Global Information Assurance Certification Paper

Securing the Network with VLANs (A Case Study)

Anthony Perri

February 14, 2003

GSEC Version 1.4b, Option 2

Abstract

I recently worked with a client of mine, Acme Financial Systems (*name changed*), on a project to improve the overall security of their network. My role was to determine an area of the network that was at significant risk, evaluate potential solutions and recommend one, and help implement the solution.

Acme's perimeter defenses were pretty sound. Internally, however, I determined their "wide-open" network left them vulnerable to well-known threats. We wanted to deploy an internal firewall to control the flow of traffic between different zones on the network. After evaluating several options, we decided implementing VLANs and using a router with access lists to control the traffic flow was our solution of choice. While there is still work to do, as is the case with any security solution, we were able to improve the overall security of their network.

Before

Acme Financial Systems operates a data center that provides data processing services to its member financial institutions. These services include access to a mainframe-hosted banking package, check processing, Internet access, and email. Acme depends on their network to deliver these services to their customers. Any disruption of service on the network poses a significant threat to Acme's business, since the network **is** their business. Therefore, Acme is continually looking to enhance the security of their network.

Acme had applied a good defense-in-depth strategy in securing the network perimeter. The primary defense, a Raptor firewall, had been installed. The firewall rules had been carefully crafted, permitting the necessary traffic and denying everything else. Internet-accessible servers that provided web banking, file transfer, and other services were installed on a DMZ segment. The one exception was the email server, but the Raptor was running a SMTP proxy, so outside users did not have direct access to the mail server. Net Prowler, an intrusion detection system, had been installed and was actively monitoring the firewall logs and alerting network administrators of any suspicious activity. Lastly, regular penetration testing performed by a third party confirmed the defenses were working. Overall, the network perimeter seemed pretty secure.

Our focus next shifted to the internal network, where very little had been done to improve security. Acme's LAN consisted of a single subnet on which all hosts, regardless of function, resided. These hosts included:

- Router with frame relay connection to connect to customers' networks.
- Routers with leased-line connections to other customers' networks.

- Mainframe that was accessed by some, but not all, customers.
- Raptor firewall with connections to DMZ and Internet.
- Acme employees' workstations.
- Email server accessible by customers.
- Novell server and Win2K servers accessible only to Acme employees.
- SQL Server accessible by customers.

These hosts were physically connected to a Cisco Catalyst 2924XL switch, either directly or through a hub with an uplink to the switch. Acme's WAN connected its customers to the data center via frame-relay and leased-line connections. This is illustrated in Figure 1.

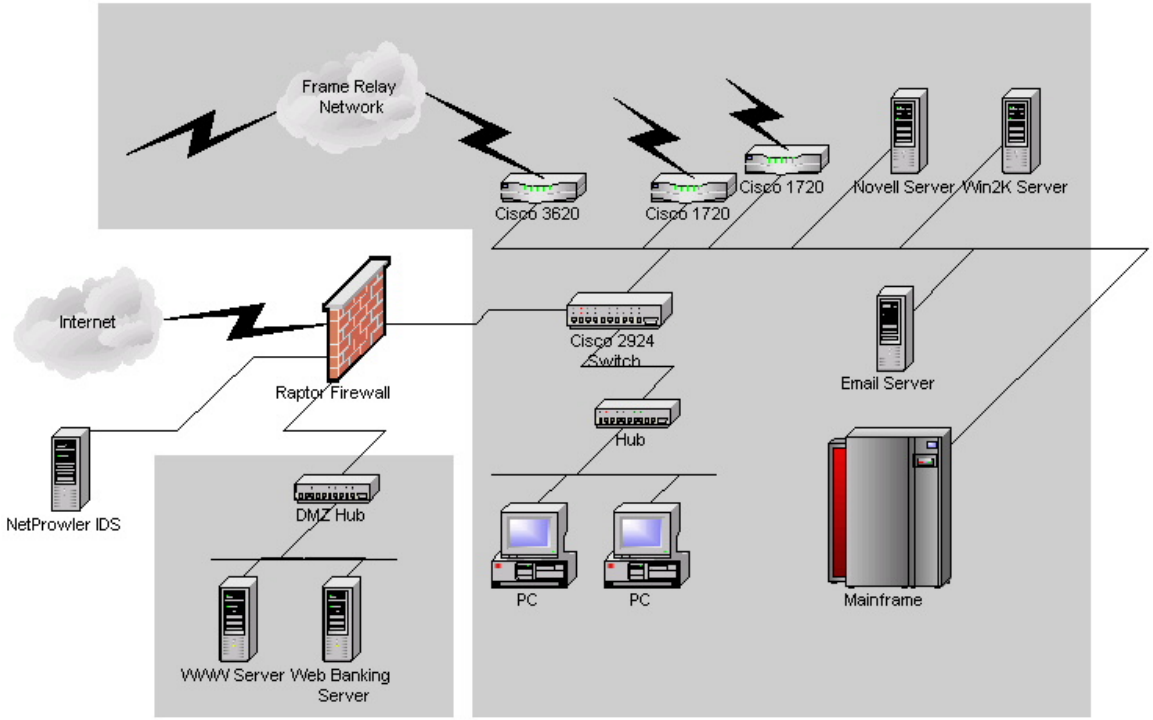

Figure 1

Some attempts had been made to make the network more secure. Physical access to the computer room was restricted, however, since Acme is a small operation, most employees had access. Although mostly for performance, using a switch did make the network less vulnerable to packet sniffing. However, since most of the workstations were connected to a hub rather than a switch, a large portion of traffic was still vulnerable to sniffing. Access lists had been configured on the frame-relay router to prevent traffic from flowing between Acme's customers. However, all traffic was allowed to flow between Acme and it's customers. Critical servers were protected with username and password, but these were seldom if ever changed.

It was becoming apparent that, despite some effort at improving the security of the internal network, some serious vulnerability existed, placing the network at serious risk. Especially of concern was the "wide-open" nature of the network. This compounded the risk of any threat because, once inside, the threat would meet minimal resistance accessing any host on the network. While practices such as installing patches, keeping anti-virus software current, implementing a strong security policy, etc. can eliminate many vulnerabilities at the host level, it is difficult, if not impossible, for administrators to stay on top of this task at all times. Furthermore, Acme was depending on its customers to properly maintain its systems as well.

In order to illustrate the current level of risk Acme was assuming with their "wide-open" network, we examined some well-known threats and theorized about how an attack from that threat might play out.

The first threat we considered was an attack from a malicious hacker. Acme was confident a hacker would have trouble gaining access to their network through their perimeter defenses. Furthermore, because their building was secure, they weren't concerned about a would-be attacker accessing the network from inside the building. However, Acme couldn't vouch for the physical security of all of their customers' locations. Additionally, they were dependent on their customers for securing workstations, dial-in modems, etc. It started to seem more likely that a determined hacker could gain access to the network. Since the network was wide-open, that attacker would be able to communicate with any host on the network, potentially cracking the host-based security or exposing an un-patched vulnerability.

Next we considered the case of a specific virus, Nimda. The Nimda worm is a blended threat that can spread through mass-mailing, unpatched IIS servers, and open network shares. We figured the most likely entrance into the network would be a user opening an infected email message. Nimda can be prevented from spreading by keeping anti-virus signatures up to date, keeping IIS patched, and having restrictions on network shares. However, mistakes do happen, and with our wide-open network, Nimda or a Nimda-like worm could spread very quickly throughout Acme's and Acme's customers' networks. [5]

Finally, we considered the threat of a user running a program that, unintentionally, causes harm to the network. Perhaps an administrator for one of Acme's customers tries out a new network scanner. Unfamiliar with the software, he elects to scan the entire network. Since the computer he is running the program on has access to the entire network, the scan generates lots of network traffic, potentially resulting in a denial of service to Acme's customers.

While certainly not a comprehensive list of all the threats the wide-open network leaves Acme's network vulnerable to, it is clear the network is at significant risk.

During

We decided that limiting the flow of traffic between different zones on the network would significantly enhance the security of the internal network. Essentially, we wanted to deploy a firewall on the internal network. While firewalls are not the holy grail of network security, they can go a long way towards blocking access to computer systems. Its much more difficult to exploit a vulnerability on a host if you can't communicate with that host.

Our top priority was controlling access to Acme's servers from their customers' networks, since Acme had limited control over the security of their customers' networks. To accomplish this, we determined what access customers required, so we could permit this traffic and deny everything else. Specifically, customers needed access to the following systems:

Host	Reason	Protocol	Port
Mainframe	Telnet access for terminal access to banking applications.	TCP/IP	23
	HLCN access for terminal access to banking applications. HLCN is being phased out in favor of Telnet, but is still required.	IPX	
	FTP access to transfer files to/from mainframe.	TCP/IP	21, 20
SQL Server	SQL access for a banking application.	TCP/IP	1433
Email Server	SMTP to send email.	TCP/IP	25
	POP3 for customers not running their own email server.	TCP/IP	110
Raptor	Customer's path to the Internet. Customers don't access the Raptor, but traffic destined for the Internet must reach it.	TCP/IP	80, 443

Since computers outside of Acme's control would access these hosts, we wanted to place them on a separate network segment, much like a DMZ. That way, if one was compromised, it couldn't be used as a jumping point to compromise hosts the attacker wouldn't have direct access to. The hosts on the network seemed to fall into the following segments.

- **Private** – network segment containing Acme employees' workstations and printers
- **Intranet** – network segment for Acme's *in-house* servers
- **Extranet** – network segment for servers accessed by Acme's customers
- **Customers** – segment for the various routers that connect Acme's customers

Like any networking challenge, several alternatives existed for implementing our requirements. Although an exhaustive search of all alternatives was not conducted, we did consider adding access control lists to our routers, deploying a PIX firewall, and configuring VLANs and a filtering router.

The first alternative we considered was configuring access control lists (ACL) in the existing routers. The advantage to this approach is it could be implemented with no additional equipment. However, this would mean maintaining ACLs in several different places, increasing the chance for error. Also, we felt this would make troubleshooting more difficult, as there would be multiple places to look when someone was unable to access a host.

The next alternative we considered was deploying a Cisco PIX firewall. The PIX is available with 6 ports, so it met our requirements. Additionally, the PIX is a very capable firewall up to the task of protecting our network segments from each other. However, the PIX doesn't support IPX, therefore, deploying the PIX would require tunneling IPX over IP. Also, this option would require an investment in separate switches for each network segment.

The final alternative we considered was configuring VLANs on the existing Cisco 2900XL switch and using a router with ACLs to route/filter traffic flowing between the VLANs. The advantages to this alternative are limited investment in new equipment, virtually unlimited number of isolated network segments, and a single location to maintain the ACLs. Additionally, this alternative could handle IPX traffic. The main disadvantage is this alternative may not provide as much security as the other alternatives.

The VLAN alternative seemed like an attractive solution, but we were worried about how much security this solution would provide. In Day 1 of the SANS GSEC training materials, in an overview of VLANs, it is noted "VLANs are a management/performance issue, they are not a security solution." [1] Further research turned up evidence of VLAN-hopping, essentially *tricking* the switch into transmitting packets onto a network segment the user shouldn't have access to. [2]

Some further research, however, turned up a recent report from @stake, which concluded, "VLANs on Cisco Catalyst switches, when configured according to best-practice guidelines, can be effectively deployed as security mechanisms." [3]

We decided to implement VLANs in order to enhance the security of our internal networks. While SANS and our research raised some concerns about using VLANs to secure our network, we felt this approach added an adequate layer of security to our defense-in-depth strategy.

It was time to implement our solution. We used, "Configuring InterVLAN Routing and ISL/802.1Q Trunking on a Catalyst 2900XL/3500XL/2950 Switch Using An External Router" as our guide. [4]

Since we needed to implement this solution with minimal disruption to the production network, our plan was:

1. Upgrade the existing switch off-hours.
2. Configure the switch for our Private VLAN, #2, and put a test workstation on this subnet.
3. Configure the router to route between the VLANs.
4. Move the workstations and printers to VLAN #2.
5. Create the Intranet VLAN, #3, and move Acme's in-house servers to that subnet.
6. Create the Customers VLAN, #4, and move the routers with connections to Acme's customers.
7. The remaining computers on VLAN #1 would comprise the Extranet containing the mainframe, some servers, and the Raptor firewall to the Internet.
8. Implement ACLs on the router to control the flow of traffic between VLANs.

The 2924XL switch was running an older version of IOS than what was required for trunking, so we needed to upgrade to version 12.0(5)WC(1). Also, the switch had previously been configured to have some ports be members of different VLANs, and some ports were members of multiple VLANs. However, this configuration wasn't compatible with the trunking configuration that we wanted, so we removed the current VLAN configuration.

Now we needed our filtering router. Since a majority of our network traffic would be passing through this router, we decided to purchase a new router for this task, rather than configuring the trunking functionality into one of the existing routers. We chose a Cisco 2620 with a single Fast Ethernet port for this task. Since we needed IP and IPX capabilities, we loaded the IP/IPX/AT IOS feature set onto the router.

Configuring VLAN 2 for our Private subnet was very straight forward. On the switch, we simply added the VLAN as follows:

```
2900xl#configure terminal
2900xl(config)#vlan database
2900xl(vlan)#vlan 2
VLAN 2 added:
Name: VLAN0002
2900xl(vlan)#exit
```

Next, we chose a port and assigned it to VLAN 2. This is the port we would plug our test workstation into.

```
2900xl(config)#int fastethernet 0/5
2900xl(config-if)#switchport access vlan 2
2900xl(config-if)#spanning-tree portfast
```

We also needed to configure a trunk port for our router. This port would allow our router to route between the different VLANs without requiring a physical Ethernet port for each VLAN.

```
2900xl(config)#int fastethernet 0/1
2900xl(config-if)#switchport mode trunk
2900xl(config-if)#switchport trunk encapsulation isl
```

At this point, we decided to test things out. We plugged a workstation into port 5 and tried to ping around. However, since that port was configured to be part of a different VLAN, we weren't able to ping.

Configuring the router entailed defining a separate sub-interface for each VLAN, complete with its own IP address. At this point, we needed to interfaces, one for VLAN 1 and one for VLAN 2. Since VLAN 1 already had hosts attached to it, we needed to use an IP address that was valid for that subnet. We also used the correct IPX network since we needed to route IPX.

```
c2620(config)#int fastethernet 0/0
c2620(config-if)#no shut
c2620(config-if)#exit

c2620(config)#int fastethernet 0/0.1
c2620(config-subif)#encapsulation isl 1
c2620(config-subif)#ip address 10.0.0.201 255.255.255.0
c2620(config-subif)#ipx network 1
c2620(config-subif)#exit

c2620(config)#int fastethernet0/0.2
c2620(config-subif)#encapsulation isl 2
c2620(config-subif)#ip address 10.0.50.201 255.255.255.0
c2620(config-subif)#ipx network 2
c2620(config-subif)#exit
```

At this point, our test workstation was able to ping other hosts on the network, access the Novell server, and access the Internet. The only problem was accessing Windows 2000 servers on VLAN 1. This had to be resolved by deploying a WINS server. Once completed, we moved our workstations and printers over to the new VLAN. We don't use DHCP, so this required changing the IP address on each workstation, but overall things went smoothly.

The rest of the VLANs were created just as this one. We changed our original plan and decided to put each customer's router in its own VLAN. We thought this might aid with configuring ACLs, since each customer would have its own interface in the router, and therefore could have its own access list.

With our VLANs implemented, we were now in a position to "lock-down" access to the various subnets using Access Control Lists. Initially, we wanted to use all ingress filters, so we could stop unwanted traffic before it entered the router. However, because customers accessed the Internet through this network, we needed to allow http, ftp, telnet, etc. to virtually any host on the Internet, but block that traffic to any internal hosts. Therefore, we thought it would be simplest to use egress filters on each VLAN interface, only allowing the traffic to those hosts that was allowed.

We decided to use named access lists to simplify management. Our access list limiting traffic to VLAN 1, our Extranet subnet, was as follows:

```
ip access-list extended extranet-out
  permit tcp any host 10.0.0.6 eq telnet
  permit tcp any host 10.0.0.6 eq 21004
  permit tcp any host 10.0.0.6 eq ftp
  permit tcp any host 10.0.0.6 gt 1024
  permit tcp any host 10.0.0.10 eq 1433
  permit tcp any host 10.0.0.15 eq smtp
  permit tcp any host 10.0.0.15 eq pop3
  deny ip any any log-input
```

This access list goes a long way towards limiting the exposure of hosts on our Extranet to unwanted traffic.

We also moved the Raptor to its own VLAN. That way, we could allow all traffic destined for the Internet onto that VLAN, and the Raptor could decide what to do with it.

After

After implementing our solution, the network was very different than it was before we started. Rather than a single subnet, the LAN was comprised of multiple VLANs. Access control lists controlled what traffic was permitted to flow between these VLANs. Also, having computers on smaller network segments reduced the amount of broadcast traffic any computer would see and the number of computers any single computer could communicate with without going through the filtering router. This is illustrated in Figure 2.

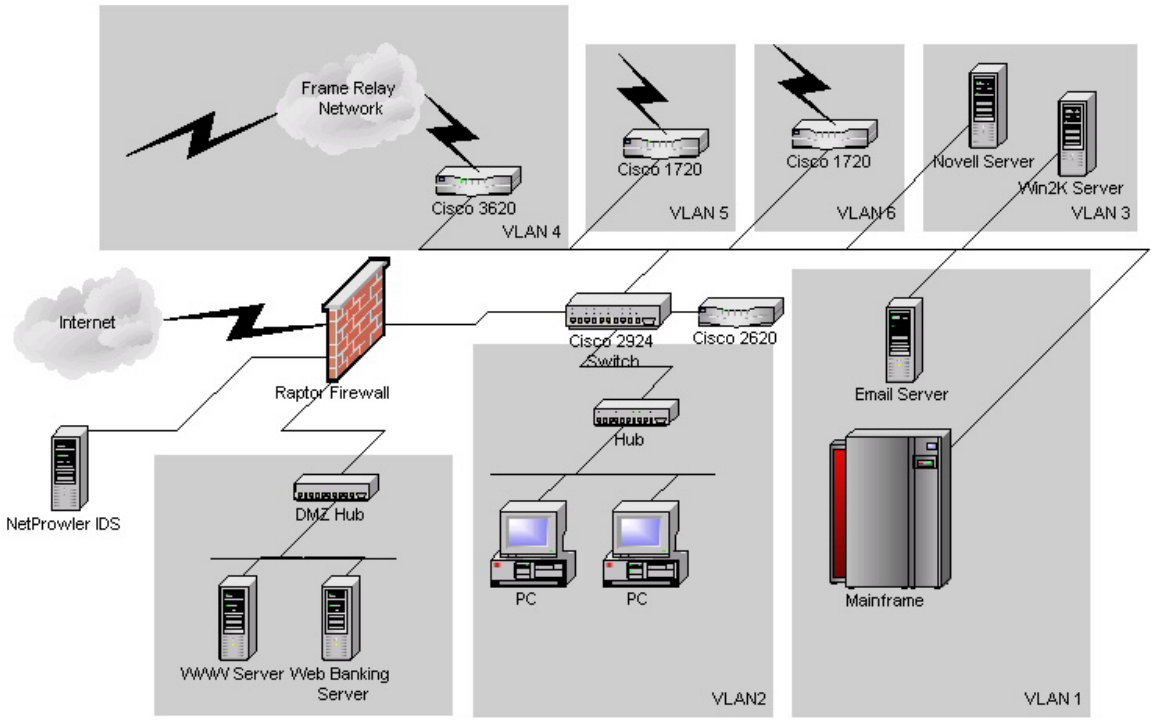

Figure 2

To determine if the network was more secure, we revisited the threats we had identified before this project began to determine if we were now less vulnerable to these threats and therefore had reduced our risk.

The first threat we had considered was an attack from a malicious hacker. Since our solution did not address any vulnerabilities at our customer sites, a hacker could still gain access to Acme's network by compromising a customer's network, perhaps through an unsecured modem or by physical access to the customer's network. Before, we were concerned this hacker would be able to communicate with all of Acme's computers, including the mainframe, workstations, routers, etc. If any vulnerability existed on one of these systems, the hacker could potentially exploit it. With our access control lists in place, however, this hacker would have limited access to only a few hosts on Acme's extranet segment. Since only minimal services would be accessible, the chances of finding an un-patched vulnerability were greatly reduced. Also, since the extranet hosts are on their own segment, if one was compromised, it can't be used as a jumping point to compromise the rest of the network. Therefore, I would say this risk has been significantly reduced.

The second threat we examined was the threat from a virus such as Nimda. We were concerned that if a user downloaded Nimda in an infected email, it would be able to spread throughout the network by exploiting un-patched IIS servers and un-secured network shares. Now, however, access control lists block access to most IIS servers and network shares, so the spread of the virus would be

significantly slowed. Computers that were on the same VLAN as the infected computer would still be vulnerable, and email could still spread the virus, but the overall impact of the attack would be reduced.

The final threat we examined was the threat from an employee running a program that could generate a lot of network traffic, like a port scanning utility. With the network wide-open, the program would have been able to scan a large number of computers, potentially causing a denial of service attack. Now, however, our access control lists have significantly reduced the number of Acme computers that any other computer can access. Of course, the traffic would still be making it to the filtering router, so this threat could still result in a denial of service if the router is unable to keep up with the amount of traffic.

Like most security solutions, this is not a "put it in and forget it" solution. Continual enhancement and tweaking is the only way to stay ahead of the bad guys. Some areas where this solution can be further refined are:

- Implement reflexive access lists to "temporarily" open up ports for return traffic rather than using other, less secure mechanisms.
- Implement the IOS Firewall Feature Set, which provides greater firewall capability than access lists alone.
- Since the frame relay customers terminate in the same frame relay router, they can communicate with each other without going through our filtering router. This needs to be addressed.
- Determine what IP addresses should be allowed to enter any interface and configure ingress access lists to only allow those valid IP's.

Overall, though, I'd say our solution added to the overall security of the network, and was an important part of a defense in depth strategy.

References

1. SANS, "SANS GSEC Security Essentials", Day 1, Page 1-24
2. Dave Taylor and Steve Schupp, "VLAN Security", http://online.securityfocus.com/archive/1/26008
3. David Pollino and Mike Schiffman, "Secure Use of VLANs: An @stake Security Assessment", http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/tech/stake_wp.pdf
4. Cisco, "Configuring InterVLAN Routing and ISL/802.1Q Trunking on a Catalyst 2900XL/3500XL/2950 Switch Using An External Router", http://www.cisco.com/warp/public/473/50.pdf
5. Symantec, "W32.Nimda.A@mm", http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html

6. Tony Bourke, “VLAN with a plan”, http://www.hostingtech.com/nm/01_08_vlan.html