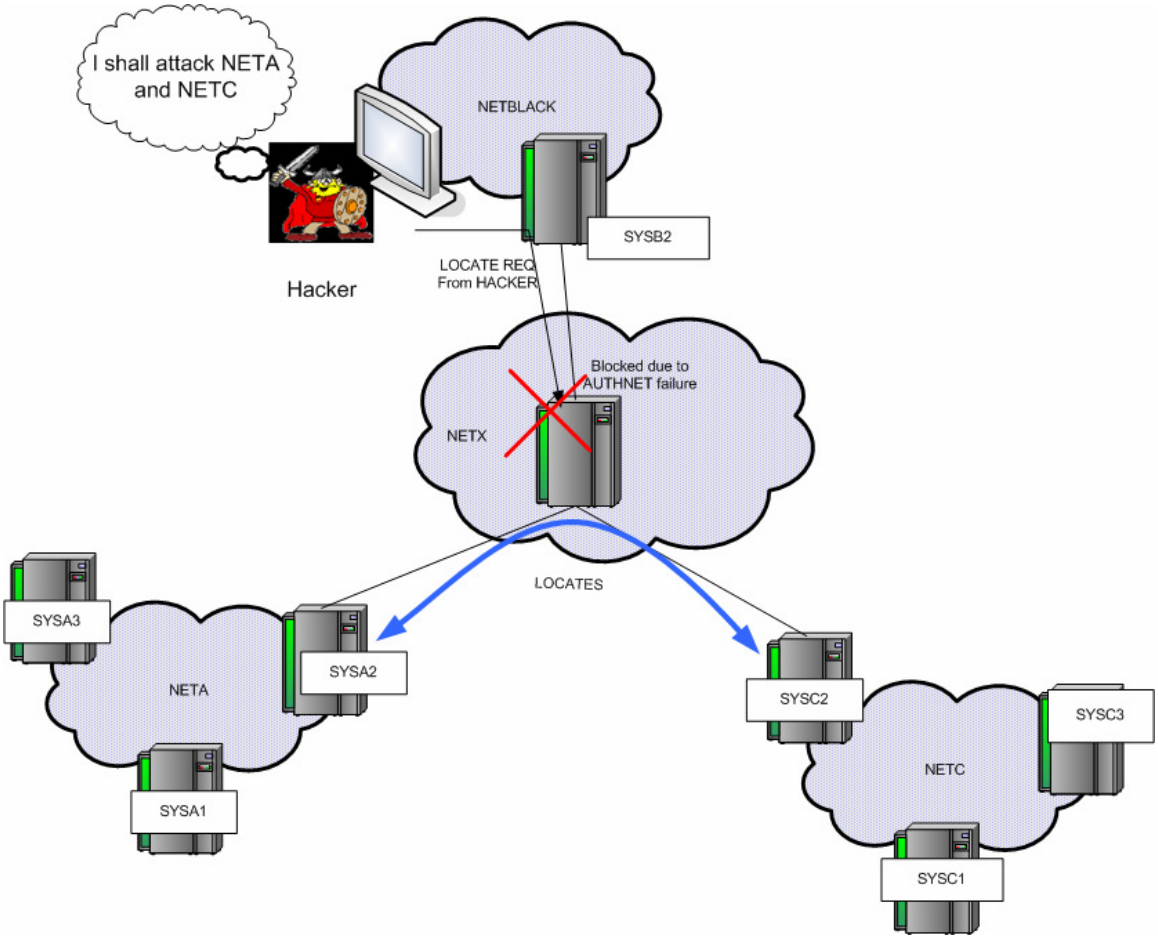


Figure 8 Example of Authnets

This ADJCP table will prevent any search requests coming from NETBLACK while allowing NETA and NETC to perform searches on each other

Chapter 7 Application Security

SNA provides the same basic security functions as IP does, namely data confidentiality, data integrity and partner authentication. The major difference is SNA has no asymmetric encryption, which means:

- Key distribution and key management must be done manually.
- All partner authentications must be done via the possession of shared symmetric keys.

Data confidentiality and data integrity work in similar fashion whether the session concerned is dependent LU (z/OS to z/OS) or independent LU 6.2 (where one partner need not be owned by z/OS). The same is true whether the API used is the Record API or the APPCCMD API. Authentication is somewhat different in the LU 6.2 environment, where VTAM provides extra facilities for applications using the APPCCMD API.

Session level encryption (data confidentiality)

Overview of SLE

The shared symmetric key used in data encryption is exchanged securely between partners during the session setup flow. Typically, this is what happens. The example is the most complex case, namely an SLU-initiated dependent LU session:

- The administrator installs shared symmetric keys (only DES and Triple DES[6] are supported) on each partner node. These are the master keys that are used to encrypt the session keys, and there should be one key per physical node. Each VTAM node will have installed:
    - Its own key.
    - The keys for all partner VTAMs and independent LUs with which it is expected to communicate securely.
    - The keys for all the dependent LUs it serves, that are capable of SNA encryption.
- The VTAM definitions determine whether encryption is to be performed on a particular session. If it is, the SLU VTAM:
    - Generates a symmetric key for use in the session.
    - Encrypts this key using one of the master keys installed by the administrator, twice: once using the key for the SLU node and once using the key for the partner VTAM node.
    - Sends both the encrypted keys to the PLU VTAM in the CDINIT or Locate request.
    - The PLU VTAM then decrypts its own copy of the session key, but sends the other copy with the BIND to the SLU itself.
    - The SLU device decrypts the session key in the BIND, so that now both partners have the same key. The session data now flows in encrypted form.

This is the way it works within an up-to-date APPN network as long as you have installed the appropriate keys (for the SLU node and for the PLU's CP) in the SLU VTAM's key database. The Locate and/or CDINIT flow end to end without any encryption or decryption on the session path, even if there are subarea hops on the path.

If the partner's key cannot be found, or if VTAM is a subarea node, the session key in the CDINIT must be decrypted at each intermediate node, and re-encrypted using the master key of the next node on the path. Thus, the adjacent node's key is required to be in the SLU VTAM's key database instead of the partner node's key.

Even if the network topology permits end to end crypto session setup, an alternative is to let the NN servers of the session endpoints take part in the process. The setup flows are

---

[6] It is recommended to use Triple Des in all cases

then in three parts, with encryption and decryption being done at each NNS as well as at each endpoint.

Given the choice, the installation should determine which session initiation method (end to end, host by host, or NNS to NNS) is used based on the complexity of the network and the effort required to setup the key databases. End-to-end means less overhead in setting up the session but more key administration. Host-by-host means more work in setting up the session but fewer keys defined in each host. Letting one or both NN servers take part is a compromise between these two options.

Whether the session initiation is done end-to-end or host-by-host, the actual session data always flows encrypted end to end. The whole idea of the initiation flows is to send the session key securely from SLU to PLU.

This scheme works for both record API and APPCCMD API. With LU 6.2, the actual encryption options are negotiated on the BIND and BIND response.

Setup of SLE

By default, VTAM supports SNA session encryption, but you can turn it off, or restrict the cryptographic products to be used, using the ENCRYPTN start option.

As to the session itself, the definitions that determine whether or not encryption is used are coded in:

- The APPL definition of the PLU
- The APPL or LU definition of the SLU
- The mode table associated with the SLU

The APPL definition has two relevant keywords: ENCR and ENCRTYPE. ENCRTYPE is easy – the options are DES or TDES24 (triple DES). VTAM uses the higher of the SLU's and PLU's values. Triple DES is preferable since DES is no longer considered a secure cipher.

ENCR can take the following values:

- COND means try to secure every session but set up a clear session if the negotiation fails.
- REQD means that sessions must be encrypted; session setup fails if the negotiation fails.
- OPT means we don't care, let the partner decide whether sessions will be secured or not.
- NONE means there will be no encrypted sessions with this application.

- SEL means the application will specify (using an appropriate API) which messages are to be encrypted. As with REQD, if the crypto negotiation fails then the session setup fails.

The LU definition has the same values and the same meaning for ENCR and ENCRTYPE, except that the SEL value is meaningless and thus not available. In addition, the LU definition has a CKEYNAME keyword that points to the master key associated with this LU. An application acting as SLU always uses its node's master key. The key database must have a suitable key filed in it under that name. The default for CKEYNAME is the LU name.

Use the mode entry associated with the SLU to override the values in the VTAM definitions. However, security cannot be decreased. For example, if the VTAM definition says Triple DES then you cannot override that with DES in the mode entry.

The keywords in the mode entry are:

- ENCR defines a bit string that corresponds to the ENCR value in the LU or APPL definition.

- ENCRTYP has only one valid value, namely TDES24. Omitting ENCRTYP defaults to DES.

- In addition, there is a CKEY keyword that allows you to use an alternate key name for this session. You can define alternate (master) keys may be defined for temporary use while the primary master key is being updated.

Installing Keys

All the setup tasks above are business as usual for a VTAM systems programmer. What may be unfamiliar is the process of installing those master keys on the VTAM nodes.

If triple DES is to be used (the only serious option), a product implementing the CCA (Common Cryptographic Architecture) must be installed. These days, that means ICSF. VTAM issues ICSF calls to generate, encrypt, and decrypt the session keys.

ICSF generates the master keys for you. The local copies of the keys (both for this node and for partners) are stored in the secure key database, while at the same time clear keys are produced for export. Thus, you can generate keys for all nodes on the same VTAM and simply export the clear versions to the desired places, where they too will be placed in the secure database.

Message authentication (data integrity)

SNA message authentication works in almost exactly the same way as data confidentiality (encryption), except that instead of encrypting the message itself, it encrypts a hash value (message authentication code) appended to the message. The same algorithms (DES or triple DES) are available, and the same methods are used to generate the keys.

In fact, VTAM's message authentication offers an alternative to crypto techniques, namely a message digest created by an internal VTAM algorithm instead of DES or triple DES. This alternative cannot be regarded as truly secure.

To invoke message authentication for a session or sessions, once again we have additional keywords on the APPL definition (not LU definition) and the mode entry. On the APPL definition:

- MAC specifies whether message authentication is required (MAC=REQD), preferred (MAC=COND) or not supported (MAC=NONE).

- MACTYPE is the method of producing the message digest: MACTYPE=CRC (VTAM's home made code) or MACTYPE=DES (the real thing). If MACTYPE=DES then the value of ENCRTYPE determines whether DES or triple DES is actually used.

- MACLNTH is the length of the message digest: 4, 6 or 8 are the valid values for DES. The longer, the better.

On the mode entry at the SLU end, exactly the same keywords are used to override the MAC specifications for a given session.

LU 6.2 session-level authentication

In addition to the encryption and data integrity features described above, LU 6.2 resources have an extra authentication feature available to them. The partner LUs authenticate themselves to each other by exchanging encrypted random data on the BIND and BIND response. The encryption is done using DES (note : not triple DES) under a shared symmetric key.

The key itself (referred to as a password in the VTAM literature) is held in RACF, and you need to define the appropriate RACF profiles in the APPCLU resource class to hold the passwords. A password is required for each LU-LU pair that will engage in session level authentication.

Since APPN control points are LU 6.2 resources, the same principles apply to CP-CP sessions as for any other LU-LU sessions. This is a particularly useful feature when you need to be sure that nobody has introduced a rogue APPN node into your network, which is a lot easier to do in these days of ubiquitous EE.

Configuring LU 6.2 Authentication

To protect a VTAM application program against an impostor partner, you need to:

- Choose an application program that uses the VTAM APPC API, not the record API. Applications that use the record API are on their own and must code the authentication algorithms themselves. CICS is the major LU 6.2 user of the record API.

- Code the VERIFY and SECLVL keywords as appropriate on the APPL definition, whether SLU or PLU or both. APPC=YES is a prerequisite.

- Code the appropriate RACF resource profiles.

The VERIFY keyword determines whether authentication is required (VERIFY=REQUIRED), optional (VERIFY=OPTIONAL), or not supported (VERIFY=NONE).

Use SECLVL to indicate either the basic or the enhanced level of authentication is to be used. The choices are LEVEL1 (basic), LEVEL2 (enhanced) or ADAPT (pick the higher one of the two partners' choices). The difference between the basic and enhanced algorithm is that the enhanced algorithm sends an encrypted amalgam of three values (two random numbers and the SLU name) instead of just one random number.

If the sessions to be protected are CP-CP sessions, there are no APPL definitions. The VERIFYCP and SECLVLCP start options do the job of the VERIFY and the SECLVL keywords, respectively.

The name of the RACF profile depends on whether:

- This profile is for CP-CP sessions, or

- The local LU supports network qualified names (the ACB has NQNAMES=YES).

In this case, the name of the profile must be of the form:

```
localNetID.localLUname.remoteNetID.remoteLUname
```

If neither of the above conditions is true, the profile has only three parts:

```
localNetID.localLUname.remoteLUname
```

Thus, you would code the RACF definitions as below:

```
RDEFINE APPLCU NETX.VTAM1.NETY.VTAM2 UACC(NONE) +
 SESSION(SESSKEY(X'8B44D208C1AA'))
```

Note the use of the SESSION optional segment to define the DES key used in the authentication. The same key must be installed in the partner node.

Don't forget `SETROPTS CLASSACT(APPCLU)` to activate the class.

LU 6.2 conversation-level authentication

Session-level authentication is designed to verify the identity of a partner LU before an LU 6.2 session starts. However, an LU 6.2 session typically carries many conversations, each of which may represent a different individual user. Conversation-level security is designed to authenticate the user on any given conversation.

Conversation-level security is completely under the control of the LU 6.2 application. If the application uses the record API, VTAM has no involvement in the process. If the application uses the APPCCMD API, there is one VTAM keyword that you can use to influence the matter. This is SECACPT on the APPL definition:

- SECACPT=NONE means that conversation level security is not supported.
- SECACPT=CONV means that conversation level security is supported.
- SECACPT=ALREADYV means that the "already verified" option of conversation level security is allowed, in addition to the CONV level. This is used when an application program passes an incoming request on to another program. The first program has already verified the user ID and password, so it tells the second program not to bother.
- SECACPT=PERSISTV means that the "persistent verification" option is supported. This allows two application programs to have multiple conversations with each other, but perform the verification only once.
- SECACPT=AVPV means that both ALREADYV and PERSISTV are supported.

Because conversation level security implements a user ID and password, it is not to be regarded as secure unless the session itself is encrypted.

Chapter 8: Conclusion

This paper has outlined many techniques that can be used to improve the overall security of the SNA Network. The suggestions in this paper should assist any IT professional to perform the "Due Diligence" necessary to secure their SNA environment. Below are a set of suggested activities one can use to secure their SNA network.

Policy Recommendations

- Do not allow mixed security environments (for example, production and test) to be connected via SNA
- Make sure to separate System Programmer and System Operation duties between 2 or more individuals
- Only allow as much access to any resource as is required to perform tasks
- Ensure that all resource definitions are purged as equipment is removed from a SNA environment

Network Recommendations

- Secure SNA links using IP with some type of encryption and authentication (IPSec or SSL/TLS)
- Review the TN3270 settings that can control access to VTAM applications
- Evaluate the use of multiple TN3270 servers to separate those applications that hold sensitive data from those that do not.
- Do not allow non-qualified searches to enter the SNA environment from other NetIDs
- Set BNDYN=NONE and use ADJCLUST tables to secure Network searches

Platform Recommendations

- Set up protections for VTAM datasets using RACF or equivalent SAF interface product
- Code Adjacent CP table definitions for any non-native nodes to restrict access
- Code Adjacent SSCP tables to control searching
- Modify DYNADJCP, DYNASSCP, and SSCPDYN start options to NO

- Code ALIASRCH to NO and use the new V1R10 option AUTHNET in the ADJCP definitions for cross-network connections

Application

- Evaluate the use of session level encryption and authentication techniques to secure those sessions that carry critical data.

APPENDIX A: Useful Links

Communication Server Links

IBM Communications Server family page

http://www.ibm.com/software/network/commserver

IBM Mainframe Servers:

http://www.ibm.com/systems/z/

Networking: IBM System z servers:

http://www.ibm.com/systems/z/hardware/networking

z/OS Communications Server

http://www.ibm.com/software/network/commserver/zos/

Communications Server for Linux on System z

http://www.ibm.com/software/network/commserver/z_lin/

Communication Controller for Linux on System z

http://www.ibm.com/software/network/ccl

Documentation

Communications Server products - white papers, product documentation, etc.

http://www.ibm.com/software/network/commserver/library

ITSO Redbooks

http://www.redbooks.ibm.com

Technical support documentation (techdocs, flashes, presentations, white papers, etc.)

http://www.ibm.com/support/techdocs/

Support

Communications Server technical Support

http://www.ibm.com/software/network/commserver/support

DSME Exit Example

http://www-1.ibm.com/support/docview.wss?rs=852&uid=swg24014056

Appendix B: Trademarks and Additional Disclaimers

IBM Corporation

Department AQYA

3039 Cornwallis Road

Research Triangle Park, NC 27709

Printed in the United States of America

9-00

IBM, OS/390, S/390, and S/390 Parallel Enterprise Server are trademarks of International Business Machines Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others..