Global Information Assurance Certification Paper

GSEC Version 1.4b – option 2

Securing a Previously Open Private Network with Cisco ACLs

Aron Smith – July 2003

Table Of Contents

INTRODUCTION....3
BACKGROUND....3
The DECnet Issue....3
BEFORE....4
Before Diagram....5
PLANNING....5
Management Assigned Objective....5
Risk Assessment....6
WHAT HAPPENED....7
The Approach....7
Three Layers....8
The New Diagram....9
The Access Lists....10
IP....10
Non-IP....11
The Interface....11
Securing The Routers....12
Data Center Routers....12
Client Routers....12
TO DO LIST....14
CONCLUSION....16
SOURCES....17

INTRODUCTION

My company is a financial institution that develops Credit Union processing software and offers online processing services for our clients. We have two categories of customers, "in-house" customers that purchase a host system and our software. These in-house clients manage their own networks and systems. The other category is what we refer to as "online" clients. Online clients connect to us via a private Frame-Relay network to hosts in our data center.

It is the online Frame-Relay network that I will focus on in this paper. The before picture and associated configurations was a case study in bad security. The business needs, application requirements, and historical lack of scrutiny resulted in a very insecure environment. I will describe here how I transformed a 3Com-router based network to a Cisco-based network with multiple security layers using Access Lists.

BACKGROUND

The primary processing systems are based on VMS systems. The MOP and LAT (DECnet) protocols are heavily used to provide connectivity to remote DECServers. DECServers have historically provided remote clients with many services through serial ports, such as:

- VT400 terminals for tellers, loan officers, and back-office staff
- Printing for transaction receipts, cashiers checks, and general ledger reports
- Application interface to Real-Time Audio Response
- Application interface to ATM processors (like MAC, Star, Visa, and EDS)

Obviously these are critical services for a credit union, and while there are IP-based devices to handle all of these services, there are thousands of deployed DECServers in use. This requirement to support legacy DECServers results in the need for Transparent Bridging, more detail is below.

The DECnet Issue

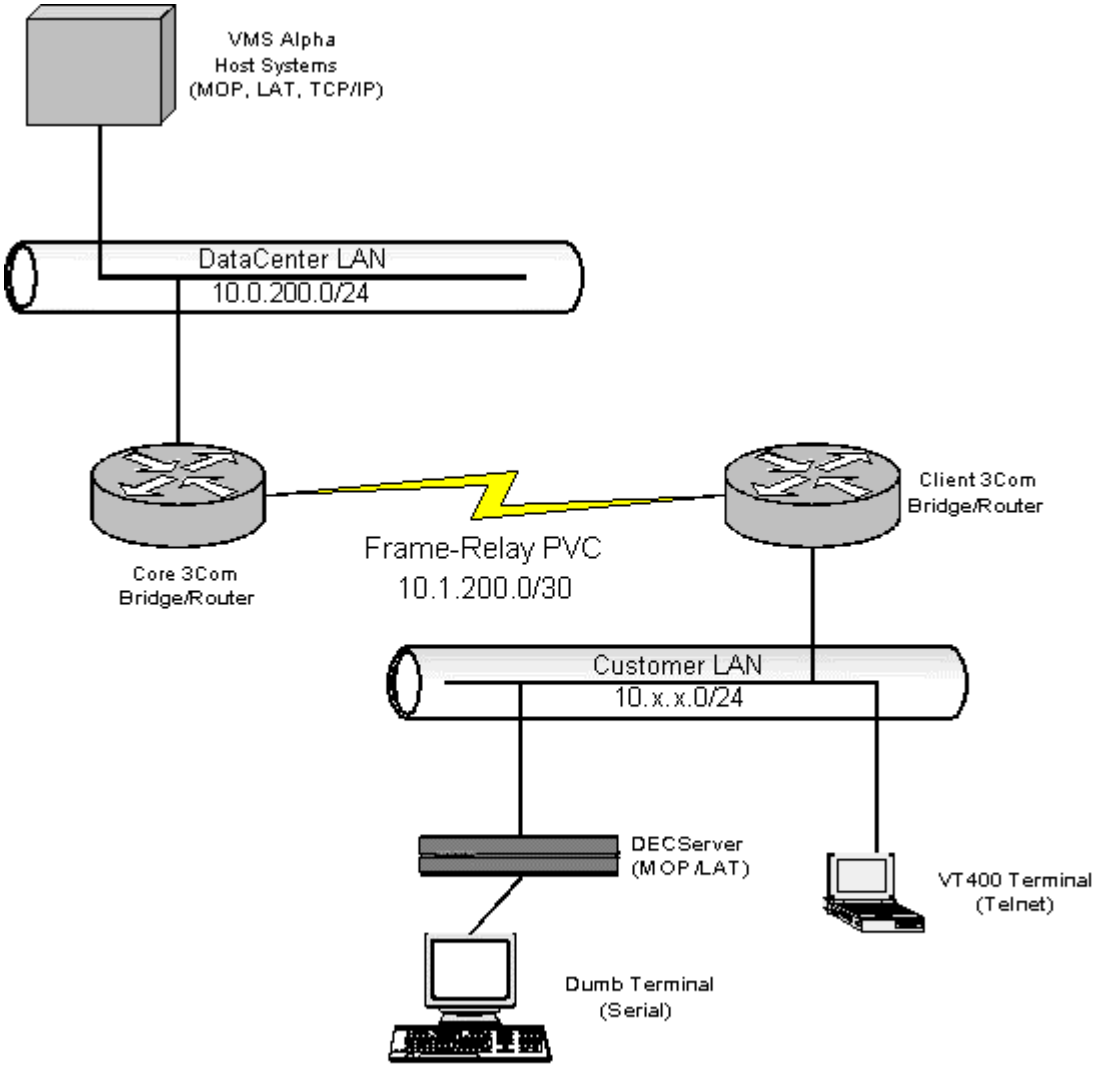
MOP performs a similar function for DECServers that BOOTP and DHCP performs for IP based devices. Like BOOTP/DHCP, unless a router acts as a proxy for the remote destination, the traffic will never leave the LAN segment the router feeds. These DECnet features are expensive to purchase in Cisco's IOS (Requiring the "REMOTE" or "ENTERPRISE" feature sets) and complicated to configure. A simple "bridge-group 1" command on all connected interfaces on the router allow all non-routable/non-IP traffic to pass through the WAN. Older DECServers load their operating system and configuration through MOP traffic to the VMS host. Once the MOP bootstrap load is

complete, the DECServer communicates to the host through LAT/DECnet. While DECnet is routable to some degree, this network was not designed into DECnet segments, and addresses were assigned to the same "LAN", thereby eliminating any possible routing that might have been done. Rebuilding the DECnet network would be disruptive, and the direction is away from that protocol.

BEFORE

All core data center routers, and nearly all deployed client routers at the beginning of this project were 3Com routers. No meaningful filtering of any kind prevented IP or non-IP traffic from reaching the core host network, or from crossing over from one client to another. A simple print server in its default configuration with IP, IPX, and DLC enabled resulted in broadcasts transmitted to every part of the network, including the core host LANs, and other clients. Below is a basic network diagram of how these networks connected.

Before Diagram



PLANNING

Management Assigned Objective

There were three primary objectives provided to me by management at the beginning of this project:

1) Convert from 3Com hardware to Cisco hardware
2) Improve the performance/technology used

3) Improve the overall security of the network

The first two objectives, while broad, were fairly easy issues to deal with.

The hardware conversion consisted mainly of deciding what router models would suit our bandwidth/growth needs best. The technology issue was somewhat more involved with the major shift being from an all Frame-Relay network to a mixed ATM-to-Frame-Relay network with the FRF.8 standard (ATM to Frame Relay Interworking).

ATM to Frame Relay Interworking is a service offered by many carriers, where a PVC is built between a Frame-Relay DLCI and an ATM VPI/VCI. The Frame-Relay side must use IETF Frame-Relay encapsulation, but the carrier does the conversion to ATM transparently. For the head end, ATM IMA (Inverse Multiplexed ATM) was chosen to allow the purchase of WAN bandwidth in 1.536Mb increments (one T1 at a time) in order to scale more gracefully.

The security management goals were stated broadly as I indicated above. The only specific I was given was that a partner company had called with a warning. Apparently, an engineer that was on-site at one of our client sites, accidentally telneted to a router at another client.

Since I had no more direction than "improve overall network security," and this one specific example, I moved on to do a risk assessment and break down those risks into projects/layers that would provide the biggest wins first. Throughout, I had to keep in mind that any direction I moved in had to be maintainable by relatively inexperienced network engineers (at least on Cisco networks) until they could be brought up to speed on the new technology.

Risk Assessment

Here my goal was to identify our most significant areas of concern from a network standpoint. The fact that a partner company let us know about the client-to-client telnet capability was obviously a top priority for management. While less obvious, the transparent bridging was resulting in the same kind of risk, if someone connected a VMS host, or a PC with DECServer Manager installed, they could potentially reconfigure any DECServer at any client site. To make things worse, the DECServers were left with the default password they are shipped with—being hard-coded into the management application makes this impossible to address in the near term.

Below is the prioritized list of risks I felt I could address with Cisco Access Lists, and a brief description of the problem or an example exploitation scenario.

1. Client-to-Client IP connectivity unrestricted (self-spreading Windows worms, active hacking—whether accidental or malicious).

2. Client-to-Client transparent bridging (remote exploitation of DECServer configurations, print servers, NetBEUI PCs, etc).
3. Client-to-Data Center transparent bridging (any non-IP protocol could potentially be used to gain access to misconfigured servers, e.g. NetBEUI, or IPX/SPX).
4. Client-to-Data Center IP connectivity unfiltered (access to hosts clients don't need, and no restriction on what ports are available).

Clearly I focused on network connectivity as my primary area of concern. Application security was outside my purview, not to mention incredibly difficult to resolve since it is primarily a development issue. In addition, not all security issues can be addressed at the application layer (e.g. client-to-client IP connectivity)

With the conversion to a Cisco infrastructure, access-lists and potentially PIX rules figured prominently in nearly any security policy implementation plan. This was the basis of my focus on them, and drove the research.

WHAT HAPPENED

The Approach

There were two primary limiting factors that influenced the technical details of this implementation.

1) Limited staff resources – being the only "Cisco" resource meant any solution needed to be straightforward and not interfere with day-to-day network support.
2) Applications using custom ports are in widespread use, and there was—and is—no mechanism that informs the Network Services area of new network access requirements. Things "just work," similar to how most people think about plumbing.

Application tools to manage access lists like CiscoWorks ACL Manager might have offered a relief from the staffing issue. Unfortunately that option would require fast PCs on the network being managed for the CiscoWorks Java web applets.

The only access most of the support staff has to the production network at issue, is via a 9600bps serial DECServer port at their desk, or terminal dialup. Anything that could not be done inside a telnet session, was not a realistic option for day-to-day activities. Some staff members still use dumb terminals to support the network infrastructure, IP access to the production network from a PC is only available to most staff through a few older, shared PCs, or walking into the data center itself to access the console of a server. This eliminated virtually any GUI tool that might be implemented to simplify management of ACLs.

Three Layers

With the new network layout, there were three opportunities to implement access controls.

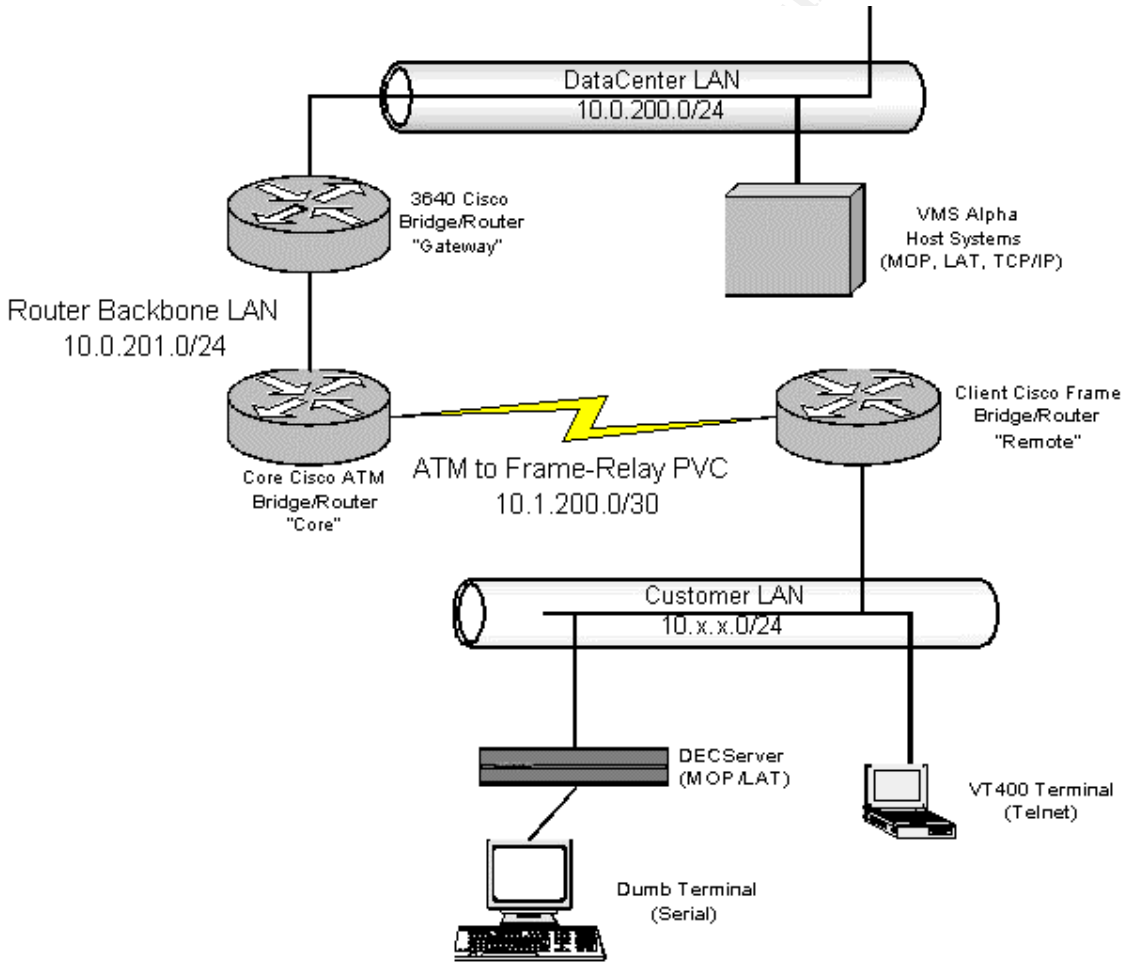
1) The Remote / Client Routers
2) The Core WAN Routers
3) The Gateway / Choke Point

I have focused on Layer 2 here, but the remote routers are certainly capable of access-lists, and the gateway routers were purchased for the specific purpose of providing a dedicated checkpoint to control traffic flow even if those controls were not part of the initial implementation.

As the current network support staff becomes more comfortable with ACLs, the realistic potential to implement Layer 1 and Layer 3 will improve. By focusing on Layer 2, I was able to move ahead with securing the environment by creating reusable access-lists, and making their implementation procedural for new client connections.

The New Diagram

The new infrastructure features an additional layer as a choke point for future access-lists (thus providing Layer 3 described above). I decided on routers for this gateway layer instead of firewalls due primarily to the integration these devices would need to have into the OSPF environment. This means no stateful packet inspection, but better redundancy through HSRP (Hot Standby Router Protocol) and better initial integration since they could be left without access-lists initially—thus allowing their installation into the network without having to immediately face the task of building the highly complex access lists they need. Here is the basic flow of network traffic after the initial implementation.



The Access Lists

IP

In the new diagram there are numerous places to potentially place access lists. I considered placing extremely restrictive client-specific lists exclusively at the remote router. This would, in fact, produce the most secure client-sourced traffic pattern, unfortunately, physical access to a router means it can be compromised, not to mention the fact that, with hundreds of clients, this method would take quite some time to implement. Eventually, very tight client-side access-lists will be implemented as the first layer of security in this three-layer approach. At this point, I focused on what will actually be "Layer 2" access-lists even though it was the first one implemented.

Fortunately, one thing that had been done right in this network was to use the RFC1918 10/8 networks for all locations. All data center-side subnets fall under 10.0/16. All WAN segments between routers fall under 10.1/16. Any other subnet can be assigned to a client network. With the assumption that no client needs to—or should, communicate with another, the simplest of initial access lists became obvious.

This list is what was applied to all client PVCs coming into the Core ATM routers.

access-list 115 remark Client PVC Inbound ACL
access-list 115 permit ip any 10.0.0.0 0.0.255.255
access-list 115 permit ospf any any
access-list 115 permit icmp 10.1.0.0 0.0.255.255 host 10.2.53.7
access-list 115 deny ip any any

This access-list allows any source network to reach the data center, necessary OSPF traffic, and ICMP inside the WAN segments for a network monitor application.

I tested this access list with the last statement including the 'log' keyword so that a syslog message would be generated for rogue IP traffic. This turned out to be a bad direction to go, there was so much bad traffic from so many clients I could not begin to attack this problem. In any case, a 'show access-list' command will display how many "hits" there are for each statement. Also, even though there is an implicit "deny any any" statement at the end of all access lists, by explicitly specifying it, you can obtain some basic statistical information about how much traffic is being dropped. Here is an example of an access-list counter to show just how much bad traffic was being dropped at this point in the network.

Router#show access-list 115
Extended IP access list 115
permit ip any 10.0.0.0 0.0.255.255 (32491789 matches)
permit ospf any any (1910653 matches)
permit icmp 10.1.0.0 0.0.255.255 host 10.2.53.7 (157063 matches)

deny ip any any (1124210 matches)

Clearly, out of roughly 32 million packets, over 1 million were dropped, and not one of the approximately 200 clients this list was applied to, called or complained. Only one "odd" entry for 10.2.53.7 was required (this is a monitoring system that only needs to ping routers and so was restricted by protocol to ICMP). While this list could further be tightened down, I felt comfortable implementing it, and was able to do so across the entire WAN within a week once the decision was made to move forward.

Non-IP

To prevent clients from reaching one another through the transparent bridge presented a different problem. The MOP loads required by DECServers meant I could not restrict inbound traffic without programming the MAC address of every DECServer on the network into individual access lists, restricting by protocol offset is very CPU intensive for a router to do and was not preferred either. Due to staffing concerns described previously, I opted to continue operating on the assumption that I should implement simple, static, reusable access lists for the core ATM WAN routers.

Again I lucked out to some degree in this case because the MAC addresses of our VMS systems all begin with bbbb.0400.* for MOP/LAT traffic. The following outbound MAC address access list was applied to the sub-interfaces on the core ATM routers. This list allows DECServer broadcasts (and, admittedly, others as well) to come into the network for MOP loads, but only bridged traffic from the VMS systems can leave. This prevents client-to-client non-IP communication, as well as any two-way communication to potentially misconfigured systems on the protected subnet in the data center.

access-list 1115 permit bbbb.0400.0000 0000.0000.ffff 0000.0000.0000 ffff.ffff.ffff

The Interface

Here is an example of how these two access lists appear when applied to the ATM IMA PVC sub-interface with the access lists underlined:

interface ATM1/ima0.999 point-to-point
description AAA DHECxxxxxxATI
mtu 1500
bandwidth 128
ip address 10.252.7.93 255.255.255.252
ip access-group 115 in
pvc AAA 1/999
vbr-nrt 6000 143 32
oam-pvc manage
bridge-group 1
bridge-group 1 output-pattern-list 1115

Securing The Routers

While security and control of network traffic is the primary focus of this paper, securing the routers themselves is necessary to prevent tampering with any router-based security measure. There are two basic classes of routers in this network, each with different security risks and administration requirements. Here are the two classes, and what was done for each.

Data Center Routers

These routers are physically located in a data center, with all the security and environmental controls associated with a "data center." The potential for a hacker to gain physical access to a router to circumvent the passwords is highly unlikely. The primary risk here is accidental or intentional modification of the configuration that results in a denial of service until the proper configuration is restored. Therefore, auditing on a per-user basis for administrators, and configuration audit are the primary goals.

In order to provide per-user access control, Cisco ACS was implemented through the TACACS+ protocol. The TACACS+ servers provide individual administrator logins, logging of accesses, and access times. Instant removal of access is possible simply by terminating a user ID in Cisco ACS. Reliable LAN-access to the TACACS+ server helped make this a viable option.

To audit configuration changes, CiscoWorks 2000 is used to track different versions of configurations as they are changed. This also allows (within reason) a point-in-time recovery of the configuration for all data center routers.

Between Cisco ACS, and CiscoWorks 2000, it is very easy to discover who made a change, and when. Auditing of failed logins is also available when Cisco ACS is used as an authentication service (unlike a router with only internal telnet and enable passwords).

Client Routers

These routers are physically located at a client site, and can easily be tampered with, without our knowledge. Despite this, since our customers are credit unions, the physical locations are actually quite secure against outside intrusion. The remaining issues are thus:

1) What if a client gets control of a remote router through the console and removes any network access controls?
2) What if there are WAN connectivity issues and someone on-site requires access?

In the first case, the Layer 2 protection is the only active network access control. However, even with control of a remote router, if the Layer 2 controls are tight enough, access to the on-site router doesn't get the hacker very far, and CiscoWorks configuration audit reports will reveal these configuration changes as well.

In the second case, reliance on a network-based authentication tool like Cisco ACS will hamper troubleshooting. While a "backup" user ID can be setup on the router, once someone has access to the router configuration, discovering that ID gains the same access as in the case of a static enable password with the added penalty of significantly more complex router configurations. For these reasons, the remote routers were given static passwords. With the client-to-client IP traffic halted in ACL Layer 2, discovery of a static password will still only provide access to the one compromised router, even if that password is reused elsewhere on the network. The other consideration for in-router passwords is encryption, while decrypting a telnet password is easy, decrypting an enable password that allows configuration change is not (they use very different encryption methods).

TO DO LIST

In practice, these simple access lists virtually eliminated the illegitimate traffic for both IP and non-IP packets. It is not difficult to poke holes in this configuration, however. Which explains why this project is not truly complete, even if management's immediate concerns have been addressed.

In most cases, a given customer only needs to actually communicate with one or two hosts in the data center. The remaining issue is how to further lock down permitted traffic. There are two more initiatives remaining to protect the host systems from unauthorized communications. These steps will complete the ultimate three-layer approach we are moving to.

1) "Layer 1" – Specify by host and port permitted destinations for traffic at the remote router. Even though the remote routers can be compromised, this will eliminate the ability of someone with only local network access to get very far.
    a. This is simple, but time consuming to accomplish and is under way. The main obstacle I have run into so far has been support and time. Staff must be trained to better understand access-lists in order to allow them to make the appropriate changes when a client requires different, or new access, to hosts on the protected network.
    b. The other obstacle to moving forward with Layer 1 is that many remote client routers are still 3Com. This in itself does not prevent work from being done to improve security on the remote Cisco routers, but it does create a real-life barrier to completing this phase until the remote 3Com routers are replaced.
2) "Layer 3" – Specify permitted source networks, and destination ports at the gateway routers.
    a. This is not simple because these routers will process all traffic from all sources and these access lists will need to be carefully written lest I generate a major outage at implementation. Wide deployment of applications using non-standard ports means a lot of research must be done before port-specific access-lists can be safely implemented at all, this is under way.
    b. Another problem with this layer is complexity, as order of operation inside the access list will likely have a mix of permits and denies in a certain order. Nearly all other locations on the network can accept a simple "permit, permit, permit, deny all" pattern. These access lists must also be permissive enough that a simple change in access needs by one client does not require rewriting the gateway ACLs.
    c. Due to access list complexity, attempting to implement this layer has also shown me that while using routers here allows simple OSPF integration, a PIX might be better suited to this in the long run. Access-lists are all or nothing, that is, you cannot insert a rule in the middle, you can only append. To insert a rule, one must remove the access list completely and

rewrite it to the router. If not done carefully, it would be easy to shutdown all traffic during the rewriting process. A PIX does not operate this way and would handle single rule injection and removal gracefully.

CONCLUSION

Access-lists are a powerful tool in Cisco routers to control traffic flow, and can produce a very secure environment for IP and non-IP traffic. They can also generate huge headaches during troubleshooting, especially if you are working with a large access list or inexperienced staff. They are not a panacea, and there is no substitute for early-stage security evaluation when an application and network are being built. Some of the compromises I made would not have to be made if the applications were more inherently secure. As it stands, there are insecurities here that only very intrusive, potentially expensive, and definitely time-consuming processes can solve. On the other hand, no application security measure could accomplish on its own, what controlling traffic as it traverses the network can do.

Starting with a completely open network presents its own set of challenges, since the Availability bedrock issue never existed—everything was available all the time.

There are obviously other issues with this network, such as the clear text protocols in use like telnet and LAT sessions to DECServers. Fortunately, (or unfortunately) those issues are far beyond my scope and are dictated by the development teams and business requirements. I briefly considered trying to use IPSEC to encrypt traffic between the remote routers and the last hop before the protected subnet, however that would add huge performance and cost penalties, without actually hiding the data at its most likely interception point (inside the protected net or on a client LAN).

SOURCES

Allen, Julia. "Improving the Security of Networked Systems" Oct. 2000. The Journal of Software Defense Engineering. Jul. 3 2003. <http://www.stsc.hill.af.mil/crosstalk/2000/10/allen.html>

Configuring IP Access Lists Jun. 13 2003. Cisco Systems, Inc. Jul. 14 2003. <http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a00800a5b9a.shtml>

Cole, Eric. SANS Security Essentials II: Network Security Overview. Publish location unknown: 2003.

Crane, Mike. Cisco CID. Indianapolis: Cisco Press, 2001.

Hernan, Shawn. "Security Often Sacrificed for Convenience" Oct. 2000. The Journal of Software Defense Engineering. Jul. 3 2003. <http://www.stsc.hill.af.mil/crosstalk/2000/10/hernan.html>

Improving Security on Cisco Routers. Jun. 25 2003. Cisco Systems, Inc. Jul 14 2003. <http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080120f48.shtml>

Morrisey, Peter. "Demystifying Cisco Access Control Lists" Unknown Publish Date. Network Computing. Jul. 7 2003. <http://www.networkcomputing.com/907/907ws1.html>

Security Policy Management Jan. 2002. Solsoft Executive Whitepaper. Jul. 12 2003. <http://www.solsoft.com/library/wp_solsoft_012002.pdf>

Last Updated: May 25th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event