22

Controlling Internet Access at an Educational Institution

W. Olivier and H. van de Haar
Department of Information Technology
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth 6000
South Africa
Tel: 041-5043279
Fax: 041-5043313
e-mail: wernero@iaccess.za helen@ml.petech.ac.za

Abstract

Internet usage is spreading as widely and as densely, as the personal computer which has settled in the homes of millions of people all over the world. As television created invisible glued threads onto children's eyes in the past, so the Internet and its weblike access to exciting opportunities and worlds, is sticking like glue to the minds of today's youngsters. The educational institutions are not immune to all of this, and especially not the computer studies students. All day and every day, one can walk into the laboratories and find groups of students glued to the screen, the browsing facilities, and the downloading activities. This is both good and bad for the students. Being late for classes, missing classes altogether, disrupting practical laboratory sessions, using workstations which could have been used by other students who wished to do valid practical work, chatting and gathering around the workstations and so on, are everyday occurrences due to the arrival of Internet. A friendly solution was deemed necessary, one that would still allow access at certain times, yet curtail such access during valid practical sessions in relevant laboratories.

Keywords

Internet, controllin g access, TCP/IP, firewalls, sockets, packet filters

INTRODUCTION

This paper attempts to define a solution to a particular problem, that of controlling access to the Internet at certain times of the day from certain computer workstations in the laboratories at a typical educational institution. The task of finding a solution to this problem, was given to a fourth year student, who subsequently delved into the intricacies of the Internet, the TCP/IP protocols, firewalls, sockets and packet filters. The result was, that for the particular problem, in the particular given environment, there could be at least four solutions, some better than others. This paper details the research and the suggested solutions as investigated by the student.

STATEMENT OF PROBLEM

The Internet is a global network of interlinked computers, allowing users all over the world to communicate with each other. It is constructed in such a way that the technology hides the details of network hardware and enables computers to communicate even though their respective physical network designs are different. Most educational institutions are obviously linked to the Internet to allow access to this wonderful resource. However, allowing the students access can become a problem due to the magnetic appeal of having such vast realms of information available on the Internet. The students tend to disrupt practical sessions in computer laboratories and use computer facilities for 'playing on the Internet', thus removing the capacity for other students to conduct their own valid practical sessions in order to complete assigned tasks.

A BRIEF LOOK AT THE INTERNET

The Internet itself is totally decentralized, in that the machines and networks taking part are managed and paid for locally. The Internet network itself, the mesh of dedicated telephone lines that connect all of these networks, is owned by no one, but used by all. The Internet Society (ISOC) is a voluntary non-governmental international collection of researchers, academics and users who determine the survival and future of the network. They cooperate and coordinate networking technologies and applications for the Internet and are bound by a common stake in maintaining the viability and global scaling of the Internet. Within the Society is the Internet Architecture Board, the IAB. The main function of the IAB is to maintain the Internet through the creation and enforcement of international networking standards, as well as to make sure that no two users have the same Internet address (Carvin) (Internet Society, 1995).

The Internet began with the birth of the ARPAnet in 1969. Commissioned by the U.S. Department of Defense, ARPAnet was a communications network which allowed computers at separate locations to communicate with each other in order to exchange military and national security data. With this new technology, the data from one computer could be formatted into an electronic bundle or packet and then

addressed to another computer by way of the ARPAnet. This method of sending and receiving electronic information became known as the Internet Protocol, or IP for short. If a computer had the IP software implemented, it could in theory communicate to any other computer in the world, as long as that other computer had similar IP software and was on the ARPAnet (Carvin).

The rules formulated are officially named the TCP/IP Internet Protocol suite. This protocol is used by many organizations including the Department of Defence, National Aeronautics and Space administration (NASA) in America. After this initial development of the ARPAnet, additional networks branched out from defense research to general, scientific and academic use. Universities and research groups began to develop smaller networks specific to one site known as Local Area Networks, or LAN's. A LAN would have the ability to interconnect all the computers in a building using the correct network communication software. Using IP software, however, a LAN could connect with other LAN's, in other words a network within a larger network which formed the basis for the Internet. The National Science Foundation created another network, called the NSFNET, which would allow researchers and scientists to access their supercomputers by means of high speed phone lines. The NSFNET was so useful that very quickly other universities around the USA began to connect to the NSFNET. The BITNET is another network used by universities (Carvin). The building of networks throughout the country, using TCP/IP, in universities, government institutions and private industries, was happening at an unbelievable rate. TCP/IP was very popular, not because it was considered the best method for shipping data from computer to computer, but because of the fact that it was one of the first proven methods for delivering data. Today, this international network of networks is known as the Internet, and it is the most popular computer network in the world (Carvin).

A CASE STUDY NETWORK AND INTERNET SERVICE

Currently, the network setup for the case study problem at a South African educational institution, consists of multiple Novell 4.1 and Unix servers. All the workstations can access the network via DOS and Windows 3.1. Two of the Novell servers and one of the Hewlett-Packard unix servers, are used to house all the students' applications. To access Internet, the students may use Trumpet Winsock (written by Peter Tattum), which is available on any workstation. This Winsock software allows any Winsock compliant Internet software to access the Internet via Windows 3.1. The organisation connects to the Internet through UNINET, which is the local backbone network to which all universities and technikons in South Africa are linked. Routing on to the Internet takes place via a Cisco router.

The students may start telnet sessions to the unix machines, if they have a valid unix account. They have access to World Wide Web browsers, ftp, mail, chat, freetel and any other facilities that they manage to acquire. The main problem identified at the case study site, is in controlling the students' Internet access from the practical laboratories. The required level of control depends on the time of day and the type of access, since some practical sessions do require Internet access to continue. It is not

viable to restrict access completely, because the students do benefit from the knowledge obtained by browsing the Internet. It is, therefore, envisaged that the restrictions will apply only to certain times of the day, and must be automatically and dynamically altered by some controlling mechanism.

A BRIEF LOOK AT TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) has become the standard communications protocol for the Internet since its creation by the US Department of Defence (DOD). For machine connectivity, one could call it the software solution. On the battlefield a communications network would have to sustain damage, so the DOD designed TCP/IP to be robust and automatically recover from any node or phone line failure. This design allows the construction of very large networks with less central management (Gilbert).

TCP/IP is not a single protocol as its name suggests, but rather it is a collection of related protocols designed to provide the ability to transfer information across a network and includes the provision of information about the network itself. The collection of TCP/IP programs (protocols) enables the user to send email messages, transfer and share files and the remote execution of applications across both LANs and WANs. One of the most important aspects of this software solution, is that it allows communications between heterogeneous computers and operating systems used on the Internet. Unix, VMS, Macintosh, Intel based personal computers and others can talk to each other, regardless of the differing hardware of the machine. The most common hardware solution is Ethernet, but TCP/IP will also run on Token Ring and Serial lines (modems, serial connections) and other systems as well. For a full installation of TCP/IP, one will need a hardware driver, a TCP/IP stack and the TCP/IP applications themselves (TCP/IP).

On Macintosh systems, the hardware drivers are built into the system or are provided by the board manufacturer. On a personal computer system, there are different types of hardware drivers available, both commercially and via public domain/shareware including the Packet driver specification by FTP Software, Inc., Microsoft's Network Device Interface Specification (NDIS), and Novell's Open Datalink Interface (ODI). Drivers for OS/2 systems are available from IBM and/or the board manufacturer (if they support OS/2).

The TCP/IP stack is package specific and usually comes with every product. Each such stack has its own requirements for hardware drivers. One has to find a combination of driver and TCP/IP stack which is compatible with the rest of the environment. Personal computer systems have something close to a standard in TCP applications called the Windows Sockets API (Winsock). (Note: This is not specific only to TCP/IP but it is a general standard for networking on personal computers irrelevant of the transport protocol.)

One would wish to have all the TCP/IP application programs such as Telnet, FTP, mail, etc. Just about every TCP/IP package has a corresponding set of applications but perhaps not every TCP/IP package contains all the different applications that are available.

The Transmission Control Protocol (TCP) part of TCP/IP treats the data as a stream of bytes. It logically assigns a sequence number to each byte because it is responsible for verifying the correct delivery of data from client to server. The TCP packet has a header that says, for example, that the packet starts with byte 532456 and contains 200 bytes of data. The receiver can detect missing or incorrectly sequenced packets. TCP acknowledges data that has been received and retransmits data that has been lost (TCP/IP).

Simply put, the Internet Protocol (IP) is the Internet's universal method of addressing and forwarding data from node to node. Every computer on the Internet has its own address, which is a series of four numbers each below 256, such as 101.231.03.56. This is called the IP number or IP address. The Internet authorities assign ranges of numbers to different organizations who in turn, assign groups of their numbers to departments. IP operates on gateway machines that move data from department to organization to region and then around the world. When a user sends data to another user, such as an email message, IP transmits the data in snippets of information known as packets (packet transmission is much faster than sending one's data as a single chunk). TCP/IP creates and uses what is known as a checksum to ensure correct ordering of packets (Internet Society, 1995).

Every time a message arrives at an IP router, it makes an individual decision about where to send it next. Traffic can be routed by the 'clockwise' algorithm, or the routers can alternate, sending one message the one method and the next by the other method. More sophisticated routing methods measure traffic patterns and send data through the least busy link. If one phone line in this network breaks down, traffic can still reach its destination through a roundabout path. This kind of recovery is the primary design feature of IP, and provides continued service though with degraded performance. The loss of a line is immediately detected by the routers, and somehow this information is sent to the other nodes. Each network adopts some router protocol which periodically updates the routing tables throughout the network with information about changes in route status.

There are three levels where knowledge of TCP/IP intrinsics become important. Those individuals who administer a regional or national network must design a system of long distance phone lines, dedicated routing devices, and very large configuration files. They must know the IP numbers and physical locations of thousands of subscriber networks. They must also have a formal network monitor strategy to detect problems and respond quickly.

Each large company or university that subscribes to the Internet must have an intermediate level of network organization and expertise. A half dozen routers may be configured to connect several dozen departmental LANs in several buildings. All traffic outside the organization will typically be routed via a single connection to a regional network provider.

However, the end user can install TCP/IP on a personal computer without any knowledge of either the corporate or regional network. Three pieces of information are required:

- the IP address assigned to this personal computer ;

- the part of the IP address (the subnet mask) that distinguishes other machines on the same LAN (messages can be sent to them directly) from machines in other departments or elsewhere in the world (which are sent to a router machine) ;
- the IP address of the router machine that connects this LAN to the rest of the world (Cedeno and Osborn, 1996) (Comer, 1991).

A BRIEF LOOK AT SOCKETS

Sockets is a name given to the package of subroutines that provide access to TCP/IP on most systems (Gilbert). WinSock is short for Windows Sockets. Today's most popular Internet applications for Microsoft Windows and IBM OS/2 are developed according to the WinSock standard. Berkeley Sockets is the standard programming model for TCP/IP networking under Unix. Windows Sockets was actually designed to be very similar to Berkeley Sockets so that those experienced in programming with sockets in Unix will be able to easily make the transition to Windows Sockets. WinSock is a .DLL (Dynamic Link Library) and runs under Windows 3.x, Windows for Workgroups, Windows NT, and Windows 95. The WINSOCK.DLL is the interface to TCP/IP and, from there, on out to the Internet. WINSOCK.DLL actually acts as a layer between the WinSock applications and the TCP/IP stack. The WinSock applications tell WINSOCK.DLL what to do, WINSOCK.DLL translates these commands to the TCP/IP stack, and the TCP/IP stack passes them on to the Internet (Cedeno and Osborn, 1996).

A BRIEF LOOK AT ETHERNET

Ethernet is one of the most popular network cabling schemes in use. The original ethernet specification was developed by Xerox. A second version (Ethernet II) was made with the efforts of Digital Equipment Corp., Intel, and Xerox. The Institute of Electrical and Electronics Engineers (IEEE) standardized a separate form of ethernet which has come to be known by the standards document number: IEEE 802.3. Both Ethernet II and IEEE 802.3 are compatible on the same wire so hardware utilizing either can work in the same network. Both these standards also specify a hardware protocol which describes each 'frame' of data. Ethernet hardware use CSMA/CD (Carrier Sense Multiple Access/Collision Detection) which says that only one machine on the ethernet can speak at any one time and if two or more try to do it at once, the packet frames sent will collide and the machine has to resend the frame of data at a later time.

Ethernet is a hardware and data link specification. Other software network protocols run above this such as IP, IPX and NetBEUI etc. In turn, other protocols can run over those: TCP & UDP over IP, SPX over IPX, etc. So TCP/IP will work fine with ethernet and this is also how the problem case study network is set up.

Personal computers and Macintoshes connect to an ethernet via a network interface card, which fits into the machine's bus (eg. ISA or PCI for personal computers) and require a network driver to function (Gilbert).

FIREWALLS

Packet filters are exactly what their name says: devices that filter the packets moving across a certain point in a network. Packet filter applications and mechanisms have become known as firewalls. A network firewall has the job of keeping unwanted visitors away from the network. Firewalls are therefore an excellent way to control Internet access on a network. The actual mechanism whereby this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one which exists to block traffic, and the other which exists to permit traffic. Some firewalls place a greater emphasis on blocking traffic, while others emphasize permitting traffic (Ranum, 1995).

A firewall can also act as the corporate voice to the Internet. Many corporations use their firewall systems as a place to store public information about corporate products and services, files to download, bug fixes, and so forth. Some firewalls permit only email traffic through them, thereby protecting the network against any attacks other than attacks against the email service. Other firewalls provide less strict protections, and block services that are known to be problems. More elaborate firewalls block traffic from the outside to the inside, but permit users on the inside to communicate freely with the outside (Ranum, 1995). Firewalls provide a single point where security and audit can be imposed. In a situation where a computer system is being attacked by someone dialing in with a modem, the firewall can act as an effective tracing tool.

There follows a definition of three basic types of firewalls: packet filters, circuit level gateways, and application gateways. Of course there are also hybrid firewalls which can be combinations of all three.

Packet filter gateways are usually comprised of a series of simple checks based on the source and destination IP address and ports. However, there is no way for the filter to securely distinguish one user from another. Packet filters are frequently located on routers and most major router vendors supply packet filters as part of the default distribution. Smart packet filters are really not very different from simple packet filters except they have the ability to interpret the data stream and understand that other connections which would normally be denied should be allowed. Smart packet filters, however, still cannot securely distinguish one user on a machine from another.

Circuit level gateways are much like packet filters except that they operate at a different level of the OSI protocol stack. Unlike most packet filters, connections passing through a circuit level gateway appear to the remote machine as if they originated from the firewall. This is very useful to hide information about protected networks. Socks is a popular de facto standard for automatic circuit level gateways.

Application gateways represent a totally different concept for firewalls. Instead of a list of simple rules controlling which packets or sessions should be allowed through, a program accepts the connection, typically performs strong authentication on the user which often requires one time passwords, and then often prompts the user for information about the destination host. However, for most environments it provides

much higher security because unlike the other types of gateways, it can perform strong user authentication to ensure that the person on the other end of the IP connection is really who he/she says that he/she is. Additionally, one can perform other types of access checks on a per user basis such as what times they can connect, what hosts they can connect to, what services they can use, etc. Many people consider application gateways to be the only true firewalls, because of the lack of user authentication in the other two types.

Hybrid gateways are ones where the above types are combined. Quite frequently one finds an application gateway combined with a circuit level gateway or packet filter, since it can allow internal hosts unencumbered access to unsecured networks while forcing strong security on connects from unsecure networks into the secured internal networks (Ranum, 1995).

Application level or proxy type of filtering. The main principle of an application level filtering firewall is, that it blocks all IP level traffic between the internal network and the Internet. No IP packet from the internal network will ever reach the Internet and no IP packet from the Internet will ever travel the internal network. It therefore avoids much of the security related problems of the IP protocol which was not built with security in mind. The principle of a proxy based firewall is, that an internal client connects to the firewall and talks to a server on the firewall and not (directly) to the server on the Internet. This server on the firewall is called a proxy. The proxy on the firewall understands the client/server protocol and acts as an intermediate: when it decides that the client is allowed to do a certain type of operation, the proxy on the firewall connects to the server on the Internet and will execute that operation on behalf of the client. The filtering and screening of a proxy can be threefold.

- IP level information: source address, destination address, destination port, in fact the same type of information an IP level filtering firewall is filtering on.
- Additional authentication information: the client can be prompted for a user name and a password before the proxy allows a client to do something. Because user names and static passwords are dangerous to use on the Internet (passwords travel unencrypted on the Internet), more secure mechanisms can be used: challenge/response mechanisms using a dongle.
- Screening on the client/server protocol itself: sometimes the client is allowed to use the proxy in a limited way. For instance, the client may use an ftp proxy only to import files, or an http proxy which denies general clients access to private html pages and only allows privileged clients to get them (Bellovin and Cheswick, 1994).

The proxy type of firewalls are considered to be the most secure. However, there are complications and disadvantages. There is no general proxy: a proxy type of firewall runs a telnet proxy, an ftp proxy, an http proxy and so on. A proxy is, in general, a complex piece of software which is specifically designed for a certain type of client/server protocol.

There are advantages to be gained when using a proxy type of firewall. In principle they offer the highest level of security. It is not necessary to worry about security holes in the IP protocol since the firewall blocks all IP traffic between internal

network and Internet. Also they allow for screening on application level. Sometimes a proxy can do more than offer security. In fact only a very limited Domain Name System (DNS) zone can be run (on the firewall).

Firewalls cannot protect against attacks that do not go through the firewall. Many corporations that connect to the Internet are very concerned about proprietary data leaking out of the company through that route. Firewall policies must be realistic, and reflect the level of security in the entire network. For example, a site with top secret or classified data should not be hooking up to the Internet in the first place. To set up a firewall, one has to first decide if it reflects the policy of how one's company or organization wants to operate the system.

DOMAIN NAME SYSTEM ISSUES

Some organizations want to hide Domain Name System (DNS) names from the outside. This approach is one of many, and is useful for organizations that wish to hide their host names from the Internet. The success of this approach lies upon the fact that DNS clients on a machine do not have to talk to a DNS server on that same machine. In other words, just because there is a DNS server on a machine, there is nothing wrong with (and there are often advantages to) redirecting that machine's DNS client activity to a DNS server on another machine.

First, one sets up a DNS server on the host that the outside world can talk to, such that it claims to be authoritative for one's domains. In fact, all this server knows is what one wants the outside world to know: the names and addresses of the gateways and so forth. This is the 'public' server.

Then, one sets up a DNS server on an internal machine. This server also claims to be authoritative for one's domains. Unlike the public server, this one is telling the truth. This is the 'normal' nameserver, into which one puts all the 'normal' DNS stuff. One also sets this server up to forward queries that it can not resolve to the public server (using a forwarders' line in /etc/named.boot on a UNIX machine, for example).

Finally, one sets up all the DNS clients (the /etc/resolv.conf file on a UNIX box, for instance), including the ones on the machine with the public server, to use the internal server. This is the key.

An internal client asking about an internal host asks the internal server, and gets an answer. An internal client asking about an external host asks the internal server, which asks the public server, which asks the Internet, and the answer is relayed back. A client on the public server works just the same way. An external client, however, asking about an internal host gets back the 'restricted' answer from the public server.

POSSIBLE SOLUTIONS

Possible solutions will be sought on the server side as well as the workstation side.

Non Centralized, Non Firewalling Solutions (Windows 3.1 / 95)

There is a non firewalling solution that can be implemented on each workstation that needs to be controlled. Almost 90% of all Internet software at the case study site is Windows based. With this knowledge, it is possible to create a small application that will close down the offending application depending on the time of day. Programming languages such as Visual Basic and Visual C have the capability to access the Windows Task list. The task list is a list of applications currently running on the workstation. If the list of running applications is available it is very easy to determine if any Internet applications, such as Netscape, are active and simply shut the application down depending on the time of day. There are, however, quite a few inherent problems with this solution.

Firstly, students will very soon realize that they can still access Internet by merely changing the system time within the control panel of windows itself. The solution is to update the time of the workstation from an Internet time server. This application connects to an Internet Time Server that has the correct time and updates the workstation's time accordingly.

Secondly, in Windows, nothing stops a student from merely closing down the application which checks what Internet software is running. The way to overcome this in the coding, is that there is a procedure saying that if the application is being shut down, to start up a new instance of the application. This is also very easy to accomplish in Visual Basic.

Thirdly, this Internet checking application must start up every time Windows is run. The normal way to do this is by creating an icon for it and placing a copy of the icon in the startup program group. Any program that has an icon in the startup program group will be run when windows starts up. Another place to start an application from within Windows is with a run command in the win.ini file.

However, a knowledgeable student may figure out which application is prohibiting him/her from accessing the Internet and from where it is being activated when Windows starts i.e. in the startup program group or win.ini file. Once this is known it is very simple to remove the icon from the startup program group or to edit the win.ini file and remove the run command starting up the Internet checking program.

To stop this from happening, it becomes a bit more complex and possibly more costly. A very secure method will be to have all necessary installed software on the C drive with a separate partition for space where the students can save all their files. The C drive then has to be secured so that no files can be deleted from there. This is done by special software that 'locks' the hardware so that only users with the right password will be able to change anything on the locked C drive. So Windows, its win.ini file and the Internet checking program are safe from deletion and tampering by students.

The current method of implementation forces each laboratory to have its 'version' of the Internet checking program since different labs have different requirements on time restraints and also different applications to terminate. All the information for each lab can be stored in a database on the Novell network. When Windows starts up, the Internet checking program will access the database on the network and from there read all the information pertaining to the laboratory. All the application needs to know is in which laboratory it is running and this can be stored in a file on the C drive which is locked.

This method will force the student to log into the Novell network in order for the Internet checking program to access its database and allow or disallow Internet access. Forcing students to log in will enable future monitoring and logging of what students are doing on the network.

Non Centralized, Firewalled Solution (based on server)
The case study site will probably migrate to a Windows NT network, leading to more possible solutions. Each laboratory can have its own dedicated firewall workstation set up to filter out its Internet connectivity depending on the time of day. The rules for the firewall in the laboratory will be simpler since one will have to restrict only a certain number of workstations at a time e.g. only the number of workstations that reside in that laboratory. This solution has a few drawbacks. If a student disables the controlling machine in a laboratory by switching off the workstation, then the whole firewall is shut down and the laboratory has full Internet access, thus implying extreme measures to secure physical access to each firewall machine in each laboratory.

Centralized, Firewalled Router Solution
The case study site uses a Cisco router to forward Internet traffic. Cisco routers can be used to do basic packet filtering. Since the router is the single entry and exit point for all Internet traffic, this is the natural point to install a firewall. As said previously, the Cisco router software is capable of basic packet filtering. The main problem with this solution is that at certain periods of the day the rules file or script for the Cisco router has to change, as the usage rights of different laboratories change during the day. The current Cisco software is unable to perform this function.

It is theoretically possible to enable the Cisco router to do the rules file updating procedure needed, but that entails the installation of remote management protocols not in use at the case study site. The installation of such protocols will increase the maintenance effort of the network of an already over stressed administration department.

Centralized, Custom Firewall Solution
The final option and probably the more popular one, is to install a central server solely and exclusively as a firewall. In other words, it has to be positioned in such a manner that all Internet traffic on way to the router or from the router will pass through the firewall. The firewall will have more than one rules file covering all the various periods of the day when different rules have to be enforced. The firewall server will have a timer running to trigger at the right time of the day to copy the correct rules file to the firewall. This way the correct rules will always be available to the firewall.

This firewall server can also be integrated with a proxy server. The proxy server will intercept all packets outbound on the Internet and change the packets' source address to that of the proxy server. When the return packet arrives the proxy will forward the packet to the original sender within the local network. This way all external networks will only see the address of the proxy server and in this way the rest of the internal network is hidden away from the outside world. The firewall to control the laboratories can then reside on the proxy server.

The Cisco router can also be brought into this solution. If, for instance there are a few unsavory sites to be completely banned from the network such sites that have previously initiated hacking attacks against one's network, the Cisco router's filtering capabilities can be used to filter out these unwanted sites completely, while the custom firewall keeps control on the laboratories.

This entire setup has one inherent flaw. The filtering of the firewall is based on IP addresses that are software generated addresses by Internet. A student may, however, change his/her workstation's IP address and in this way bypass all the security measures installed. This is a very serious flaw that needs attention. The student may have discovered one of the IP addresses of the administration or lecturing staff who perhaps have no limitations on their Internet access and usage.

Even though Internet uses IP addresses to establish a link between sender and receiver, these are merely logical addresses. In fact, these addresses are converted to the physical hardware addresses of the network cards themselves. So obviously in TCP/IP there already exists something that can translate an IP address to its hardware (MAC) address. It is in fact called the ARP protocol. One can use the capabilities of ARP to sample all the IP addresses and hardware addresses of the laboratories which must be controlled. The Internet firewall controlling the access to the laboratories will need a small customized add on facility to house a list of all the IP addresses and hardware addresses of the workstations in the laboratories. Once every half hour, for example, the ARP protocol will be used to query all the workstations in the labs and compare the addresses returned by the query to those housed in the table.

If the results do not match up, a student has most likely changed an IP address illegally. The network administrator must have access to the table housing the IP and hardware addresses to update the table if a workstation configuration changes, e.g. a network card is replaced. If an address does not match up, the custom ARP application can very easily notify the network administrator via email.

CONCLUSION

This last solution seems to be the best way to go for the particular case study site. Various alternatives have been suggested, but not all are viable, due to wastage of dedicated equipment, and decentralized control, which in turn places more burden on administrative staff. A centralized solution will allow for easier administration, and will cancel out the possibility of tampering by students. If the case study site replaces their network operating system with a Windows NT networking environment, then the first idea of a decentralized controlling mechanism may become popular, especially in view of the fact that it will then be possible to lock out certain sections and directories on the hard drives. Preemptive multitasking is a feature which can well by used in the Windows NT environment.

REFERENCES

Bellovin, S.M. and Cheswick, W.R. (1994). *Firewalls and Internet Security: Repelling the wily hacker.* Addison-Wesley Publishing Company.
Carvin, A. *EdWeb: Exploring Technology and School Reform.* http://edweb.cnidr.org:90./ibahn.int5.html
Carvin, A. *Network Wildfire.* http://edweb.cnidr.org:90./ibahn.int4.html
Carvin, A. *NSFNET.* http://edweb.cnidr.org:90./ibahn.int3.html
Carvin, A. *Paving the First Path: The Internet.* http://edweb.cnidr.org.90./ibahn.int1.html
Cedeno, N and Osborn, K. (1996). *The alt.winsock FAQ (Frequently Asked Questions).* http://www.well.com/user/nac/alt-winsock-faq.html
Comer, D.E. (1991). *Internetworking with TCP/*IP. Vol 1. Prentice Hall.
Gilbert, H. *Introduction to TCP/*IP. http://pclt.cis.yale.edu/pclt/comm/tcpip.htm
Internet Society. (1995). *What is the Internet Society?* http://info.isoc.org:80/whatis/index.html
Ranum, M.J. (1995) *Internet Firewalls Frequently Asked* Questions. http://www.greatcircle.com/firewalls/info/FAQ.html
TCP/IP - Short Description. http://www.webpress.net/ib/ibm/tcpip.htm

BIOGRAPHY

Werner Olivier is a full-time student at the Port Elizabeth Technikon, and is currently studying towards his Masters Technical Degree in Information Technology, in the field of Information Security.

Helen van de Haar began her computing career in 1972 as a computer programmer, and has been writing programs in various languages ever since. She has a B.Sc from the University of Port Elizabeth and a Masters Diploma in Information Technology from the Port Elizabeth Technikon where she is a Senior Lecturer in charge of Operating Systems. She is busy working towards a PhD at Rhodes University, in the field of parallel and distributed processing and debugging.