Global Information Assurance Certification Paper

SANS Security Essentials Practical: Securing a Wide-Open Computer Network

Mark Andrich

Abstract:

Instead of choosing one single aspect of this course, I will be outlining how I've applied many of the security principles discussed in this paper.

Selection Criteria:

While there are many highly qualified IT professionals in the field today, there are also a large number of unskilled people who used the media's certification hype as leverage to gain employment in the IT industry. The best example of this is Microsoft's MCSE which teaches nothing in terms of security concepts. Having dealt with this situation myself, I though it might be helpful to do an overview of the steps that I've taken to secure a wide open network from almost zero knowledge.

Background:

In late 1999 I joined the ranks of the "paper MCSE's". I had no prior experience and no real computing knowledge. Armed only with a degree in electronics, I jumped on the MCSE bandwagon with countless others in the quest for easy money. Fortunately a consulting company hired me before I even completed my certification. Most of what I was taught was menial tasks dealing with e-mail, toolbars, and Windows problems. My computing universe did not extend beyond Microsoft products and some occasional connectivity issues.

About six months into this, the company that hired me split from a former merger and many of the client sites were left without service. I received an offer to become a full time employee from one of the companies I worked onsite at. Not just as a network administrator, but as THE information services department. The situation I stepped into covered every aspect of planning, implementation, support, etc. I can't stress enough how narrow the scope of training is for the MSCE and how poorly prepared and foolish it was for me to accept (and them to offer) so much responsibility. Looking back now it was only through common sense and luck that something really bad didn't happen. Overwhelming to say the least...

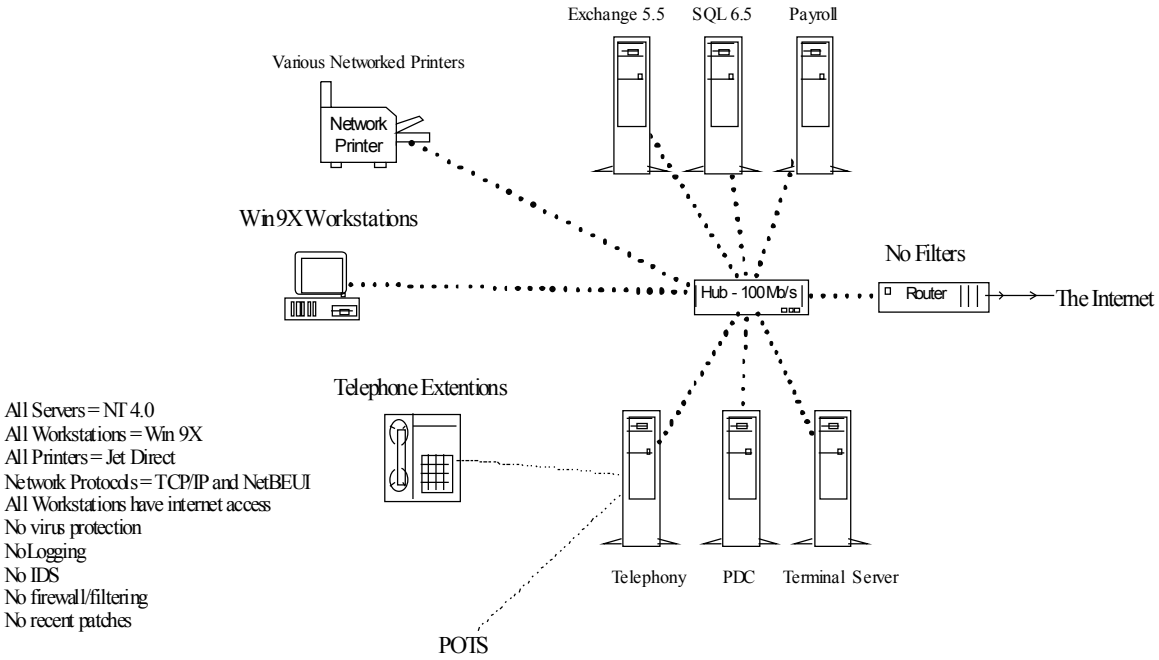
After a time of treading water, I realized the need to document the network. I also began to get the idea that my former employer (who had built this network from the ground up) had not taken a very textbook approach to things. While I did not realize the extent of neglect that was going on, I did get a feeling that I should start researching how networks should be setup and what the best practices were. Everything listed in this paper is an outline to the path that I took from total ignorance to at least having some of the right

ideas. While some of the choices here would not have been a choice I might have made now, they were the best I could do with the knowledge and experience that I had to work with at the time. Under these sections, I've taken the time to outline what I might have done differently now in hopes that anyone in a similar situation can take both approaches and weigh them against their current needs.

Part one: The original network

As you can see from the diagram below, this network was completely vulnerable. Every host had a public IP address and was connected directly to the Internet. Among these were an Exchange server, a SQL server which held all the financial information for our organization as well as all of our affiliates across the country, a Terminal Server, a dedicated server that housed all of our payroll information, our telephony server, our PDC, eight networked print devices, and about 45 Win 9X workstations. This network had been running exposed like this for about three years as the various components were added. Though this was a TCP/IP network, all workstations and servers also had the NetBEUI protocol installed for redundancy. None of the hosts on the network had any kind of personal firewall, IDS, or virus protection, and logging was disabled on all servers. Lastly, a few of the employees thought it was fun to install Microsoft's Personal Web Server and leave their machines up 24 hours a day/seven days a week logged into the network.

The Original Network



\

Part 2: Where to begin

My first goal was to put a barrier between our network and the Internet. Due to budget constraints and limited knowledge my choices seemed pretty sparse. It had to operate on an NT 4.0 operating system, have a point and click interface, and cost around $500.00. Being in "Microsoft mode", I pretty much defaulted to Microsoft's Proxy Server 2 to serve as our firewall. The machine was re-formatted and re-installed from scratch as a stand-alone server. The most recent service pack and the all-subsequent patches were then applied. I removed any unwanted services from the external NIC. (I.e. NetBIOS)

Additionally, this server was to run IIS for a small extranet site. I made a full back up of the server and followed Microsoft's "Internet Information Server 4.0 Security Checklist" - http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/tools/iischk.asp. I will warn from experience that it is very important to make a copy before you go mucking about in the registry or any changes for that matter. It's something that everyone hears over and over again, but it's really easy to skip when you're in a rush. Did I mention that I had to do the above twice because I didn't make a back up or an ERD prior to making changes the first time? Thankfully, it was only a loss of time.

When the server was ready, I installed Proxy Server 2 and configured the following:

1. Applied NAT and changed our internal IP addresses to a non-routable scheme.
2. Used the "Access Rights" feature to limit access to the various Internet services user by user using a "denied by default" approach.
3. Applied port/packet filters to ensure that only the necessary services were available.

In addition I also applied many of the same filters to our router for redundancy. The end result was to limit open ports to specific services that we were using in the "well known" port range. When everything was finished, the Proxy Server was dropped in behind our router with mail forwarding enabled to our Exchange server which was still located on our internal network.

If I were to go back and do it again, there are a number of things I would have done differently. First, I would have moved the Exchange and IIS server onto individual machines and then off into their own domain and set up a DMZ. Having a web server on the same machine as your firewall on your network is not the best choice as it complicates the configuration of the machine and leaves more oporotunities for mistakes. Additionally, having a straight shot through the firewall on port 25 to a machine behind your first tier of protection is just a downright bad decision. I'm in the process of correcting this. I guess hindsight is always 20/20. Secondly, I would have a dedicated firewall on a *nux/BSD platform. I am not a zealot, but I do think open source OS's have

some advantages (when properly configured) over Windows NT 4.0. I wouldn't suggest any Windows NT/2000 admin jump into implementing any OS without taking a few months to really learn it. After all, if you don't know how to configure it, you can't rely on it to protect you.

Part 3: Cleaning up the network

Once we had something between the outside world and us, our next step was to clean up the internal network. We purchased a popular anti-virus program and installed a licensed copy on each of our servers and workstations with the most current definition files. We scanned every single drive and cleaned or re-formatted when necessary. In order to be sure that definitions were being updated on a weekly basis, I wrote a small batch file to check for and download the most recent definitions to each workstation at logon from a network share. While making the rounds and installing the anti-virus software I went through each workstation and removed any personal software. There were tons on screensaver programs, joke files, games, a few web servers, and some file sharing. Once everything was uninstalled, the CD and floppy drives were disabled and unplugged. Each machine received any appropriate patches and browser software was updated.

Once the workstations were taken care of it was time to move to the servers. I made a list of all accounts and the groups they belonged to. Using this list, I sat down with each department manager and our CEO to determine which user accounts were no longer needed as well as updating which current employees should belong to which groups. All unneeded accounts were deleted or disabled, all groups were updated. This was done for individual programs (SQL Server) as well as for the domain itself. While most of the deleted accounts were ex-employees, there were a few accounts that concerned me. Namely were names that no one recognized, and one account listed in our SQL server named "probe". Once the domain accounts were groomed we changed passwords for all system/service accounts. Finally, we renamed the domain and each local administrator account and created fake ones in their place.

Next we implemented account policies. The password restrictions being a minimum of seven characters, changes every 45 days, and the inability to use the previous three passwords. I considered using the strong password feature (article Q161990) but management felt that the previous changes were enough and didn't want to inconvenience the employees further. Once password policies were in place auditing was enabled on all servers. All failures are logged as well as successful logons and privilege changes. Lastly we set the account policies for logons. The number of unsuccessful attempts was set to five. The system would lock the account forever, and the count was reset every 360 minutes.

This is one area where I had thought I'd encounter the least amount of problems and was proven wrong. Employee understanding and cooperation are vital to the success of any attempts to secure a network. With the new policies in place, many long-time employees very unhappy about having to deal with all this new "big brother" stuff. Most employee's

had had the same simple password from day one, and were outright angry that they had to change to a long one, remember it, and then change it again to something new in a relatively short time. I wound up having to negotiate with management and find a compromise that made everyone happy. The lesson I learned here is that while you know how important what you're doing is, the average employee or manager does not. Most people don't understand what it is you're doing and most don't have the ability to see beyond the immediate inconvenience.

Once we had reached an acceptable compromise efforts were made to put some unwritten policies into black and white. While I have yet to get permission to write out a full policy, I was able to get the following listed in the employee handbook:

```
For those of us in the National office our policies are as follows:

   1. Don't download anything from the Internet. This includes any
   games, pictures, documents, screensavers, joke files, etc.

   2. Don't open attachments that you receive via e-mail unless it's
   the exact attachment that you were expecting. Even if it's someone
   you know they may have opened an infected e-mail and inadvertently
   triggered an infected e-mail to be sent to you. If you receive an
   attachment that you were not expecting, please check with
   Information Services to find out if it's safe.

   3. Don't install any software on your workstation without clearing
   it with Information Services first. In the past we've seen
   everything from  "productivity enhancing software" for the Palm
   and Visor PDAs to games and screen savers that have been installed
   without permission. Installing any software (especially custom
   software, games, and screensavers) increases the risk of
   installing trojans (hidden programs), worms, and viruses onto our
   network.
```

Granted, this is just a quick blurb so that we have something in writing. In the near future I hope to create a written policy that covers all aspects of our network resources and their uses in depth as well as a "quick read" version that must be signed. The problem of not having a written policy is twofold. First, not having policies in writing makes them very easy to dispute or just plain ignore. Secondly, without written policy outlines, it's not just hard to enforce the rules, but it's also hard to keep a focused view of exactly what your company's stance is on many different issues.

Our next addition to network protection was the purchase of a mail gateway scanner. Though most people were pretty good about not opening attachments, there is always those one or two people who will open anything without regard. Weeks after the "I Love You" outbreak we still had two people who's machines suddenly got infected, even though they didn't open anything. (They swear, so it must be true.) The scanner updates two virus definition files and a worm definition file daily, and notification is sent to all parties involved upon detection, including administrators. By stopping known infected documents at the mail server, we've dramatically reduced our chances for infection.

However, because there's always new stuff that's yet to be discovered, we still remind everyone on a regular basis about not opening unexpected attachments without having them checked first.

Part 4: Education

Education is more than just pursuit of certifications. Education (like security) is a daily process that must be pursued on a continual basis. As I began making these changes, I would have been totally lost without the following informational resources:

Web Sites –

1. www.sans.org - The Security Reading Room and Security Digest are excellent.
2. www.securityfocus.com – A great resource for industry news and alerts. Also, the incidents and security basics mailing lists are full of good reading.
3. www.packetstormsecurity.org – Excellent forums and tool libraries.

Training –

In my opinion the Sans/GIAC certifications are in depth technical training that focuses on practical information. In the two SANS/GIAC certifications I have pursued so far, I have come away with a ton of knowledge that I could put to use right away. These are the only security specific certifications I've trained for so far and they've been some of the most worthwhile training I've come across.

Reading -

When I first began looking into security, I was drawn to anything with the word "hack" in the title. Many of these books were more of an overview or a "how to" and lacked technical content. I found the following books to be of great benefit to my understanding of TCP/IP, and would strongly recommend them to anyone who really wants to learn. A solid understanding of protocols and packets is a good foundation to start building on.

1. "TCP/IP Illustrated Vol. 1" by W. Richard Stevens, Addison Wesley Longman, Inc.
2. "Network Intrusion Detection: An Analysts Handbook" S. Northcutt, J. Novak, D. McLachlan; New Riders Press
3. "Intrusion Signatures and Analysis" by S. Northcutt, K. Frederick; New Riders Press

Part 5: Detection

Until this point I had been putting up barriers with only a vague idea about what I was trying to keep out. I decided to try out some of the free programs that are available in order to "see" what I was trying to defend against, and if I was successful. I downloaded Windump and Snort Win32 V 1.7 and took some time to get familiar with them.

WinDump is the windows version of TCPDump and was ported over from *nix. WinDump is a bare bones packet sniffer that works with WinPcap drivers to pull raw data off the network. The available command line switches make this program flexible and useful for a wide range of tasks including data collection and communications troubleshooting. Due to the real-time display of information, it's useful to use the –w (filename) switch to write the collected information to a file for later review. The –r (filename) switch will read from a file instead of the network interface. After experimenting with Windump for a few days, I had a fairly good idea of what normal traffic was on our network. Both Windump and WinPcap can be downloaded from NetGroup at http://netgroup-serv.polito.it/netgroup/tools.html. The educational value of this program is enormous!

Like Windump, Snort is free and has been ported over from Unix. Originally written by Martin Roesch, this program is a fully configurable NIDS that is customizable through dynamic rules that work in conjunction with the downloadable static rule sets.

According to www.snort.org: "Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient."

Using Snort made a huge difference in my understanding of what I was defending against. Instead of "trying to keep hackers out" I was able to see what exploits were being attempted, and verify that I had the proper settings/patches in place. On the third day of using snort I noticed the alert log had the following entry repeated many times:

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-10:53:19.871932 198.xxx.xxx.xxx:1363 -> 216.xxx.xxx.xxx:80
TCP TTL:128 TOS:0x0 ID:30537 IpLen:20 DgmLen:953 DF
***AP*** Seq: 0x125B68C  Ack: 0xE78DB6DE  Win: 0x2238  TcpLen: 20
```

First, I had thought that someone might be running a script against our server. After taking a moment to actually look at the log files, I noticed that our server was the aggressor. I blocked all outgoing packets to their address at the router, sent them an e-mail explaining the situation and that I had blocked the activity, and even called and left a message after failing to get any response from them. In the end, there were a total of 255 alerts for this attack, which is a nice number for a script. They never returned my e-mail or phone call, and I never found any trace of what files might have been planted.
While I was left with a lot of unanswered questions, I realized that without some kind of

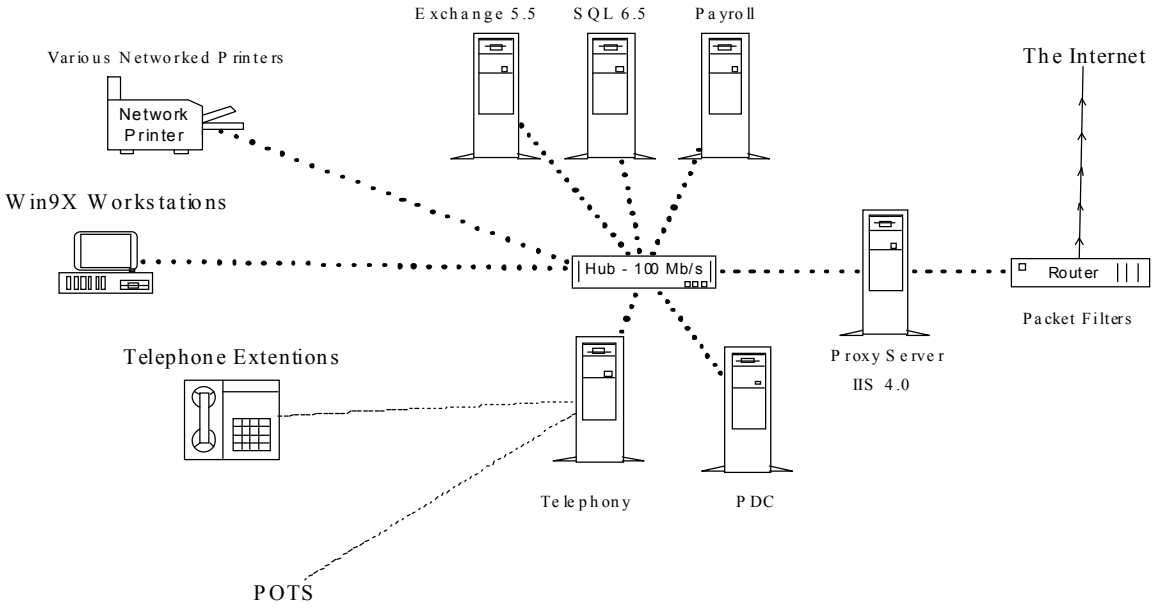
IDS that I never would have been completely unaware that my server was being used to attack another. Snort is an excellent way to begin learning about intrusion detection, as well as to be more knowledgeable about what's happening on your network or gateway.

Snort and WinDump are just two of many useful tools available for the Windows platform. It is important to note that while the number of available Win32 programs is growing on a daily basis, there are a mind boggling number of these programs for Unix based systems. In fact, most of the more popular freeware programs (Snort, WinDump, Nmap, Tripwire) were initially developed on open source OS's and tend to offer more advanced functionality in their native platform.

Part 6: Daily Practices

In addition to the changes made to the network, I've learned to change my daily routine. As stated previously, education and security are an ongoing process. Each morning I start by reviewing the log files, checking my favorite websites for news and patches, and then review e-mail received from the various mail-lists for developing news or warnings. This enables me to keep on top of what's happening both on our network and in the outside world. Additionally, Keeping up on current issues on a daily basis makes it much easier to stay current in what patches you might need for your machines. As perfectly outlined by the recent "Code Red" scare, an unpatched machine can be a liability to others as well.

While there's still a lot of work to be done, many of the basics that were previously missing are now in place. The current network:



All Servers = NT 4.0
All Workstations = Win 9X
All Printers = Jet Direct
Network Protocols = TCP/IP only with non-routable IPs on internal network
All Workstations have internet access through a proxy server
All Workstations and servers have anti-virus programs that are updated weekly
Exchange Server has mail scanning anti-virus program to scan all mail and attachments
Auditing has been enabled on all servers
Snort IDS on external and internal interfaces logging to the NT event log
Proxy Server 2 firewall configured with access rights
All patches up to date
All accounts cleaned up, organized, and monitored
All logs monitored on a daily basis

Sources Used while researching solutions:

Websites:
Packetstorm.securify.org
www.snort.org
www.securityfocus.org
www.google.com

Books:

Windows NT/2000 Network Security Schultz / New Riders Publishing / July 2000
Building Internet Firewalls Zwicky, Cooper, Russell (Editor) / O'Reilly / September 1995
Network Intrusion Detection, An analyst's handbook Northcutt, Novak, McLoughlan Sept 2000
Intrusion Signatures and Analysis Northcutt, Fearnow, Federick, Cooper / New Riders/ Jan 2001