A Secure Station for Network Monitoring and Control

Vassilis Prevelakis
vp@unipi.gr
Network Management Centre
University of Piraeus, Greece

Key words: VPN, IPsec, OpenBSD, network security

Abstract: The proliferation of computers has spurred the creation of large networks even in small organisations. These networks comprise great numbers of elements such as routers, switches, servers etc. located in multiple locations. The administration of these elements has to be carried out usually from a central location over the existing network infrastructure.

Starting from the premise that the organisation LAN or MAN cannot be assumed to be secure, we created a network of stations that communicate via a secure VPN. Each station provides a secure bridgehead into one or more remote parts of the network. From this bridgehead the administrators can monitor and control nearby network elements in a secure way.

In this paper we present the architecture of the monitoring and control stations that have been deployed within the University of Piraeus network. We also describe how such stations have been deployed in a pilot project for the management of the Greek University Network (GUNET). [†]

1. INTRODUCTION

The management of a complex multi-vendor network provides many challenges to the network administrators. The issue is too complex to be dealt within the confines of a single presentation, so we will concentrate our

[†] This work has been carried out under the auspices of the GUNET/EPEAEK programme which is jointly funded by the Greek Ministry of Education and the European Union.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35575-7_19

J. H. P. Eloff et al. (eds.), *Information Security Management & Small Systems Security*

attention to the aspects of establishing secure connections to the various active elements that comprise the network infrastructure.

Why should network administrators worry about secure connections to routers, switches and other network assets?

- **Security**: through the use of access lists (for routers) or VLANs (for switches) these elements play an important role in the implementation of the security policies of the organisation. If they are compromised they may serve as a bridgehead for attacks on other network assets [Garf96].
- **Points of attack**. In most cases attacks come from within. So the assumption that the internal network is safe from snoopers may be very optimistic [Chap95].
- **Damage control**. In cases where the network has been infiltrated, or some asset is under attack, the network administrators must intervene to limit the damage and perhaps identify the intruder. If their actions can be monitored by the hostile party, then the effectiveness of their manoeuvres will be diminished.
- **Peace of mind.** In most cases, network problems are not caused by hackers, but by other, rather mundane, reasons such as bad configurations, buggy software or hardware failures. Nevertheless, being able to quickly eliminate hostile action from the probable causes helps identify and correct the problem faster and with less mental anguish.

Given the above, it is not surprising that the need for effective protection of network elements has long been identified and a number of solutions have been proposed, such as:

- New, more secure protocols such as the SNMPv2 [RFC.1910] which offers a number of enhancements over the SNMPv1 regarding the integrity of the information that is exchanged between the managed network element and the Network Management Station.
- Replacing hubs with switches to reduce the possibility of wiretaps.
- Use of proprietary security features of the equipment. Since vendors can react to market demands faster that standards bodies, a veritable arsenal of software and hardware solutions has been made available to network administrators. One important category that is becoming increasingly popular is intrusion detection systems such as the ISS RealSecure (*http://www.iss.net*) [Fort99].

However, none of these approaches provide a complete solution. While various security features are now provided with new products, most organisations already have large installed bases of older kit that have not been designed with strong security in mind.

Another handicap, shared by organisations outside the U.S, is that it is difficult to get strong security out of the box on products that originate from the U.S. due to export restrictions on cryptographic mechanisms.

To make matters worse, even in cases where vendors offer security features and strong authentication for their products, these are usually applicable only to the product range of the particular vendor. Thus, the configuration and management of the security features is a constant headache for the management team. [Shah97]

2. REQUIREMENTS

About two years ago, the University of Piraeus upgraded its connection to the outside world to 2Mbps and initiated a programme for the complete upgrade of its networking infrastructure. One side effect of this decision was a temporary freeze on spending on new equipment since they would all be replaced when the new network became operational. Given the progress of government procurement in Greece, the new network is still under construction (but will become operational before the summer of 1999).

Faced with a networking infrastructure that could not be upgraded, the staff at the network management centre at University of Piraeus decided that instead of trying to make the network elements communicate securely with the central network management systems, it would be more cost effective to provide specially configured PCs that would act as mediators and provide the necessary security. The result was a number of vendor independent secure network monitoring stations, that were cheap and versatile so that they could be placed in various parts of our network and interfaced to different networking gear. The requirements for a machine of this class were as follows:

- Low cost, preferably constructed from parts taken from decommissioned PCs.
- Minimal administrative overhead. This implied easy configuration and no administrator intervention after installation. Moreover, the bulk of the work for the construction of the software distribution for the network monitoring station should be devoted to integration of

existing tools and packages, rather than the development of new code that would have to be maintained.

- Offer secure (encrypted) network connections with other similar stations and with the workstations of the network management staff.
- Be resistant to tampering. In cases where there are indications that the station has been hacked, its original configuration must be easily restored.
- Offer a standard platform for the execution of common network management and monitoring tools. It must also support the SNMP protocol.
- It must offer ways of establishing connections with network elements of various vendors for the purposes of administration and configuration.
- Finally, for troubleshooting purposes, it must be able to be deployed with minimal overheads in any part of the network.

In short, our intention was to construct something that could be used like meteorological balloons or sonar buoys: off-the-self and easily redeployable after use.

3. NETWORK MONITORING STATION

From the very beginning, the design team wanted a platform that could accommodate a large number of tools for network monitoring and management. The requirement that the station should operate in wiring closets without a monitor, keyboard or mouse effectively disqualified all Windows platforms. From the available UNIX or UNIX-like systems we eventually chose OpenBSD 2.3 for the following reasons:

- Like other free UNIX-clones, a large number of programs like tcpdump, snmpd, ssh, etc. are either supported in the base release or can be easily ported.
- The designers of OpenBSD have paid a lot of attention to the security profile of the system, creating a robust environment that is resistant to security related attacks. In fact, on the OpenBSD web site (*http://www.openbsd.org/goals.html*) it is claimed that OpenBSD passes Ballista's (now called Cybercop Scanner by Network Associates, *http://www.nai.com/products/security/cybercop_scanner*) tests with flying colours.

However, perhaps the most important consideration was that the system supports the transport layer security protocols (IPsec) that offer secure communication channels between stations. Since these channels are created by the networking code in the kernel, the encryption is transparent to applications. Thus, programs like rlogin that have no encryption facilities can take advantage of the built in security offered by IPsec without any modifications to the application code.

3.1 Examples of Use

In figure 1 we show an actual configuration where two remote network elements are located in the same LAN as a secure network station (SNS). The station has an encrypted tunnel with another station located in the NOC local network. A user working at a regular workstation (Windows or X11), initiates a telnet session with destination one of the remote elements. The packets flow through the local SNS and are encrypted. They then flow through the University LAN towards the remote SNS. At that point the packets are converted to cleartext and injected into the local Ethernet that links the remote SNS with the telnet destination [Oppl98].

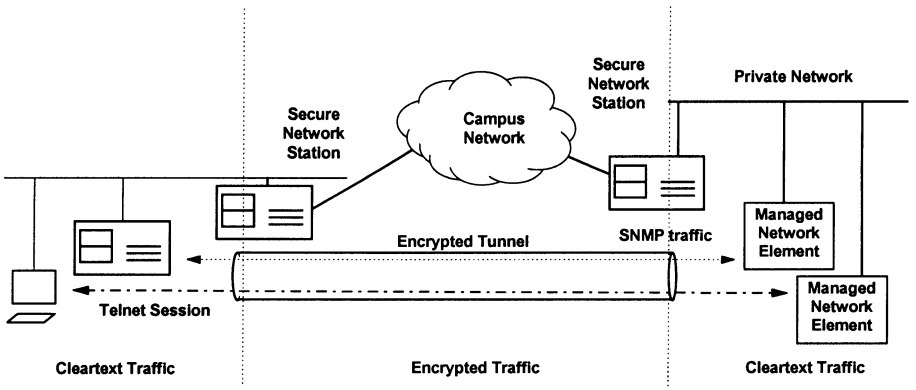

Figure 1. SNMP and telnet traffic passes via the secure tunnel

In a similar manner the SNMP traffic from the NOC NMS, gets encrypted and exits on the other side of the tunnel.

Note that the Ethernet segments used to connect the various elements with the SNSs are private in the sense that no one else may join them. For this reason it is be necessary to have more than one network interfaces on each managed asset.

In cases where there are network elements that lack a second Ethernet interface, the private LAN solution mentioned above cannot be used.

Instead, control is exercised via the serial console ports that are standard features on most network elements (see figure 2).

The remote SNS runs a process called the *console server* (conserver) that manages the local serial ports. Another process, the *remote console*, connects to the console server via TCP/IP and allows access to the serial port from any point of the secure network. Thus the system administrator wishing to configure a remote network element uses ssh to log on the local SNS node and from there runs remote console to connect to the serial port on the remote machine.

In this way, machines even in geographically remote locations can be fully reconfigured. Access to the console port gives the administrator numerous benefits, which include full administrative access to the remote system even if the network interfaces are down (as in the case of routers being reconfigured or UNIX servers running in single-user mode).

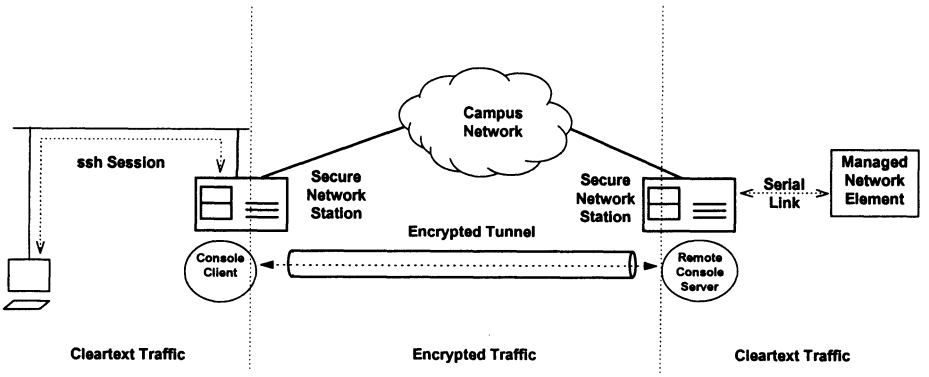

Figure 2. Serial ports on the Secure Network Station can be accessed remotely

3.2 System Architecture

The prime considerations in the design of the SNS has been simplicity and security. In this section we will elaborate on these two issues and examine their impact on the design of the SNS operating environment.

In order to satisfy the security requirement we used the following techniques:

3.2.1 Firewall

SNS nodes must be able to access external network devices while at the same time should allow only a very restricted set of incoming connections. On the other hand, connections from other SNS nodes must be accepted.

In the SNS design we have used the packet filtering functionality of the OpenBSD kernel with a configuration that blocked **all** packets **except**:

- IPsec packets, since IPsec has its own security mechanisms.
- Outgoing TCP/IP connections, to allow connections to network elements.
- Packets from established TCP/IP connections.
- Outgoing SNMP requests (UDP).
- Incoming ssh connections, to provide secure access to administrators from hosts outside the secure network.
- ICMP echo and reply messages but excluding the other ICMP control messages.

Given that there are no access restrictions within the secure network, we were extremely concerned about allowing access to the SNS nodes from outside workstations. When considering security mechanisms there is always a need to strike a balance between security and convenience. Making life difficult for the administrators would only mean that they would avoid using the secure network or find ways to disable or bypass various security mechanisms thus compromising the security posture of the entire network.

In the end we came up with two scenaria of use. One is more restrictive and specifies that access to SNS nodes is only allowed via the secure shell. The other scenario provides transit from the network administrator VLAN to the private networks via the secure network (see the telnet connection in figure 1).

3.2.2 Secure shell server

The secure shell (ssh) [SSH98] is primarily used as a secure replacement for the rlogin and rsh UNIX command. In other words, it allows secure access to remote hosts.

The secure shell system comprises of a server process (sshd) that runs on the remote stations and the client (ssh) that runs on the local workstation. The secure shell system offers secure (encrypted) connections and strong authentication (RSA with a the private key protected by a passphrase).

Ssh clients exist for both UNIX and Windows platforms. In the case of Windows, there exists a very comprehensive commercial version (*http://www.ssh.com*), but we chose to use the free version which is

distributed as a plug-in for the Teraterm free telnet program (*http://www.zip.com.au/~roca/ttssh.html*).

The secure nodes contain only the sshd server since they are only expected to accept connections and not initiate them. Users within the secure network use the normal (insecure) telnet over the secure links.

The use of the secure shell was mandated because on one hand we wanted to protect the communication path between the administrator's workstation and the secure network, while on the other we felt that running IPsec on the Windows platforms was not advisable: existing IPsec implementations for Windows platforms were not mature and the platforms themselves are full of security holes.

3.2.3 IPsec

IPsec is a series of protocols [RFC.1825] that aim to provide encryption, authentication and integrity checking at the network layer. The secure network employs IPsec in tunnel mode with encryption (ESP) [RFC.1827]. Tunnelling consists encrypting and then encapsulating a normal IP packet within a IPsec packet (see figure 3) . Since both the header and payload of the original packet are encrypted, the internal structure of the private network is concealed from intruders [Murh98].

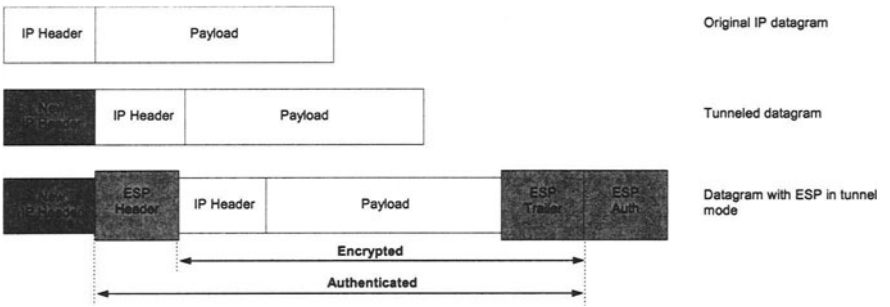

Figure 3. IPsec using ESP in tunnel mode

The use of tunnel mode also allows us to use the SNS nodes as gateways routing packets from private LANs (like the one in the SNMP example in Figure 1) to remote workstations or other SNS nodes [Scot98].

3.2.4 Simplicity

We attempted to keep the complexity of the SNS platform as low as possible for the following reasons:

- A complex design is difficult to verify and control. This implies that maintaining the security posture of the platform after its original roll out will be difficult.
- Network administrators come and go. A non standard tool like the SNS will have to be easy to master, otherwise new staff will not be able to use it effectively.
- The SNS is primarily a tool for network troubleshooting. The administrators must be confident of the platform, otherwise instead of troubleshooting the problem they will be troubleshooting the platform.

In order to comply with the simplicity requirement we decided to dispense with the hard disk. The reason behind this decision was twofold, reliability and support. Older equipment, like the ones we use, tend to have problems with their hard drives, especially in the kind of hot environments that we let them operate. Greece is pretty hot in the summer and these PCs are left running in offices without air-conditioning for extended periods of time. Hard disks contribute a fair amount of heat and are also more prone to failure in these conditions.

The second and more important reason was related to the way that these machines were intended to be used. For our purposes, hard disks are already huge and are getting bigger all the time. This free space can cause all kinds of trouble; for example, it can be filled with data that should not be stored in the monitoring station in the first place. This means that stations can no longer be redeployed easily because this information must be backed up, or processed. Secondly, if a station is compromised, the intruders will be able to use this space as a bridgehead, transferring and installing tools that will enable them to attack other network assets.

On the other hand, diskless machines bring with them a whole collection of problems and administrative headaches. They are also basically incompatible with our intention of using standalone machines with encrypted tunnels for all communications between the monitoring stations.

Instead, we adopted the techniques used by the PICOBSD project which is a collection of FreeBSD configurations that can be accommodated within a single boot floppy (*http://www.freebsd.org/~picobsd*). The PICOBSD project provides configurations for a dial-up router, dial-in router (ISP access server), general purpose router and firewall. The PICOBSD technique links the code of all the executables that we wish to be available at runtime in a single executable using the *cruchgen* utility. [Silv98] The single executable alters its behaviour depending on the name under which it is run (argv[0]). By linking this executable to the names of the individual utilities we can

create a fully functional /stand directory. The root of the runtime file system together with the executable and associated links are placed in a ramdisk that is stored within the kernel binary. The kernel is then compressed (using *gzip*) and placed on a bootable floppy. This floppy also contains the /etc directory of the running system in uncompressed form to allow easy configuration of the runtime parameters. At boot time, the kernel is copied from the floppy disk to main memory, uncompressed and executed. The file system root is then located in the ramdisk. The floppy disk is mounted and the /etc directory copied to the ramdisk. At this point the floppy is no longer needed and may be removed. The system is running entirely off the ramdisk and goes multi-user running the /etc/rc* scripts. Once the boot process is complete, user logins from the console or the network are allowed. The floppy is usually write-protected so changes in the system configuration do not survive reboots. However, there exists a utility that can copy the contents of the ramdisk /etc directory to the floppy, thus making the running configuration, permanent.

The aggregation of the system executables in a single file and the compression of the entire kernel allows a surprising number of facilities to be made available despite the small size of the boot medium. The SNS nodes have been deployed in various roles such as:

- Controller: managing network assets like the ones we mentioned in the example in section 3.1.
- Traffic monitor: using the tcpdump utility in conjunction with the syslog facility allows suspected hostile activity in remote parts of the network to be monitored.
- Router: By adding high speed serial cards to SNS nodes we have created emergency routers that can provide up to 8 Mbps links with nearby buildings. Routing software (e.g. gated) can support both BGP and OSPF which are the protocols used by University of Piraeus routers.

Since the systems do not have any permanent storage, we have to send the system logs to the central monitoring station. This is a conventional (with disk) OpenBSD system that has IPsec links to all the other stations. The transfer of logging information to the central station is performed using *syslogd* over the IPsec links.

3.3 Monitoring stations in a WAN

Another example of the use of the SNS platform concerns the Greek University Network (*http://www.gunet.gr*). In Greece, all public institutions

of higher education have received funding in order to connect to a single high speed network (GUnet). Funding has also been provided for a local router, a server (in most cases a SUN workstation) and a network administrator stationed in each institution. These network administrators need to be in contact with the central network administration team in Athens.

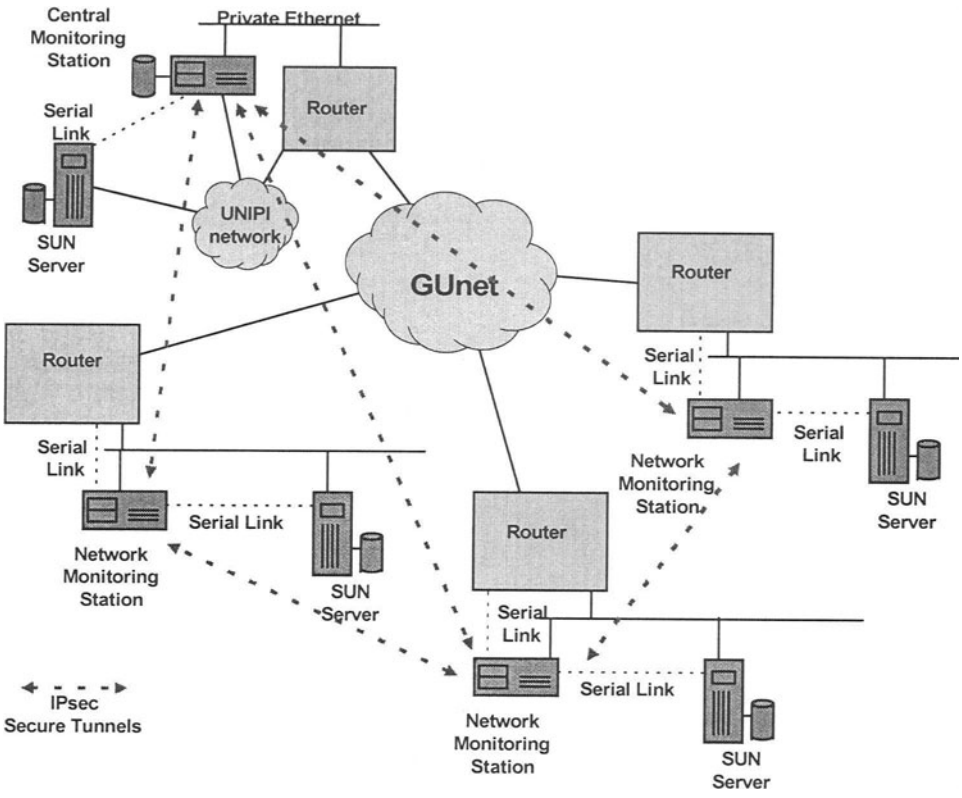

Figure 4. GUnet Network Monitoring Stations

Traditionally, these communications are carried out using PGP or some form of encrypted email scheme. However, for network monitoring purposes the administrators in the central site also need to be able to monitor the state of the routers located in the remote institutions. In cases of network problems, after consultation with the local network staff, the central site administrators may also need to run tests or reconfigure the GUnet routers and servers in the remote institutions. These operations would normally be run over insecure links, thus, creating opportunities for various exploits.

Due to the above considerations a small scale pilot has been commissioned by GUnet to investigate the possibility of employing network

monitoring, stations similar to the ones mentioned in the previous section, in all the GUnet sites. The pilot project involves the installation of a central monitoring station at the University of Piraeus and remote stations in four other institutions.

The configuration of the remote stations is shown in figure 4. The central site is hosting the logging facility and provides email and bulletin board services (for CERT and other security advisories) only to hosts within the secure network.

The serial links to the routers and hosts allow remote reconfiguration of the devices through a secure connection via the IPsec links. The primary service provider for GUnet is GRNET (*http://www.grnet.gr*). However, most institutions have backup links with other ISPs (not shown in the diagram). Through these backup links, the secure network can be used to debug and reconfigure even the GUnet routers via their serial consoles.

Currently secure links have been set up between four of the institutions participating in the pilot, while work is being carried out for the production of the final software distribution for the GUnet monitoring station. The pilot is expected to be complete by the end of May 1999, at which point a study will be carried out to determine the feasibility of creating a production system covering all institutions participating in the GUnet project. If all goes well, the full system will go on-line by the end of this year.

4. CONCLUSIONS – FUTURE PLANS

Most of what the tools used in our project are well known and widely used. There are numerous network monitoring programs, and the notion of having a monitoring station installed in a LAN, has been proposed before [Ches94]. Also the use of IPsec, ssh and the other tools mentioned in this paper is common practice.

However, combining all these disparate services in a floppy-based distribution and deploying numerous stations both in our university network and in the institutions participating in the GUnet pilot is to our knowledge the first clear demonstration of such a network monitoring station.

It is difficult to stay still in an ever changing world, so we intend to keep maintaining the software distribution for the network monitoring station and adding features to make our lives easier. Taking advantage from the fact that the entire system is based on free software, we plan to make the entire system (executable floppies and the development system that produces them) available on our ftp site so that other users may benefit from our efforts.

We are also looking into ways of abandoning the floppies as distribution media and replacing them with flash RAM cards because they offer higher capacity then floppies and due to the complete lack of moving parts are far more reliable.

Finally, as vendors slowly adopt the IPsec protocol, we hope to integrate equipment that support the protocol into our secure network. This will give us first hand experience with interoperability and integration between different implementations of the IPsec protocol.

REFERENCES

[Chap95] Chapman D.Brent and Elizabeth D. Zwicky, "Building Internet Firewalls," Second Edition, O'Reilly & Associates, Inc. 1995.

[Ches94] Cheswick, William and Steven Bellovin, "Firewalls & Internet Security, Repelling the Wily Hacker," Addison-Wesley Professional Computing Series, 1994.

[Fort99] Forte Dario, "Intrusion-Detection Systems: Guaranteeing the Safety of a Network Beyond Using a Firewall," ;login: The USENIX Association Magazine, Vol 24, No 1, February 1999.

[Garf96] Garfinkel, Simpson and Gene Spafford, "Practical UNIX and Internet Security," Second Edition, O'Reilly & Associates, Inc. 1996.

[Oppl98] Oppliger, Rolf "Security at the Internet Layer", IEEE Computer, Vol. 31, No. 8, pp. 43-47, Sept. 1998.

[RFC.1825] Atkinson, R. "Security Architecture for the Internet Protocol," Internet Engineering Task Force, August 1995.

[RFC.1827] Atkinson, R. "IP Encapsulating Security Payload (ESP)," Internet Engineering Task Force, August 1995.

[RFC.1910] Waters, G. Ed, "User-based Security Model for SNMPv2," Internet Engineering Task Force, February 1996.

[Scot98] Scott, Charlie, Paul Wolfe and Mike Erwin, "Virtual Private Networks," O'Reilly & Associates, Inc. 1998.

[Shah97] Shah Deval and Helen Holzbaur, "*Virtual Private Networks: Security With an Uncommon Touch*," Data Communications, Sept. 97,

[Silv98] Silva James da, "Cruchgen," OpenBSD User Manual, 1998.

[SSH98] SSH Communications Security, "SSH IPSEC – White Paper", Ver. 1.0, Jan. 1998, <http://www.ssh.fi>, SSH Communications Security Ltd., Finland.