A Network Firewall

Marcus J. Ranum

mjr@dco.dec.com

Digital Equipment Corporation

Washington Open Systems Resource Center, Greenbelt, MD

June 12, 1992

Abstract

Information is the lifeblood of the computer age, and network connectivity is crucial to day-to-day business. Connecting a private, corporate network to the Internet is not acceptable without some form of secure gateway acting as a firewall between the two networks, to prevent miscreants and unwelcome visitors from accessing hosts on the private network. In the case of a software or hardware vendor, source code, CAD diagrams, and other product-specific information must be kept secret. Hospitals and insurance companies, that maintain confidential information, or pharmaceutical research labs with patent applications cannot afford to take chances with data theft. A break-in over the network could do incalculable damage in a very short time.

Digital has implemented several gateways between its corporate network and the Internet, which provide a high degree of access while maintaining excellent security. The gateways are composed of multiple machines acting together, and a specially configured packet-screening machine that provides discretionary TCP/IP access control. Software is configured across the gateways to provide transparent USENET, SMTP mail, FTP, and name service, while preventing direct connections between internal machines and external machines. This paper discusses the overall configuration, software used, and some of the security measures that are in place.

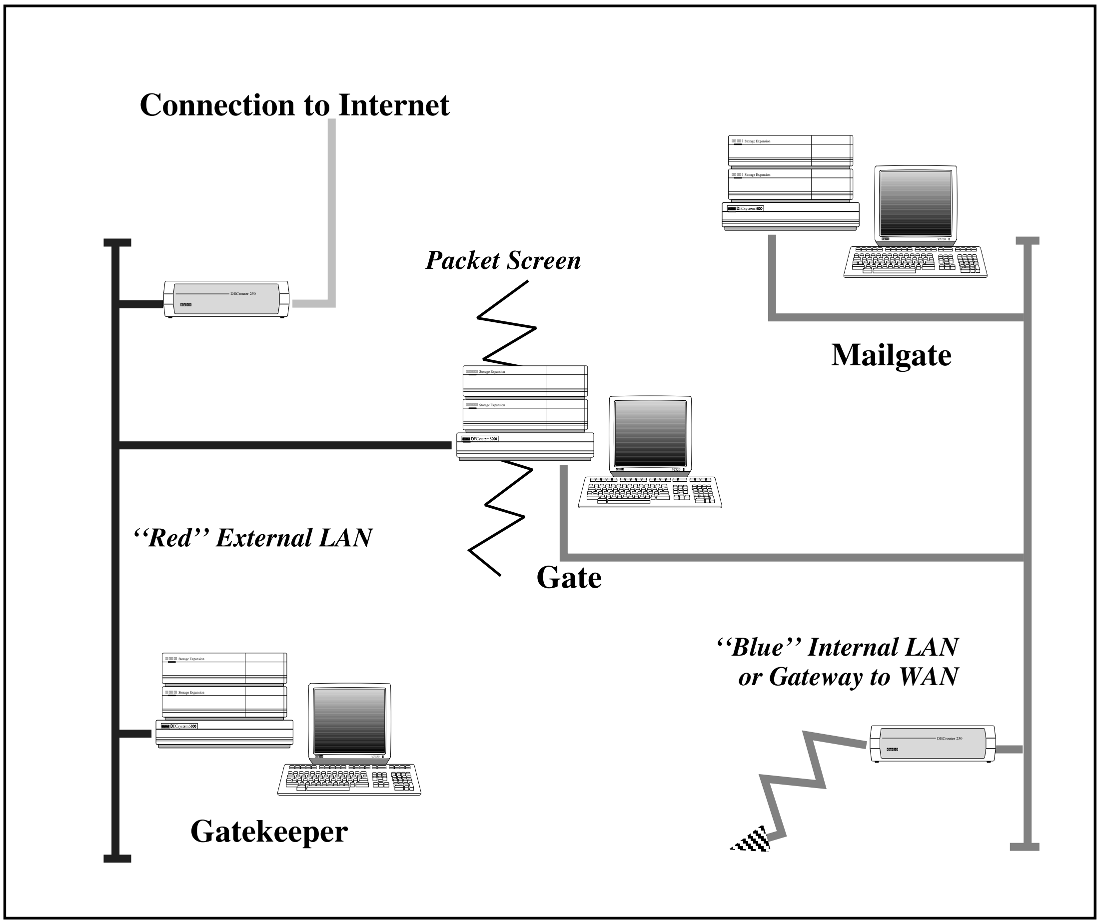
These three gateways have been in operation for over six years, and to date no (discovered) break-in has occurred. The importance of the gateways is hard to estimate, since it provides a crucial link between Digital sales and their customers, as well as maintaining an important presence on the network.

1. Design Goals

The key criterion for success for the Digital corporate gateways is preventing an unauthorized or unnoticed leak of data to the outside. There is no restriction on incoming data whatsoever. Clearly the gateways themselves must be as secure as possible, since if they could be penetrated and reconfigured, they could permit open access to any node within the private network. A more general design goal is that the gateways should be relatively unobtrusive -- services should be made to work as if the gateway is not there are all, wherever possible. As always, there is a constant trade-off between ease of use and security. Digital corporate policy solves the trade-off by assuming that security is always the higher priority.

2. General Configuration

Each gateway is actually made up of three hosts, acting in concert to provide service as if they were a single system. For the sake of terminology, these systems will be referred to as Gatekeeper, Gate, and Mailgate. Gatekeeper and Gate are on a short network segment that supports the router connecting to the Internet. This is known as the ‘‘red’’ network and is considered untrusted. Gate has dual network interfaces, the second of which is connected to the ‘‘blue’’ network -- Digital’s private internet. Mailgate is a purely internal host and is considered to be on a trusted network.



The ‘‘red’’ and ‘‘blue’’ networks are only connected by Gate, which acts as a router and packet screen. The packet screening is implemented using the ULTRIX[1] packet screening software, which is a supported option in the ULTRIX kernel. An application level program (‘‘screend’’)[1] is given veto authority over all IP packets that try to leave Gate’s IP layer. Screend supports a simple configuration language that allows exact specification of source host

[1] ULTRIX is a trademark of Digital Equipment Corporation

or network, destination host or network, source port or destination port, packet type (UDP, ICMP, etc) and can be configured to either accept or reject traffic meeting a given specification. Thus it is possible to permit only certain types of traffic between certain sets of hosts. Rejected or accepted packets can be logged or summarized if desired. These capabilities, except for the logging, are not unlike the capabilities present in most commercial routers, but using a host for routing is preferred as it is easier to secure and maintain the system.

The packet screen is configured such that Gatekeeper can communicate freely with any node within Digital. It would be possible, if desired, to restrict this to a subset of hosts, but security seems good enough as to make this unnecessary. No other system on the ‘‘red’’ network is permitted to send data through the packet screen. This effectively leaves Gatekeeper as an isolated point of contact that is reachable by both the Internet and Digital’s network. Gatekeeper acts as a moderator for services, receiving and forwarding mail, news, etc. between the two networks. Each of the services that Gatekeeper provides is examined separately for its security implications, and is either disabled, or made secure.

To further improve security, IP routing is carefully configured such that it is impossible to route TCP/IP traffic directly between the private network and the Internet. This requires using static routing and careful manual configuration of the routing tables on all the machines comprising the gateway. Gatekeeper is configured with a default route out to the Internet, and hard-coded routes to the private network through Gate. Gate is configured with a default route to the private network, and a route to the ‘‘red’’ untrusted network. Gate has no route to the Internet, and does not run any software that accepts RIP information. Hosts on the private network can route traffic to the ‘‘red’’ network through Gate, but if someone attempts to route something through Gate with an address on the Internet, Gate will fail to route it, since its default route leads back to the private network. This additional complication is probably gilding the lily when added to the packet screen, but since it costs nothing to implement and does not remove any functionality, it is worthwhile.

The gateway also takes advantage of the fact that the hosts comprising it are running a full-fledged UNIX installation. Anything of interest that occurs on the systems is logged to a pair of designated hosts on the private network as well as to local disk. The system logs are automatically processed at intervals, and the systems administrator is notified if something untoward is discovered. Failed or successful login attempts and attempts to FTP out the password file are logged, as are all mail transactions, FTP file access, remote telnet access, and uses of the ’su’ command.

To further restrict access to Gatekeeper and Gate, a ‘‘wrapper’’ program is interposed between all connections over the network to remote login or terminal services. The ‘‘wrapper’’ program checks the origin of all incoming TCP connections that come in though the Internet, and checks a simple access control list that causes the service request to be either denied or accepted [2]. Both Gatekeeper and Gate are set up to deny network login attempts from any host that is not on the private network, with Gate being limited to accepting telnet connections for a small list of trusted hosts. Every access, whether permitted or denied, is logged in case the systems manager needs an audit trail. To break into Gate and reconfigure the packet screen, a systems cracker would have to first somehow break into Gatekeeper, then break into one of the trusted hosts on the internal network, and then from there break into Gate. Each of these steps is protected with passwords and a degree of physical security.

This configuration provides excellent security, since it is only necessary to ensure that one system is as secure as possible. If a security hole is identified in the mailer software, patching it immediately in one place will secure the entire network, whereas in a completely connected Internet site, it is necessary to patch the hole in every system on the network separately. This tends to create what Bill Cheswick refers to as ‘‘a sort of crunchy shell around a soft, chewy center’’ [3] where the network is extremely hard to penetrate, but is lax in security inside. For Digital’s internal internet, with over 30,000 nodes running TCP/IP it is not reasonable to assume that they are all secure - the shell needs to be very crunchy indeed.

3. Electronic Mail

Electronic mail was the original raison d’etre of the gateways, and they continue to process over 22,000 messages/day apiece. Since Digital has a large DECnet installed base, interoperability with VMS[1] Mail-11 and ALL-IN-1[2] are considerations, and are addressed by the simple means of handing mail to be delivered to VMS nodes off to an internal system that runs DECnet: Mailgate. Mailgate is the designated forwarder for VMS mail and mail to ALL-IN-1, and can also act as a recipient for outgoing mail that uses DECnet as a transport. In the real production environment at Digital, there are multiple hosts that fulfill this role.

Gatekeeper publishes an MX (mail exchanger) record for each DECnet host in the company, advertising that it will accept mail for hostname.enet.dec.com. The ‘‘*enet.dec.com*’’ is just a syntactic namespace for the DECnet hosts, that permits gatekeeper to recognize it and forward it to mailgate for delivery. Mailgate re-writes the addresses such that *user@host.enet.dec.com* is converted to a DECnet-style *host::user* address and delivered over DECnet. Similar support for ALL-IN-1 is provided, based on the pseudo namespace ‘‘*mts.dec.com*’’ where mail addressed to *firstname.lastname@sitecode.mts.dec.com* is passed to Message Router (Digital’s X.400 mail product) for delivery. Since Digital has three gateways connected to the internet, the MX records list all three in descending priority, so that in the event of one gateway’s being unreachable, the mail will still go through.

This system of handling electronic mail is quite secure, since it prevents external systems from delivering mail directly or potentially probing for security holes in sendmail on internal machines. The version of sendmail that is run on Gatekeeper has been enhanced to improve logging and all known sendmail security bugs are fixed.

4. Telnet and FTP

Telnet and FTP are important services that must also be supported without compromising security. This is done through trusted application gateways - software specially designed to mediate a particular type of traffic and to prevent unauthorized use. Gatekeeper runs an FTP protocol gateway to provide interactive remote access to FTP on the Internet. This gateway has a

[1] VMS is a trademark of Digital Equipment Corporation

[2] ALL-IN-1 is a trademark of Digital Equipment Corporation

variety of logging and permissions checking features that allow the administrator to selectively permit or deny incoming or outgoing FTP on a host or network basis, and can selectively enable or disable specific FTP commands (E.g.: STOR, preventing file export).

The telnet gateway provides similar functionality, with a special ‘‘rate limiter’’ built into it, which limits data going out the connection to 1200 baud, while permitting data to flow in through the connection at any rate. The rate limited telnet gateway is completely invisible for interactive terminal sessions, but quickly puts a halt to users attempting to copy large volumes of data out of the company through their terminal session.

Digital’s FTP gateways are configured to permit any amount of incoming data, and to block all outgoing data. A single one of the gateways handles about 200Mb/week of FTP traffic. The telnet gateways are presently under review prior to entering common use.

Recently, the telnet and FTP application gateways have been enhanced to allow authentication using a ‘‘smart card’’ crypto key. The permissions files for these applications now have extensions so that connections can be permitted from external systems to internal systems, after a user authenticates himself to the network. This system is currently under evaluation. When a user attempts to perform an operation that would require authorization, the operation is denied until the user authenticates himself with the hand held cryptographic calculator. This is done using a challenge/response mechanism, in which the server challenges the user with a random number, which the user encrypts with their calculator and returns over the network. Security is very strong in a system like this, since the response key is entered into a trusted hand-held unit, and not into a potentially booby-trapped host system as with Kerberos or other authentication systems that rely on a user entering their key into software. It is easy to imagine a system in which the terminal driver logs all keystrokes from a user. A secret password could then be unobtrusively captured in plaintext form when the user entered it while getting their Kerberos ticket. Since the handheld calculator is presumably completely insulated from the network, it is not vulnerable to snooping via that avenue.

5. *Other Services*

All three gateways act as a domain name servers for ‘‘*dec.com*’’ with the primary name server being the gateway in Palo Alto, CA; the others are secondaries. All act as forwarders, so internal systems can still resolve names correctly. Since the gateways are authorities for ‘‘*dec.com*’’, name service queries from the Internet are handled correctly.

USENET news passes through the gateway using NNTP (Network News Transfer Portocol) and propagates itself within the corporation’s USENET community. This service is completely transparent to the gateway. NTP (Network Time Protocol) is used to synchronize clocks between hosts on the Internet and the private network.

None of these services actually passes data directly between the Internet and the internal network, and therefore are not considered security risks. Since these services might contain security holes, the gateway managers maintain the software carefully, keeping an ear to the ground for any security loopholes as they are discovered. Generally the gateways do not run the ‘‘latest greatest’’ version of software - debugged and stable versions are preferred.

6. Existing Practice, Different Approaches, and Tar Baby

Several papers have been published that describe various approaches to configuring network firewalls. Most of them advocate using a commercial router that permits screening within the router [4] [5]. These configurations can be made very secure, but in the event that the router is compromised, the network can be reconfigured such that the screening is disabled and the private network is completely exposed. It may take some time to detect if this occurs. Other configurations, with a single host on both networks, and TCP/IP packet forwarding disabled, present similar risks, in that if the host can be compromised, TCP/IP packet forwarding can be re-enabled by a miscreant, throwing the network gateway wide open.

The AT&T corporate secure gateway is similar to the Digital one, in that it implements outgoing connections using proxies that are available only to hosts on the private network. The AT&T gateway uses a switching connection server through datakit, and provides a raw connection to the outside, unlike the Digital gateway which has proxies that ‘‘understand’’ each protocol and may optionally block or enable certain types of traffic. The AT&T approach permits more general access, which would not be in line with Digital corporate policy for exporting data.

Some functionality from the AT&T gateway was added to the Digital gateway[6]. This functionality consists of a variety of ‘‘sucker traps’’ for the unwary systems cracker. Rather than simply serving as a form of cheap humor, ‘‘sucker traps’’ are a useful tool for telling the systems administrator that they are actually being probed. In network security, 95% of the problem is detecting that someone is testing your defenses in the first place; if they walk into a trap and give the systems manager a warning, they are that much easier to contain. These ‘‘sucker traps’’ consist of simple changes to the ‘‘login’’ program that will attempt to run a ‘‘finger’’ against the host originating the connection. When multiple failed login attempts come from a given host, one can often identify the user originating the login attempts by examining process idle times. Some initial work was done by the author to have the login program attempt to contact the calling machine’s SNMP service and to examine its connection tables. This would be useful since many system crackers will ‘‘springboard’’ from one host to another to obscure their tracks. Unfortunately, many Internet backbone segments filter out SNMP traffic, so this proved impractical. Other "sucker traps" on the gateway log attempts to steal the password file via FTP or TFTP, and present a false view of the system’s user activity when queried with the "finger" command.

7. Experiences and Observations

DEC’s gateways have been in place guarding DEC’s network for over six years and have proven robust in the face of adversity. Recently, Digital has installed similar gateways for customers that have a business need for Internet access and a requirement for security.

Keeping users off the gateways is a good idea. When users are permitted on the gateway, steps need to be taken to ensure that the users cannot unintentionally (or otherwise) compromise the system. Stories abound of users who choose consistently guessable passwords, who use the same password on every system to which they have access, or who have .rhosts files and .netrc files containing more passwords. In an attempt to prevent these kinds of problems, the systems manager will quickly get bogged down in attempting to identify potential bad things that a user might do, and rushing to patch each potential hole. Having users on the gateway defines the problem domain as: *''That which is not explicitly prohibited is permitted.''* Throwing them off and using trusted application gateways enforces the policy: *''That which is not explicitly permitted is prohibited.''*

Log everything that shows when connections were made, and from where. This is especially useful in tracking back a series of connections to determine whether someone is probing you seriously, or whether they are just sniffing around. If an actual incident occurs it is important to have time-stamped records. Logs on the Digital gateway are saved on archival media for over a year.

Gateways will get probed, but most probes are not serious. Some are even well-intentioned. Several individuals have triggered traps on the gateway by FTPing out the password file and running ''crack''[7] against it. Crack finds several of the ''sucker trap'' passwords - well-meaning individuals have warned the gateway administrators of this apparent weakness. This is somewhat akin to running around your neighborhood with a crowbar, testing your neighbors' window-locks and door-hinges, then letting them know when you find an open window, or a door that can be pried off; it is ''neighborly'' but somewhat suspect. The rate at which the gateways get probed rises sharply at the beginning of the fall semester, and tends to tail off towards the end of each semester and final exams. Most of these probes are inept and are not worth noticing other than to send mail to the manager of the originating site: an FTP capture of the password file, followed 2 hours later by a series of bad login attempts against one of the dummy accounts. These are not considered threats.

8. ''Dances with Turkeys''

One important part of the gateway administrator's job is to make at least a cursory effort to understand how an attack is launched. It is not sufficient to simply sit back and watch, assuming that the attacker is going to try the same old tricks. Ideally, watching an attacker and monitoring thier behavior will give a good idea of where defenses can be improved. The following are a sequence of entries from one of the Digital gateway's logs: *(All names have been changed)*

```
Feb 5 08:22:56 18391 tftpd: connection from spud.cs.college.edu
Feb 5 08:22:56 18392 tftpd: spud.cs.college.edu opens /etc/passwd (r)
Feb 5 10:18:15 20409 inetd: as /usr/etc/telnetd gomer.eng.bsu.edu
Feb 5 10:18:31 login: BADLOGIN ttyp0, sucker
```

Note how the log entry shows a connection to inetd for the telnet service, from a host at Big State University (`gomer.eng.bsu.edu`). This connection occurred approximately two hours after the password file was captured from the host at College College (`college.edu`) -- running ''crack'' with a reasonable dictionary on a slow machine might take that long. The login attempt ''sucker'' is an account that exists in the dummy password file, with a non-obvious password that *is in* commonly used ''crack'' dictionaries. Unfortunately for the perpetrator, the tftp daemon and the login program are part of the ‘‘sucker trap’’ and the following information appears in the log:

```
Target machine: finger output from gomer.eng.bsu.edu /usr/etc/telnetd
Login     Name              TTY      Idle     When      Where
usr1      Sum Gai           p3                Wed 10:19 spoof.cs.coll
```

There is only one user on the machine at the time of the login attempt, and there is no idle time. Even more interestingly, the user is logged in from a node at College College (`spoof.cs.college.edu`), where the password file was originally stolen from. At this point email was sent to the manager of the node at Big State University, informing them of the problem and requesting that they log connections to the account that was being used (it turned out to be a ‘‘cracked’’ account).

Later the system underwent another probe:

```
Feb 5 23:46:13 5634 inetd: as /usr/etc/telnetd nemo.math.yau.edu
Feb 5 23:46:46 login: BADLOGIN ttyp0, sucker
Feb 5 23:47:03 login: BADLOGIN ttyp0, guest
Feb 5 23:47:24 login: BADLOGIN ttyp0, demos
Feb 5 23:47:36 login: BADLOGIN ttyp0, lp
Feb 5 23:47:47 login: BADLOGIN ttyp0, lpd

Target machine: finger output from nemo.math.yau.edu /usr/etc/telnetd
Login     Name              TTY      Idle     When      Where
emal      Electron Mic      p0                Wed 23:39 spud.cs.college
root      Root              co       2:16     Wed 19:00
```

Clearly this individual is not a threat. They tried once, and then were unable to realize that they had been detected, and tried again, giving away another node that had already been compromised. The systems manager at Yet Another University (`yau.edu`) was notified, and the compromised account was monitored. A few network queries were made against the systems at College College that were hosting the attack and it became quickly apparent that the node the attacks were being launched from was some kind of terminal server or other node that did not run SNMP, and had no finger, ruptime, mail, or other services to examine for useful information. At this point, a summary of the activity was mailed to the systems managers at College College, and the attacker was ignored henceforth.

Was this worth expending effort on, or was it simply a case of ‘‘pulling the wings off flies’’? On one hand, it strengthened the security of the gateway by giving a clear picture of how a moderately inept Internet cracker might probe a system, and it encouraged two other sites to improve their security - on the other hand, the attacker was apparently not a real threat to the integrity of the gateway at any point in time, and expending the 30 or so minutes of effort to reconstruct events from the system logs was a waste of time. It does show clearly that a little knowledge can be a dangerous thing, and it leaves one convinced that having some kind of firewall is worth the time and effort required to set it up.

9. Future work

The Digital corporate gateways continue to evolve and be strengthened as new ideas are proposed, and new bugs and attacks are found. The author carefully monitors security-related newsgroups and mailing lists on the internet, and immediately verifies that any holes mentioned are already plugged and/or that a ‘‘sucker trap’’ is ready to fire off alarms if someone attempts to probe through a well-known hole (like the sendmail debug hole). This is an ongoing process that will never be completed, and Digital’s gatekeepers and Digital’s customers that run Digital gateways keep each other up-to-date on changes and potential threats.

Future enhancements to the gateway will probably include a more widespread use of hand held cryptographic calculators, and the development of new application gateways, such as an X-window protocol gateway. The feasibility of making the entire gateway software reside on a bootable CDROM or other fixed media is being examined as a possible means of eliminating worries about trojan horses and easing release control of the software. The gateways presently have a system running NNStat [8] that gathers statistics about IP traffic over the network link. Based on findings by Steven Bellovin [9] the statistics-gathering configuration will probably be extended to look for ‘‘mystery packets’’ and flag their apparent sources.

10. Conclusions

Experience managing the Digital gateway encourages the author to attempt to list a few important rules derived for managing a gateway:

- Keep the users off the actual gateway machines.
- Provide gateway services that permit controlled access to the Internet from hosts on the private network. Ensure that all the gateway services generate logging information that can be summarized and examined if necessary.
- Keep an audit trail of everything that connects or disconnects from the gateway. This is absolutely vital for reconstructing an attack.
- If you do not have logs of an attack, you cannot know if it succeeded.
- 99% of the probes against your system do not represent a real threat.
- Lean on the experience of others. Implementing a gateway is hard - it only takes one mistake.
- Traps and bait are useful warnings that someone is sniffing around. Silent alarms give you a better chance to watch and see if there is in fact a weakness the attacker can exploit.
- Turn off all non-essential services. It is hard to break into a machine using bugs in software that is disabled.
- Do your research. Read USENET news and monitor systems management mailing lists.
- Configure the system to mail the systems manager anything that looks unusual. Footprints in your system may not look like much - you need to be prepared to peer at system logs if something looks odd.
- A sneaky, devious mind is the best weapon.

11. Availability

The gateway software is composed of a mix of freely available software from the internet with various bug-fixes and modifications, and software written by Digital technical staff. The packet screening router software is a part of the basic ULTRIX operating system; no kernel modifications are required. Digital will configure the Digital Internet gateway for customers as a consulting offering; contact the author for details.

Acknowledgements

The first Digital internet gateways were set up by Richard Johnsson and Brian Reid. Jeff Mogul developed the packet screening software, and Paul Vixie re-configured the gateway to use the screen. Subsequent refinements have been provided by many individuals. Jody Patilla helped make this much more readable.

About the Author

Marcus Ranum is a software specialist in Digital's Washington, DC, Open Systems Resource Center. He specializes in a wide variety of UNIX-related topics, mostly dealing with systems security, network security, systems management, and TCP/IP network management. Marcus runs and maintains one of Digital's three corporate internet gateways (decuac.dec.com) and is the technical lead for the SEAL (Screening External Access Link) internet gateway product, which is based on work presented herein.

References

[1] Jeff Mogul, *Simple and Flexible Datagram Access Controls for UNIX-based Gateways*, USENIX proceedings, Feb 1989

[2] Wietse Venema, *log_tcp*, USENET archives, comp.sources.misc Volume 23, Issue 77, Oct 1991

[3] Bill Cheswick, *The Design of a Secure Internet Gateway*, USENIX proceedings

-see also- Bill Cheswick, Steven Bellovin, and Doug McIlroy, communications on USENET newsgroup Sun-nets, Apr 1991 - May 1991

[4] Smoot Carl-Mitchell, and John Quarterman, *Building Internet Firewalls*, UNIX World, Feb, 1992.

[5] Simson Garfinkel and Gene Spafford, *Practical UNIX Security*, O'Reilly and Associates, June 1991

[6] Bill Cheswick, *An evening with Berferd in which a cracker is Lured, Endured, and Studied*, USENIX proceedings, Jan 20, 1992

[7] Alec David Muffett, *crack - The Password Cracker*, USENET archives, comp.sources.misc, Volume 22, Issue 49, Aug 1991

[8] Bob Braden and Annette DeSchon, *NSFnet Statistics Collection System NNStat*, USC Information Sciences Institute, published on USENET archives, Mar, 1991

[9] Steven M. Bellovin, *Packets found on an Internet*, AT&T Bell Labs, published on internet archives research.att.com, May, 1992.