Global Information Assurance Certification Paper

Mike Powell
GSEC Practical version 1.4b, option 2

Setting Up Controlled Virtual Private Networks Using Microsoft's Proxy Server and Routing and Remote Access Service

Abstract

Sometimes it seems like business needs are not in harmony with, or even run counter to, basic security requirements. I have written this paper to show how I was able to provide for a business need while still maintaining an acceptable level of security.

In the **Before** section, I give a brief description of our network. I describe the clients, servers, services, and connections that make up the systems that I support at my company. I explain how I felt that our systems were quite secure, and how satisfied I was to be able to make that claim.

I then go on to describe how, while demonstrating the amazing array of services that could be available to even a remote user, I was asked by our management to do the unthinkable: allow home users to access our systems remotely, from their personal computers that were completely outside of the company's control.

In the **During** section, I describe my situation in a bit more detail. I outline my thought processes in determining what the minimum access was that would be needed by our users.

Next, I explain how I began to make a plan for setting up such a connection in a secure manner. I detail my plans for securing the actual clients, ensuring that data is transmitted securely, and finally authenticating users before they are allowed to make a connection to our network.

Then I arrive at the heart of the problem. Allowing remote users on home computers to connect to our file-sharing server using the standard Windows methods was not an acceptable situation. Too many Windows services use the same ports for communication, and I was concerned that allowing machines outside of my control to communicate to our internal systems without strict limitations on what they could access would be asking for an attack similar to Nimda. Unfortunately, there was no simple and clean way to separate these Windows services that was within my limited budget.

Moving along, I explain my decision to look for alternate services to provide the functionality that my users required while still allowing for the traffic to be limited to only a very narrow channel of communication. My research led me to choose FTP to provide file access for users, and Windows Terminal Services for remote access to our ERP application.

Setting up the VPN connection hit only a minor snag, but the issue that almost stopped me was when I found that Microsoft Proxy Server, the product that controlled and monitored our Internet traffic, did not support filtering traffic that was inbound to the internal network.

After some research, I overcame this limitation by using the packet filtering capabilities that are built-in to the Windows Routing and Remote Access Service. I was able to get the results that I was looking to achieve with only a slight modification to my rule base.

The end of this section is an overview of how I tested the new configurations to ensure that unwanted traffic was not being allowed onto the network, while still allowing the functionality that I needed.

The **After** section details the steps that we now go through to set up users with remote access, and how well it has worked out.

This section also goes into some detail about some of the plans I have for continuing to make sure that remote access, and our Internet access in general, remain secure. Some of these items include replacing our Proxy server with a Cisco PIX firewall, using the Proxy server behind the PIX for outbound user authentication and logging, and researching systems that would be able to check remote computers for required security software before allowing them to communicate on the network.

And, as always, user requests for new functionality ensure that my job is never done.

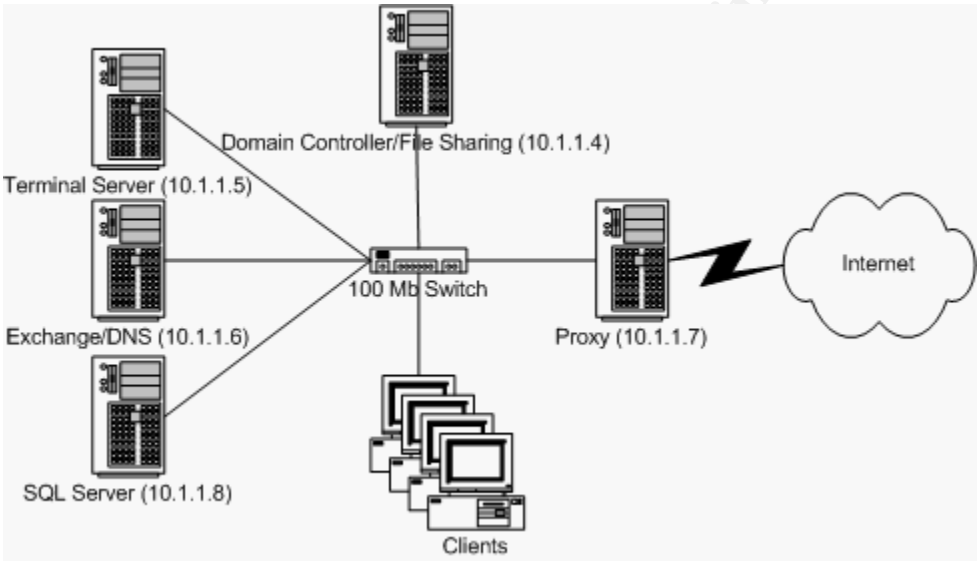
Before

I work as a Network Administrator for a small manufacturing company. The title of Network Administrator is the quick way of saying I handle everything from A–Z, including the kitchen sink. While it can be a challenge, it can also be advantageous since I am able to make decisions and configure the systems without having to rely on other groups being responsible for various parts of the network.

Our network environment is quite simple. We have about 60 client computers, all Windows 2000 Professional except for the Marketing department's Graphic Designer, who runs a Macintosh OS 10 machine.

We also have several servers. First, we have a Windows 2000 Server running Microsoft SQL Server that is the back end for our ERP (Enterprise Resource Planning) software. Next, we have another Windows 2000 server that serves as both a domain controller and provides file-sharing services. Then we have another Windows 2000 server that runs Microsoft Exchange server for email and groupware services and also provides DNS name resolution. Finally, our network clients connect to the Internet through a Windows 2000 server running Microsoft's Proxy server. The Proxy server controls client's access to our Internet connection and uses Microsoft's SMTP server to relay mail between our Exchange server and the Internet.

Other than the Proxy server, none of the servers can access the Internet and no machines external to our network can touch any of our internal machines, except through SMTP relay to our Exchange server. Clients can only access the Internet by going through Proxy server. All clients and servers are connected via switched 100-megabit connections. A simple network diagram illustrates our network:



While there are many additions and changes that I would like to make to our systems, I have to deal with outside constraints such as the tight budgets and stringent ROI requirements of a small business. This is why our Internet connection is still handled by Microsoft Proxy server instead of a dedicated device like a Cisco PIX or Checkpoint Firewall-1.

Instead of installing many of the products that are on my security "wish list", I have contented myself with making sure that the systems we currently have in place are configured for the best security possible. While no network or system is bulletproof, I felt that I had reasonably secure systems in place. In fact, I would venture that our systems were more secure than many other businesses of our size.

Then I made a terrible mistake. No, I didn't leave a key port open, or forget to disable an unused service. I demonstrated Windows Terminal Services to several of our managers, and talked about how this could be used in the future to give remote employees access to our ERP application, which is otherwise very bandwidth intensive.

The managers seemed impressed, and I thought that I had scored a few brownie points for "planning for the future". Little did I know that this demo would come back to haunt me very soon.

For some time, my manager had been trying to convince the top management at our company to provide laptops for the members of the management group. This would allow them to work on documents at home, and also to be more flexible while at work. The top management didn't buy it. Laptops for 10 people at $2500 each were just too much money to spend on the potential of a slight increase in productivity.

However, my demonstration with Terminal Services got them thinking. Most of the managers had personal computers at home. Why couldn't we just let the managers have remote access to our systems from their home computers? After coming to this obvious (to them) solution to the problem, they announced the decision that they had come to without any input from the IS group: All managers would be given remote access from their home computers to everything that they normally had access to while on site. And the best part about their solution is that we, in IS, should be able to implement it with almost no cost!

My challenge had been set before me: give end users remote access from their home computers at near-zero cost without sacrificing the security of our network.

During

What To Do?

At first, my mind boggled when I began thinking of the many possibilities to provide users with remote access to our network and the data that they needed. In order to provide a more clear-cut choice, I decided to define exactly what I needed to provide to our remote users in order to accomplish my goal.

The primary access that was needed was to user's personal files. At our company, each user has a personal directory on the file server. I needed to provide each user with access to their individual files. Additionally, everyone wanted to be able to access our ERP application. Fortunately, the managers had been warned during my initial demonstration that remote access to our ERP system would require the purchase of an additional Windows 2000 server in order to provide access via Terminal Services, plus licenses for any clients that were not running Windows 2000 or Windows XP Professional.

Since I had to begin somewhere, I decided that I would start at the remote clients and plan my way back into our network. Therefore, the first step in my plan was to put in place policies that would help to ensure that remote users would not become unwitting backdoors into our network.

I began with a policy that addressed requirements for remote access by home users, and the first item in the policy was a requirement for any remote users to be running anti-virus software that was kept up to date with new virus definition files. Also required was personal firewall software, along with a hardware firewall for DSL, cable modem, or other "always on" users. Any remote users would be required to submit the details of their software and hardware choices to the IS department for approval before they would be granted permission for remote access.

The next part of my plan centered on the actual user connection to our network. I did not want just anybody who was connected to the Internet to be able to send traffic onto my internal network. Only authenticated users should be allowed to access the network. Also, I did not want the information that passed back and forth between my network and remote users to be transmitted in clear text. Sensitive documents, in addition to usernames and passwords, would be traveling across the Internet, so I wanted to make sure that any data passed would be encrypted. This led me to choose a Virtual Private Network (VPN) connection as the method to get users onto the internal network.

Since Windows 2000 Serer includes a built-in VPN server using existing Windows authentication, and because all of our remote users would be using Windows on their remote machines, this decision was a fairly easy one.

Moving finally to actually connecting to servers on the internal network: this last step caused me a great deal of difficulty. I wanted users to access their files and other services, but in a typical corporate environment such as ours, Windows file sharing operates over certain well-known ports, such as TCP 135 (RPC), TCP/UDP 137-139 (NetBIOS), and UDP 389 (RPC). I felt that it would be a security weakness to allow remote machines that are outside of my control to send traffic over these ports. Many viruses, such as Nimda, have the ability to scan for and infect any Windows file shares that the user has access to. Also, these ports are used for numerous Windows services that I did not want remote machines to have access to.

Even though the remote users would be running both anti-virus and personal firewall software, in order to have defense in depth, I needed to strictly limit what they would have access to. I did not feel that I would be able to limit access to services effectively if I utilized the standard Windows services. Therefore, I needed to find alternative services for users to access their files through, and I also needed to limit incoming VPN traffic to only allow the ports that corresponded to these alternate services.

Alternate Services

An alternative file service was easy to locate: FTP. Conveniently, Windows 2000 Server includes an integrated FTP server. Since the file server was already sharing files with Window's built-in capabilities, I felt that allowing users to also access their files using FTP would create relatively little security risk, especially when comparing it to the alternative of allowing users to connect via typical Windows file sharing. Allowing FTP traffic to our file server would allow users to only connect to the FTP service, no other services would be accessible over those ports. Also, all of our prospective remote users use some flavor of Windows as their operating system, so they could all be given shortcuts to open their respective FTP directories using Internet Explorer. All I needed to do was install Microsoft's FTP services on the file server and allow VPN clients to pass FTP traffic onto the network.

With file access under control, I moved to the final challenge, access to our ERP system. Since our ERP application is far too data-intensive to run over any type of connection short of 100 megabit Ethernet, the only solution was to run the application remotely via some sort of remote desktop software. I had already set up a server with Windows Terminal Services to provide a demo for several people in the company and I was satisfied with the performance that I received, so it seemed like the logical choice. Fortunately, I had already informed our management that getting access to our ERP application for any significant number of users would require the purchase of a more powerful dedicated server and licensing as appropriate.

All I needed to do was to determine how Terminal Services communicated over the network. After some searching of Microsoft TechNet, I found that Windows Terminal Services used a specific port to operate over, port 3389. I only needed to allow this one port into the network to allow users to connect via Terminal Services. I also felt fairly secure about allowing this traffic onto the network since connecting to the Terminal Server required users to authenticate with the server before actually being logged in. I felt that this provided another layer of defense against both human and automated attacks.

That was it! I had my plan, now all I needed to do was implement it.

Setting Up Services and RRAS

The actual execution of my plan began with installing and configuring Window's built-in FTP service. This was relatively straightforward. I configured the FTP server's root directory to run from the directory that housed all of the individual user's directories. This way, I could simply make a shortcut to ftp://ftpserver/userdirectory, and the user would be able to access their files using Internet Explorer by simply double-clicking on the shortcut and authenticating when prompted. Also, all of the files and folders already had the correct permissions configured to only allow each user access to their specific files.

I already had Terminal Services running on a low-end test machine, so no additional configuration was required for this service. Also, our management was aware that a

beefier server and additional licensing would be required to offer our users remote access to our ERP software using Terminal Services.

I next configured Routing and Remote Access on the Proxy Server. At first I encountered problems when following the RRAS configuration wizard. After researching and reading an article in the Microsoft Knowledge Base, which recommends manually configuring the RRAS services and enabling Proxy server's default PPTP Receive filter, I was able to successfully configure the RRAS and make VPN connections across the internet from a test machine.

Once I was able to connect, I then verified that I had good network connectivity. To do this, I performed several tasks that I might perform at work. I first used ping to verify that I had connectivity to the network. I then mapped a drive to one of the file shares on the file server and copied several files to my VPN client. I also configured Outlook to connect to our Exchange server and verified that I could successfully connect. Finally, I connected to the Terminal Server using the Terminal Services Client. Everything seemed to work perfectly.

Setting Up Filtering

Now I needed to filter the traffic to only allow what I needed. Looking over my plan, I was able to list my desired packet filtering behavior using the following "pseudo rules":

Output from inside interface

Allow/Deny	Source IP	Dest. IP	Source Port	Dest Port
Allow	Any	Terminal Server	TCP > 1023	TCP 3389
Allow	Any	FTP Server	TCP > 1023	TCP 21
Allow	Any	FTP Server	TCP > 1023	TCP 20
Allow	Any	FTP Server	TCP > 1023	TCP > 1023
Allow	Any	DNS Server	UDP >1023	UDP 53

Input to inside interface

Allow/Deny	Source IP	Dest. IP	Source Port	Dest Port
Allow	Terminal Server	Any	TCP 3389	TCP > 1023
Allow	FTP Server	Any	TCP 21	TCP > 1023
Allow	FTP Server	Any	TCP 20	TCP > 1023
Allow	FTP Server	Any	TCP >1023	TCP > 1023
Allow	DNS Server	Any	UDP 53	UDP > 1023

The first rule in each table would allow traffic to the Terminal Server over the port used by Terminal Services. The next three rules allow FTP connections. The first FTP rule allows traffic to the FTP server for the data connection. The second FTP rule allows for an active-mode FTP connection. The third FTP rule allows data to pass for a passive-

mode FTP connection. Finally, the last rule allows clients to pass traffic associated with DNS queries.

Technically, I thought that I might not need the second FTP (third overall) rule since an active-mode connection actually requires the server to initiate the data connection over port 20, and Proxy server includes a dynamic filter that automatically allows traffic that is initiated by the Proxy server. However, I decided to include the rule just in case.

Now I was eager to set up my filters and test them out. My energy was quickly sapped, however, when I opened the management snap-in for Microsoft Proxy server. Packet filtering for Proxy server is configured through the Internet Services Manager (ISM), which is also used to manage IIS. Specifically, opening the ISM, double-clicking on the server's name next to any of the Proxy services, selecting the Services tab, then clicking on Security under Shared Services, accesses it. Normally, you select the typical options for a packet filter such as source port, destination port, source address, and destination address. However, when you are setting up a new packet filter, the UI does not allow you to create a filter that has an internal IP address in either the source or destination address field. An error message pops up that reads: "An Invalid Local Host Address was specified for a packet filter".

I needed to filter packets being sent to the internal network by the inside interface of the Proxy server, because the VPN traffic all looked the same to the outside interface since all of the IP traffic is encapsulated and transmitted over TCP port 1723. But Proxy Server would not allow me to set up the filtering that I needed.

I had a terrible feeling that all of my planning and previous work was about to go out the window. In desperation, I combed the Microsoft Knowledge Base, only to find an article informing me that Microsoft Proxy server did not support packet filtering on the inside interface. However, my search also pulled up another interesting article, one that dealt with the use of the Routing and Remote Access Service's packet filtering functionality to filter traffic arriving over a VPN or dial-up connection. My plan was saved!

I quickly dug into the Routing and Remote Access interface, following the steps in another helpful knowledge base article. Routing and Remote Access Service gives you the option of two filtering philosophies, either "deny all except…" or "allow all except…" Since I only wanted to allow a narrow range of traffic into the network, I chose the "deny all except…" option for both the input and the output filters. Below this option, you are then allowed to enter traffic that will be the exception to the basic behavior. I entered the filters that I had previously defined to allow the very specific traffic that needed to get through to my internal network, with two exceptions.

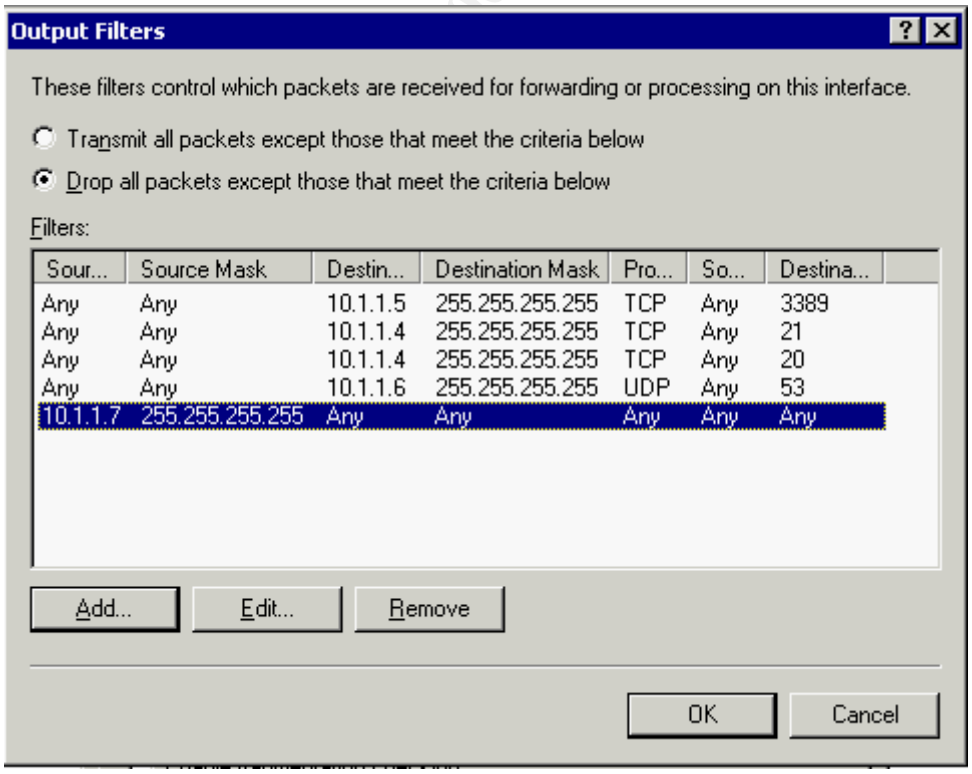
Both exceptions are based on the fact that RRAS packet filtering does not allow you to input a range of ports; you only get a choice of either a single port or all ports. The first exception was with the filters that had a source port defined as >1023 (dynamic port), I chose to set the filters up with "any" in place of the ">1023". This was a slight decrease in the security of the VPN, but I felt that I had no other choice. I also felt that the

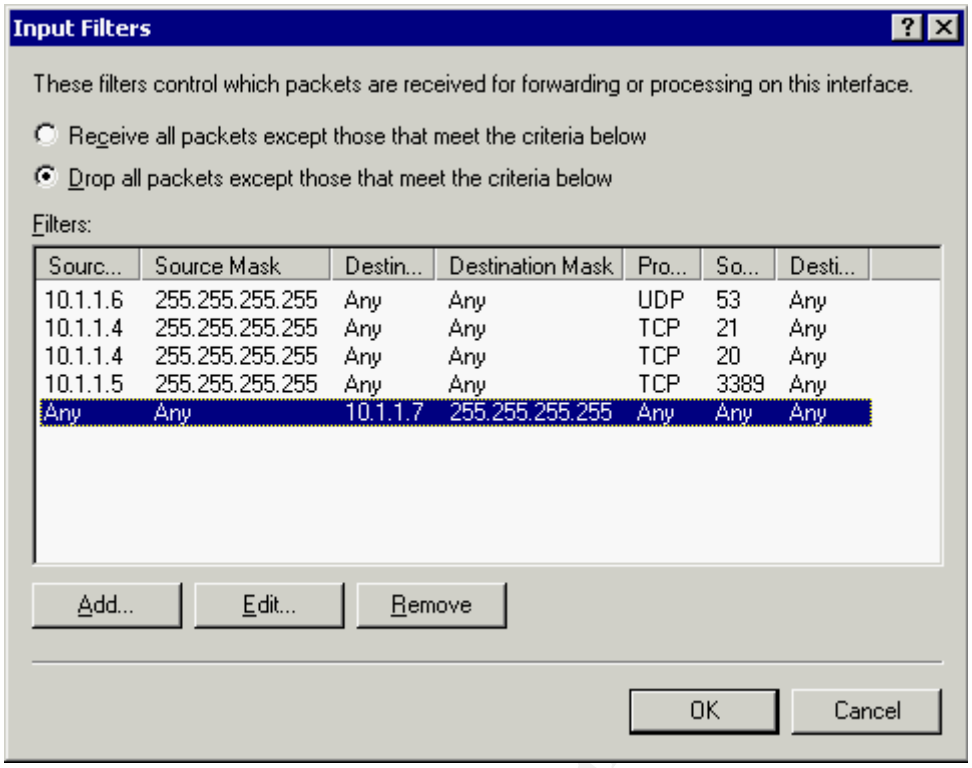
decrease in security was not enough to warrant scrapping the entire project. Clients would still be limited to contacting only the addresses and ports that I allowed; there would just be the additional possibility that client traffic could come from a source port below 1024. Since we were only giving access to Dri-Eaz employees, I felt that opening the door just this small additional amount would not create a serious vulnerability.

The second exception dealt with the third FTP (fourth overall) filter. Since I could not create a filter using ">1023", my only option to allow passive mode FTP would be to create a filter allowing traffic from any port to any port of the FTP server. This was an unacceptable situation, especially so since our file sharing server was also a domain controller. Therefore, I chose not to set up this rule, and to only allow active mode FTP using ports 20 and 21. After making this decision, I felt that this would actually be a more secure setup than my initial plan since clients would not be able to send traffic to any dynamic port of the FTP server. I would simply advise all of our VPN users to make sure that they were running at least Internet Explorer 5, since Internet Explorer 5 and later have the option of using either active of passive mode FTP.

Finally, I added an inbound and outbound rule to allow incoming and outgoing internal network traffic that was generated by the Proxy server or going to the Proxy server. This is because the RRAS filters actually affect all traffic sent to the server, not just traffic being forwarded by the RRAS service. These two rules were required to allow the server to be able to function normally on the network while still strictly controlling forwarded VPN client traffic.

Here is what the finished product looked like:





(Note: Rules have been automatically re-ordered by Routing and Remote Access Service)

Testing

After setting up my filters, I returned to my VPN test client. I connected to the VPN and tried several functions that had previously been successful. I was glad to note that I was unable to map a network drive, or to connect to the Exchange server using Outlook.

However, I was able to perform name resolution using nslookup. I was also able to connect to the file server via FTP using Internet Explorer. Finally, I was able to connect to the terminal server using the Terminal Server Client.

I also included more in-depth testing to make sure that the RRAS filters were working correctly. Using Rafale X on my VPN client and Ethereal on a client machine connected to the monitoring port of our internal switch, I generated a number of packets that had a range of addresses and ports in both the source and destination fields. I was pleased to find that any traffic that should have been caught by the RRAS filters did not get through to the network. The only way I was able to get traffic into the network was to send packets that had both a destination address and a destination port that were set up to be allowed into the network.

After

The rollout of the services to users was a complete success. When a new user needs to have remote access, we have an easy to follow procedure. We first go over the policy that requires them to have the anti-virus and personal firewall software. Once we have verified that they have met this requirement, we flag their user account to allow VPN connections and distribute a document and a floppy disk that guides them through setting up the VPN connection, setting up Internet Explorer for active mode FTP, creating a desktop shortcut to their personal files, and installing the Windows Terminal Service client.

Users are happy that they can have remote access to the functions that they need, and I am happy that user's remote connections to our systems are limited enough that I am able to sleep at night without nightmares about arriving at work to find a rampant Nimda infection on our network.

Even though I was pleased with the progress that I had made, I was not about to rest on my laurels. Due to the success of the remote access and also my SANS training, we have been able to get funding approved for a number of upgrades to our systems. Primary among these is approval for an upgrade to our firewall, which will allow us to replace our Microsoft Proxy server system with a Cisco PIX device. However, since not all of the users at our company are allowed to have Internet access, we will continue to use the Proxy server behind the Cisco PIX firewall to provide user authentication and logging of outgoing Internet traffic. I feel that this will give us a great deal of added flexibility and security for our present and future needs.

Also, we are looking at systems that would be able to verify the installation of anti-virus and personal firewall software on remote user's computers when they connect to the VPN. While we do have a policy for all remote users to run these programs, I would rather be safe than sorry. If we have a system in place that will only allow users to connect to the VPN if the prescribed security software is in place, it will be just one more layer in our defense in depth strategy.

Now that our users have remote access to the critical items that they need, they are clamoring for remote access to their email. My work is never finished…

References

1. Internet Assigned Numbers Authority; "Port Numbers"; 03/04/2003; URL: http://www.iana.org/assignments/port-numbers

2. Microsoft Knowledge Base; "Windows NT, Terminal Server, and Microsoft Exchange Services Use TCP/IP Ports"; 10/14/2002; URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;150543

3. Microsoft Knowledge Base; "Enabling VPN in RRAS Causes Connection Issues to Remote Networks"; 10/10/2002; URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;243374

4. Microsoft Knowledge Base; "Err Msg: Invalid Local Host Address Specified for Packet Filter"; 1/14/2000; URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;176376

5. Microsoft Knowledge Base; "Using PPTP, RRAS, and Proxy Server 2.0"; 12/11/2002; URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;176924

6. Microsoft Knowledge Base; "Enable PPTP Filtering Option No Longer Works"; 12/12/2002; http://support.microsoft.com/default.aspx?scid=kb;EN-US;169890

7. Microsoft Knowledge Base; "HOW TO: Configure Internet Explorer to Use Both the FTP PORT Mode and the FTP PASV Mode in the Windows Server 2003 Family"; 1/20/2003; http://support.microsoft.com/default.aspx?scid=kb;en-us;323446

Last Updated: August 21st, 2017

Upcoming Training

Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401**	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
SANS Tysons Corner Fall 2017	McLean, VA	Oct 14, 2017 - Oct 21, 2017	Live Event
SANS Tokyo Autumn 2017	Tokyo, Japan	Oct 16, 2017 - Oct 28, 2017	Live Event
CCB Private SEC401 Oct 17	Brussels, Belgium	Oct 16, 2017 - Oct 21, 2017	
Community SANS Omaha SEC401	Omaha, NE	Oct 23, 2017 - Oct 28, 2017	Community SANS
Community SANS New Orleans SEC401	New Orleans, LA	Oct 23, 2017 - Oct 28, 2017	Community SANS
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201710,	Oct 30, 2017 - Dec 06, 2017	vLive
San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
Community SANS Colorado Springs SEC401*	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS