Centralized Administration of Distributed Firewalls

Mark Miller & Joe Morris
Bell Atlantic

Centralized Administration of Distributed Firewalls

Mark Miller & Joe Morris – Bell Atlantic

ABSTRACT

Allowing the connection of various networks to one's internal network while keeping the internal network protected from unauthorized access is key to success in this information age. Firewall systems are currently the best known way of achieving this objective. Management of these firewalls in a secure, consistent and reliable manner is a strategy that is imperative to the successful interconnection of diverse networks. This paper describes the technical implementation of a paradigm enabling the administration of distributed firewall systems from a centralized location.

Justification

In November of 1995 the Business Continuity group was formed at Bell Atlantic, and charged with providing security for all systems connected to the Bell Atlantic internal network. One key responsibility was to provide firewall support, including support for existing firewalls, as well as the development of standards for the support of any new firewalls.

Prior to the formation of the Business Continuity group, a total of eight people in four different groups spread throughout the seven state region of Bell Atlantic performed maintenance of existing firewalls. System administration policies were inconsistent, and management felt a certain lack of control over the state of our connections to other networks. In response, a standard firewall architecture was developed using a combination of vendor supported software, public domain software and programs developed in-house. Currently, only three people in the Business Continuity group are needed to provide support for all existing firewalls, with less than five percent of their time dedicated to ongoing firewall management.

This change was seen as a very positive step by all parties involved. From an executive perspective, there was a reduction in the number of resources dedicated to performing firewall support. Those previously performing the function were generally elated as this was never meant to be their full time job, and this move enabled them to focus on their primary responsibilities. From a security perspective, it was advantageous because consistent repeatable processes were in place for building and managing the firewalls. This was a major step that allowed our security stance to go from reactive to proactive.

Defining the Architecture

Bell Atlantic has an internal TCP/IP based network consisting of 25 class B address spaces assigned based on geographic locations and connected in a hierarchically manner. This internal network, known as BAInet, is connected to several other networks, including, the Internet, other Bell Atlantic networks, and a few customer sites. The manner in which these interconnections are constructed ranges from simple filtered routers to full-fledged DMZ lans with bastion hosts. The services provided through these interconnections range from standard Internet services to application specific communication channels.

To aid in the ease of administration, a standard firewall architecture was adopted. This standard defines the guidelines to be used in the design of new firewalls. The key principles used in defining the architecture are:

- All firewalls are designed to eliminate any Single Points of Failure.
- All management functions are performed centrally.
- No management functions are done in the clear over the network.
- Firewall rules are built on the philosophy that "whatever is not expressly permitted is forbidden."

This architecture also specified hardware and software standards which were to be used.

Selecting A Standard Hardware Architecture

Firewall systems can contain both filtered routers and firewall computers (generally Unix). In developing a standard firewall configuration, the first decision was that of the hardware platforms to be used. Sun was chosen as our firewall computer vendor and Cisco was chosen as our router vendor.

Sun was chosen for the following reasons:

- There is a high level of familiarity with the hardware and company.
- Tools are, in general, most readily available on Suns. Stated another way, tools are generally not unavailable for Sun.
- Bell Atlantic has a pre-existing support contract with Sun.

- The scalability of Sun's hardware platforms enables the selection of correctly sized system for individual firewalls.
- Existing Unix support groups in Bell Atlantic are most experienced with Suns. This enables us to pass certain support functions to this group and exploit economies of scale.

The firewall servers are designed to be redundant, and to eliminate single points of failure. For example, duplicate network interfaces are used, and mirrored internal disks are configured to allow booting from either disk. High availability is greatly enhanced by these measures as well as the use of Uninterruptible Power Supplies at all sites.

Cisco was chosen as our router vendor for the following reasons:

- Administrative access can be provided in a secured manner.
- There is pre-existing expertise in writing Cisco rules.
- Cisco is the de facto Internet standard.
- Firewall/1 allows creation of Cisco rules.

One concern voiced over the decision of using only one router vendor is the argument that, "if a weakness is found in the operating system of the router, then all firewall routers from that vendor can be subverted." This problem was avoided by always including a firewall server sandwiched between two routers. This design necessitates that both the filtered router and a firewall be compromised to gain unauthorized access. Also, possible confusion caused by using multiple vendors' filtering languages is avoided.

Securing the Operating System

The decision to use Sun as our hardware base lead us to use Solaris as our operating system. A decision was made to use Solaris 2.5 based primarily on the fact that it includes most security patches to Solaris 2.4, and it has numerous performance enhancements over Solaris 2.4. Also, by using the latest version of Solaris, we avoid the common vendor retort of "load the latest patch" when confronted with problems.

Once the operating system was selected, a process was developed for hardening the operating system. The process consisted of:

- Loading the base operating system.
- Removing unnecessary or dangerous programs (e.g., sendmail, ftpd, telnet).
- Securing permissions on key files and directories (e.g., removing setuid and non-owner write permissions).
- Installing various software packages (e.g., ssh, tiger, tripwire, FDDI package).

The entire process was scripted and documented so it could be performed in less than an hour. The philosophy used in the design of the secure operating system installation process was threefold in nature:

- Prevent and detect unauthorized access to the firewall.
- Prevent and detect unauthorized root access of the firewall.
- Prevent further unauthorized access via a compromised firewall.

Installing the Firewall Software

The firewall product which was selected was Checkpoint's Firewall/1. The reasons for using Firewall/1 were:

- Firewall/1 is a highly respected firewall product.
- Firewall/1 rules are built using a GUI. Rules could be discussed with appropriate management, without requiring management to posses a high degree of knowledge in a filter rule language.
- Remote centralized administration functionality is inherent in its design. This function is not supported by many firewalls.
- We inherited it from existing firewalls. We had paid for the licenses, and it was deemed most cost-effective to use the pre-existing licenses.

There are also several features of Firewall/1 which are promising to us for future applications. First, support for SecurID authentication is built into the Firewall/1 proxies. This was promising to us as we already have a large installed base of SecureID card users. The inclusion of NAT and VPN functionality is also encouraging to us for future use.

Installing Monitoring Software

Monitoring of the firewall systems is performed by several means. Firewall/1's built in logging facilities are used for monitoring the activity of the firewall filters. Custom developed software is used for the monitoring of system resources. Finally, several public domain utilities are used for monitoring security related aspects of the system. Procedures have also been established and implemented for both regular review of the logs by a person, and automated analysis and reporting of error messages via scripts.

Firewall/1 allows the reporting of critical conditions by several methods including writing to a log file, popping up an X Window alert box, or specifying a program to run. We opted to run a custom program; specifically Firewall/1 is configured to use the syslog facility to enable real time message forwarding. Messages are generated by Firewall/1 whenever an inappropriate packet is detected.

System resources are monitored using a modified suite of scripts developed at Bell Atlantic for monitoring Unix systems. These scripts monitor memory, disk, and processor resources and send messages when certain thresholds are surpassed. All scripts are written in Bourne shell and report error or threshold conditions via the syslog facility.

Two public domain security programs are also used to monitor the system. We chose Tiger to perform regular audits of the system. Tiger is a tool from TAMU which verifies that permissions and ownerships of files on the system are not overly permissive. Tiger was chosen over several other available packages due to its ease if use and configuration. Any error messages which are generated are compared to a file containing expected messages, and any differences are forwarded to the central collection system for analysis.

The other public domain security tool we use is Tripwire from Purdue University. When initialized, Tripwire creates a database of permissions, ownerships, modification times and digital signatures of key files on the system. Upon subsequent runs, Tripwire will report any changes made to the attributes of those files. After Tripwire has been run for the first time, the database is copied onto a floppy diskette. The diskette is then made read-only, and mounted on the system, thus preventing tampering with the initial database.

One difficult selling point was the use of publicly available tools on the firewalls. The tools we decided to use are ones that have been reviewed and discussed, in depth, in the Internet community. They also have the benefit of being free. As Brent Chapman and Elizabeth Zwicky point out,

> “Some people feel uncomfortable using software that’s freely available on the Internet, particularly for security-critical applications. We feel that the advantages outweigh the disadvantages. You may not have the “guarantees” offered by vendors, but you have the ability to inspect the source code and to share information with the large community that helps to maintain the software. In practice, vendors come and go, but the community endures.”

A modified version of syslogd is used to transport messages from our firewalls to our logging system. The modifications restrict syslogd to receiving messages only from the local system, i.e., it will not accept syslog messages from other systems. Syslog was chosen as our transport mechanism for several reasons:

- Syslog is easy to configure.
- Other alternatives for message transport are equally or less trustworthy (sendmail, snmp).
- Syslog is standard on all versions of Unix which we were planning on using.
- Commercial monitoring packages are more expensive and an “unknown quantity”. The source code for syslog is freely available and reviewable.

Syslog is configured on the firewall to write the message to a log file, display the message on the console, print the message on a directly connected printer, and forward the message to a central syslog message collector.

The central error message collector receives the syslog messages and processes them using a software system called BASEMM (Bell Atlantic Server Error Message Monitor). BASEMM is an application designed to parse system-generated error messages. In essence it functions as a wrapper around the syslog log file (/var/adm/messages). Because of its design it will also parse any application message that uses the system “logging vehicle”: syslogd. In general terms, BASEMM refers to the collection of applications that collect and report on data parsed by the central application, the BASEMM daemon (BASEMMD).

The various components of BASEMM are written mostly in Perl 5. Other than the various programs and scripts on the BASEMM client machines the only other BASEMM application not written in Perl is the one that does alpha-numeric paging, which is written in expect.

BASEMM parses incoming messages and deals with error conditions in a variety of methods. If the message is deemed critical, the message is logged, and optionally, key individuals are paged. If the message is an unusual message, i.e., one which has not been seen before, it is written to a log file for later review. If the message is a well known message, and not deemed critical to the continued operation of the system, the message is dropped. This data reduction capability is one of the most powerful functions of BASEMM. As an example, on one day BASEMM received 17,000 messages and reported on 120 of them. Without the data reduction capability of BASEMM the firewall managers would be swamped with data.

Regular review of log file is performed on a 24x7 basis by the Network Management group. This group has been provided with procedures for reviewing the log files and looking for anomalous messages. In addition, throughout the business day, a summary of unreconcilable messages is emailed to appropriate Business Continuity individuals.

Defining System Administration Processes

All traditional system administration functions on the firewalls are performed by the Business Continuity district except those operations requiring physical access to the systems not physically co-located with members of the district. For those remote systems, there is a limited set of individuals (site administrators) at the remote site which have physical access to the firewall. Physical security is strongly enforced at all firewall locations, requiring, at a minimum, a card-key to enable auditing of physical access.

The root password is controlled and only known to the Business Continuity district. Any administration requiring root authority is the responsibility of Business Continuity. Root logins are only permitted from the console. For remote sites, the root password is sealed in a tamper-evident container, and stored in a safe located in a physically secured area of the building. This is to allow the site administrator to have root access in an emergency situation in which no one from the Business Continuity district is available. Examples of this may include, a power outage, or an emergency change to the firewall rules. Once this password has been compromised, a new root password is generated and secured.

The ssh tool is installed on all firewall systems. Ssh is a replacement for the set of Berkeley r-commands, with the enhancement that all sessions can be encrypted, and no passwords are sent in the clear over the network. Ssh is really the key to enabling remote administration of the firewall systems. The use of ssh combined with token authentication also enables secure system administration to be performed from home. Ssh also allows for use of "one-time" passwords and because all traffic is encrypted greatly reduces the chance of a compromise. All methods of remote access are removed from the firewall machines leaving ssh as the sole method of remote access.

Regular scans using several IP Scanners and intrusion attempt tools are also performed. Bellcore's Pingware is used to scan for general Unix security holes. Netcat (or nc) is used to scan servers protected by the firewall to assure they only offer appropriate services. IP Watcher from EnGarde monitors specific connections and traffic patterns when required.

A change management process has also been put into place to assure proper notification of planned changes. There are two classes of changes which can be made to the firewall, confidential and non-confidential. Confidential changes include security related changes, and any firewall rule changes. Confidential changes are communicated in a secure manner and require approval from both the Business Continuity director, and the director of the client organization. Non-confidential changes include general hardware changes and changes which affect connectivity to BAInet. Non-confidential changes are scheduled via the Bell Atlantic standard change control process, and require approval from only the director of Business Continuity. The Bell Atlantic standard change control process requires review of a change by a change control board. After a change has been completed a standard checklist is used to verify that the proper levels of functionality and security are in place.

Summary

Our experience has shown that distributed firewalls can be securely and effectively managed by a centrally located organization. It is hoped that other individuals can benefit from our experience.

Material discussed in this paper can be obtained at ftp.bell-atl.com:/BC. There are ASCII, postscript and MS Word versions of the entire paper and a copy of the slides in Powerpoint format at this site. The paper can also be viewed in HTML format at http://www.bell-atl.com/BC.

Acknowledgments

Thanks to John Conner, Steve Geria, Bill Hill, Bill Irwin, Mark Krause, Harry Leung, Ming Ma, and Brad Passwaters for providing suggestions which greatly enhanced this paper.

Author Information

Mark Miller is currently employed at Bell Atlantic in Silver Spring, MD as a Distinguished Member Technical Staff. He has worked for six years in the areas of Unix system administration, Unix system programming and Internet security. While at Bell Atlantic, he was awarded the Presidents Award for Outstanding Service for work performed to enhance network survivability. He was co-creator of the first Bell Atlantic Web server. He has done graduate study at James Madison University, and received his bachelors degree in Mathematics and Computer Science from Eastern Mennonite University. Reach him via US Mail at 13101 Columbia Pike, Room 208H, Silver Spring, MD 20904, USA. His electronic mail address is <marcus@bell-atl.com>.

Joe W. Morris retired from Federal service after 20 years. While assigned to the National Security Agency, he was a senior computer security officer, senior system administrator. At the agency, he won a Scientific Achievement Award in 1991, and the prestigious Directors' Productivity Improvement Award the following year. Both awards were for network firewall solutions. He has won numerous other DoD awards as well. Since joining Bell Atlantic in January 1994, Joe has introduced a number of technical innovations. He was co-creator of the first Bell Atlantic Web server. He created the Bell Atlantic Server Error Message Monitor (BASEMM, used to monitor major projects) and the NOVA Client Outage Notification Service (NCONS), and also automated metric creation from the NCONS database. Joe is currently assigned to firewall and other security technology projects. Reach him via US Mail at 13101 Columbia Pike, Room 209B, Silver Spring, MD 20904, USA. His electronic mail address is <joe@bell-atl.com>.

Bibliography

Centerpoint Software. Firewall-1 2.0 documentation. Available with the Firewall-1 product.

D. Brent Chapman and Elizabeth D. Zwicky. *Internet Firewalls and Network Security*. Oreilly & Associates, Sebastopol, CA 12995.

William R Cheswick and Steven M. Bellovin. *Firewalls and Internet Security, Repelling the Wily Hacker*. Addison-Weley Publishing Company, Reading, MA, 1994.

R. Christopher E. Hare and Karanjit Siyan. *Internet Firewalls and Network Security*. New Riders Publishing, Indianapolis, IN 46290.

Dan Farmer and Eugene H. Spafford. "The COPS security checker system". In *USENIX*

Conference Proceedings pages 165-170, Anaheim, CA, Summer 1990. *A package to audit systems for vulnerabilities. The paper is available for anonymous ftp from ftp.cs.purdue.edu as /pub/spaf/security/COPS.PS.Z.*

Gene Kim and Eugene H. Spafford. *The design and implementation of Tripwire: A file system integrity checker.* Technical Report CSD-TR-93-071, Purdue University, 1993. *A package to audit systems for vulnerabilities and evidence of hacking attacks. This paper is available for anonymous ftp from ftp.cs.purdue.edu as /pub/spaf/security/Tripwire.PS.Z.*

Joe Morris. "Implementing Active Data Network Security at Bell Atlantic". In *Computer, Network and Information Security Symposium Proceedings,* Baltimore, MD, June 1996.

Joe Morris. "The Bell Atlantic Trusted Network Segment: A Firewall Solution". *Practicum for System Analysis and Design*, Potomac College, Washington, D.C., June 1996.

David R. Safford, Douglas Lee Schales, and David K. Hess. "The TAMU security package: An ongoing response to Internet intruders in an academic environment". In *Proceedings of the Forth Usenix UNIX Security Symposium*, pages 91-118, Santa Clara, CA, October 1993. *A detailed look at a hackers activities in a university environment – and what they did to stop them. The paper is available for ftp as part of the TAMU security package (Tiger).*

Wieste Venema. "TCP WRAPPER: Network monitoring, access control and booby traps". In *Proceedings of the Third UNIX Security Symposium*, pages 85-92, Baltimore, MD, September 1992. Available for ftp from ftp.win.tue.nl at /pub/security/tcp_wrapper.ps.Z .

Tatu Ylonen. *Secure Shell*. A package that replaces the Berkeley "r commands" (rsh, rlogin) with cryptolgic enhancements. Available from http://www.cs.hut.fi/ssh .