Global Information Assurance Certification Paper

Network Security Concepts and Essentials: A University Overview
Matthew Wu Leng
Version 1.2e
September 2001

Abstract

Have you ever tried looking at your company's network from a different perspective? You work daily, usually long hours, drink high caffeine drinks, looking after your company's prized possession – network and system infrastructure. But just for a moment, imagine yourself as a hacker trying to get into your network. Could it be done? Have you tried it? What are the possibilities?

We could say that secure networks are those that combine skill and knowledge with successful implementation and management direction. But imagine a university network for a second…

How many computers would a university have? Estimate the number of users? A very large number is the answer, probably in the 10's of thousands and possibly higher. Do all these users authenticate with appropriate encryption techniques? How do users understand Internet security awareness? What does this all mean?

And imagine no perimeter firewall and intrusion detection system in place - it's an attacker's dream place to commence creating havoc amongst the university community. And this can all be done from the convenience of your next-door neighbour's home computer.

Now think from the inside for a moment. What level access do your employees have? Could they potentially do damage to your company? Who is to stop them? And who chooses your staff's passwords? Are they i) strong enough; ii) encrypted well enough; iii) accessible by others?

Using my experience from working at an Australian university, I will discuss how the number of internal and external threats is increasing and providing intruders with a vast array of "ways to compromise university machines."

Introduction

A network can be defined as two or more computers connected together so they can share resources easily and with high reliability. They can also provide cost saving benefits to companies. Joining two or more networks together is known as internetworking. This means that the Internet is just an internetwork – a collection of interconnected networks.

How big or small are typical local area networks (LAN)? How many computers make up one network? I work at a university that has over 28,000

students and over 6000 full time staff members. Assuming that each staff member has a dedicated workstation, that department has some dedicated servers and then there are approximately 5,000 workstations on campus for students alone - that's approximately 12,000 computers in total.

That's 12,000 computers directly connected to the Internet. Chances are, these computers are not all centrally managed, nor are they encompassed by perimeter protection. They would utilise a variety of operating systems, hardware, software, protocols, with varying user education and security policies.

Now, imagine attackers choices for a moment. Wouldn't this sort of network be a great place to start? Different attackers have different motives. Typically a network becomes a target when it:

i) holds information that others would consider extremely valuable;
ii) displays weaknesses and vulnerabilities that others can take advantage of.

This paper presents a description of the major internal and external threats, along with their remedies. The target audience is any company that has a presence on the Internet.

Public Address Space

Because of the vast size of universities and the amount of computers used within them, a "Class B address" (e.g. X.Y.0.0/16) is generally assigned to them. This provides them with 65,025 possible IP Addresses. That's 255 [1-254] subnets and each subnet can have up to 255 [1-254] hosts. Which is just 255 x 255 = 65,025 (not including the highest IP address in any network, which is reserved as the broadcast address). IPv4 address space refers to the IP addresses that we commonly see today (e.g. a.b.c.d), which was managed solely but the Internet Assigned Numbers Authority (IANA), but is now delegated to various international registries.

It's unlikely that a university will ever have that many machines, but it makes it easier to allocate subnets to departments. It leaves plenty of room for expansion. A lot of other companies thought similarly and decided to register large amounts of address space. That's one reason why the IANA are running out of IPv4 address space (hence the introduction of IPv6). IPv4 allows the Internet to have up to 255 x 255 x 255 x 255 = 4,228,250,625 (approx 4.2 billion) IP Addresses. Considering there are six billion people on Earth and certain IP address ranges are reserved, you can now see why IPv4 address space is becoming a shortage. IPv6 not only provides more addresses, but also offers improvements like routing and network auto-configuration.

Companies are realising that they really don't need to use lots of public address space. Administrators are also realising that using public address space can open up your network to a variety of attacks and denial of service

threats. Lets face it, once computers are assigned IP addresses, they become a static entity on the Internet. Everyone knows who you are and how to get to you.

Instead of using public address space, companies can purchase a few IP addresses, have our LAN's assigned reserved or internal IP addresses and use Network Address Translation (NAT) techniques to connect our LAN to the Internet. Obviously network architecture (size of enterprise) and performance issues need to be considered. If a hacker does manage to compromise a system with an IP address of 192.168.1.2, they would be far more restricted in using that machine as a launch pad to attack other networks.

Routers and switches can be configured to drop packets with a source or destination IP address of reserved IP addresses. Firewalls are generally set up to prevent this as well. Also, an attacker cannot easily distribute information about this machine, because which 192.168.1.2 is it?

Another good strategy is to make your internal network use random reserved IP addresses. Use the 10.218.167.0/24 subnet for issuing IP's to machines. Make 10.218.167.39 your default gateway. Many administrators use conventional 192.168.1.{1|2|3} for internal servers and 192.168.1.254 for default gateways – this can be easy for an attacker who manages to find themselves on this sort of network.

Central DNS Services

Most companies have central servers for different Internet services – which is fine and probably make life easier for everyone. We must understand that these become targets if someone has a motive to attack your company. Generally a central server provides a service to all LAN users and needs to be connected to the Internet. Because this has a big presence and is a necessary resource for the day-to-day operations, when it fails it becomes a problem. Work ceases, staff get frustrated and cranky and companies stop making money.

Lets take a central Domain Name System (DNS) server for example. The DNS server would provide naming services for the entire university campus. If an attacker wanted to stop the university machines from connecting to other (internal or external) machines, they would probably try to compromise the central (primary and secondary) DNS servers.

By the way, if you wanted to try and stop the Internet, this is probably a good place to start (bring down the DNS root servers). This would eventually prevent networks from communicating with each other. On a side note – what if this really happened? Would the Internet stop? It is certainly possible! Lets say it did happen; people would panic and start to use the telephone. But how many people use the Internet today? According to NUA, there are over 400 million on-line users [1]. Are there enough phone lines to

let all these people make phone calls?

Back to our central DNS scenario, we've just seen how an attacker can try to stop a network from functioning properly by trying to break DNS. The most common DNS software used today is Berkeley Internet Name Domain (BIND). According to Computer Emergency Response Team (CERT), BIND is one of the most vulnerable services on the Internet today [2]. Because DNS is relied upon globally, BIND is a potential target. Many releases of BIND contain exploits – some resulting with remote root compromisation.

Some techniques to help protect yourself against DNS attacks are to:

i) run the named daemon as a non-privileged user;
ii) install BIND in a chrooted environment;
iii) get some documentation about securing a DNS server (A Linux reference is http://www.linux.ie/articles/tutorials/dns-tsig.php [3] and a Solaris reference is http://www.everythingsolaris.org/articles/dfd/frameset.html [4])
iv) do not run or keep the BIND source code on any other machine that should not be doing DNS duties;
v) keep a look out for improved DNS protocols including DNSSEC and DNSsafe (http://www.toad.com/~dnssec/)

Central E-mail Services

An extremely common and important communication service found on the Internet today is e-mail. Larger enterprise networks are now using "messaging" type features, which are built on top of existing e-mail services. A common e-mail server/client software package is Microsoft's Exchange server and Outlook client. This product can provide excellent benefits to companies but is also renowned for actively propagating destructive e-mail viruses.

Today we see harmful attachments that are executable within the Outlook client. That means the attachment simply runs when the user reads that e-mail message. Recently, infected attachments have been configured to e-mail everyone in that victim's address book This can be very bad if your company uses public address books or folders – imagine how fast the virus could infect so many users. Do you think these people are alerted to the virus beforehand? Probably not. Usually the virus starts somewhere, spreads rapidly and then security analysts discover and document it. This process can take days, weeks or even months before any virus is discovered. So waiting for the patch or mail server filter to come out isn't always going to prevent you or your company from being infected.

A classic example is the recent W32 SirCam virus. I was first alerted to this virus when two strange, but similar e-mails arrived in my mailbox one morning. Searching the Internet for other occurrences only revealed discussions about it and still no advice for my local AusCERT, SANS or Anti-Virus vendor. So I took it upon myself to examine the e-mails and

attachments. I determined that the only common thing about them was the attachment. The subject was different, the body of the e-mail was in a foreign language and the filename extension of the second attachment was not familiar. I made a mail filter that scanned for the first few bytes of the attachment. To my amazement, within 24 hours I had blocked over 35 (virus infected) e-mails from various domains. After 48 hours, this number increased to over 110.

According to securitytracker.com, Microsoft recently reported vulnerability in the Outlook Web Access (OWA) service of Microsoft Exchange Server (5.5 and 2000) that may let a remote user execute malicious code on the OWA user's web browser [5]. This code could manipulate the user's e-mail (e.g. adding, modifying or removing data in the mailbox). This can be disastrous.

And if you think it's just Exchange with problems, think again. Sendmail, a popular Unix Mail Transfer Agent (MTA), is not without problems. Over time, Sendmail has had a combination of local and remote exploits throughout its versions. In Red Hat 6.2 (Linux 2.2.14-5) the version of Sendmail came with a local buffer overflow, which if executed correctly gave the local user a root shell. With some earlier versions of Sendmail, "intruders may be able to force Sendmail to execute arbitrary commands with root privileges" [6]. Some other versions provided an open mail relay with the default install.

Finally, remember that e-mail will always get to the mail server – even with a firewall in place. So your last line of defense is Anti-Virus software. Make sure you have console Anti-Virus software on every machine (including laptops) and associated with mail clients. It's also vitally important to make sure it's kept up-to-date. Your mail server should have Anti-Virus scanning abilities too – this will help detect viruses before they get delivered to the user's mailbox or spool folder.

Departmental Web Services

Typically in a large company each department does a different job and may have a different web site. A university is no exception. There are many departments and most of them have a different web server to meet their needs. Web services are a vital part of any company's presence on the Internet. Without them, some companies would cease to operate (e.g. www.amazon.com). So doesn't that make web servers a big target? It sure does – especially when there are so many on the Internet today and the number is growing at phenomenal rates.

According to Netcraft, there are two major web servers that operate today, Microsoft Internet Information Server (IIS) and Apache [7]. They have a web server market share of 25.87% (over 8 million) and 58.74% (over 18 million) respectively. That's many machines running with known HTTP Port 80 open and waiting for connections. There are two important security issues to note with these web servers, As of June 23 2001:

i) "Microsoft have released 21 patches for IIS 5.0 alone and this

number is increasing at a rate of 1 every 3 weeks" (insiderreports.com) [8];

ii) "The last serious hole in Apache was reported and fixed in January 1997 and since then the only Apache security holes have been Denial of Service and unauthorized listing of filenames" (insiderreports.com) [8].

If an administrator needs web server software, the choice is clear. Consider Apache like everyone else and for the above (and very important) security reasons. But why do over eight million web servers run IIS if it has so many problems with it and will continue to do so?

To start with IIS is very easy to install and setup – it comes with the Windows NT and 2000 operating systems. Should administrators consider migrating from IIS to Apache? Managers might say there is a cost involved – well software wise, no. Apache is free and according to the Apache Software Foundation it will continue to remain a non-profit organization [9].

Apache doesn't install a lot of extra programs like IIS does. Installing Apache gives you a web server and not much else. By default, Windows 2000 and IIS install seven external Dynamic Link Library (DLL) files plus FrontPage server extensions. Every one of these eight components has had security updates since Windows 2000 was shipped.

If you choose to install extra Apache components, they are run as a non-privileged user. If a buffer overflow was to occur an attacker could only do minimal damage. Microsoft IIS allows "System" level access, thereby potentially granting Administrator rights. Any user, local or remote, who was able to get these elevated privileges could access, modify and delete files anywhere on the machine. Lastly, Apache gets all of its configurations from one file, http.conf. Microsoft IIS gathers configuration data from many files. This makes Apache easy for webmasters to maintain and manage.

Operating System Choice and Patch Distribution

Today there are so many operating systems to use and choose from. The one that is right for your company is dependent upon many issues, including: skills, administrators, knowledge, hardware, software and business requirements. So how do you choose which one to use? Do you select an operating system because other companies are using it? Possibly, but it's important to keep in mind that ALL operating systems have security problems and will continue to do so. This means that an operating system vendor should always promptly release patches to its customers when a security problem is identified and fixed.

How fast do you expect the patch to become available? How will you gain confidence that the patch you download is the correct one? Which site is the authoritative site? These are just some of the questions an administrator should ask when choosing an operating system. Patch distribution and

application are extremely important and play a vital role in the Internet today. Do you really think that every Windows 2000 machine currently has Service Pack 2 installed on it (correct patch requirement at time of writing)? Similarly, do you think most Unix machines have patched the recent telnet buffer overflow vulnerability that may give a local use root privileges? Definitely not.

So how does the external attacker try to break into machines? One certain way is to scan your network to determine what type of operating systems you are using. Once identified, a hacker could search the vendor's web site for known problems with that operating system. Then the attacker can do further research throughout the Internet to discover the malicious code or sequence of events that would provide them with elevated privileges. Sound easy? Well it certainly is and this is exactly what is happening daily.

According to CERT, more than 50 percent of vulnerabilities found in operating systems are due to buffer overflows [10]. A buffer overflow occurs when someone inputs more data into a field than the field expects. An article by Robyn Weisman describes a release of a virus that can infect Windows and Linux machines! Called W32.Winux, "It's dangerous but doesn't carry a highly destructive payload" [11]. Currently it may only affects machines with an Intel Pentium processor.

To make matters worse, once an attacker has access to your system, they could possibly install a "Root Kit" which enables them back door access to your machine for their own convenience. This root kit would also cover up any tracks left behind by the intruder – this makes it extremely hard to catch them now!

Secure Communications

At some point in time we need to make connections to other machines within our network and outside our network. Sometimes we, committed workers dial up from home after hours and connect to machines back in the office. Currently, many computer users are performing business functions on-line (e.g. Internet banking, Stock broking, etc.). Every time we accomplish one of these tasks, we are establishing a connection with a remote machine. We are authenticating to their host or application based environment. But what exactly are we authenticating? Who is verifying the integrity of the connection? Worse yet – is someone eavesdropping upon your Internet connection?

Many people believe that the small gold padlock in your favourite browser's status bar is a sign of secure communications and "Trust me - everything will be alright." This false sense of protection is becoming vastly exploited and the repercussions are causing more inconvenience to all Internet users.

Today we find ourselves using network protocols (e.g TCP) that were basically designed without security in mind. Most modern network services

use a simple TCP or UDP connection. Some examples are:

i) Telnet
ii) R Services (rlogin, rsh, rexec, etc.)
iii) POP
iv) IMAP
v) FTP

All the above services allow users to make insecure connections to remote machines. This means that all authentication tasks carried out are done in plain text over the network. Thus allowing a user to "sniff the network" or perform "session hijacking" and possibly discover your account details. Session hijacking is when an attacker steals your IP address and pretends to be you. Thus all incoming data destined to you is now going straight to the attacker. Has your computer connection to the Internet ever just stopped, without notice or warnings? This could be a case of someone hijacking your session.

One way to prevent this from happening is to use newer protocols that achieve the same result as the existing ones, but in a secure fashion. That is, an environment where a third party cannot intervene. Common protocols are called Secure Shell (SSH) and Secure Sockets Layer (SSL). These attempt to provide users with confidence that the data they are transmitting will not be disclosed to another party. Does this mean that it will guarantee the correct recipient is receiving your data? Is the integrity of the encrypted session verified? Does it mean that my sensitive data is protected at all times? No, no and no are the respective answers. Is it really secure then?

These are some of the known problems with SSH and SSL today. Here's a tip, next time you see an on-line site requesting credit card information, view the source and see what is there. If the form uses the "GET" method, you can forget it – any data you submit will just be passed along as the query string (and you will see your credit card details as a part of the URL). If the form uses a relative link, to call a script that processes your credit card details, e.g.:

```
<form method="POST" action="/card_details.cgi">
```

You should make sure this is a secure server site and verify the site's certificates. If the above link is absolute and points to an IP address, be very suspicious. The form syntax should look like:

```
<form method="POST" action="http://www.some-shop.com/cgi-bin/card_details.cgi">
```

"Most SSL attacks involve fooling the user, rather than breaking technology" (Seifried) [12]. SSL certificates contain various pieces of information that the end user should read and validate for their own piece of mind. Many Internet companies today used signed certificates to prove their authority. The certificate of information is signed by the secret key of a trusted third party (e.g. VeriSign). The trusted third party's key is usually built into software like web browsers. Assuming the user has trusted web browser software, they

can verify that the certificate the server sent them is valid and belongs to whom it claims to.

But as Seifried points out, there is one missing part to the equation, "While SSL requires the server to authenticate to the user, it is usually only an option for the user to authenticate with the server" [12]. And since not many users have personal certificates, we seldom see a user prove their identity to a server – thus leaving the connection open to attack.

This problem exists with SSH too. Generally, to establish a secure, encrypted connection over the Internet, you must initiate the connection over a public and potentially hostile network(s). Usually with public key encryption, two parties exchange their public keys. "Because this is done over a public network, it is possible for an attacker to intercept the key exchange and subvert it" [12]. This is better known as the man in the middle attack. SSH 1.x, SSL 1.0 and SSL 2.0 are all vulnerable to the man in the middle attack.

TCP and IP

The Transmission Control Protocol (TCP) and Internet Protocol (IP) are two very common and extremely important communications protocols that exist today. IP can be considered as the glue that holds the Internet together. According to Tanenbaum, "IP was designed with internetworking in mind. Its job is to provide a best-efforts way to transport datagrams from source to destination, without regard to whether or not these machines are on the same network, or whether or not there are other networks in between them" [13].

TCP is a connection-oriented service, which means that it is responsible for getting the data from source to destination correctly. And there are many different ways, techniques and policies to do this. There are also many different problems that are associated with transmitting data from source to destination and it's TCP's jobs to resolve all of these as best as possible without error.

With all this in mind, it sounds like we have a very good combination of protocols that could provide us with top quality of service. And with over four billion people on the Internet today – lets give the designers credit, because I think they got it right. But these two protocols lack one important property, security. Twenty years ago, no one would have expected packet sniffing to be a cause for concern. And who would want to hijack other people's sessions by predicting TCP initial sequence numbers? Similarly, no one in their right mind would try to get through firewalls by altering one of the TCP header six 1-bit flags.

This is now the year 2001 and where there is a potential for someone to exploit a service or gain control of assets, it will become reality. Bring in IPSec (IP Secure) – the new protocol for eliminating our network security

problems. This is designed to prevent network sniffers, session hijacking and server spoofing. Also, DNSSEC (DNS Secure) will be introduced soon which is designed to prevent domain name spoofing and redirecting people to fake websites.

IPSec is being migrated now into IPv4 LAN's in gateway-to-gateway environments. This will prevent data being transmitted without being encrypted and authenticated first. This does involve time, installing IPSec software to gateways and possibly certificate based authentication. Don't forget that the process of encrypting and decrypting packets will take time and I assume decrease network performance. Servers are bound to need hardware crypto accelerators; otherwise we would see servers falling over from not being able to handle loads.

These cards are likely to come at a medium to high cost depending upon functionality. Operating system compatibility will be an issue for administrators. Finally, there are vendor extensions to IPSec that can be confusing at first. These extensions are designed to provide better functionality and features, but are going to be different from vendor to vendor and are unlikely to be compatible across the board. This problem will (unfortunately) slow down the widespread deployment of IPSec.

IPSec is still to be proven and I imagine it will be another decade at least before the majority of the Internet is using it. There are still various methods for ways to authenticate IPSec connections, including Pre-shared secrets, X.509, PGP and other commercial solutions (e.g. Indus Rivers). Each of these has its limitations. So we must be extremely careful when going down this track. The last thing you want to do it replace or undo your IPSec infrastructure later on.

To finish off, lets talk about TCP and its security. When we make a TCP connection, we send many packets back and forth. And we use sequence numbers to differentiate who is talking to who and the order of communication. These sequence numbers usually increment by some constant amount. This should straight away ring bells - if an attacker can guess or predict the Initial Sequence Number (ISN), they could possibly manipulate or spoof TCP connections. This is not true for all operating systems (Linux isn't vulnerable to this attack), but you should consult CERT Advisory CA-2001-09 [14] to see if your systems are affected.

Attacks against TCP ISN generation have been debated before and in reality, these attacks led to the development of Pseudo Random Number Generators (PRNG's). Its job was to introduce some randomness while producing ISN's used in TCP connections. As Dave Wreski discusses, "Because of the implications of the Central Limit Theorem, adding a series of numbers together provides insufficient variance in the range of likely ISN values allowing an attacker to disrupt or hijack existing TCP connections or spoof future connections against vulnerable TCP/IP stack implementations. Systems relying on random increments to make ISN numbers harder to guess are still vulnerable to statistical attack" [15].

Berkeley-derived kernels are suspect to this because their ISN's are incremented by a constant every second and by another constant for each new connection. Systems using PRNG's were once thought to be safe, but as CERT Advisory CA-2001-09 [14] describes, some PRNG's are not completely random.

Wreski also points out that it only takes a couple of ISN samples to be collected by an attacker from a previous connection made to a victim's site. So what is the impact of all this? Well, if an attacker can guess or predict an existing ISN, they could choose to close or hijack sessions. If they can guess or predict a future ISN, they can complete a TCP 3 way handshake, establish a phantom connection and spoof TCP packets delivered to a victim. Jon Postel's RFC 793 describes ISN randomness, reliability and assurance [16] – this is a must read for anyone involved with network security.

Knowing what an attacker can do, how do we prevent it? In a nutshell, if the attacker can see unencrypted TCP traffic generated from a site, that site is vulnerable to various attacks. So the only definitive way to prevent this is to use end-to-end cryptographic solutions, such as ones provided by IPSec. Use of SSH and SSL are not fully protected. They operate above the transport layer, which means that can prevent arbitrary packets from being inserted into a session but they cannot prevent a connection reset (e.g. Denial of Service), since connection handling is done by a lower level protocol.

Conclusion

This paper has discussed external threats to network and system administrators. It seems like a never-ending battle at times and with various vendor specific advisories being released as often as daily, it's no wonder the administrators seem busy. This paper presents my view of common external threats to University networks (and other networks) and sample solutions. All solutions have advantages and disadvantages associated with them and require careful consideration and planning most times.

After reading this paper, you may wish to challenge the statement made by various international security bodies, "The majority of network and system breaches are internal ones." When upper management told me this was the reason for not allowing our department to deploy and administer a firewall, I almost fell unconscious. According to CERT, reported incidents are at an all time high and don't look like dropping [17]. When you hear that on average it takes just over 72 hours before a newly installed, default Red Hat Linux machine directly connected to the Internet is broken into – you start to wonder if it is safe to leave your computers on over the weekend while no one is at the office.

Overall there are many ways to obtain and maintain a presence on the Internet and each come with potential problems. There are so many choices

to make in regards to network security and system administrators will only learn with hands on experience. There is no simple guide to making your network secure because over time, things change. Technology changes, hardware and software changes, protocols change and more importantly society changes.

So secure networks is not just about skill and knowledge, it's about being vigilant and managing change. We need to keep our computer users aware of security threats and be proactive rather than reactive.

Remember, as a systems or network administrator you have the power to control your network and it's our duty of care to maintain an effective level of network and Internet security throughout our organizations on a daily basis.

References

[1] **How many Online?** [Online], Available: http://www.nua.ie/surveys/how_many_online [2000, Dec.].

[2] Lanza, J. P. et al, 2001, **CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND** [Online], Available: http://www.cert.org/advisories/CA-2001-02.html [2001, Aug. 07].

[3] Raferty, J., 2001, **Securing DNS with Transaction Signatures** [Online], Available: http://www.linux.ie/articles/tutorials/dns-tsig.php [2001, Jan. 17].

[4] Holve, M., 2001, **DNS for Dummies** [Online], Available: http://www.everythingsolaris.org/articles/dfd/frameset.html [2001, Mar. 02].

[5] **Microsoft Exchange Server's Outlook Web Access (OWA) Lets Remote Users Execute Arbitrary Code on the OWA User's Web Browser** [Online], Available: http://www.securitytracker.com/alerts/2001/Jun/1001696.html [2001, Jun. 07].

[6] Allman, E., 1997, **CERT Advisory CA-1997-05 MIME Conversion Buffer Overflow in Sendmail Versions 8.8.3 and 8.8.4** [Online], Available: http://www.cert.org/advisories/CA-1997-05.html [1997, Sep. 26].

[7] **July 2001 Netcraft Survey Highlights** [Online], Available: http://serverwatch.internet.com/netcraft/200107netcraft.html [2001, Aug. 24].

[8] Murphy, D., 2001, **Comparing Microsoft IIS And Apache HTTP Server** [Online], Available: http://www.insiderreports.com/storypage.asp_Q_ChanID_E_WB_A_StoryID_E_20001773 [2001, Jul. 23].

[9] **The Apache Software Foundation** [Online], Available:

http://www.apache.org [2001, Sep. 23].

[10] **Blame it on the Buffer overflows** [Online], Available: http://www.zdnet.com/zdnn/stories/news/0,4586,2801555,00.html [2001, Aug. 06].

[11] Weisman, R., 2001, **First Windows-Linux Combination Virus Surfaces** [Online], Available: http://www.osopinion.com/perl/story/8529.html [2001, Mar. 28].

[12] Seifried, K., 1999, **Special Kurt's Closet: Is SSL dead?** [Online], Available: http://www.securityportal.com/closet/closet19990930.html [1999, Sep. 30].

[13] Tanenaum, A. S., 1996, **Computer Networks 3rd Edition**, Prentice-Hall, United States of America.

[14] Havrilla, J. S. et al, 2001, **CERT Advisory CA-2001-09 Statistical Weaknesses in TCP/IP Initial Sequence Numbers** [Online], Available: http://www.cert.org/advisories/CA-2001-09.html [2001, May 10].

[15] Wreski, D., 2001, **CERT: Statistical Weaknesses in TCP/IP Initial Sequence Numbers** [Online], Available: http://www.linuxsecurity.com/articles/security_sources_article-2968.html [2001, May 05].

[16] Postel, J., 1981, **Transmission Control Protocol, Darpa Internet Program, Protocol Specification** [Online], Available: http://www.ibiblio.org/pub/docs/rfc/rfc793.txt [1981, Sep.].

[17] **CERT/CC Statistics 1988-2001** [Online], Available: http://www.cert.org/stats/cert_stats.html [2001, Jul. 18].