Case Study - Implementing Secure HTTP-to-HTTPS Redirection

Robert Hercey

GIAC Security Essentials Certification (GSEC)
Practical Assignment v1.4b - Option 2
June 15, 2004

Table of Contents

Acknowledgements

I have found myself in the fortunate position of working for a company full of bright, hard working people. While standout individual performances are encouraged and recognized, what makes our company successful is the ability for everyone to come together as a team when a crisis happens. This paper would not have been possible without the efforts of my co-workers in the Hosting, Operations, and Security departments who provided most of the muscle during the implementation of the redirection project. It was their attention to detail and drive to meet the customer requirements that allowed this project to succeed.

Special thanks go to Network Engineers Jeff Thompson, who created the original Visio diagrams that were modified for inclusion in this paper, and Darrin Wiens, who wrote the change control for the 7120 replacement. Both of them also took time out of their busy schedules to make sure I didn't misstate anything regarding the design flow. Editorial review from Systems Engineer Geoff Bard and my wife Keri Carkeek were also appreciated. Finally, without the encouragement and support of the Management team, my GIAC certification attempt would have never gotten far.

Abstract

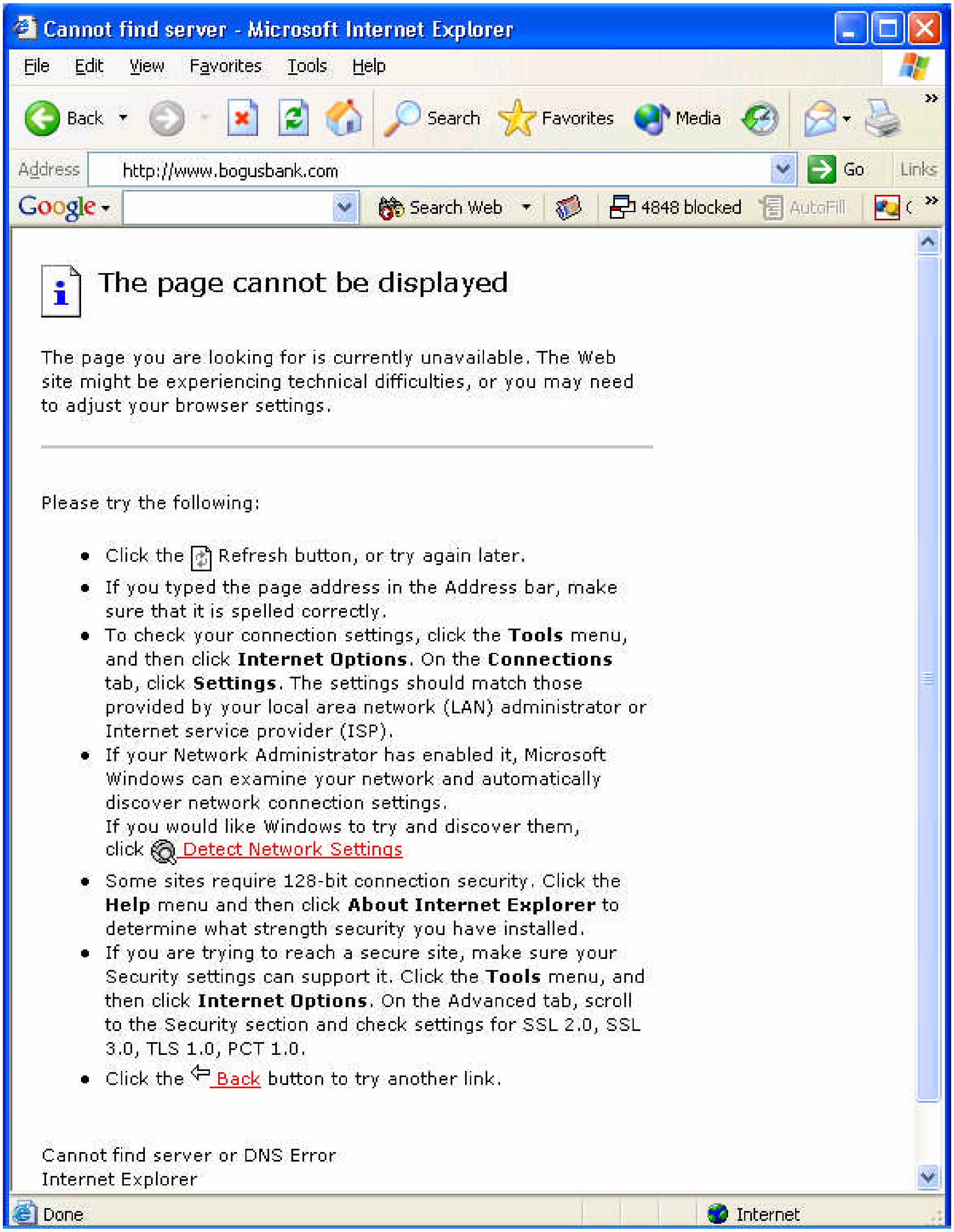
The setting is the Hosting department for a company that provides services and software to companies in the Financial Services industry. An issue came up that forced us to reconcile a long standing Internet Security-related de facto company technical standard with a conflicting customer requirement in a growing live production environment under strict time deadlines. The extreme urgency and wide scope of this change required a "Tiger Team" approach requiring input from Applications, Networking, Management, and Security groups. The real challenge from a security point of view was less a matter of enhancing a security posture and more a matter of fighting to maintain high security standards in the face of the Almighty Dollar.

Specifically, we had an existing customer ("Bogus Bank", one of many Financial Institutions or "FIs" the Hosting department manages) demand that port 80 HTTP traffic requests from end user Internet Banking customers result in a secure HTTPS login page on port 443. This paper documents the steps taken during this multi-phase implementation en route to a final elegant solution that enabled us to achieve all of our objectives, including positioning the department for continued secure and profitable growth for years to come. This entire multi-month project provided great benefits as an educational exercise at both a business and technical level.

Before

Our "before" configuration was a study in the widely accepted Defense In Depth principle, where our strong perimeter defenses were supplemented by additional protections throughout the system. Our customer problem was caused by our first layer of defense - border routers using strong ingress filters designed as a protection for their data which was now causing them grief. All inbound network traffic except the protocol TCP destined for port 443 was dropped on the edge of the Hosting department network space. This long-standing configuration minimized exposure of our Application web servers to the Internet by preventing access to any ports not directly tied to the service of HTTPS-enabled web pages, and was validated by independent auditors as an effective and prudent front line defense mechanism. Note that some additional products, services, and methods used in our implementation are outside the scope of this document and will not be mentioned henceforth, but you can read Clayton Dillard's SANS submission for a nice discussion of Defense In Depth in an eCommerce environment.[1]

While providing an excellent foundation as perimeter security, for some people this configuration resulted in a poor browsing experience. Unless the end user remembered to type the "s" in "https://www.bogusbank.com" in their browser, they would see a very unsettling "This Page Cannot Be Displayed" error.


Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
Go
Links
Search Web
4848 blocked
AutoFill
Done
Internet

http://www.bogusbank.com

The page cannot be displayed

The page you are looking for is currently unavailable. The Web site might be experiencing technical difficulties, or you may need to adjust your browser settings.

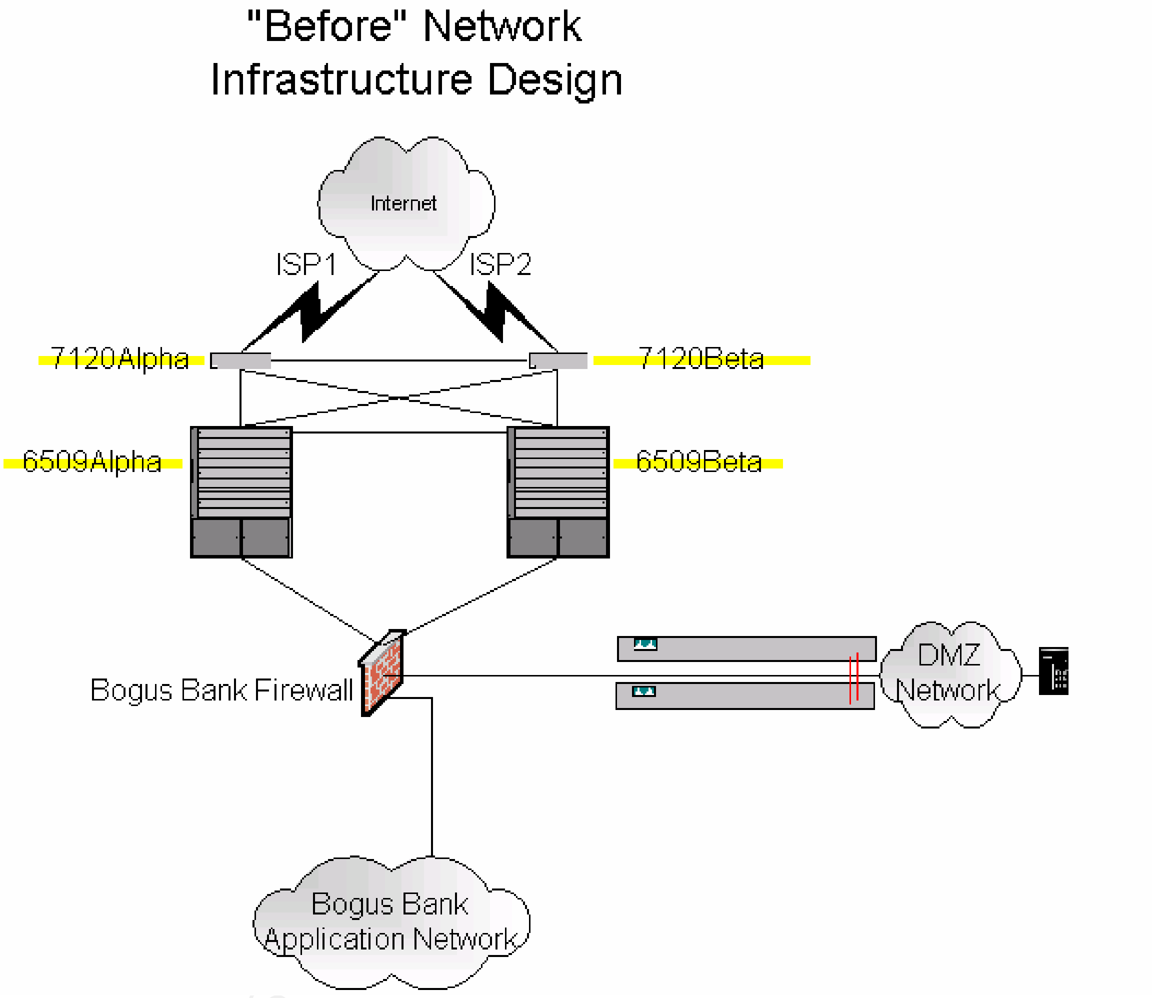
Please try the following:

- Click the Refresh button, or try again later.
- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- To check your connection settings, click the **Tools** menu, and then click **Internet Options**. On the **Connections** tab, click **Settings**. The settings should match those provided by your local area network (LAN) administrator or Internet service provider (ISP).
- If your Network Administrator has enabled it, Microsoft Windows can examine your network and automatically discover network connection settings.
  If you would like Windows to try and discover them, click Detect Network Settings
- Some sites require 128-bit connection security. Click the **Help** menu and then click **About Internet Explorer** to determine what strength security you have installed.
- If you are trying to reach a secure site, make sure your Security settings can support it. Click the **Tools** menu, and then click **Internet Options**. On the Advanced tab, scroll to the Security section and check settings for SSL 2.0, SSL 3.0, TLS 1.0, PCT 1.0.
- Click the Back button to try another link.

Cannot find server or DNS Error
Internet Explorer

Confusion reigns supreme for the uneducated user when this page appears…

This experience was unacceptable to the FI, who lists among their end users non-technical grandma and grandpa-types who don't know or, with all due respect to Lemieux[2], necessarily even care about the difference between secure and non-secure browsing. To build upon Randy Stauber's Lord of the Rings analogy, our ingress filters effect upon the "http://" request were akin to that first volley of arrows taking out grandma and grandpa before they even got close to the walls of Helm's Deep – they were dead before they knew what hit them.[3]



The precarious path grandma and grandpa's "http://" request must take…

To understand how the single "s" in a URL Address field can affect the end user experience so drastically, let us look at what actually happened when the end user tried to get to the login page of Bogus Bank in two ways. Requester one typed "http://www.bogusbank.com" (or just www.bogusbank.com) in the address field of their Internet Explorer browser window and requester two typed "https://www.bogusbank.com" in their address field. In our original design:

1) For both requests (packets), a Domain Name System (DNS) server would translate the name "www.bogusbank.com" to an Internet resolvable IP address. The DNS server would respond to the end user client with that IP

address and off the TCP packets would go into the Internet cloud looking for something to respond on port 80 and port 443 respectively (Wikipedia).[4] Ultimately the cloud would deliver these packets through one of two Internet Service Providers (ISPs) into the network space managed by the Hosting department.

2) The Cisco 7120 router[5] (7120Alpha or its active twin 7120Beta) physically connected to the delivering ISP checked the protocol then the port destination of the packet against this ingress filter:

```
permit tcp any any eq 443
deny ip any any
```

   The first line of this filter specifies that the protocol TCP from any IP address going to any address on port 443 is clear to pass. The second line drops the gate down to prevent anything else from getting through. Since the end user who typed “www.bogusbank.com” or “http://www.bogusbank.com” was in reality asking for access to port 80, the deny line of the filter would kick in and drop the packet. The end user at this point was doomed to see the error page after a timeout period. The “https://” packet however was successfully passed by the router over a Gigabit connection down the line to the next device.

3) A Cisco 6509-NEB 9-slot switch[6] (6509Alpha or its twin standby 6509Beta) routed the packet to the destination FI network using the following modules:

   a. A 48-Port Ethernet Switching Module[7] Model WS-X6348-RJ-45 provided a mechanism for Layer 2 routing and provided a termination point for Layer 1 connectivity to each of our customer FI networks via Fast Ethernet cables.

   b. A Supervisor 1A Engine Model[8] WS-X6K-SUP1A-2GE provided Layer 2 VLAN routing.

   c. A MultiLayer Switch Feature Card (MSFC) Model WS-F6K-MSFC provided Layer 3 routing distribution to each FI.

4) A Cisco PIX 515E[9] Model PIX-515E-FO-BUN firewall (or its twin running in failover mode) existed for every FI as an additional level of security in the Defense In Depth design. On the 6509-side interface, the PIX ingress filter again blocked all but TCP 443 traffic. In this way, the PIX functionally acted like another perimeter device, with the only real difference being the location in the traversal path of the traffic from the Internet. Unique firewalls in every FI network space were designed originally to keep network traffic from any one FI from getting to any other FI, but they had the added benefit of doubly screening Internet traffic. Network Address

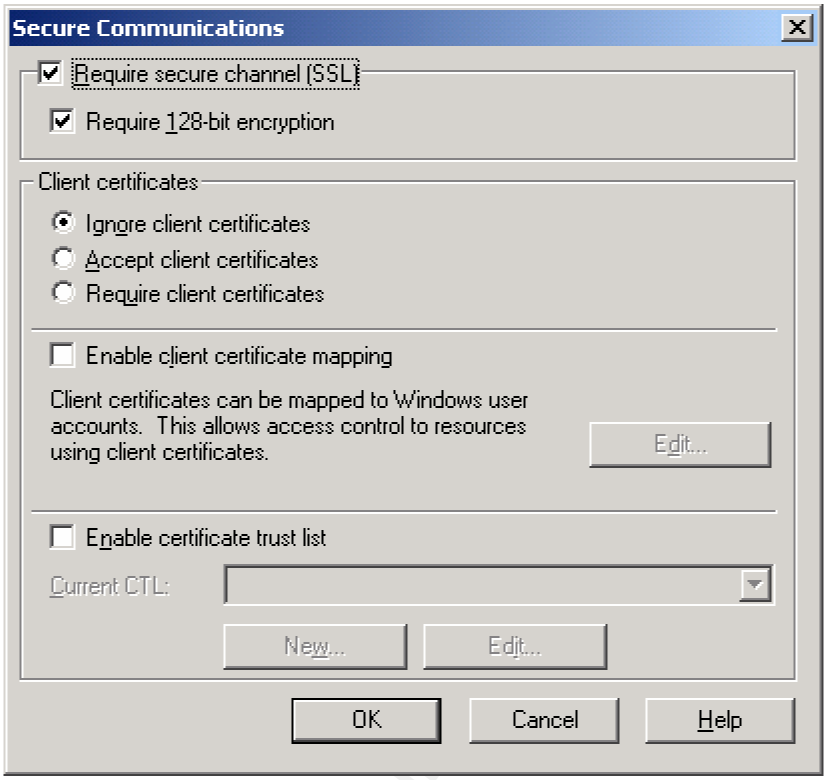
Translation (NAT) was used to create a DMZ of non-Internet routable Bogus Bank IP address space at 10.7.14.0/24 populated by web servers, a load balancer, and the infrastructure to connect them together.

5) Two Cisco 2950 Catalyst® 2950-24 Switches[10] were used to provide the redundant infrastructure for the DMZ over Fast Ethernet.

6) A Radware Web Server Director (WSD) Pro + Application Switch[11] served as a load balancer, providing a virtual IP address on the 10.7.14.0/24 network in front of multiple web servers. The packet would enter the WSD VLAN and the WSD application would be responsible for finding a web server (generally speaking the least-busy one) to handle the request. With an actual termination IP address now specified, the packet was sent on its way to the Internet Information Server (IIS) web server application on that server for processing.

7) The designated web server on the Web Application VLAN would accept the "https://" request on port 443 and establish a session with the end user browser. At this point, the end user browser would display the secure login page and further transactions between the Web Application and end user would proceed in a Secure Sockets Layer (SSL) context. Success!

Interestingly enough, even if the "http://" packet had found a way through the multiple ingress filters and reached the web server, the end user would still have seen the "This Page Cannot Be Displayed" error page. As good Defense In Depth practice tells us, we don't rely on just one layer of protection so on the web server itself the setting "Require 128-bit encryption" is turned on. There is no web site configured on port 80 on the web servers, so requests for non-secure pages would be ignored.



Secure Communications

☑ Require secure channel (SSL)

☑ Require 128-bit encryption

Client certificates

● Ignore client certificates

○ Accept client certificates

○ Require client certificates

☐ Enable client certificate mapping

Client certificates can be mapped to Windows user accounts. This allows access control to resources using client certificates.

☐ Enable certificate trust list

Current CTL:

This IIS web site setting is just one part of a Defense In Depth approach…

During

The constant conflict between ease of use/functionality, security, and profit motive was never more clearly apparent than during the brainstorming of solutions to this issue in the hours following the customer request.

“Nothing is as simple as we hope it will be.”
(Horning)[12]

On the one hand, there was an elegance and simplicity to the “allow only port 443 and deny everything else” model. By putting up those simple solid walls on the perimeter of our network, we were able to defend against the huge amount of traffic generated by worms, scanners, and the like unauthorized noise with a minimal chance of making a configuration error. This was a time tested model that all were reluctant to change. In fact, several previous requests to alter this design to provide “http://” access for other FIs had been considered but denied on the basis of potential security impact and cost.

On the other hand, it was known not just to our customer base but internally as well, that the most secure site in the world wasn’t as good as it could be while some percentage of the end user base couldn’t access it. The long standing port 443-only design was signed off on by the FI as part of the original project scope, but only after the new Internet Banking site went live and other senior members of the FI went through the experience firsthand were the ramifications of the secure-but-simple design fully realized. High level discussions inside the FI and then between the FI and senior management in our company took place, at which point it became apparent that the issue was of severe enough importance to once again consider making a change to the de facto departmental security standard. We were able to demonstrate in the scope documents that this functionality was not correctly able to be called a bug, so the project was determined to be an enhancement request. As such, any change in end user experience would potentially be a source of increased revenue for the department. More high level discussions ensued, and with the understanding that the FI would subsidize any permanent secure "http://" redirection implementation as a new monthly service charge on top of their existing contract, it became Hosting's highest priority project.

At that point, technical brainstorming discussions commenced in earnest to determine potential solutions including associated costs and timelines. Technical representatives from our department began looking each piece of the entire global design to discover a way the requirement could be met in a timely and secure fashion. The hope was that we already had a piece of equipment that could be leveraged to perform this functionality without incurring additional costs. If we could find that device, and it could implement redirection in a manner consistent with our security policies, we had a clear win-win. The Network team was tasked with exploring the capabilities of the Cisco-based networking equipment already being employed. An Application team was formed to look into options involving the rest of the hardware and software elements in the existing implementation.

As is often the case for technical professionals these days, Google searches were put into motion looking for answers from people who may have had to deal with this problem in the past. A count of the number of items returned from variations on searches for "http to https redirection" showed, not surprisingly, that many others have had to deal with this issue. Many of the offered solutions were confined to performing redirection on the web servers themselves. From functional and cost perspectives, these solutions might have gotten the job done, but the SANS-educated representatives from the Security department flatly rejected any option that would allow any currently employed web server be configured to listen on port 80. The promise of quick and recurring revenue had caused the Security department to reconsider a long standing policy, but it was clear that some things were non-negotiable. 128-bit encryption was still going to be required on every existing web server, and IIS was going to continue to serve up pages on port 443 only. Programming or scripting-based solutions on the existing web servers themselves were ruled out. This stance framed the investigation to follow.

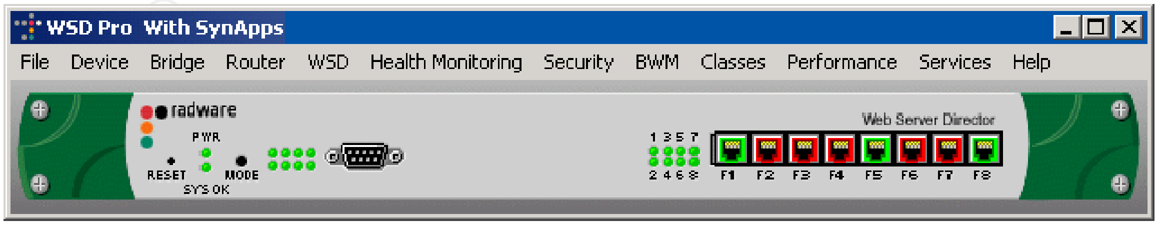
The Networking team began their investigation into their devices one by one. The most secure place to make the change appeared to be on the 7120 routers. Given their status as the first line of defense on the perimeter, the Security department wanted to see the redirection take place out there, where port 80 penetration into the network space would be kept to an absolute minimum. The Network group investigated the router capabilities but found no mechanism for implementing protocol redirection. An “http://” packet coming in from the outside interface stayed an “http://” packet, though it could be successfully moved to the 443 port. Strike one.

Looking down the packet traversal path, the 6509s and in particular the MSFCs were the next logical device to consider as they were already performing routing on Layer 3. Testing however revealed again that though we could successfully redirect the “http://” request to the secure port 443, the web server itself would only return “https://” responses in response to “https://” secure requests. Strike two.

The PIXes were the last infrastructure piece in the design, but we knew by this point that any Layer 3 solution alone wasn’t going to solve our problem. At this point due diligence was considered complete and it was concluded that none of the currently installed Cisco infrastructure network pieces were capable of meeting the requirement. Strike three!

A call to Cisco was scheduled to look for Cisco-supported designs that could do port and protocol redirection. In particular, Google searches had returned some promising results around the Cisco Local Director product that needed to be discussed.

The Application team had a couple of items to look at in the current implementation. Only one non-Cisco, non-Microsoft device existed in the path from end user browser - the WSD. Protocol redirection seemed to be a natural extension of the load balancing functionality already in use. The fact that “Application Switch” existed in part of the name of the device suggested higher-than-Layer 3 redirection. With an abundance of menus and sub-menus in the graphical Configware software interface, it was difficult to locate exactly how to configure this desired functionality.



Many of these menu items have six or more submenus – GUI overkill!

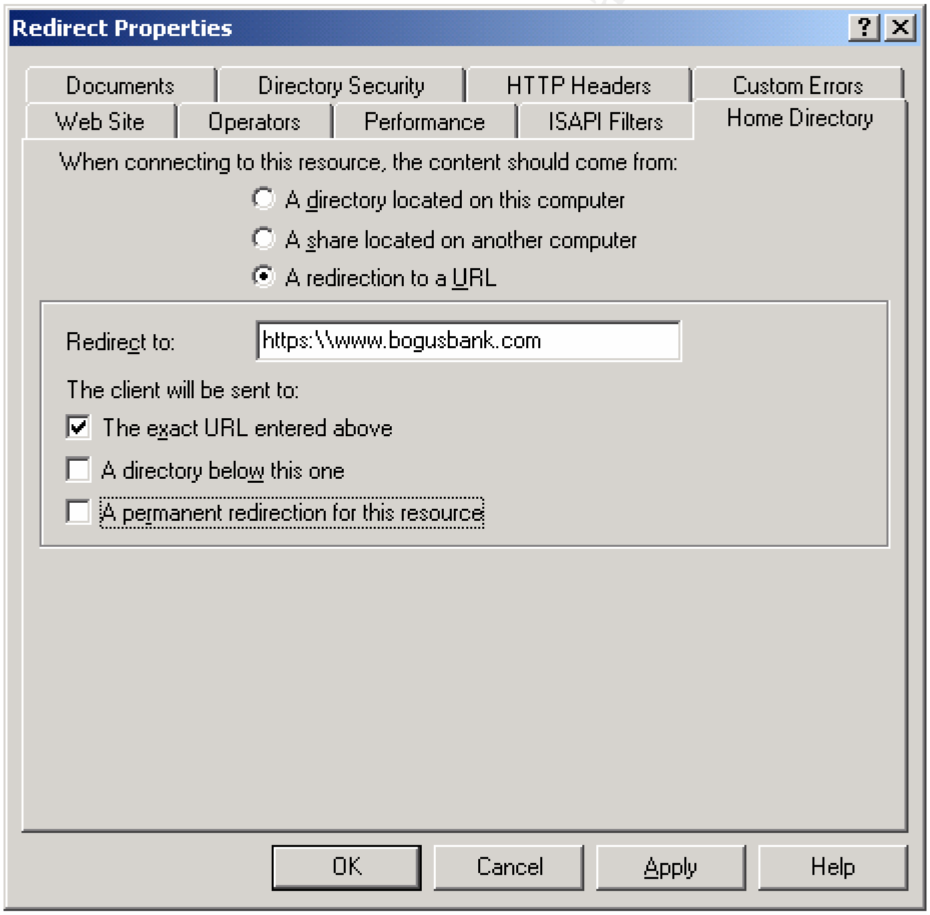
Eventually, Radware was engaged directly to determine if the product supported protocol redirection and how to configure it. Much to our pleasure, we were told that indeed protocol redirection was an offered feature of their product. We thought we had found our silver bullet. Closer inspection revealed otherwise.

It turned out that the software level we purchased for the WSD originally did not support protocol redirection. Radware representatives said they could offer us this functionality for an upgrade fee. This option was considered but ultimately dismissed on several grounds. A big strike against the WSD upgrade option was the stated goal of keeping implementation costs to a minimum. Any up front cash expenditures were viewed as something to be avoided if possible, and delays on getting the redirection service up meant delays on when the new revenue could start flowing in. The Radware solution was especially expensive because there were no discounts being offered either on trade-ins or for volume discounts. Another big problem with the WSD was its limited logging ability. The Security department was especially interested in having only the highest possible logging level on any device inside the FI network that would be the recipient of the port 80 "http://" traffic, and the WSD didn't offer much in this area. Fortunately, as the Application team continued its investigation, simultaneously a different and essentially cost-free option was making itself known – namely, the Redirecting IIS server.

In the end, there were two implementation phases we went through in configuring our environment to handle the FI requirement of HTTP access to the Internet Banking login page in a secure manner.

The new Redirecting IIS server was formally proposed as Phase I. This design was put forth by the Application team as a concession to the urgency and up front cost requirements of the problem at hand. A potentially always-under-attack bastion host Windows 2000 server was envisioned, its sole purpose being a focal point for providing IIS redirection to the "https://www.bogusbank.com" virtual IP address. The server would be truly stand-alone, with no connectivity to the FI domain in any way so if the system was compromised the damage would be minimal. Just about every service that wasn't in direct support of serving up IIS redirection content was to be shut down or disabled. Logs would be shipped to a central storage location for daily review. All the latest Microsoft service packs and patches were to be applied in accordance with industry best practices (K. Dillard).[13] As a Hosting department, we already had an established procedure and automated tools for helping secure and monitor production grade Windows 2000 servers, and this experience was a strong argument in favor of the design. Another benefit of this design was the way we could setup this server behind the perimeter walls and test out the functionality right on our Bogus Bank network before allowing any exposure from the Internet. The Security department gave their approval to test this as a temporary solution. An existing but unused Compaq ProLiant DL360 server was requisitioned, configured, and setup to be monitored for signs of attack at DMZ address 10.7.14.49. Physically, this server was directly connected to the WSD device via a crossover cable and it was given a "real" IP address or 172.16.1.1, not on the same network as the web servers to which it would redirect. The WSD served up 10.7.14.49 to the PIX-side interface while keeping "real" IP address hidden.

Unlike any previous device on a Hosted network, once the World Wide Web Publishing Service on the Redirecting IIS server was started we had an IIS 5.0 web server listening on port 80. This web site contained no content, only built-in IIS functionality for providing redirection to port 443. Setting it up was simple. We created a new site called "www.bogusbank.com", selected Properties for it, navigated to the Home Directory tab, and selected the radio button "A redirection to a URL." In the "Redirect to:" field we entered the existing secure page for Bogus Bank, "https://www.bogusbank.com", and checked the box next to the "The exact URL entered above" text. Selecting "OK" to get out of the screen made our changes effective and local testing was successful immediately thereafter.



When connecting to this resource, the content should come from:

○ A directory located on this computer
○ A share located on another computer
● A redirection to a URL

Redirect to: https:\\www.bogusbank.com

The client will be sent to:

☑ The exact URL entered above
□ A directory below this one
□ A permanent redirection for this resource

IIS 5.0 has port and protocol redirecting capability built in…

In order to let Internet traffic get to the redirecting IIS server, network infrastructure reconfiguration was required. It was necessary to poke a hole in the border routers ingress filters to allow TCP port 80 traffic into the Bogus Bank network space only as follows:

```
permit tcp any host {Bogus Bank Internet-routable IP address} eq 80
```

The PIXes then needed to have a route defined to point http port 80 traffic to the Redirecting IIS server at 10.7.14.49 via the WSD. Still, even with the route in place, the http traffic would be blocked unless the following line was added to the ingress filter of the PIXes:

```
permit tcp any host 10.7.14.49 eq 80
```

It was understood that this design had a real pitfall. Of gravest concern, this design allowed the unsavory traffic that transmits on port 80 to get closer to the real crown jewels of the operation - customer data that resided on the even more interior Application Network on the other side of the PIXes. The Security department blessed this configuration but in the same breath mandated a more permanent solution that better protected the customer data be implemented as soon as possible.

The design for Phase II came about after direct consultation with Cisco. The security requirement called for a move back to a configuration that kept port 80 on the edge of Hosting network space. Hosting Management had the additional request of being able to extend the redirection functionality (and thus revenue potential) to all Hosted customers. We learned quickly that the Local Director solution previously "Googled" didn't have the horsepower or scalability we were looking for as a departmental solution, and Cisco shut the door on that solution completely when they told us they were phasing out the Local Director product. They steered us towards the general solution they termed "Content Networking."[14] There were a number of devices in this realm that met our requirements at a functional level. Some of them were discounted because they were blade solutions that would have fit into the 6509s, not where we needed them to be at the edge of our network. Others, like the SCA 11000 Series Secure Content Accelerator,[15] decrypted SSL traffic before it arrived on the actual web servers. Given these Security-related constraints, the best fit looked to be a product called the Cisco CSS 11501 Content Services Switch.[16]

We liked the CSS because it was designed with Layer 4-7 traffic management in mind and could handle decent traffic loads. We saw how we could limit its role in the network path to provide port and protocol redirection for customers willing to pay for it, and place it next to the border routers to meet the security requirement. Though the CSS devices (we'd want two for redundancy) weren't inexpensive, they were less expensive than the WSD and the monthly revenue Bogus Bank was to give us for the solution projected to offset the cost of the CSS switches by the end of the existing contract. This looked promising indeed, though we anticipated installing the devices might be a little tricky. All the of NAT configurations described on the PIXes on the Bogus Bank network would need to be moved up to the border routers when the CSS was implemented.

We were then forced to look at the proposed CSS connection point on the edge, the 7120 border routers themselves. Projections of continued growth in bandwidth usage and a desire for more redundancy in connectivity to the Internet had us considering adding more ISPs to our design even before the redirection project fell into our laps. The connectivity requirements of the additional ISPs would require additional physical interfaces on top of those needed by the CSS devices, and we already were out of those resources. Additionally, we had hit the ceiling regarding memory expandability on the 1720s which, in our eyes, made them obsolete. In fact, over a year had passed since Cisco had End of Life'd the line. The decision was made to put these workhorses out to pasture as part of the redirection project. The new edge devices would be 7206VXR/NPE-G1s[17] with beefed up memory and plenty of physical interfaces to handle ISP growth and content redirection. Between the 7206s, CSSs, and supporting infrastructure all the desired requirements would be met as part of what we were now calling the Front End Redesign.

Negotiation between our company and vendors offering us the required Cisco equipment continued for some time before a final price (including a nice return for our existing trade-in equipment) was agreed upon. For a complete list of the equipment purchased and costs, please see Appendix I. Note that actual costs were lower due to our company's existing reseller relationships. Eventually, the devices started arriving onsite and over the next several months things fell into place through a series of change controls that were scheduled for our department's weekly maintenance window.

In the first change control, the 7206s were moved into production. Our redundant connections to the Internet allowed us to take one 7120 offline without impacting service for our customers. Bandwidth to the Internet was reduced during the time of this change control but at no time was service availability an issue because of the low traffic volumes during the implementation. Configurations were backed up to a central management storage location. Border Gateway Protocol (BGP) was stopped on the first router to stop advertising routes to and from the Internet. This bled off traffic from that router over the course of the next several minutes. Once traffic was confirmed to have stopped on the router, the device was powered off. The High-Speed Serial Interface (HSSI)[18] card was removed from the 7120 and installed in first 7206VXR router. Cables were taken from the 7120 and added to the 7206 which was then powered up. The configuration information for the HSSI interface was loaded and connectivity tests were performed. Finally, BGP was enabled, advertising the routes to the Internet, and traffic then began to flow through the first 7206. The process was then repeated on the second 7120/7206 pair. With this out of the way, the department was in a position to physically connect the CSS devices, and just as importantly support department-wide network growth with no degradation of our perimeter defenses.

The second change control is pending implementation, and calls for the CSS redirecting devices to be connected to the 7206s and the NAT to be moved from the FI PIX to the 7120s.

The third and fourth change controls are also pending implementation, and will utilize the extra interfaces on the 7206s for connections to two more ISPs.

After

Let's look how the Front End Redesign will treat the "http://www.bogusbank.com" request from grandma and grandpa's computer:

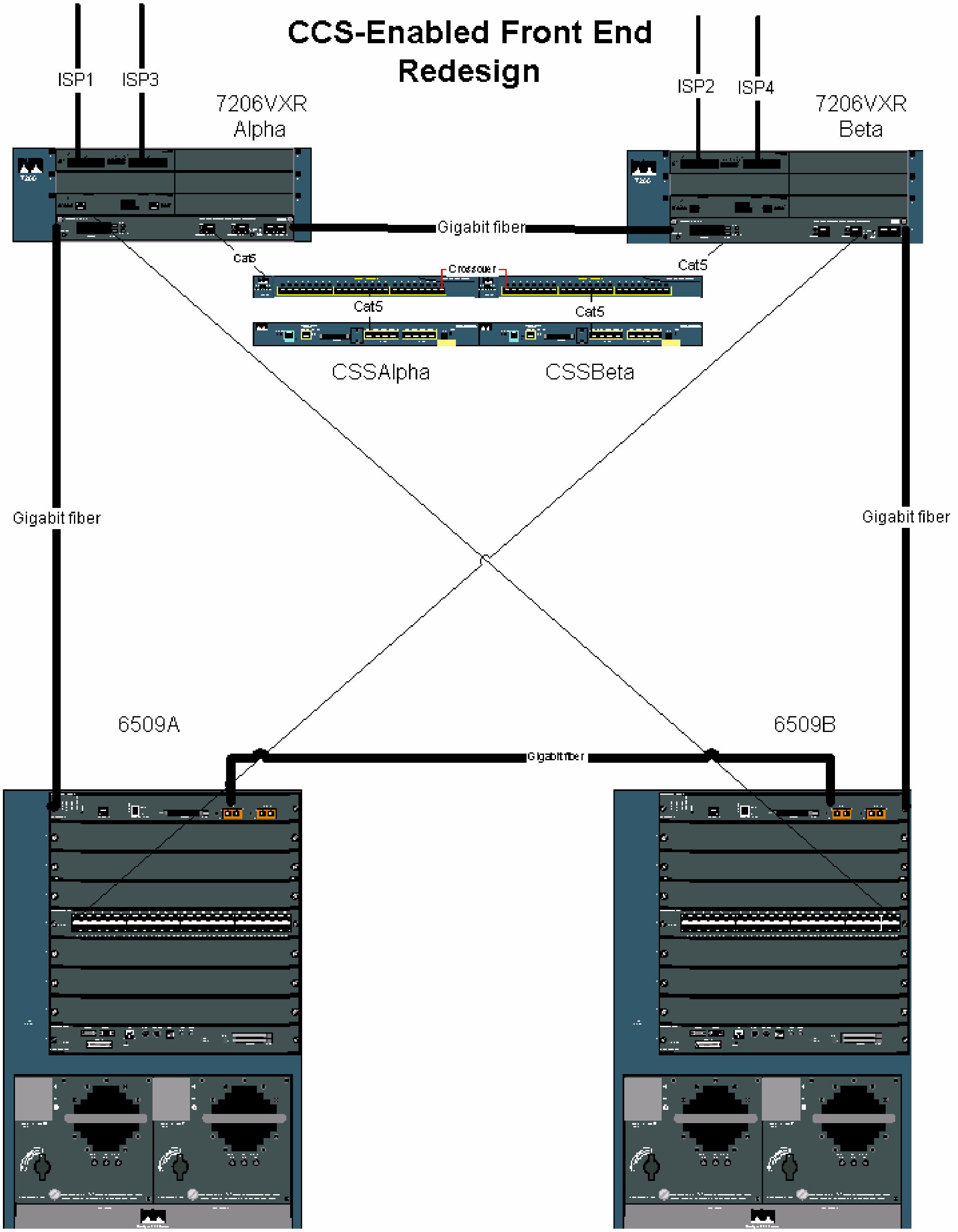
1) Any one of four ISPs delivers the request to the Hosting department network space. Not only is there more redundancy in place to handle the request should one or more of our upstream providers fail, but with potentially fewer Internet router hops in the path many requests should get to our network space quicker.

2) One of the new Cisco 7206 routers checks the protocol and port destination of the request against the ingress filter:

   ```
   permit tcp any any eq 443
   permit tcp any host {Bogus Bank Internet-routable IP address} eq 80
   deny ip any any
   ```

   Any TCP packets going to port 443 are allowed straight through the device towards the 6509 just as before. The second line allows any TCP packets going to port 80 at Bogus Bank's network space. The routing table on the 7206 then shunts this port 80 traffic to the connected CSS device. The final line of the filter again drops all other traffic.

3) The destination CSS accepts the port 80 request and responds back to grandma and grandpa's browser with the redirecting URL "https://www.bogusbank.com". This is the same functionality that the Redirecting IIS server was performing in the Phase I solution. The subsequent automatic request from grandma and grandpa's browser will be an appropriate "https://" request and thus flow neatly through the network to the secure Bogus Bank login page.



Scalability, redirectability, redundancy and security have been achieved…

In our “after” configuration, the metaphorical sentries on our walls have been trained to recognize grandma and grandpa’s requests to Bogus Bank’s network as appropriate, and will hold fire before directing them towards a satisfactory end user experience via the CSS devices.

Despite significant pressure to relax our long term security standards in the face of increased profit margins on Bogus Bank, a high level of security has been maintained and arguably improved through expanded logging capabilities on the 7206s. All our other customers now have the option to upgrade their service to use port and protocol redirection capability, so a new revenue-generating opportunity exists on those and future customers. Ultimately, our department has put ourselves in a position where we can continue secure and profitable growth for years to come.

Along the way, we got some other nice benefits as a result of this project. The technical knowledge gained on the capabilities of our devices was invaluable. This project illustrated perfectly the way even the best designed systems have to be able to evolve to survive in the face of growth and new customer requirements. Finally, at a process level it was great seeing firsthand the way a good checks and balances system can revolve around the core of a strong organization and result in a better customer offering in the end.

Appendix I

Part#	Description	MSRP	Quantity	Total
	Hosting Hardware			
7206VXR/NPE-G1	7206VXR with NPE-G1 includes 3GigE/FE/E Ports and IP SW	$22,000.00	2	$44,000.00
PWR-7200	ADDITIONAL CISCO 7200 POWER SUPPLY 280W	$3,000.00	2	$6,000.00
MEM-NPE-G1-1GB	TWO 512MB MEM MODULES 1GB TOTAL FOR NPE-G1 IN 7200	$7,500.00	2	$15,000.00
MEM-NPE-G1-FLD256	CISCO 7200 COMPACTFLASH DISK FOR NPE-G1 256MB	$1,990.00	2	$3,980.00
C7200-I/O-2FE/E	CISCO 7200 INPUT/OUTPUT CONTROL W/ DUAL 10/100	$3,400.00	1	$3,400.00
WS-G5484	CATALYST SERIES 1000BSX GBIC MOD	$500.00	8	$4,000.00
Cables	Fiber cables 1 meter	$38.00	4	$152.00
Cables	Fiber cables 3 meter	$48.00	4	$192.00
	Total of Hosting Hardware			**$76,724.00**
	Hosting Maintenance			
CON-SNT-7206	SMARTnet 8x5xNBD for Cisco 7206 Modular Router	$3,150.00	2	$6,300.00
CON-SNT-WSSVCNAM1	MAINTENANCE 1YR NBD 8X5 SMARTNET	$715.00	2	$1,430.00
	Total of Hosting Maintenance			**$7,730.00**
	Total of Hosting Hardware and Hosting Maintenance			**$84,454.00**
	Bogus Bank Required Expense			
	Bogus Bank Hardware			
WS-C2950-24	Cisco 2950 switch	$995.00	2	$1,990.00
CSS11501-2PK	2PK CISCO CSS11501 SERVICES SWITCHES	$19,995.00	1	$19,995.00
C7200-I/O-2FE/E	CISCO 7200 INPUT/OUTPUT CONTROL W/ DUAL 10/100	$3,400.00	1	$3,400.00
	Total of Bogus Bank Hardware			**$25,385.00**
	Bogus Bank Maintenance			
CON-SNT-C2950-24	8x5xNBD Svc, C2950: 24 port 10/100 autosense/autonego	$55.00	2	$110.00
CON-SNT-CSS115012	SMARTNET 8X5XNBD Cisco CSS11501 Two-Pack	$2,419.00	1	$2,419.00
	Total of Bogus Maintenance			**$2,529.00**
	Total of Bogus Bank Hardware and Bogus Bank Maintenance			**$27,914.00**
	Total of Hosting totals and Bogus Bank totals			**$112,368.00**
	Trade In credit received			**$10,320.99**
	Total cost after Trade In			**$102,047.01**

Trade In	Description	Quantity	
CISCO7513	Cisco 7513 13-Slot, Dual Bus, 1 RSP2, 1 PS	2	$1,547.68
PWR/7-AC=	Cisco 7000/7507 AC Power Supply, US (Spare)	4	$345.47
VIP2-40=	VERSATILE INT. PROCESSOR-2,MODEL 40	2	$709.35
RSP2=	CISCO 7507/7513 ROUTE SWITCH PROCESSOR SPARE	2	$575.77
CX-AIP-DS3	ATM Interface, DS3 Coax, 45 Mbps A	2	$1,031.78
CX-EIP2	2-Port Ethernet Interface Processor	1	$451.41
CX-EIP6	6-Port Ethernet Interface Processor	1	$1,031.78
CX-FSIP8	8-Port Serial Interface Processor	2	$1,031.78
CX-TRIP2	Two Port Token Ring I/f Proc.	2	$741.60

References

[1] Dillard, Clayton T: "eCommerce and Defense In Depth." Oct. 21, 2001. URL: http://www.sans.org/rr/papers/index.php?id=571 (14 Jun. 2004).

[2] Lemieux, Kimberly: "Shopping for Security." Aug. 2002. URL: http://www.sans.org/rr/papers/index.php?id=869 (14 Jun. 2004).

[3] Stauber, Randy: "Defense In Depth." May 28, 2004. URL: http://ebcvg.com/articles.php?id=219 (14 Jun. 2004).

[4] Wikipedia: "Domain Name System." Jun. 3, 2004. URL: http://en.wikipedia.org/wiki/DNS (14 Jun. 2004).

[5] Cisco Systems: "Cisco 7120 Router." URL: http://www.cisco.com/en/US/products/hw/vpndevc/ps333/ps334/ (14 Jun. 2004).

[6] Cisco Systems: "Cisco Catalyst 6509 Switch." URL: http://www.cisco.com/en/US/products/hw/switches/ps708/ps711/index.html (14 Jun. 2004).

[7] Cisco Systems: "Cisco Catalyst 6500 Series 48-Port 10/100/1000 WC Ethernet Module." URL: http://www.cisco.com/en/US/products/hw/modules/ps4835/ps5171/index.html (14 Jun. 2004).

[8] Cisco Systems: "Cisco Catalyst 6500 Series Supervisor Engine 1A." URL: http://www.cisco.com/en/US/products/hw/modules/ps2797/ps4342/index.html (14 Jun. 2004).

[9] Cisco Systems: "Cisco PIX 515E Firewall." URL: http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/ps4094/index.html (14 Jun. 2004).

[10] Cisco Systems: "Cisco Catalyst 2950 24 Switch." URL: http://www.cisco.com/en/US/products/hw/switches/ps628/ps627/index.html (14 Jun. 2004).

[11] Radware: "Radware WSD-Pro+." Aug. 3, 1998. URL: http://www.radware.com/content/document.asp?_v=about&document=2636 (14 Jun. 2004).

[12] Horning, Jim. "Jim Horning's Home Page." May 19, 2004. URL: http://home.comcast.net/~jhorning4/ (14 Jun. 2004).

[13] Dillard, Kurt: “Intrusion Detection FAQ: What is a bastion host?” Jun. 12, 2003. URL: http://www.sans.org/resources/idfaq/bastion.php (14 Jun. 2004).

[14] Cisco Systems: “Content Networking.” URL: http://www.cisco.com/en/US/products/hw/contnetw/index.html (14 Jun. 2004).

[15] Cisco Systems: “Cisco SCA 11000 Series Secure Content Accelerators.” URL: http://www.cisco.com/en/US/products/hw/contnetw/ps2083/products_configuration_guide_chapter09186a00801159f0.html (14 Jun. 2004).

[16] Cisco Systems: “Cisco CSS 11500 Series Content Services Switches.” URL: http://www.cisco.com/en/US/products/hw/contnetw/ps792/ (14 Jun. 2004).

[17] Cisco Systems: “Cisco 7206VXR Router.” URL: http://www.cisco.com/en/US/products/hw/routers/ps341/ps349/index.html (14 Jun. 2004).

[18] Cisco Systems: “High-Speed Serial Interface Port Adapter for Cisco 7000 Series - Cisco 7600 Series Routers.” URL: http://www.cisco.com/en/US/products/hw/routers/ps368/products_data_sheet09186a0080091c30.html (14 Jun. 2004).