Securing the Heart of America: The Small Business

Eric Bauer
GIAC Security Essentials Certification

November 15, 2003
Version 1.4b, Option 2

Abstract

The American economy wouldn't be stable if it weren't for the Small Business. Small businesses not only provide economic value, they tend to be more customer oriented and make it possible to live the American Dream. However, the largest hurdle small businesses face is expenditure. And in a fast growing technology market, it becomes more difficult to do business without the use of computers and, in most cases, without a broadband connection.

I recently had the privilege to improve system and network security for a friend whom owns a retail jewelry store. He traded goods with a "Buddy" of one of his employees (whom happens to work for a local Internet Service Provider) to recommend computers for his business, network them to the retail software system, and connect all systems to the Internet via a DSL line. Now I'm all for the barter system, so long as you receive adequate services for the goods tendered!

Over the next few pages, you'll read about my horrifying discovery of these bartered services and very quick actions that were taken to help get this retail shop on their way to a more secure and, hopefully, more profitable business.

The Discovery

As I mentioned earlier, the biggest hurdle small businesses face is expenditure. The same was true for this small business. While the owner shelled out quite a bit of cash for three computers, he was not as willing to shell out even more for additional services he felt he already paid for. The new systems he bought were running slow, the connection to the retail software system intermittently dropped, and there was no means for a backup of his data. So I agreed to come in and do an assessment of his environment.

My assessment included the following:

- An old IBM PS2 with a 3 ½ inch floppy, CDROM, running Windows 95 and NetBUI for network connectivity, and a DOS based retail software application
- Three new Sony VAIO desktops with CD-RW drives, USB floppies, running Windows 98 and TCP/IP for network connectivity (using static IP addresses assigned by the ISP), and using a DOS shell to run the shared drive of the retail software application
- An 8-port 3com hub connecting all four systems which was connected to an Infinilink DSL modem/router with – you guessed it – NO FIREWALL

Unbelievable! I was staring at four exposed systems on the Internet running probably the most insecure Operating System in the world. One of which, containing sensitive business data! I think I was there all of 30 minutes when I told the owner “we have some issues”. Aside all the problems they were experiencing with their systems, I explained his vulnerability situation and requested $100.00 to get him firewalled – now.

Further Assessment

As it turns out, the Office Manager of the store was in the process of evaluating a new Windows based retail system. The system requirements of the software being evaluated did not permit it to run on the old IBM PS2. Additionally, the software vendor offered to convert the old DOS data to their format upon purchase of the software. That’s one system we don’t have to worry about securing at the network level. The other three, however, needed work. The customer complained that the systems weren’t performing the same despite them being configured identically. In my review of the systems, I verified the hardware was the same, but the software installed on each was not. I noticed various Internet applications, software known for performance degradation, which were downloaded and installed by some of the employees. Knowing that the company was not willing to spend any additional money, especially for server hardware or software, I reluctantly recommended that the systems be upgraded to Windows 2000 Professional. Even that would be an additional expense to them, but at least we could lock down the users from installing any software. Not all was lost, however, as I quickly learned that Windows 2000 was purchased in addition to the Sony VAIO systems.

Identifying Risk

After spending some time doing the assessment, I collected the $100.00 to go purchase a broadband firewall. When returning home, I decided to run nmap on the systems that were directly exposed to the Internet by the ISP assigned IP addresses. At the very least, I could show the owner how exposed his business really was. Below is a summary of the scanned results:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
 Interesting ports on  (xxx.xxx.xxx.xxx):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
17/tcp     open        qotd
19/tcp     open        chargen
135/tcp    open        loc-srv
```

```
139/tcp    open        netbios-ssn

7/udp      open        echo
9/udp      open        discard
13/udp     open        daytime
17/udp     open        qotd
19/udp     open        chargen
135/udp    open        loc-srv
137/udp    open        netbios-ns
138/udp    open        netbios-dgm

Remote operating system guess: Windows Millennium Edition (Me), Win
2000, or WinXP
```

From the results of the scan, we can see the customer has quite the vulnerability with Microsoft's NetBIOS networking protocol. There's much to understand about NetBIOS and how it works. Eric Cole's book[1], Hackers Beware, has a great overview on this protocol and its exploitability. The NetBIOS protocol in the Windows world is used within a workgroup and provides the ability to share directories among Windows computers within that workgroup. There are several exploit variants involving NetBIOS, Microsoft shares being the most common. Mr. Cole explains this exploit in detail[2] in that once a hacker obtains the name or IP address of a system, he/she can establish an anonymous connection through a null session using an automated program or the "net use" command. As we can clearly see in the above scan, any hacker can easily obtain any one of the customers' system IP addresses and use the null session exploit to gain access.

The Project

Before I returned the next day, I outlined the steps that would be taken to secure the network and systems. Below is that outline:

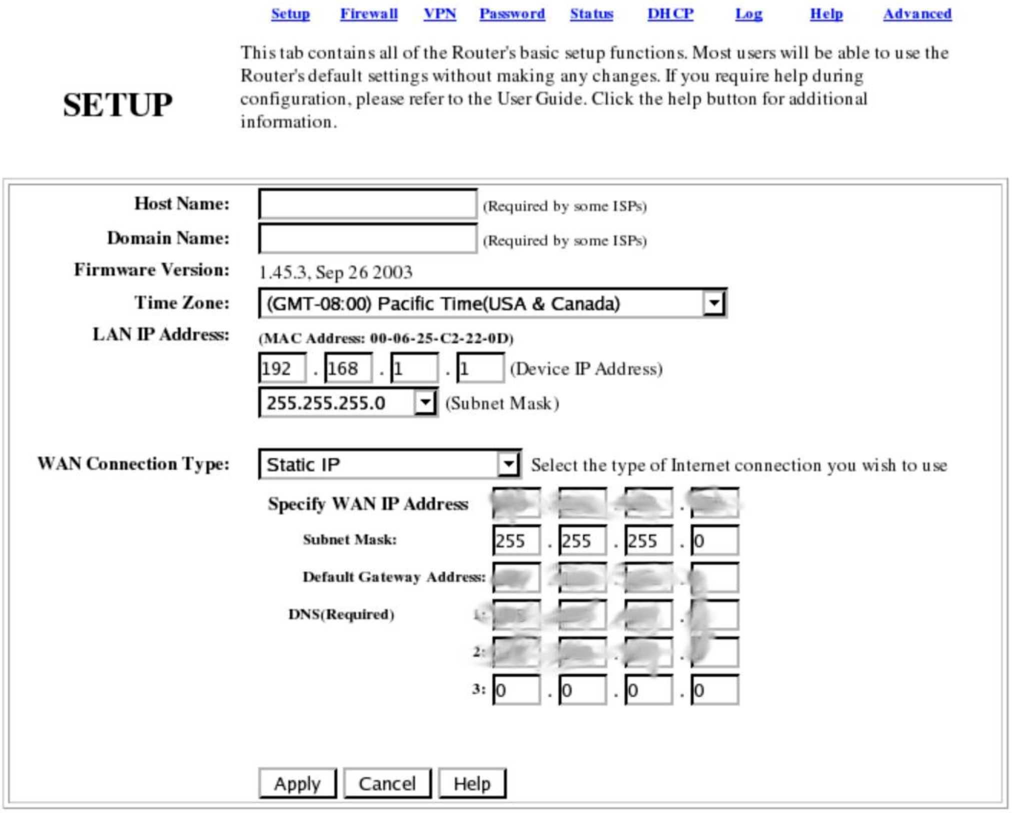
- Install and configure the broadband firewall
  - Select one of the static IP addresses for the WAN port
  - Configure DNS and Gateway IP addresses of the ISP
  - Configure the LAN IP address with a private IP address
  - Configure firewall parameters and logging
- Upgrade each Sony VAIO system
  - Upgrade to Windows 2000 Professional
  - Install appropriate VAIO drivers for W2K
  - Install Service Pack 4
  - Convert the file system from FAT32 to NTFS
- Lock down each system
  - Implement a WORKGROUP within the network
  - Network each VAIO with its own private IP address
  - Create user profiles for each employee
  - Assign each profile to the appropriate group

    - Apply any needed file/directory permissions
- Install the new retail software
    - Install software on partitioned disk
    - Share the software directory
    - Map network drive for each user on each system

Clothe the naked

The first and foremost task was to install the firewall. While I had the static IP information for the systems, I didn't have the gateway and nameserver addresses. I placed a call to the ISP and just so happened to talk to the "Buddy". He wasn't willing to provide the information at first. I guess he was worried as to why someone else was coming in behind him. After explaining I was hired by the owner to fix the insecure position of the company that should have been implemented correctly to begin with, he offered the information I requested quite offensively. OK, let's get protected!

I chose the Linksys Broadband Router with a 4-port switch. While I'm partial to Linksys products (their products have been reliable for my home network), all broadband routers work basically the same. I particularly wanted a 4-port switch on the router. The switch would offer them better performance and probably eliminate the intermittent network drops the customer was experiencing, thus allowing me to decommission the 3com hub. By default the router uses IP address 192.168.1.1 for the http configuration. Using my Linux laptop, I connected to a port on the router's switch, assigned my laptop to IP address 192.168.1.2, used Mozilla to connect to the configuration page, and entered the default admin ID and password. I was then presented with the following page:



SETUP

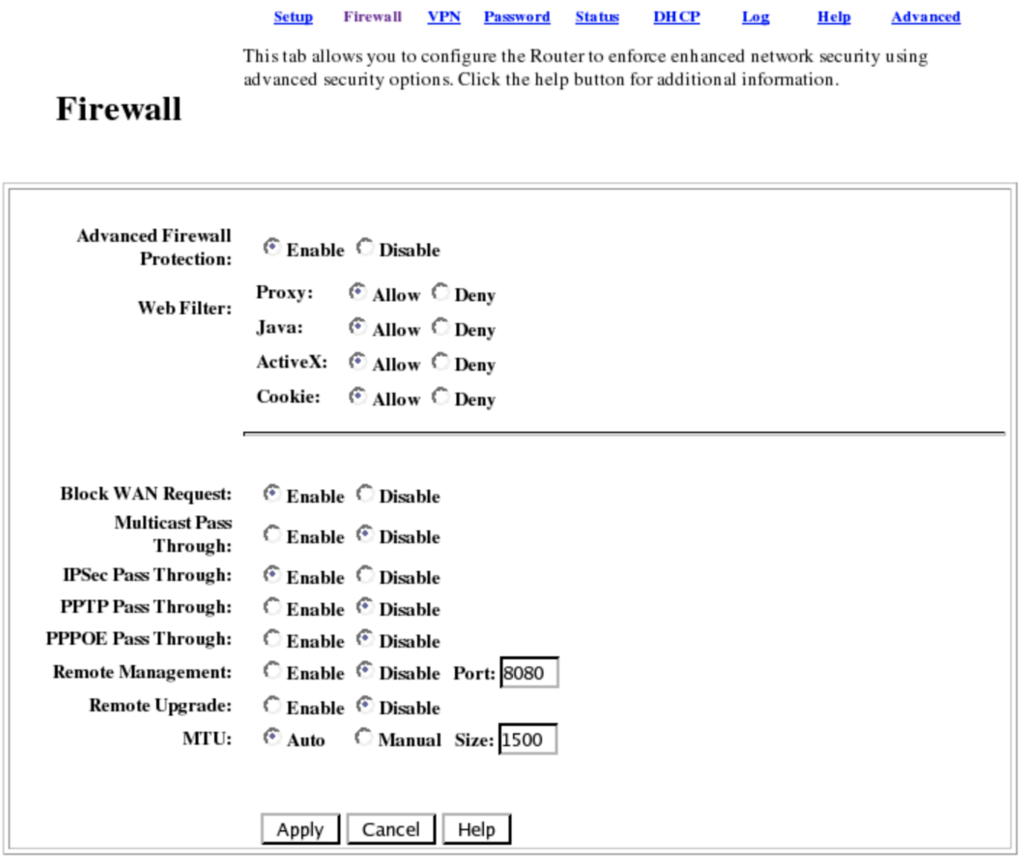
This tab contains all of the Router's basic setup functions. Most users will be able to use the Router's default settings without making any changes. If you require help during configuration, please refer to the User Guide. Click the help button for additional information.

Host Name: (Required by some ISPs)
Domain Name: (Required by some ISPs)
Firmware Version: 1.45.3, Sep 26 2003
Time Zone: (GMT-08:00) Pacific Time(USA & Canada)
LAN IP Address: **(MAC Address: 00-06-25-C2-22-0D)**
192 . 168 . 1 . 1 (Device IP Address)
255.255.255.0 (Subnet Mask)

WAN Connection Type: Static IP Select the type of Internet connection you wish to use
Specify WAN IP Address [redacted]
Subnet Mask: 255 . 255 . 255 . 0
Default Gateway Address: [redacted]
DNS(Required) 1: [redacted]
2: [redacted]
3: 0 . 0 . 0 . 0

The Host Name and Domain Name were left blank because the customer was not hosting, nor registering, a domain. I wanted to keep the LAN IP addressing simple, so I stayed with the 192.168.1.1 network. In my assessment of the environment, you will recall that the systems were lingering out on the Internet by IP addresses that were assigned by the ISP. I picked one of these IP addresses and specified it as the WAN address of the router. So, in the WAN Connection Type, I selected Static IP, entered the IP address I picked for the WAN IP address and the Subnet Mask as specified. I also entered the Gateway and DNS addresses that I had obtained from our “Buddy”.

Now that the network side has been configured, let’s move on to the firewall:



This tab allows you to configure the Router to enforce enhanced network security using advanced security options. Click the help button for additional information.

Firewall

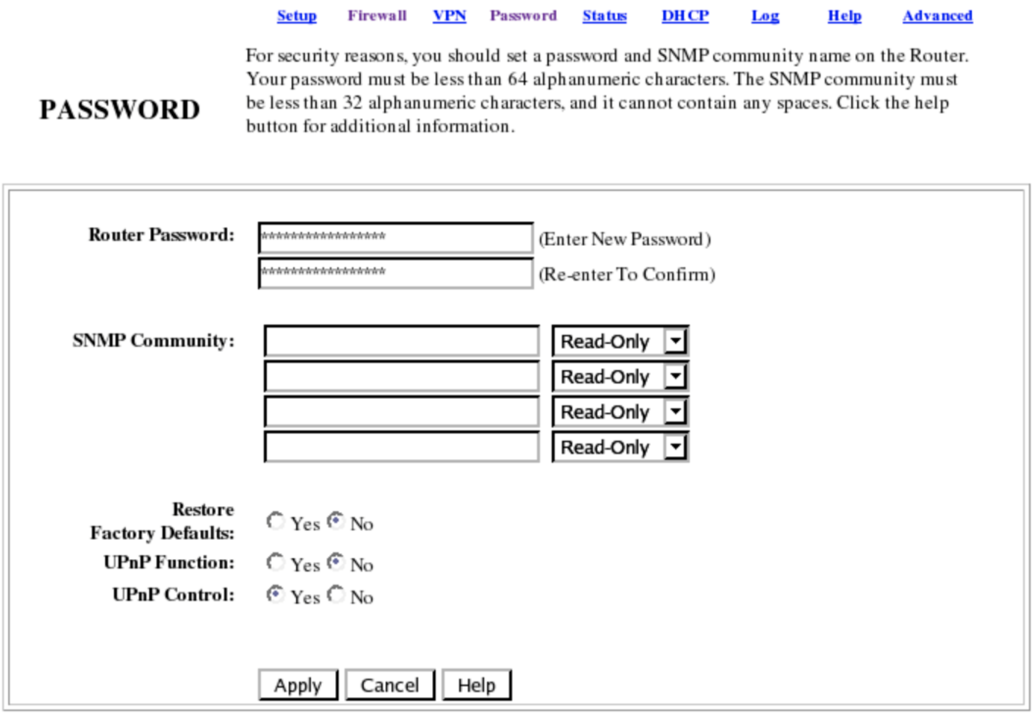
Advanced Firewall Protection:	● Enable ○ Disable	
Web Filter:	Proxy:	● Allow ○ Deny
	Java:	● Allow ○ Deny
	ActiveX:	● Allow ○ Deny
	Cookie:	● Allow ○ Deny
Block WAN Request:	● Enable ○ Disable	
Multicast Pass Through:	○ Enable ● Disable	
IPSec Pass Through:	● Enable ○ Disable	
PPTP Pass Through:	○ Enable ● Disable	
PPPOE Pass Through:	○ Enable ● Disable	
Remote Management:	○ Enable ● Disable	Port: 8080
Remote Upgrade:	○ Enable ● Disable	
MTU:	● Auto ○ Manual	Size: 1500

While your typical manufactured broadband router isn't quite as configurable as a Cisco PIX firewall or a Linux iptables firewall, it should do the trick for most small businesses. The first thing we want to do is enable the Advanced Firewall Protection. As defined by the Linksys BEFSX41 User Guide, which can be obtained from the Linksys web site[3], this function will prevent Denial of Service (DoS) attacks by using Stateful Packet Inspection (SPI). We all know about DoS and the effects it can have on an unprotected network. The way we can detect such attacks is by applying SPI. The most common form of DoS attacks is derived from many incomplete "handshakes" of network packets. That is, as a packet traverses a network to open a connection to a particular service on another network, it sends a SYN message that it wants to connect to that network. The responding network acknowledges that packet and also sends a SYN message, SYN ACK. If, however, the originating network doesn't acknowledge the SYN message from the responding network, that connection is left open, consuming valuable network resources. SPI helps alleviate this by examining the statefulness of packets and dropping them if the "handshake" isn't complete.

I left the defaults for the Web Filter section since the users will be accessing web sites of some of their suppliers whom use Java and ActiveX applications.

Although I left Proxy enabled, there was really no concern about the types of web sites being accessed by the employees. I also left the defaults for the remaining firewall settings, paying close attention that the Remote Management function was disabled.

On the password screen, I made sure to choose a strong password and document it for the Office Manager of the store. All other settings were left at default:

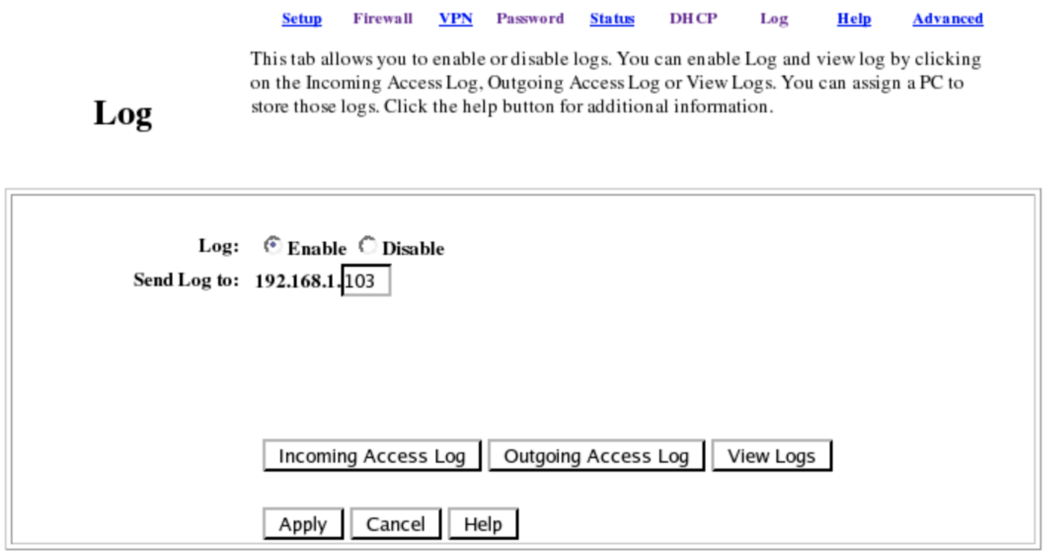


PASSWORD

For security reasons, you should set a password and SNMP community name on the Router. Your password must be less than 64 alphanumeric characters. The SNMP community must be less than 32 alphanumeric characters, and it cannot contain any spaces. Click the help button for additional information.

Router Password:	`******************`	(Enter New Password)
	`******************`	(Re-enter To Confirm)
SNMP Community:		Read-Only ▾
		Read-Only ▾
		Read-Only ▾
		Read-Only ▾
Restore Factory Defaults:	○ Yes ● No	
UPnP Function:	○ Yes ● No	
UPnP Control:	● Yes ○ No	

And since we were only talking about three systems, I assigned each a static private IP address instead of using DHCP. This was also useful when configuring the router logging to send the logs to a specified host:



Log

This tab allows you to enable or disable logs. You can enable Log and view log by clicking on the Incoming Access Log, Outgoing Access Log or View Logs. You can assign a PC to store those logs. Click the help button for additional information.

Log: ● Enable ○ Disable
Send Log to: 192.168.1.103

The logging function on these types of routers is quite limited. However, it does provide information on how much activity is happening on your network. I chose a system with adequate disk space (although these logs aren't disk intensive). Because the router applies a deny all policy to the incoming WAN connection, there was no need for Advanced Firewall settings. The customer wasn't hosting any services and didn't need any ports opened.

Now that the router and firewall was configured, I needed some downtime. With the go ahead to take the network down, I unplugged the DSL modem and disconnected the Ethernet cables from the 3com hub. I then plugged the Ethernet cable from the DSL modem to the WAN port behind the Linksys router. I powered up the DSL modem, waited for the link lights, and then powered up the Linksys router. Groovy! All was green and no red. Using my Linux laptop configured with the gateway IP address of the router and the ISPs nameserver addresses, I connected to a port on the switch and pinged the router's private IP address, as well as, the ISPs gateway and DNS addresses. Once connectivity to the outside world was verified, I reconnected the other Ethernet cables to the switch and reassigned the Windows 98 machines with private IP addresses. After several reboots, they once again were on the Net and protected. To verify this, I rescanned the router's WAN address with nmap from home:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
 Note: Host seems down. If it is really up, but blocking our ping
probes, try -P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 30 seconds

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
 Interesting ports on (xxx.xxx.xxx.xxx):
(The 1600 ports scanned but not shown below are in state: filtered)
```

```
Port        State        Service
113/tcp     closed       auth

Nmap run completed -- 1 IP address (1 host up) scanned in 764 seconds
```

The router does a fine job in concealing any information about the customer's network.

Windows what?

While Windows 98 at some point in time was functional, it was hardly a solution for any type of security…at all! So we move on to the next task by upgrading each system to Windows 2000 Professional. The first step was to be sure drivers existed for W2K and the Sony VAIO[4] systems that were purchased. This was actually quite easy to do since Sony VAIO computers are proprietary. I downloaded one set of drivers and burned them to a CD. Next, I went over each system with the Office Manager to be sure we backed up all critical user and business data to CD.

Once the backups were complete, I moved on to upgrading each machine to the new Operating System. I felt compelled to check Sony's support site[5] about upgrading this particular machine to W2K, and I'm glad I did. Some Sony VAIO computers were not recommended for a W2K upgrade. Fortunately, the customer's VAIO models were not a part of this list. Rather then performing a typical upgrade, it was recommended to perform a new install, thus creating a dual-boot system. This would allow us to decommission Windows 98 and its applications later on. I popped in the Windows 2000 CD and selected the "Install a new copy of Windows 2000 (Clean Install)". I was then presented with the typical Windows License Agreement and installation screens. Before proceeding with the install, I validated that the installation directory was \WINNT and, most importantly, selected the option to choose installation partition during setup. This was key because I didn't want to convert to an NTFS file system just yet. After Windows installed some files, it performed a reboot and began the setup. My first option was selecting the partition to install Windows in and I selected the root drive (C:). The next screen, however, presented several choices related to the file system and I was sure to choose leaving the current file system intact. I breezed through the rest of the setup (I didn't enter any networking values either), rebooted the system and was presented with a choice of which operating system to use. From here I booted into Windows 2000, installed the Sony VAIO drivers I had burned to CD, and also installed Service Pack 4 for the latest fixes and security patches.

Now it was time to clean up Windows 98. I rebooted the system to Windows 98 and reviewed the installed programs with the Office Manager. We had determined that all applications needed for business productivity existed on original CDs purchased by the company and any other applications were installed by the user (which we were going to eliminate anyway). I proceeded in removing all applications through the "Add/Remove Software" feature. Once that was completed, it was time to remove Windows 98 from the system. The Windows 98 Uninstall feature was not available to me because, well, I was in Windows 98. Simple enough. I booted the system back into Windows 2000, loaded Windows Explorer, and removed the C:\WINDOWS directory. While I was there, I cleaned up any leftover application directories I deleted from within Windows 98.

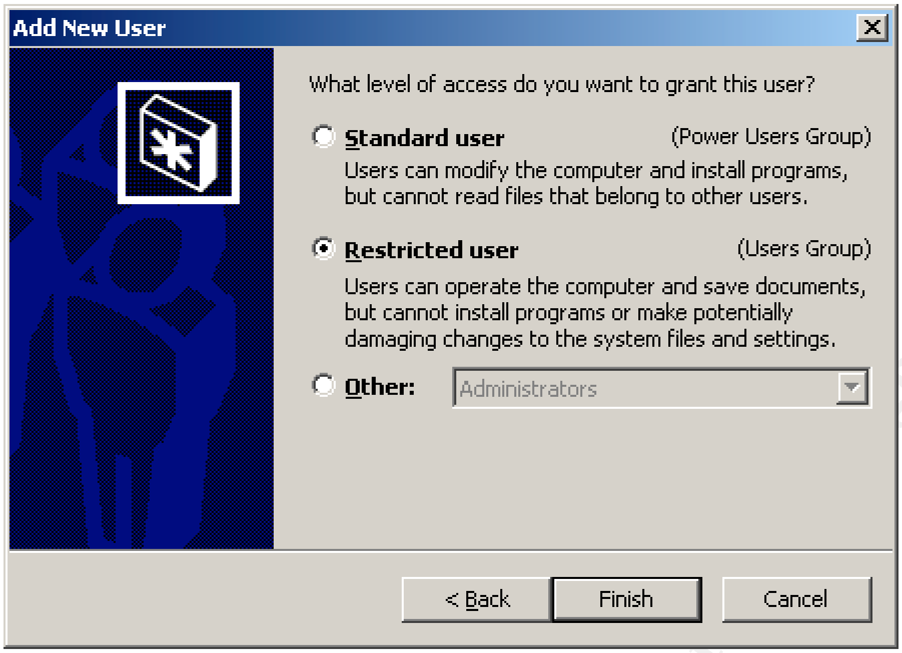
Needless to say, I was a little leery about the next step. I'd never actually done or had seen a FAT32 to NTFS file conversion. I wasn't overly worried about it because we did have the backups on CD. I just wanted to preserve as much as I could. Microsoft's support site[6] documents how to perform this conversion using the convert.exe program. And it truly was painless. After specifying the drive letter to convert to NTFS, the program did not have exclusive access to the drive and asked to schedule the conversion during the next reboot. I obliged, rebooted the system and in a matter of minutes it was converted. To verify the conversion, I went into the Administrative Tools, Computer Management, and Disk Management to validate the partition read NTFS.

Lock down: ACE is the place

Once a system was upgraded, it was time to implement the new network settings, create restricted users, and remove or assign permissions to appropriate directories or shares.

When setting up new network settings, I built a workgroup based on the name of the company and assigned hostnames to each of the three computers. Since one of the computers was to act as a "quasi" server by hosting the new retail software, I named it just that. The other two were named WORKSTATION1 and WORKSTATION2. Each system was assigned its own private IP address and its gateway to be the Linksys router (192.168.1.1).

One of the important steps in this project was to convert the file system to NTFS for more security. Not only was the Office Manager and Owner wanting to stop employees from installing software, they also did not want them to have the ability to clear their Internet cookies and history directories. They weren't so much concerned about the type of content being viewed, but rather, what surfing was going on that was not job related. To accomplish this, I created each employee as a "Restricted User":



What level of access do you want to grant this user?

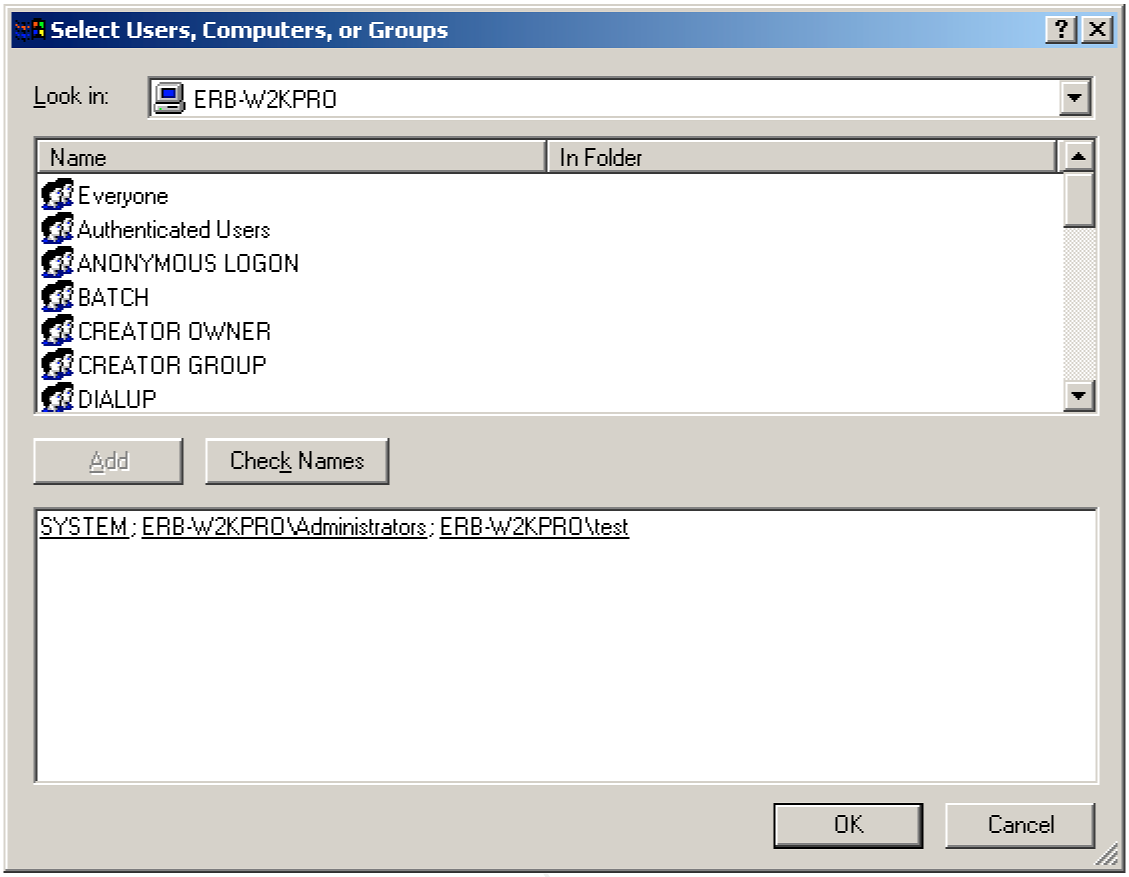
○ **Standard user** (Power Users Group)
Users can modify the computer and install programs, but cannot read files that belong to other users.

● **Restricted user** (Users Group)
Users can operate the computer and save documents, but cannot install programs or make potentially damaging changes to the system files and settings.

○ **Other:** Administrators

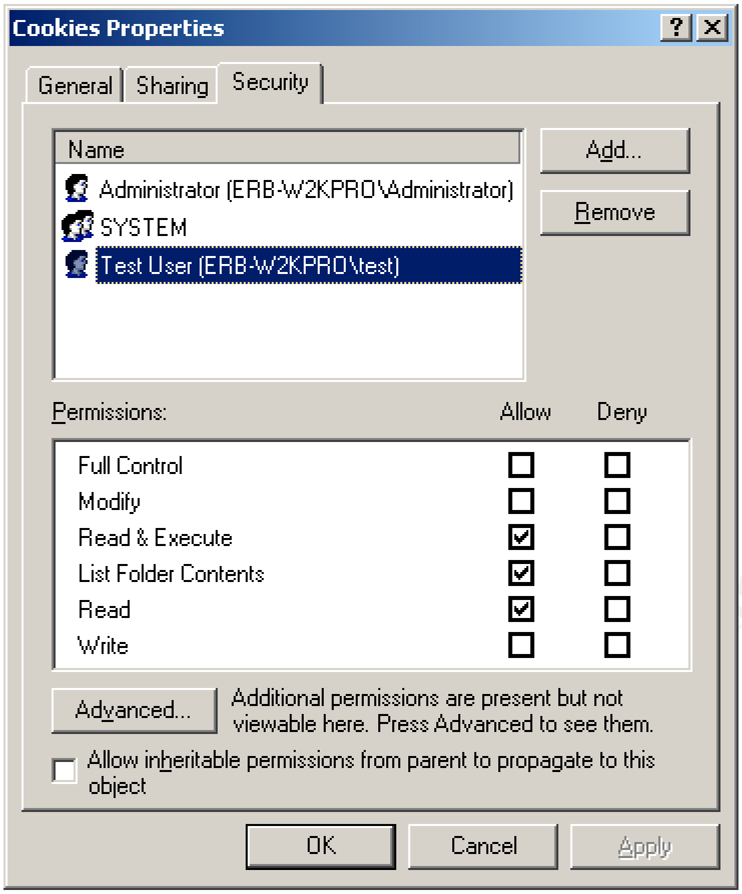
This would keep the employees from installing any programs or making any system changes. The Office Manager and Owner were made power users to have more control over the system, but were also given the Administrator password in order to have full Administrator rights when needed.

Locking down the “Cookies”, “History”, and “Temporary Internet Files” directories of each restricted user took a bit more work. Individual Access Control Entries (ACE) for each directory needed to be assigned. Additionally, “Allow inheritable permissions from parent to propagate to this object” had to be deselected. Removing this option makes permissions assigned to that directory explicit and no inherited permissions from the parent directory could be applied. Once this option was removed, the security for the object had to be rebuilt including adding users having the rights to the object. So here we go. The first step is to add “Administrators”, “SYSTEM”, and the restricted user to the list of users:



SYSTEM ; ERB-W2KPRO\Administrators ; ERB-W2KPRO\test

After I clicked OK, the users were added to the list. I left the default permissions for the "Administrators" and "SYSTEM". The restricted user, in this case "test", only had "Read & Execute", "List Folder Contents", and "Read", which is the default settings for restricted users:



Cookies Properties

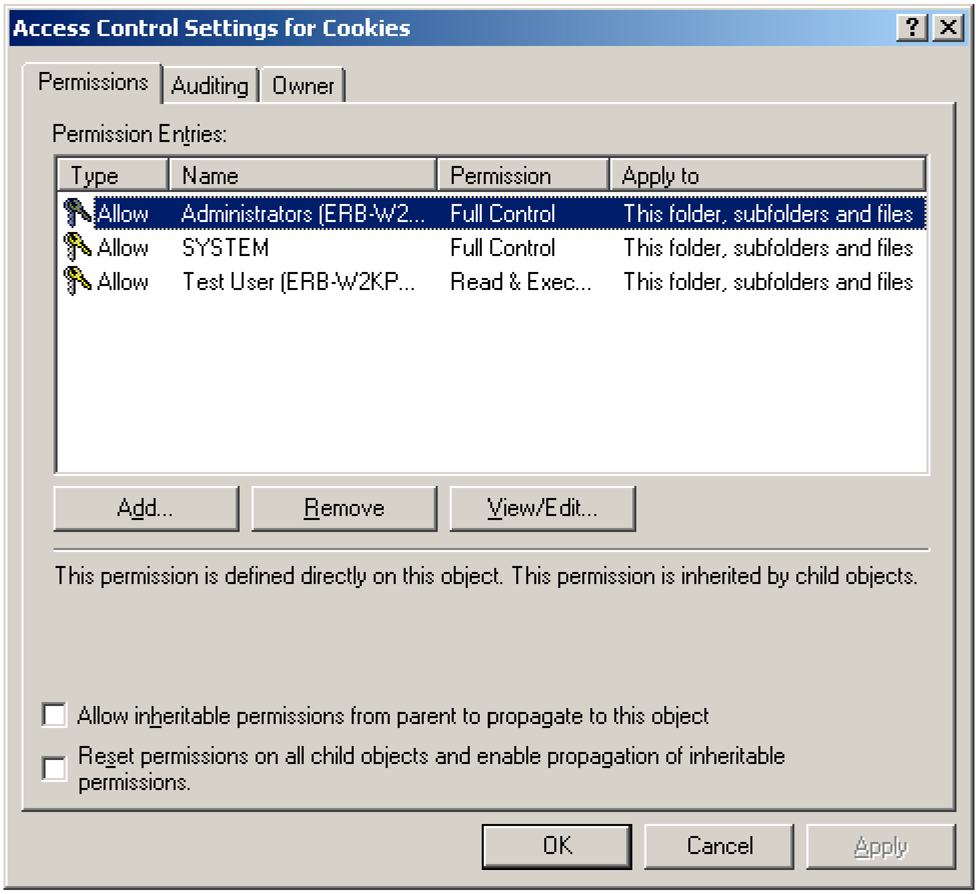
Name
- Administrator (ERB-W2KPRO\Administrator)
- SYSTEM
- Test User (ERB-W2KPRO\test)

Permissions:	Allow	Deny
Full Control	□	□
Modify	□	□
Read & Execute	☑	□
List Folder Contents	☑	□
Read	☑	□
Write	□	□

Additional permissions are present but not viewable here. Press Advanced to see them.

□ Allow inheritable permissions from parent to propagate to this object

Next, we click on the Advanced button to view the permission entries, or as I like to call it, rule set:



Permission Entries:

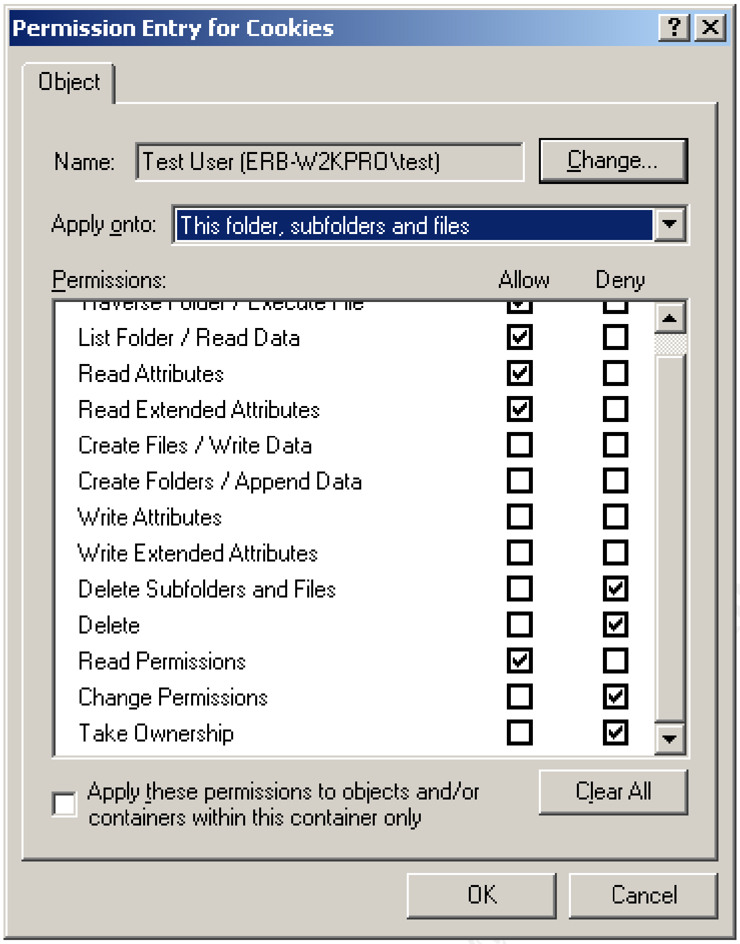
Type	Name	Permission	Apply to
Allow	Administrators (ERB-W2...	Full Control	This folder, subfolders and files
Allow	SYSTEM	Full Control	This folder, subfolders and files
Allow	Test User (ERB-W2KP...	Read & Exec...	This folder, subfolders and files

This permission is defined directly on this object. This permission is inherited by child objects.

□ Allow inheritable permissions from parent to propagate to this object

□ Reset permissions on all child objects and enable propagation of inheritable permissions.

This is important because the rule set determines which permissions are assigned. The rule set rule of thumb: the first rule match wins. And in the case of Windows, the deny rule always supersedes the allow rule, even if there is a conflict in authority (in a case where a user may belong to two groups). So, to be sure user “test” isn’t allowed to delete files from this directory, we need to highlight user “test” in the permission entries and click View/Edit:



Permission Entry for Cookies

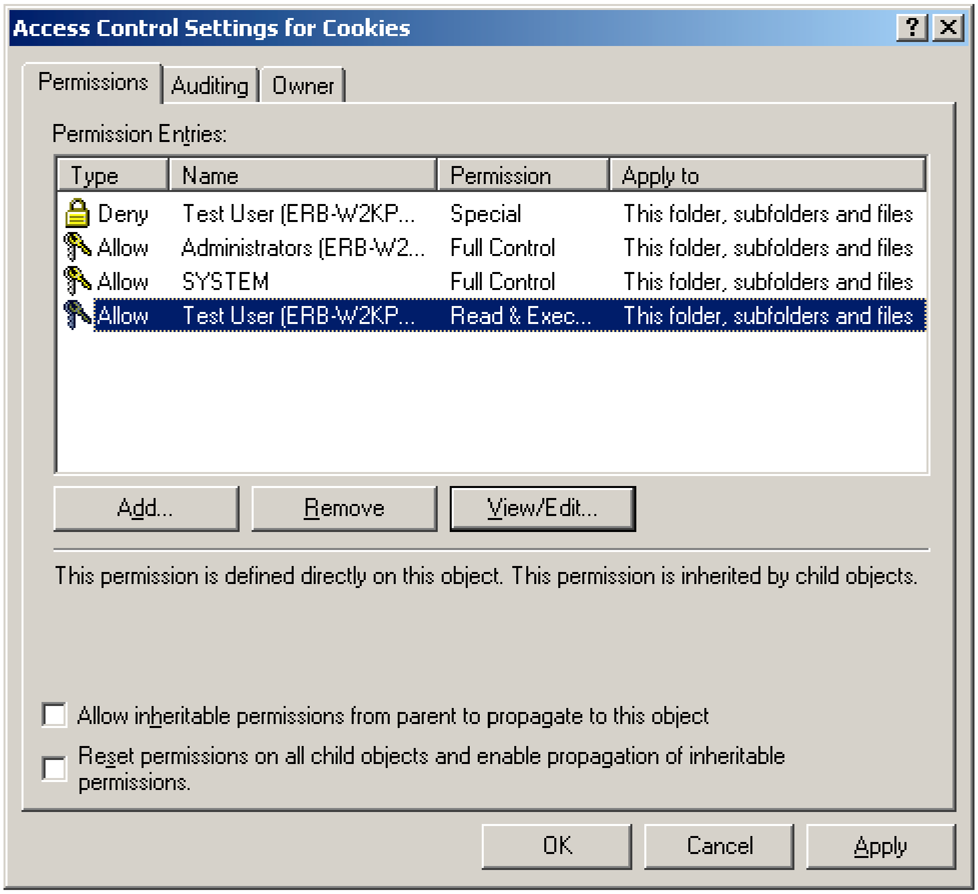
Name: Test User (ERB-W2KPRO\test)

Apply onto: This folder, subfolders and files

Permissions:	Allow	Deny
Traverse Folder / Execute File	☑	□
List Folder / Read Data	☑	□
Read Attributes	☑	□
Read Extended Attributes	☑	□
Create Files / Write Data	□	□
Create Folders / Append Data	□	□
Write Attributes	□	□
Write Extended Attributes	□	□
Delete Subfolders and Files	□	☑
Delete	□	☑
Read Permissions	☑	□
Change Permissions	□	☑
Take Ownership	□	☑

□ Apply these permissions to objects and/or containers within this container only

I selected the deny boxes for “Delete Subfolders and Files”, “Delete”, “Change Permissions”, “Take Ownership” and clicked OK.   A deny rule was added to the rule set (deny rules are automatically placed at the top of the rule set because they always supersede allow rules):



Permission Entries:

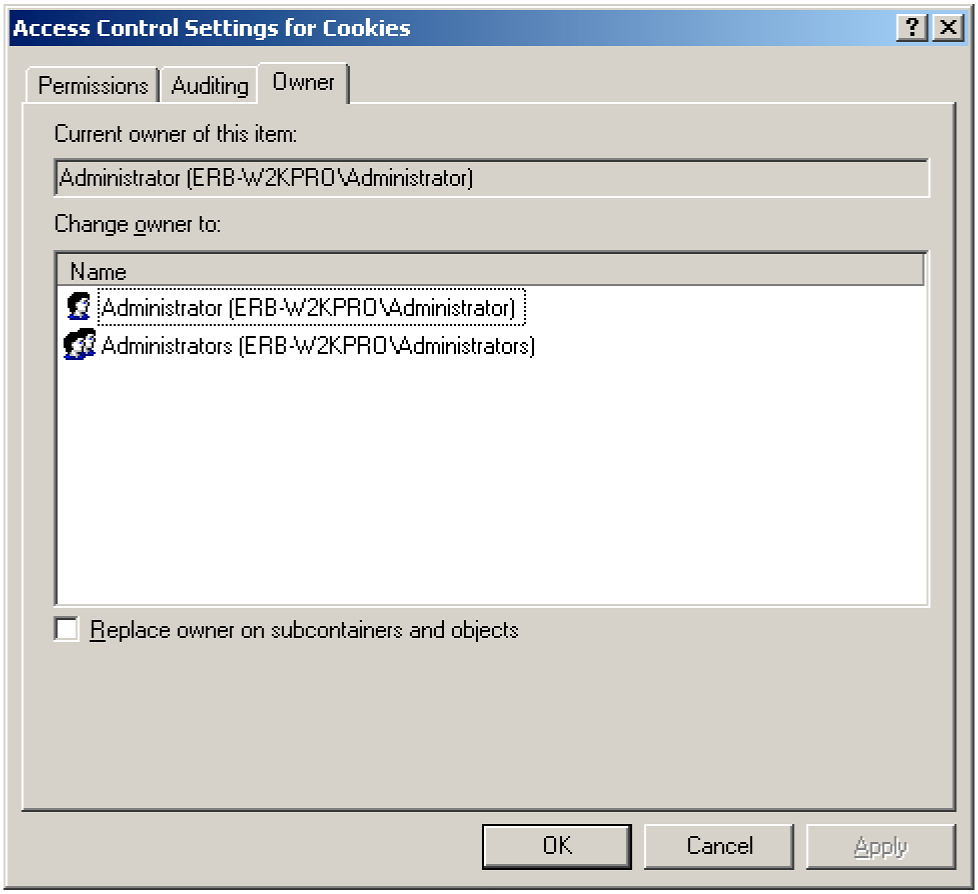
Type	Name	Permission	Apply to
Deny	Test User (ERB-W2KP...	Special	This folder, subfolders and files
Allow	Administrators (ERB-W2...	Full Control	This folder, subfolders and files
Allow	SYSTEM	Full Control	This folder, subfolders and files
Allow	Test User (ERB-W2KP...	Read & Exec...	This folder, subfolders and files

This permission is defined directly on this object. This permission is inherited by child objects.

☐ Allow inheritable permissions from parent to propagate to this object

☐ Reset permissions on all child objects and enable propagation of inheritable permissions.

Now if the user attempts to delete any files in this directory, the first rule to hit is the deny rule. You may also have noticed that the “Change Permissions” and “Take Ownership” options were denied on the restricted user’s directory. If these options were not checked, the user would still be denied from deleting any files, but would not be denied from going back into the permissions rule set and giving himself delete permissions. Therefore, it is also important to change the ownership of the directory from the restricted user to the Administrator:



Current owner of this item:

Administrator (ERB-W2KPRO\Administrator)

Change owner to:

Name
Administrator (ERB-W2KPRO\Administrator)
Administrators (ERB-W2KPRO\Administrators)

□ Replace owner on subcontainers and objects

“Quasi-serving”

The final stages in this phase of the project, was to install the new retail software and share its drive for user access. I labeled this section of the paper “quasi-serving” for two reasons. First, we don’t have a true client-server infrastructure as far as hardware and Operating Systems go. We’re simply building a workgroup among W2K workstations and sharing drives. Although this is not the most secure way to go, we are working with a very limited budget. Secondly, the application itself is not client-server technology.

Without going into boring details here, the application was installed and the default data files were overwritten with the newly converted data from the old retail system. The software’s directory was then shared to provide employee access. In order to get the software to act like a client-server application, the software had to be installed on the other two computers and the desktop shortcut was modified to run the executable program from the shared application directory on the “quasi-server”. This required that each user map a drive to the server and allow it to reconnect at logon.

Next steps

Certainly this new setup is far more secure than the original. The one thing we had working against us, as I suspect works against most small businesses, is a lacking budget. The biggest drawback in this deployment is the fact that security has to be maintained on each system. In fact, each username and password must be identical on each system since we were sharing the retail application's directory on the server. And against my recommendation, the Office Manager asked me to set each user's password to never expire (although I'm forcing her to change the Administrator password on a regular basis). Yikes! I guess it was too difficult for the user's to change their passwords three times on any given day. Perhaps a Linux Primary Domain Controller is on the horizon!

In any case, one next step is to execute a security baseline on the systems and possibly build some security templates from the baseline. The baseline should show us a better security recommendation and possibly give me some clout to get password expiration, as well as other permission assignments, implemented. This baseline study will actually be essential since we did perform a FAT32 to NTFS file system conversion and permissions were not assigned during that conversion. We can certainly deploy some Windows Resource Kit scripts to help aid us in assigning permissions once a security template is built. The customer will also be purchasing anti-virus software in the near future. This will be deployed across all systems just as an added precaution. Must keep practicing that Defense in Depth!

There is a market for securing small businesses. Most don't understand the liabilities they face with a broadband connection. Technology makes it very easy to live and do business day-to-day. It's our job to make sure we do it securely.

References

[1] Cole, Eric. Hackers Beware. Indianapolis, IN: New Riders Publishing, August 2001. 403-404.
[2] Cole, Eric. Hackers Beware. Indianapolis, IN: New Riders Publishing, August 2001. 431-438.
[3] Linksys Download Support Site, URL: http://linksys.com/download/default.asp
[4] Sony Electronics Support Site, URL: http://www.ita.sel.sony.com/support/pc/pcvlx700lx800/
[5] Sony Electronics Support Site, “Windows 2000 Clean Install”, URL: http://www.ita.sel.sony.com/support/pc/windows2000/w2kcleaninstall.html
[6] Microsoft Support Site, Knowledge Base, URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;214579