Global Information Assurance Certification Paper

PROTECTING PUBLIC APPLICATION SERVERS THROUGH DEFENSE IN DEPTH

AL BRACCO

GSEC SECURITY ESSENTIALS CERTIFICATION

PRACTICAL VERSION 1.4B, OPTION 2

JUNE 2003

ABSTRACT

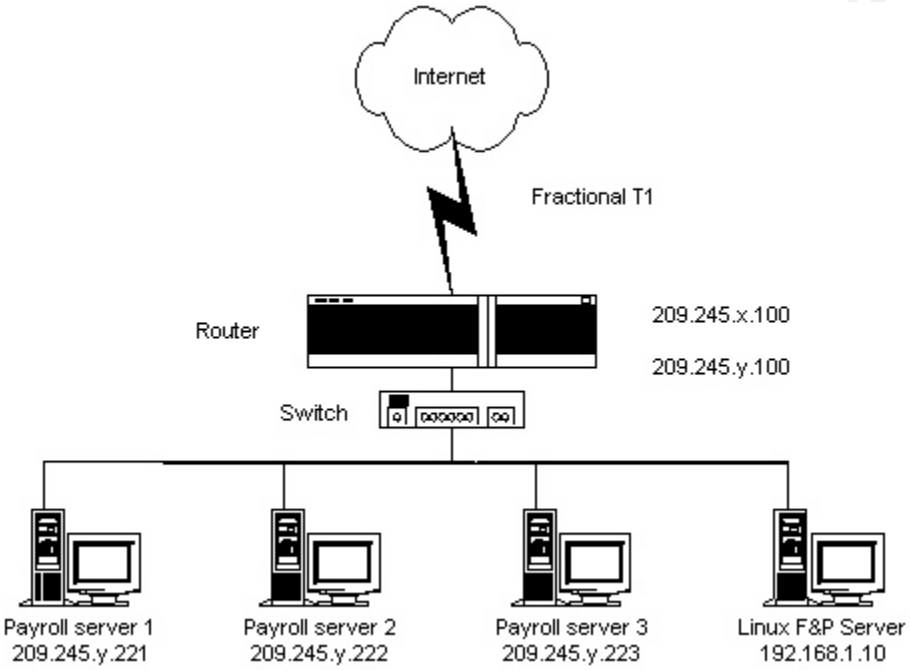
My company was hired by a small payroll company to perform an on-site installation of a new T1 Internet connection. We first did an analysis of their current network structure and found some very serious problems. This location hosted 3 public SCO OpenServer5 servers that were being accessed by their 200+ customers to modify payroll records and run their payroll. Access was provided through telnet and there was no firewall installed. Like most small companies, there was no real IT staff and little consideration had been given to security. Their current network had been setup by a "friend" who got it all working, but didn't really consider the critical need for this company to keep it's data private and secure.

This paper will first take you through a security vulnerability assessment. I will then discuss the security-specific recommendations that were made, how they were balanced versus customer needs & budget, and the final solutions that were actually implemented. All specific names, IP addresses, etc. have been changed for privacy reasons. We will call this company "Best Payroll Co."

BEFORE

The payroll company's existing network consisted of a router providing NAT to 3 servers with public IP addresses. They also have an internal Linux server, which provides file, & print services to their internal LAN.

Here is what their network looked like:



VULNERABILITY ASSESSMENT

Before any work was done, a security audit / vulnerability analysis was undertaken. The following is a brief discussion what checks were performed and what was found:

1) Visual check of physical security:

   Servers are housed in an open rack in an unlocked room where main systems printers are also located. Any employee has access to that room. One server was found to have been left logged on as root. There are no physical security measures in place.

2) Analysis of their current network usage, policies and procedures:

The most serious issue was the complete absence of a firewall. UNIX servers holding their customer's sensitive and confidential payroll data were live on the internet to anyone who could scan their IP address, see telnet running and steal or guess a password. It's very plausible to imagine an employee at one of their customers trying to break in to see what their boss' salary is or change information to give themselves a raise.

The existing Cisco router was configured with a serial interface for the Internet connection and an Ethernet interface for the company network. The public servers were connected to the same switch as the rest of the company LAN. There were no ingress or egress filters configured on the router. Lacking any type of filtering or DMZ setup, the public servers, if compromised, would provide an attacker a convenient gateway to the internal network.

Another serious issue was the use of telnet to provide access to the public servers. Because of the unencrypted nature of telnet, there is a real risk of passwords being stolen by using a sniffer program, such as Ethereal.

Each server had an external modem connected to it, left over from the days when users dialed into the servers directly. And yes, there were active phone lines connected to each. A war dialer attack would love this setup.

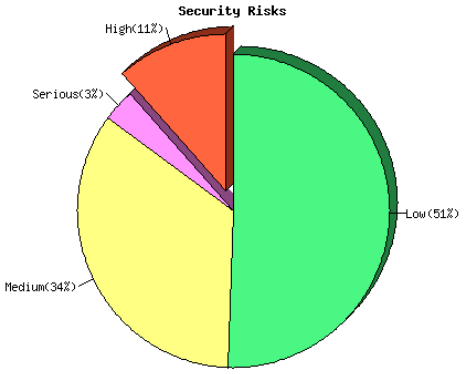
Password strength was another problem. The password-cracking program "John the Ripper" was used to do an analysis of their main payroll server. Within an hour, the program had cracked 90% of the passwords on the server. Most were only 3 characters long and easily guessable.

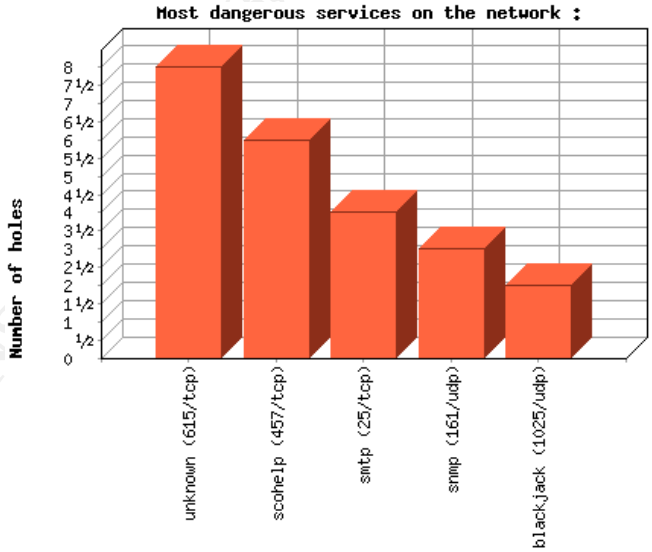
Last, but not least, there are no official security policies or procedures in place.

3) From a Red Hat Linux laptop connected to the network, a Nessus vulnerability scan was run against all servers:

36 Open ports were found on each of the SCO servers. 15% of these represented High or Serious Risk issues.

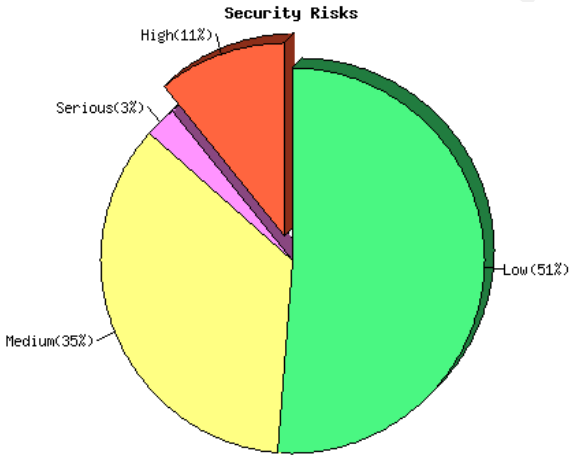
The 2 charts that follow are excerpts from the Nessus Scan Report on the 3 public servers combined.





The following pie chart and information was taken from the Nessus Report specific to the main Payroll Server. Since all 3 public servers are configured in the same manner, it accurately represents issues found on the other servers, as well.

NESSUS SCAN FOR MAIN PAYROLL SERVER (209.245.x.20)



List of open ports found (209.245.x.20):

tcpmux (1/tcp) (Security notes found)
echo (7/tcp) (Security warnings found)
discard (9/tcp)
daytime (13/tcp) (Security warnings found)
chargen (19/tcp) (Security warnings found)
ftp (21/tcp) (Security notes found)
telnet (23/tcp) (Security warnings found)
smtp (25/tcp) (Security hole found)
time (37/tcp) (Security notes found)
finger (79/tcp) (Security warnings found)
http (80/tcp) (Security warnings found)
pop3 (110/tcp) (Security notes found)
sunrpc (111/tcp) (Security notes found)
imap (143/tcp) (Security notes found)
knet-cmp (157/tcp)
smux (199/tcp)
scohelp (457/tcp) (Security hole found)
exec (512/tcp) (Security warnings found)

login (513/tcp) (Security warnings found)
shell (514/tcp) (Security warnings found)
unknown (615/tcp) (Security hole found)
unknown (620/tcp) (Security warnings found)
ftps-data (989/tcp) (Security notes found)
kdm (1024/tcp) (Security notes found)
NFS-or-IIS (1025/tcp) (Security notes found)
blackjack (1025/udp) (Security hole found)
sunrpc (111/udp) (Security notes found)
unknown (1026/udp) (Security notes found)
iad3 (1032/udp) (Security notes found)
unknown (989/udp) (Security notes found)
general/udp (Security notes found)
general/tcp (Security notes found)
general/icmp (Security warnings found)
snmp (161/udp) (Security hole found)
daytime (13/udp) (Security warnings found)
echo (7/udp) (Security warnings found)

4) Servers were manually checked for security issues:

The SANS document "SCO OpenServer 5.0.5 Security for the Systems Administrator" was used as a starting point. (This document can no longer be found on the SANS site, but we found it at the following URL: http://www.blacksheepnetworks.com/security/resources/sco.html).
Checks performed included manual inspection of all servers for security updates, unneeded services, password policies & strength, auditing features and virus protection. I will note here that I have been working with SCO Systems for almost 10 years. During this paper I may talk about SCO-specific commands or procedures. You will not see any listing of references for these instances, because they are taken from my own personal knowledge and experience.

The 3 SCO OpenServer servers were running a 6-year old version of the operating system. That version has numerous known security issues (see http://www.sco.com/support/security/) and no security updates or patches had ever been installed. An inspection of /etc/inetd showed many unneeded services active, which concurred with the Nessus Report of vulnerabilities. Some examples are: ftpd, telnetd, fingerd, pop3, echo, daytime, rlogind, and rexecd. An inspection of the /etc/rc2.d directory showed other unused services running, such as sendmail and http.

Next the SCO Security profile was checked and was found to be set to "traditional". The only positive aspects of this level are that it uses shadow passwords and logs use of the su command. Negatives are a minimum password

length of only 3 characters, 99 failed login attempts allowed, and any user can schedule jobs.

SCO OpenServer installs with auditing disabled by default. That had not been changed on any of the servers.

The Servers were found to have external modems connected and active, although no longer in use or needed.

There was no virus protection on the SCO servers, as would be expected. Anti-virus software for UNIX systems has historically been almost non-existent. Through the 1990s, I am aware of only 1 company that even sold such software. The late 1990s brought a handful of UNIX/Linux viruses/worms, thus, the 2000s have brought UNIX anti-virus products to the market, but most that I have seen are just ports of these company's DOS-based virus scanners. They are still checking for DOS and Windows-based viruses, which is only useful on a UNIX system if they are acting as file shares on a Windows network. But in that case, a Windows workstation can always be set to scan the network share for viruses. Anti-virus on UNIX is still a developing phenomenon.

Since this version of SCO OpenServer installed (5.0.5) is no longer supported, it was decided it would be best to upgrade the SCO Operating system to the latest version. Availability of current security updates was an important deciding factor.

5) Manual Inspection of Client PC's for password policies, security updates, virus protection, etc.:

All client PC's are running Windows 98, a security nightmare. Windows 98 allows easily bypassed passwords, and uses the very insecure FAT32 filesystem. Any user with access to the PC has access to the files on that drive, whether they know the login password or not.

None of the PCs examined have had any security updates or other Windows updates installed.

The PCs had a mix of various anti-virus software programs installed, but they were not setup to get virus updates, and in most cases, their license to do so had long expired.

DURING

A systematic step-by-step approach was taken to remediating their security issues. The principle of "Defense in Depth" would be applied by securing as many areas as possible on as many levels as possible. Deciding factors would be urgency, priority, budget, interference with day-to-day operations and maximum return on investment.

I personally performed the security hardening tasks below in the order listed. I will discuss each one in more detail below:

1) Installed firewall for new T1 Internet connection.

2) Installed Anti-virus on all PC's and setup a PC to act as an Anti-virus update server.

3) Upgraded UNIX servers to latest Operating System versions and installed all Security updates.

4) Closed down all unneeded services and ports on all servers.

5) Changed passwords for all users & customers and then implemented password policies.

6) Installed SSH on public servers and distributed an SSH client to all customers.

7) Installed Windows security updates on PCs and configured them to alert user of new updates.

8) Installed VPN and Encryption software on laptops of 2 remote employees.

DETAILED INFORMATION:

1) Firewall:

We chose a Watchguard Firebox III Firewall/VPN appliance to protect the network perimeter. The Watchguard is an excellent solution for small and mid-sized companies, as it is easy to maintain and has a great feature set, all at an affordable price.

Best practices would demand that all external users access the public servers via a VPN. Best practices, however, is not always possible and in this case there were several reasons for this. One reason was cost. In order to have their 200 customers all using a VPN, they would need to purchase a more expensive Firebox model that could handle that level of VPN traffic. In addition, over 200 VPN licenses would have to be purchased. These two combined would put the hardware costs too high for their modest budget. There were also practical and technical issues to be dealt with. Every customer would be required to install the VPN client software for anyone needing access to the public servers. Installation can differ greatly depending on the client Operating System. Also, running the VPN client software while still connected to a corporate network can affect local network access. These factors added up to a potential support nightmare for a company without a real IT staff. Best Practices was just not practical. We adopted the concept of MAP (Minimum Acceptable Practices) and decided customers could not be asked to use a VPN, but a more secure method than unprotected servers running telnet had to be employed. The solution (public servers in a DMZ and use of SSH) will be discussed below and again in item # 6. Remote employees or traveling employees WOULD be required to use the VPN.

As for actual deployment of the firewall, it was installed when their new T-1 line was determined to be active. It was a simple "drop-in" installation, where the firewall is placed between the Internet router and the network switch to the internal LAN. In addition, we established a DMZ zone, with a separate network switch connected to the firewall's optional interface (the DMZ), with the 3 public servers connected to that switch.

The firewall was configured to allow Internet users access to the public servers on the DMZ through the firewall's external interface while internal users could also access the public servers through the firewall's internal interface. External users can get as far as the public servers, and no further. There is no path from the DMZ to the private network. In addition, only those ports or services absolutely needed were left open or running on the external interface and the public servers. More on this later.

A remote user IPSEC VPN was setup on the firewall, using MD5 authentication and 3DES encryption. Any company employee wanting to access the company network is required to do so via this VPN. VPN client software was installed and configured on 2 laptops and a remote desktop.

The firewall also has a web-filtering feature, which allows the company to control what types of web sites its users can visit. There are approximately 30 categories that can be blocked out. The company chose to initially block access to the following: Pornographic sites, gambling sites, and violence/hatred sites.

A PC was upgraded to Windows 2000 and setup as the "management console" for the firewall. An easy-to-use graphical interface will allow the customer to check on the status of the firewall and firewall activity in real-time. In addition, 24x7 logging was setup, with a new log file being created every 24 hrs. With the firewall's

historical reporting feature, they can go back in time to a particular day's log and check specific traffic patterns or a particular user's activities on that day.

2) Anti-virus:

A multi-user license for McAfee VirusScan was purchased and the software was installed on every PC. Best Practices would call for installation and use of McAfee's Epolicy Orchestrator, an anti-virus console server that would "push" virus updates to all of the PCs and allow the PCs' anti-virus software to be managed and upgraded from the console, without having to visit each of the PCs. With the McAfee product, this calls for a separate server running Win NT 4.0 or Win 2000 Server editions. For a small company with only 15 LAN users, it is difficult to justify such an expense, so we once again employed MAP (Minimum Acceptable Practices) and instead configured the firewall console PC mentioned above to also act as the central location for virus updates. The anti-virus software on that PC was configured to go out and download updates on a nightly basis and store them in a shared directory on its hard drive. The anti-virus software on all of the other PCs was then configured to look in this shared directory for virus updates. While not as convenient as the central console approach, but a lot more affordable.

3) Upgrade Servers:

The 3 public servers were all executing a custom-written application running on SCO OpenServer 5.0.5. This version of the operating system is at least 6 years old. In those more innocent (or just more naïve) days, little worry was given to system security on UNIX systems. A look at SCO's own web site now shows a plethora of security issues that have since been found with this version (see http://www.sco.com/support/security/). As support for this version ended years ago, there are patches for only some of these vulnerabilities. Since their application requires SCO OpenServer, there was no other option but to upgrade the SCO OpenServer to the latest release, for which all security patches are readily available. This was a no-brainer, except for one thing: would their custom-written software run on the newest OpenServer? Fortunately, one of the 3 public servers is used as a development and testing server, so the plan was to run the entire upgrade process on that server, test everything, and if all was OK, continue to the production servers. Much to everyone's delight, their software ran just fine and all 3 upgrades went smoothly.

Since they were no longer used or needed, the external modems connected to these servers were removed completely. And they saved some money by disconnecting the phone lines they had forgotten they were paying for.

The internal Linux file & print server was running Caldera OpenLinux 3.1.1, which is still actively supported and has all security patches available. It was decided not to upgrade the server at this time, but rather to just apply the security patches and do some server hardening (more on this later). With a firewall solution now in place, their internal server was much less subject to attack, so it was decided upgrading this server was less of a priority.

4) Close down unneeded services and ports:

In Nessus terminology, Security Holes are high-risk, Security Warnings are medium-risk or low-risk and Security Notes are always low-risk. After discussion with the customer, they agreed that all high-risk issues should be taken care of immediately, medium-risk issues with a simple fix should also be taken care of right away, and all other medium and low-risk items would be dealt with on an ongoing basis. This was another example where the concept of MAP (Minimum Acceptable Practices) was applied.

Looking at the Nessus report, there were 5 high-risk vulnerabilities identified. (Remember that all 3 public servers were identical configurations, so these issues existed on all 3 servers. We will just discuss one server and you can assume the same actions were taken on the other 2 servers.)

smtp (25/tcp) (Security hole found):

Like many UNIX systems, the default SCO installation has Sendmail running. This leaves port 25 open, and Nessus detected several sendmail vulnerabilities. Since the company does not host their own mail servers, we could safely disable sendmail. We did this by moving the sendmail startup file out of /etc/rc2.d and into a private root-owned directory. Now SCO would no longer start the Sendmail service at startup.

scohelp (457/tcp) (Security hole found):

Nessus found 3 vulnerabilities associated with SCOhelp, which is their web-based online help system. These were all related to Javascript and involved Cross-Site Scripting vulnerabilities. Cross-site scripting (also known as XSS or CSS) occurs when dynamically generated web pages display input that is not properly validated. This allows an attacker to embed malicious JavaScript code into the generated page and execute the script on the machine of any user that views that site. This company had no use for http on these servers and was not even aware that the capability was there. The easy fix was to disable http completely. This was done by removing the line in /etc/inittab that starts it upon bootup.

unknown (615/tcp) (Security hole found):

Nessus reported 2 more vulnerabilities centered on Javascript and Cross Site Scripting. On SCO systems, port 615 is used by their X-Windows-based Internet Configuration Manager and System Administration Manager. The customer had never once used the X-Windows interface on these servers, so we decided to disable it completely. This is simply done with the command "scologin disable". (It will no longer start upon bootup, either)

blackjack (1025/udp) (Security hole found):

The next vulnerability found refers to a RPC (Remote Procedure Call), specifically rpcc.walld. The Nessus description in the report stated:

*"The rpc.walld RPC service is running. Some versions of this server allow an attacker to gain root access remotely, by consuming the resources of the remote host then sending a specially formed packet with format strings to this host. Solaris 2.5.1, 2.6, 7 and 8 are vulnerable to this issue. Other operating systems might be affected as well. *** Nessus did not check for this vulnerability,*** so this might be a false positive".*

A further examination of this vulnerability on the CERT website revealed that the vulnerability exists only in certain versions of Solaris and AIX, so this was, in fact, a false positive.

snmp (161/udp) (Security hole found):

As is usually the default with most network devices, the SNMP community name was set to the default of "public". As the Nessus vulnerability description states, besides giving away information about the server, if "writeall" access can be gained, this would be considered a huge security hole, allowing attackers to route packets and cause general havoc on the network. We could have disabled snmp completely, as it was not currently being used, but we could make the case for it's future use, so instead, we changed the default community name from "public" to a much more private and meaningful name. Any other network devices using SNMP would be changed to use this community name as they are encountered in the future.

Now lets look at the Medium-risk (security Warnings) vulnerabilities that were found:

echo (7/tcp) (Security warnings found)
daytime (13/tcp) (Security warnings found)
chargen (19/tcp) (Security warnings found)
telnet (23/tcp) (Security warnings found)

finger (79/tcp) (Security warnings found)
http (80/tcp) (Security warnings found)
exec (512/tcp) (Security warnings found)
login (513/tcp) (Security warnings found)
shell (514/tcp) (Security warnings found)
unknown (620/tcp) (Security warnings found)
general/icmp (Security warnings found)
daytime (13/udp) (Security warnings found)
echo (7/udp) (Security warnings found)

Most of these are services that are enabled by default upon installation. Most of them are not needed and thus, were just turned off. An examination of /etc/inetd showed the following:

```
#      @(#)$Id: inetd.conf,v 6.8 1996/01/09 21:48:54 aes Exp $ - STREAMware
TCP/IP  source
#
# Copyrighted as an unpublished work.
# (c) Copyright 1987-1994 Legent Corporation
# All rights reserved.
#
#      SCCS IDENTIFICATION
ftp   stream   tcp   nowait   root  /etc/ftpd   ftpd
telnet   stream   tcp   nowait   NOLUID   /etc/telnetd   telnetd
shell stream   tcp   nowait   NOLUID   /etc/rshd   rshd
login stream   tcp   nowait   NOLUID   /etc/rlogind   rlogind
exec  stream   tcp   nowait   NOLUID   /etc/rexecd rexecd
finger   stream   tcp   nowait   nouser   /etc/fingerd   fingerd
#uucp stream   tcp   nowait   NOLUID   /usr/lib/uucp/uucpd  uucpd
# Enabling this allows public read files to be accessed via TFTP.
#tftp dgram udp   wait  nouser   /etc/tftpd  tftpd
# This is the more secure method, since only files from /tftpboot can
# be accessed via TFTP.   This must be root in order to do the chroot
# to /tftpboot.  /tftpboot must be created by hand.
#tftp dgram udp   wait  root  /etc/tftpd  tftpd -s /tftpboot
comsat   dgram udp   wait  root  /etc/comsat comsat
ntalk dgram udp   wait  nouser   /etc/talkd  talkd
#
# Entries for BOOTP and DHCP servers & relay agent
#
# If running tftpd in secure mode, use bootpd with "-c securedir"
# where securedir is the argument to tftpd -s.
#
# To run bootpd by itself, use:
#bootps  dgram udp   wait  root  /etc/bootpd bootpd
#
# To run dhcpd by itself, use:
#bootps  dgram/i  udp   wait  root  /etc/dhcpd  dhcpd
```

```
#
# When running dhcpd and bootpd, bootpd must be run in "slave mode" (with the
# -S option).  In this mode, bootpd listens on an alternate port.  The port
# bootps-alt is defined to be 950 in /etc/services, but it can be anything
# < 1024.  To run dhcpd and bootpd, use the following two lines:
#bootps  dgram/i  udp   wait  root  /etc/dhcpd  dhcpd -b bootps-alt
#bootps-alt dgram udp   wait  root  /etc/bootpd bootpd -S
#
# To run the BOOTP/DHCP relay agent bootpgw, use:
#bootps  dgram/i  udp   wait  root  /etc/bootpgw   bootpgw server-name
#
tcpmux   stream   tcp   nowait   root  internal
echo  stream   tcp   nowait   root  internal
discard  stream   tcp   nowait   root  internal
chargen  stream   tcp   nowait   root  internal
daytime  stream   tcp   nowait   root  internal
time  stream   tcp   nowait   root  internal
echo  dgram udp   wait  root  internal
discard  dgram udp   wait  root  internal
chargen  dgram udp   wait  root  internal
daytime  dgram udp   wait  root  internal
time  dgram udp   wait  root  internal
pop3  stream   tcp   nowait   root  /etc/popper popper
imap  stream   tcp   nowait   root  /etc/imapd imapd
gamserv stream tcp   nowait   root  /etc/gamserv   gamserv
   bootps dgram udp wait root /etc/bootpd bootpd
```

To disable these services, you simply comment out the lines that run them in /etc/inetd.conf. They will then no longer start automatically when the system boots up. We determined that the following services could be safely turned off: fingerd, rshd, rlogind, rexecd, comsat, ntalk, echo, discard, chargen, daytime, pop3 and imap. The only things left running were FTP (needed for internal users only – blocked at firewall), Telnet (to be disabled later on when SSH is implemented), tcpmux (not sure of disable affect), time (not sure of disable affect) and gamserv (needed by RAID controller). These actions took care of 11 of the 13 security warnings found by Nessus. The 2 that remained carried a very low risk factor, so we were very satisfied that we were able to easily fix all high and medium risk vulnerabilities found on these servers by Nessus.

5) Password changes for all users:

Our Jack the Ripper results confirmed what we already knew – the passwords being used were very weak. First order of business was the root password. Presently, it

was an easily guessable (for Jack the Ripper) two-word combination and the password was identical for all 3 Payroll servers and the File & Print Server. If an attacker can guess the password on any server, they could now be root on all servers on the network. In addition, many internal employees knew the password. All it takes is one fired or disgruntled employee, and their entire network could be at risk. Much to the company's dislike, we insisted they create a different root password on each server, and specified that it should be at least 10 characters, including uppercase & lowercase letters, numbers and special characters. To help the Administrator remember the passwords, we had them use the common phrase approach, where the first word of the phrase starts with the number of the server, and the first letter of each word in the phrase is used in the password. Special characters are used to bring the password up to 10 characters, total. For example, the password for server #2 could be 2HaBt1!@#$. This stands for "Two Heads are Better than One". Since it was only 6 characters, we add the first 4 special characters on the keyboard (starting with ! and moving left to right) to bring it up to 10 characters.

Next we checked if any other user accounts on the systems had been given root permissions in any way. We did not find any, so we moved on to the last order of business for root. Since they are running public servers, it would be prudent to restrict root logins to the console. This would thwart anyone trying to login remotely by guessing the root password. If anyone legitimately needs to login as root, they can instead login as a user and use the su command. In SCO, you restrict root login by adding an entry in /etc/default/login.

Next came user passwords. A series of 3 e-mails were sent out to customers, advising them that starting with a certain date, at their ensuing login they would be prompted to create a new password for themselves. The new rules were that passwords had to be at least 8 characters long and include upper and lower case letters, numbers and at least 1 special character. While SCO does not natively have the ability to enforce all of these rules, there are some free utilities available (passwd+ and npasswd) that can overcome these limitations. Because it seemed to be well documented, it was decided to use npasswd. The program was downloaded, unzipped, compiled and installed on the servers. It was configured with the following settings: minimum length = 8, passwords checked for combinations on letters, numbers and special characters, password must be different from the last 3 chosen, and password checked to not contain the username, hostname, etc. In the SCO password settings, we set the password to expire every 60 days, and to lock out users upon 3 successive unsuccessful login attempts.

6) Install & implement SSH:

We expected to have to download the sources and compile SSH on the SCO servers. We were pleasantly surprised when we found that the new version of SCO we had just upgraded to already included OpenSSH. Therefore, there was very little we had to do on the server side. For the client side, we chose what is probably the most popular Windows SSH client, a free program called "Putty". OpenSSH and Putty are distributed under the OpenBSD and MIT licenses, respectively. Both of these licenses allow for use of the products without any royalties or license fees, even for commercial use. (The customer really liked that!)

We downloaded Putty, which is nothing more than a single executable file (putty.exe) It can be installed and run from anywhere on a Windows system. We then created a saved session for each of the three servers to be accessed, and 3 Windows shortcuts to open putty using each of the three sessions. This all made it very convenient for creating a simple installation script for the end-users. All the script had to do was copy the putty.exe file to a specific directory and move the shortcuts to the user's desktop. We chose to only copy the shortcuts to that user's desktop, so if other users log into the PC, they wouldn't see the icons. Of course, if they are using Windows 98, that does not apply, but it was worth doing it for the NT, Win2K and XP users anyway.

We were careful to use the improved SSH2 protocol. SSH2 is a significant improvement over older versions of the Secure Shell protocol. It is better designed and more flexible; but most importantly, the protocols of the 1.x series have a major design flaw that renders them vulnerable to some active attacks. SSH2 has no such issues.

We put putty.exe, the shortcuts and the install script on a CD. The customer burned 200 copies and mailed it out with instructions to it's users. Included in the mailing was a letter explaining all of the new security features that were being implemented by the company for their own protection and at no additional cost to them (the "put a positive spin on it" approach). It also informed them of the deadline date for switching over to this new access method. On the deadline date, telnet would stop working. When the deadline came, and we "turned off" telnet, there were only a handful of complaints from a few stragglers that hadn't installed Putty. Overall, this went much smoother than expected.

7) Windows 98 security issues and updates:

Securing an All-Win98 client network? Where do we begin? Well, we started with Anti-virus. A 20-user license for Network Associates' McAfee VirusScan was purchased and the software was installed on every PC. The PC acting as the firewall console PC was also setup as the anti-virus update "server". This PC checks McAfee's FTP site every night for virus and engine updates. Then all of the other PC's were setup to check the anti-virus "server" in the morning for new updates. The Anti-virus software now performs real-time virus scanning and checks all their pop3 email for nasty viruses, worms or trojans. Whew! Now we could breathe a little easier.

As far as password policies, there is little you can do on Windows 98 to stop someone from bypassing the login and accessing the local files. Fortunately, most of the PCs were rather old and would be replaced in the next year. We requested they use XP Professional or W2K with the NTFS filesystem on any new PCs they purchase.

Next came the issue of Windows updates. We downloaded most of the important Windows 98 critical updates and security updates and put them on a CD. The customer installed these updates on all of the PCs. There are some updates that are only installable through the Windows Update service, so we showed the customer how to do that and which updates to install. One of these updates happens to be a service that will let the Windows 98 PC go out and check for new updates. If any exist, it will pop up a window alerting the user there are new critical updates do be downloaded. That's the best that Windows 98 can do in this area.

8) VPN setup for laptop users:

The last issue to tackle was employees who need to access the public servers and internal network from outside the office. The Watchguard Firebox also provides an IPSEC VPN, so that was the logical choice. IPSEC is a Layer 3 method for providing tunnels. It provides data integrity monitoring, authentication and encryption.

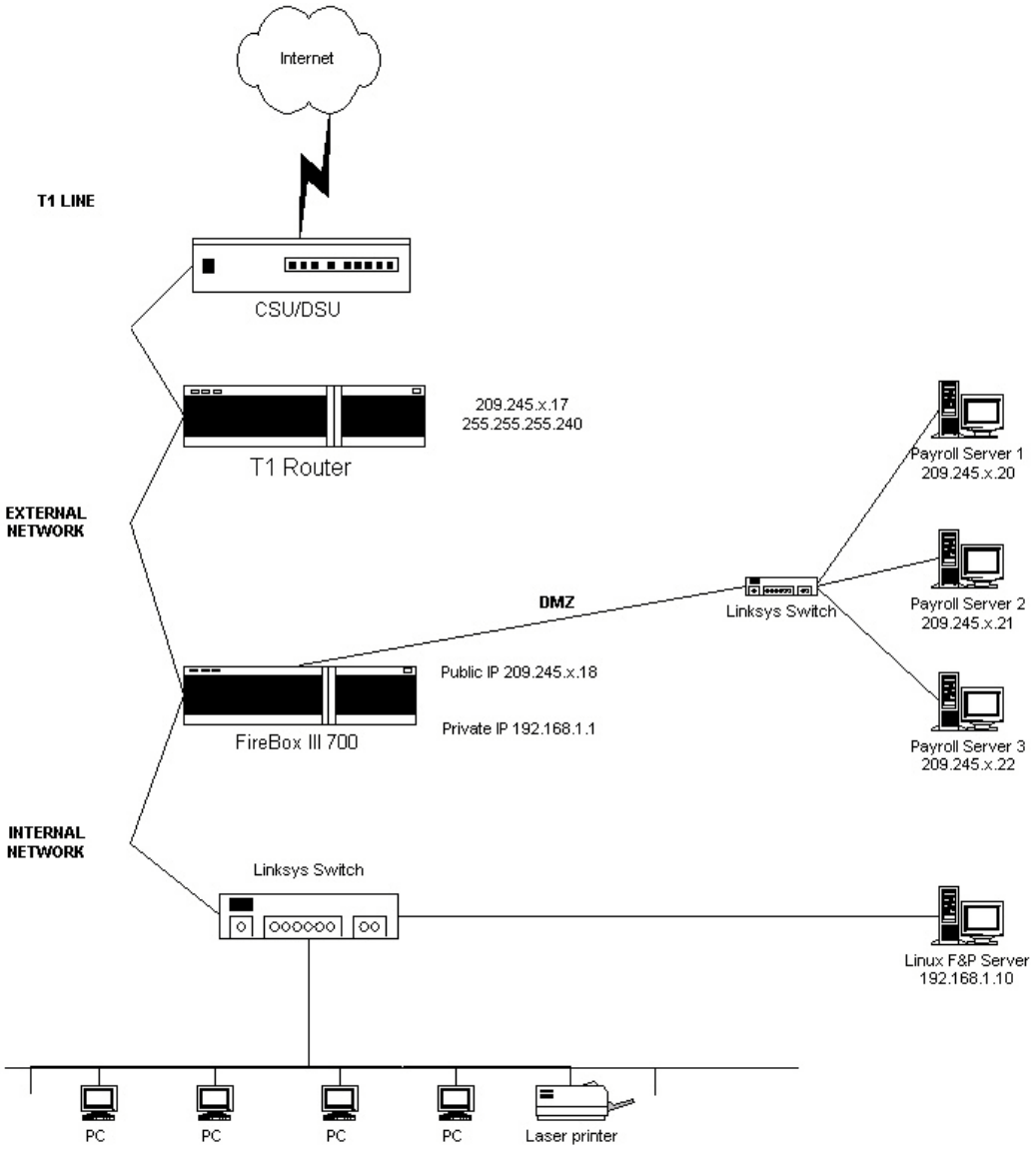
The company had 2 laptop users that often traveled and one employee who worked from a home office. They all needed access to the internal file & print server from outside the office. The first step was to setup a VPN account for each of the 3 users on the Firebox. This consisted of selecting a username, password, virtual IP address, and the encryption and authentication types. For encryption, we opted for the more secure 3-DES. Although this would slightly impact performance, it would hardly be noticeable on a small VPN such as this. The default authentication method for the Firebox was SHA-1. We stayed with this default hash function since it creates a 160-bit hash value versus 128-bit for MD5.

Upon completing the VPN account setup, the program creates a configuration file for each user. That configuration file is then loaded on the respective user's PC at the same time that the remote client VPN software is installed. Once installation is complete, the user connects to the internet, activates the VPN client and they have a

secure tunnel to the company network. This all sounds great, but there is one problem – the user is never asked for his VPN password again. If a laptop is stolen, and the thief knows how to start the VPN, he/she is into the company network. I have brought this up with the firewall vendor and they promise it will be changed in a future release. In the meantime, I can live with this only because I always insist on using encryption software for laptop users, which will make the hard drive useless without the password. The product used here was Encryption Plus Hard Disk from PC Guardian. It performs on-the fly encryption and allows you to encrypt the entire hard drive or just selected files and folders.

AFTER

Here is what their network looks like now:



The new layout of their network safely separates the public servers from the internal corporate network. With the public servers in a DMZ, and no ports open from the DMZ to the internal network, even if one of the public servers were compromised, the company's LAN cannot be reached. The new Firewall, which uses stateful packet inspection and application proxies, will protect the network from Denial of Service, Syn Flood, buffer overflow and many other types of attacks.

As for the public servers, telnet access has been disabled in favor of SSH2. This will eliminate the possibility of passwords being sniffed. We also significantly increased the password security on the servers by limiting the number of unsuccessful login attempts to 3 and imposing a 60-day expiration on all passwords. In addition, we implemented

npasswd, which adds password rules not natively inherent in SCO OpenServer. Now passwords must be 8 characters, contain a mix of upper and lower case characters, numbers and at least one special character. A re-run of Jack the Ripper after these changes produced 0 guessed passwords after 1 hour. Previously, 90% of the passwords had been guessed in 1 hour. Finally, we hardened the servers, closing down every port and service that wasn't explicitly needed. This alone will make it much more of a challenge to hack one of these servers, as most attacks look for common running services or open ports to exploit.

On the PC side, we have properly protected them with anti-virus software that is always up-to-date. This is especially important for screening e-mails for worms, viruses, trojans and other forms of malware. While the current PCs are running Windows 98, all future installations will be Windows 2000 or XP, which are much more secure.

For remote employees, a VPN is now in place for accessing the internal network. This is the only way remote users will be granted access to the company LAN.

There are no longer any external modems on the servers. Any PC's with internal modems were checked to be sure they were not connected to a phone line.

Two problems that still remain are poor physical security and lack of any formal security policy. As mentioned earlier, the servers and firewall are in a room that houses the network printers and also serves as the company mailroom. Because of a lack of available space, this room must continue to be multi-purpose. Thus, employees will continually be in and out of that room. That's the bad news. The good news is that employees will continually be in and out of that room. What I mean by that is, during business hours, it would be impossible for someone to try to break into one of the servers and not be seen trying. So the issue is, what about non-business hours? Well, for that they can at least put in a door with a combination lock. Only the two users who have been trusted with the root passwords, also have the combination to the door. They are also going to buy a new cabinet with a locking door to house the servers and the firewall, so nobody can walk by and "accidentally" hit a key on the keyboard or unplug a piece of equipment.

As for lack of any formal security policy, I have gotten them started with some sample policies taken from the SANS Web site. They are working on modifying these and adding their specific acceptable use rules and security requirements (password policies, VPN requirements, etc.)

In conclusion, the varied steps taken above illustrate the security concept of Defense in Depth. Multiple layers of security have been applied at the network, server and workstation levels. While this company is not perfect from a security standpoint (remember, MAP has been applied several times), they are much improved from where they once were. They are protecting their customers' valuable data and protecting themselves. They can show evidence of having taken multiple steps to secure the data stored for their customers. And they can also sleep better at night.

REFERENCES

1) Allen, Julia."Improving the Security of Networked Systems". CERT. 2000. URL: http://www.stsc.hill.af.mil/crosstalk/2000/10/allen.html

2) Bosworth, Kabay. Computer Security Handbook. John Wiley & Sons, 4th Edition, 2002.

3) CERT. "Keep Operating Systems and Application Software Up to Date". 2001. URL: http://www.cert.org/security-improvement/practices/p067.html.

4) CERT. "Windows 95/98 Computer Security Information". 2000. URL: http://www.cert.org/tech_tips/win-95-info.html

5) Lawrence, Tony. "Basics – SSH". 2001. URL: http://pcunix.com/Basics/ssh.html

6) McBee, Brian. "SCO Openserver 5.0.5 Security for the Systems Administrator". 2001 URL: http://www.blacksheepnetworks.com/security/resources/sco.html .

7) McClure, Scambray, Kurtz. Hacking Exposed. McGraw-Hill, 3rd Edition", 2001.

8) Mitnick, Kevin. "Committee on Governmental Affairs". US Senate, 1997". URL: http://www.senate.gov/~gov_affairs/030200_mitnick.htm

9) SANS. "Security Essentials with the Common Body of Knowledge" SANS Training Materials. 2003.

10) SCO Security Advisories. 2003. URL: http://www.sco.com/support/security/