FireWall-1

Internet Connectivity Security Solution

A White Paper

FireWall-1™

Table of Contents

Executive Summary

When you connect your local network to the Internet the single most important measure you can take to prevent break-ins is to set up a FireWall.. This white paper describes the architecture and unique characteristics of FireWall-1 Internet gateway. The document outlines the major characteristics that enable FireWall-1 to establish full, transparent, and true Internet connectivity using the entire range of Internet protocols while ensuring network security.

Two main methods are currently used to establish Internet firewalls:

- Application gateways: These are secure but inefficient, either non-transparent to users and applications or hard to set up and manage. Only a limited set of applications is supported and special tailoring is needed for each one.
- Packet Filtering: This method is insecure but more efficient. It is comprehensive and transparent to many protocols and applications. Traditional packet filters are stateless, have only a low-level protocol understanding, and are difficult to set up and verify. Lack of auditing mechanisms is also a major drawback.

FireWall-1 combines both methods to create an efficient, generic and secure packet filtering engine. It is capable of application-level security, unified support and handling of all protocols, and auditing and alerting. Its operation is also transparent to users and to system setup.

In addition to the filtering technology, FireWall-1 includes an object-oriented graphical user interface that enables simple and flexible management and configuration.

1. Internet Firewalls Technology

It is critical that you secure your network against break-ins when you connect it to the Internet or other network.. To securing the Internet link, you must put a firewall system between the local network and the Internet. The firewall's role is to ensure that all the communication between an organization's network and the Internet conforms to the organization's security policies. Additional means, such as authentication and privacy enhancements might follow and complement firewalls, but stopping the fire from spreading into the private network is the first step.

Two major techniques are used to build an Internet firewall:

- Application-level and circuit gateways
- Packet-filtering gateways

The major difference between the two techniques (which are commonly used in conjunction) lies in the flow of communication. A packet-filter gateway acts as a router between the two networks; as packets flow from their source to the destination, the gateway either forwards or blocks them. packets. With application and circuit gateways, all packets are addressed to a user-level application on the gateway that relays the packets between the two communication points. In most application gateway implementations, additional packet filter machines are required to control and screen the traffic between the gateway and the networks. A typical configuration includes two routers with a bastion host in the middle, to serve as the application gateway.

1.1 **Application-Level and Circuit Gateways**

For each application relayed, application-level gateways use a specific, special-purpose code. Application gateways can provide a high level of security, though they suffer from a number of deficits: only a limited number (usually only a small basic subset) of the applications and services are supported. In order to use the application gateways, the users must have to log into the gateway machine or install a specific client application that uses the application gateway for each application they intend to use. Each gatewayed application is a separate, proprietary piece of software and requires its own set of management tools and permissions.

Circuit gateways provide a more general way to implement application gateways. They might support some TCP applications, but not all. Circuit gateways do not support other protocols. However, users are still forced to use and install different client applications or change their work habits. Installing client applications on each internal computer could be a cumbersome task since the internal network is typically heterogeneous with respect to platforms, operating systems, versions, etc.

Network performance is also affected by both application gateways and circuit gateways; each packet must be processed twice by all communication layers, and requires user-level processing and context switching. It should also be noted that the application gateway computer itself (bastion-host or dual-homed gateway) remains exposed to the network, and additional means, such as packet-filtering should be implemented to protect it. This typically results in limiting the available service and also requires additional hardware.

1.2 Packet-Filtering Gateways

Packet-filtering technologies provide an efficient and general way to control any type of network traffic and applications. They require no changes in client applications, no specific applications management nor installation, and no additional hardware. Using a single, unified packet-filtering engine, the entire network traffic is processed and forwarded or blocked from a single point of control.

Historically, packet filtering technologies have not addressed all the security requirements. Only basic and insufficient information is available for filtering (e.g., only source and destination address and port numbers). The number of rules is limited, and a high performance penalty is paid when many rule instances are used. Lack of context or state information has eliminated the possibility of using packet filters for datagram-based protocols like User Datagram Protocol (UDP) and Remote Procedure Call (RPC). Auditing and alerting mechanisms were also missing in most cases. Previous packet-filtering technologies also suffered from poor management interfaces. Implementing them requires a high level of understanding of the communication internals and the writing of low-level bit and byte code, making these technologies very hard to change and adapt and. Some packet filters are implemented inside a router hardware, thus limiting computing power and filtering capabilities. Others are implemented as software packages that filter the packets in the user-level process. This method requires multiple data copies, expensive delays, and context switches. It also delivers lower throughput.

Most existing Internet firewalls use a combination of a packet-filter screening computer or a hardware-router for controlling the lower layers of communication, and application gateways for the enabled applications. This setup provides only limited, non-transparent and non-flexible connectivity, yet it requires a high price in setup, management, and expertise.

1.3 FireWall-1

FireWall-1, from SunSoft , was developed by CheckPoint Software Technologies Ltd. and combines the efficiency of a general-purpose solution for all network protocols, with application level-savvy. Logging and alerting mechanisms are also included, and an easy installation and setup procedures are provided. On top of this unique protocol independent technology, a simple, intuitive, object-oriented user interface enables an easy, flexible, and uniform way of implementing an organization's global security policy. The following sections detail the functions of these properties.

2. *FireWall-1 Overview*

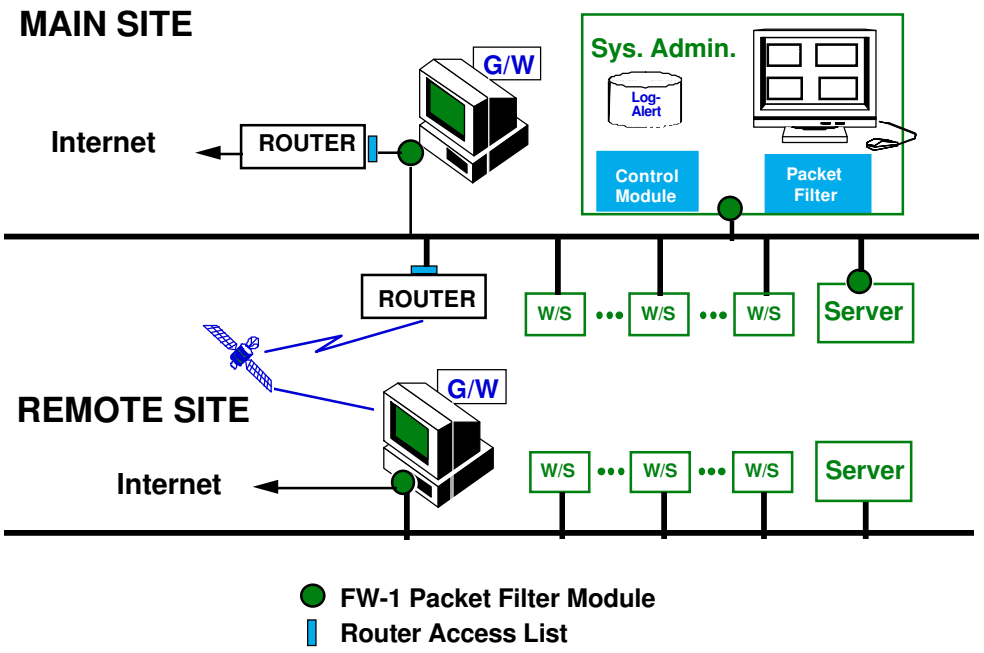
The FireWall-1 Internet Gateway acts as a router between an organization's internal networks and the Internet. All the network traffic between the organization's internal network, Internet sites, and the application gateways between them is routed through the FireWall-1 gateway. This ensures full security coverage of the entire spectrum of Internet protocols and services, and ensures that each and every packet is screened and verified to comply with the organization's security policy.

FireWall-1 is composed of two major components:

- Packet-Filter Modules
- Control Module

A single control module can control and monitor multiple packet-filter modules. The packet filter module operates independently of the control module, providing on-going, simple, powerful, and reliable packet filtering. Packet filter modules can operate on additional Internet gateways, inter-departmental gateways, and critical servers, thus providing peripheral defense as well as in-depth security and compartmentalization.

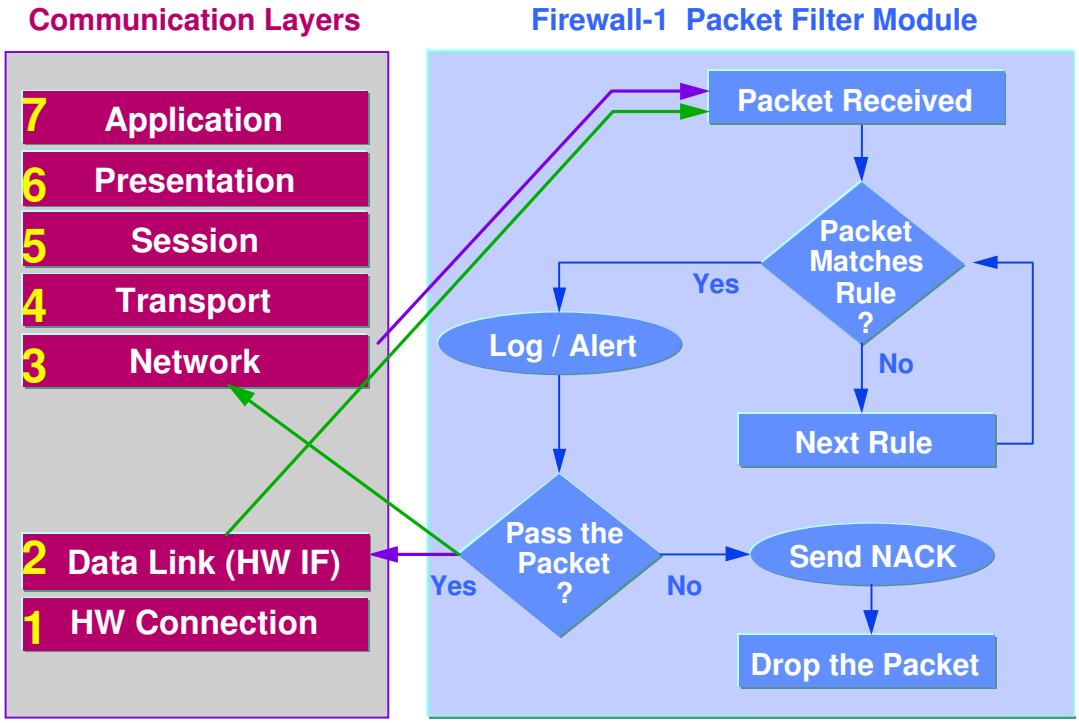
The control workstation and packet-filter module can reside on the same gateway machine or on two different hosts. In the latter case, communication between the two is authenticated, with a one-time-password authentication scheme.



3. *FireWall-1 - Packet Filter Module Architecture*

The FireWall-1 packet-filter module resides on the gateway host, and acts as a security router between the protected networks. The packet-filter module is plugged in between the Data Link and the Network layers (layers 2 and 3). The data link is the actual network interface card (NIC) and the network link is the first layer of the protocol stack (e.g., IP). Inbound and outbound packets on the gateway are intercepted and subjected to the security rules defined and installed in the packet-filter module. Filtering at this layer ensures that no packet is processed by the various protocol stack layers before it is verified as complying with the security policy.

Packets that are not explicitly to be accepted by the security policy are simply dropped. (**"That which is not expressly permitted is prohibited."**).



3.1 Packet-Filtering Technology

FireWall-1 packet filtering technology includes a number of unique characteristics:

3.1.1 Context and State Information

Context and state information can be extracted from communication packets or the operating system and can be used to securely relay protocols like UDP, RPC, etc. (Examples follow).

3.1.2 Lists and Objects

Network and services objects can be grouped together and stored in hash-based access lists (static and dynamic), providing high-level definitions, efficient management and processing, and adaptive and intelligent filtering.

3.1.3 Data Access

Full-packet context is available for screening from lower layers (like interface and address data) up to the higher application layers (layers 2 to 7 in the ISO model).

3.1.4 Programmable Filter Module

Security rules, application knowledge, context information, and packet data are combined into a powerful machine that can implement any security policy based on all needed parameters and logical expressions.

3.1.5 Protocol Independence

The generic and flexible underlying packet filter module is capable of learning and understanding any protocol, as well as adapting to newly defined protocols and applications. This is achieved by using high-level definitions, and it requires no code changes.

3.1.6 Status Reporting

The FireWall-1 packet filter module also includes robust status, auditing, and alerting capabilities. A summary of status information and network traffic is available in the System Status Monitor and can also be obtained using Simple Network Management Protocol (SNMP) for incorporation into any network management platform.

3.1.7 Auditing and Alerting

Every communication attempt can be subject to logging and alerting. Log and alert formats and actions are open and configurable. The standard formats contain the source and destination of the communication, the service attempted, protocol used, time and date, source port, action carried (communication accepted, rejected), log and alert type, and the packet-filter module originating the log. Any Information about any communication attempt can be logged or used to trigger an alert (e.g., pop up a window, send a mail message, activate a user defined action-program, or activate a trap).

3.1.8 Simple and Efficient

The FireWall-1 packet filter module is implemented as an autonomous, compact, code module. It resides inside the operating system kernel as a loadable kernel-level module. The loadable kernel module requires no system reconfiguration or even re-boot to install. Because it is inside the operating system kernel, it is efficient, introducing no additional memory requirements or context switch overheads. This configuration also makes the module harder to tamper with or bypass.

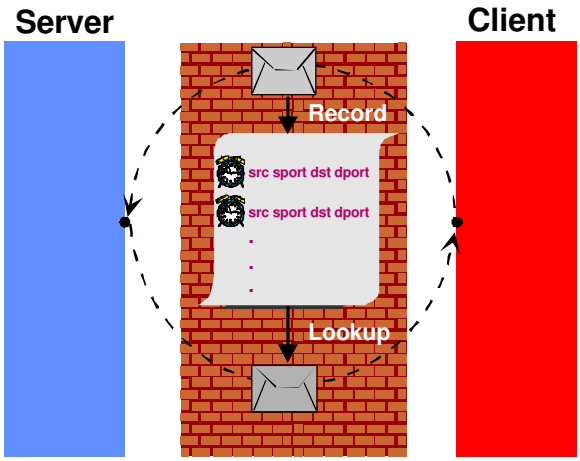
3.2 Benefits and Examples

Combining all these technological advantages enables FireWall-1 to handle securely all Internet protocols. Following are a few examples of common Internet protocols and applications and their handling by filtering technology. It should be noted that these protocols can not be secured properly using traditional packet filtering-systems.

3.2.1 UDP Applications

Problem: UDP is a packet-based, connection-less protocol. Unlike connection-based protocols (like TCP), there is no distinction between the originator of the request and the response. UDP-based applications (like WAIS, Archie, Domain Name Services) were therefore difficult to filter. Old packet-filtering techniques simply eliminated UDP connections or opened a large portion of the UDP range to bi-directional communication, exposing the internal network to attacks.

Solution: FireWall-1 solves the problem by keeping a virtual connection on top of UDP communications. This is done by keeping state information for each UDP connection on the gateway. Every UDP request packet permitted to cross the firewall is recorded. Each incoming UDP packet is looked up in the list of pending connections. Only when the packet is a response to a request it is delivered. This ensures that all attacks are blocked, while UDP applications can be used securely.



3.2.2 Outbound FTP Connections

Problem: Although the File Transfer Protocol (FTP) is one of the most basic and common TCP-based Internet protocols, it involves a level of complexity difficult for existing packet-filtering implementations to handle. After the client initiates an FTP session, the server establishes a new back-connection to the client. This connection goes from the server (outside the firewall boundaries) to a dynamically allocated port number on the client machine. Since this port number is not known in advance (and many of these clients and ports open and close frequently) old packet filter implementations open the entire range of high-numbered ports (>1023) for incoming connections. This is done to allow the back-FTP connections, but it enables numerous ways of attacks throughout most of the application spectrum.

Solution: FireWall-1 tracks the FTP session, looking at FTP application level data. When the client requests that the server generate the back-connection (An FTP PORT command), FireWall-1 records this request. Then, when the back-connection is attempted, it is checked against the pending requests and allowed only as specified by the internal FTP client. The list of connections is maintained dynamically so that only the required FTP ports are opened, but only during the FTP session and at no other time.

3.2.3 RPC Based Protocols

Problem: RPC-based services do not use pre-defined port numbers; they use mostly UDP and have no fixed packet structure. This makes them impossible to identify or handle by traditional packet filters. Application gateways usually fail to handle them as well.

Solution: FireWall-1 uses a combination of the previous techniques: RPC port numbers are tracked dynamically and transparently from the various port mappers in the system; The application information is extracted from the packet in order to identify the program used, overcoming the packet's varying structure; and a track of UDP connections is maintained, as described earlier.

3.2.4 Mosaic, World Wide Web, and Gopher

The World Wide Web project and the Mosaic front-end exhibit a new level of connectivity in cyber space by enabling users to explore Internet resources using simple windows-based applications from any desktop.

Problem: Mosaic is difficult to control because it combines many underlying protocols (HTTP, FTP, WAIS, Archie, Gopher, and others), and it has a broad range of client implementations from any windows-based desktop. It exposes network security to the risks of each of the underlying protocols (UDP, FTP, etc.) because it requires all of the services to be accessed from each and every desktop on the network.

Solution: The standard handling of all these protocols by the FireWall-1 packet filter module (as demonstrated previously) allows simple, transparent, and secure use of Mosaic by any client desktop on the network without any additional configuration or security efforts.

3.2.5 Trojan Horses and Viruses

Problem: A common way of attacking a computer system is by planting a “Trojan horse” or virus in the computer system. The subject program accepts communication from the outside world, enabling system break-ins, or it initiates communication to the outside world, exporting classified information. Trojan horses are sometimes planted in server applications. They can also be acquired when you import software on the Internet or use magnetic media. While there are good ways to minimize the problem, there is now way to eliminate it completely

Solution: FireWall-1 seals all the communication channels and logs all communication attempts, both incoming and outgoing. This enables blocking and identifying such attempts promptly, tracking the source of the problem, and eliminating it.

The above examples provide only a quick look into the difficulties of handling various Internet protocols with a firewall system and the ways FireWall-1 ensures reliable operation and system security. This is only a partial list of protocols currently supported. Other protocols are handled in a similar way. More protocols are handled along the lines of top-level network security together with correct protocol handling and user transparency. As new protocols and applications are introduced, they are being added regularly to the system, without requiring additional programming or new software modules. Incorporating new protocols in the system is done using the GUI forms, or using the FireWall-1 filter definition language.

3.3 **Performance**

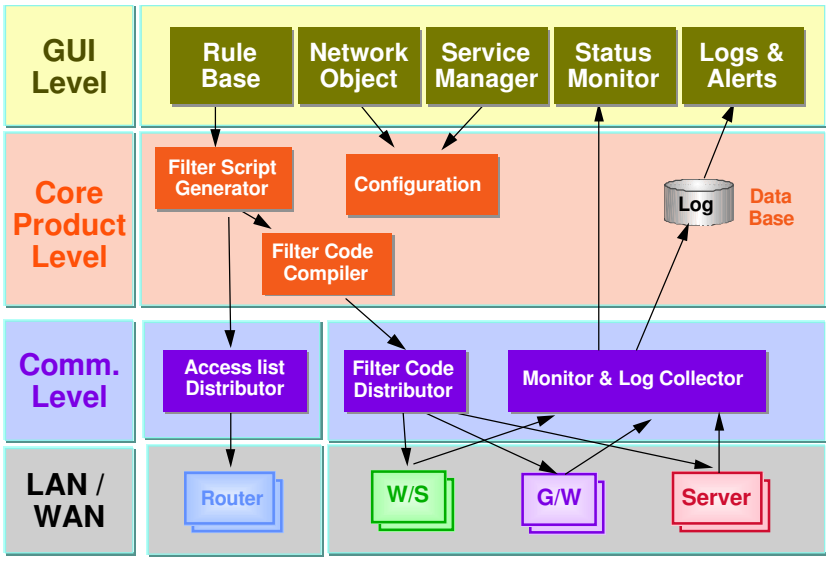
FireWall-1 was designed to be simple and effective; various optimization techniques are used to achieve high performance:

- Running inside the operating-system kernel imposes negligible overhead in processing, and no context switching is required. This achieves low-latency operation.
- Advanced memory management techniques, such as caching and hash tables, are used to unify multiple instances objects and to access data effectively.
- Generic and simple filtering mechanisms combined with a packet filter optimizer ensures getting the maximum out of today's RISC processors.

Network performance degradation was too small to be measured when operating at full LAN speed (10Mb/sec) on the lowest-end SPARCstation machine; that included communication latency and network bandwidth. Performance impact on the filtering gateway host was also negligible. Judging by these test results, we deduce the performance hit to be negligible and practically non-existent for current Internet speeds (56kb or T1 - 1.5Mb/sec speeds and up to full 10Mb/sec speed)

4. FireWall-1 Control Model

The FireWall-1 control workstation is used to configure the enterprise-wide security policy, control the communication gateways (packet filter modules), and view logging and alerting information. The control workstation runs an OPEN LOOK X11R5 GUI, supporting the full operation and verification of system operation. A set of command-line utilities enable operation from a standard computer terminal. The GUI is composed of the following modules:



4.1 Network Objects Manager

Defines the network objects directly involved in the security scheme policy:

- Networks and sub-networks
- Servers and workstations
- FireWall-1 hosts and gateways
- Routers
- Internet domains

Every object has a set of attributes that defines its characteristics, such as network address, subnet-mask, etc. Object attributes are extracted from the network databases, like the hosts and networks files, Network Information Services (NIS / Yellow Pages) network databases and the Internet domain service. SNMP agents are used for extracting additional information, including the interface and network configuration of hosts, routers and gateways. Objects can be combined in groups, creating higher-level hierarchies.

4.2 Services Manager

The services manager defines the services known to the system and addressed specifically in the security policy. It should be noted that all network services are screened and controlled, even those that are not defined. The system comes pre-loaded with a comprehensive set of TCP/IP and Internet services, including the following:

- Standard arpa-services: Telnet, FTP, Simple Mail Transfer Protocol (SMTP), etc.
- Berkeley r-services: rlogin, rsh, etc.
- SunRPC services: NIS/yellow pages, Network File System (NFS), etc.
- Advanced Internet protocols like HTTP, Gopher, Archie and many more

- IP services: Internet Control Message Protocol (ICMP), Routing Internet Protocol (RIP), SNMP, etc..

Introducing a new service to the system is done by selecting the service type, and setting the service's attributes in the proper form, service types include:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Remote Procedure Call (RPC)
- Others to enables instant definition of services and protocols that do not conform to the standard set of attributes. Services are defined using simple expressions and macros.

Services can be grouped to in families and hierarchies. Examples: NFS (The mount program, NFS-server, lock manager), NIS/Yellow-pages (ypserv/ypbind), and Mosaic (HTTP, FTP, Archie, gopher, etc.).

4.3 The Rule-Base Manager

After defining the network objects and browsing through the network services, it's time to put them together and enter the security policy. Rules expressed under the FireWall-1 rule-base use high level definitions and objects, truly representing the security policy.

Every rule is composed of four parts: match; action; tract; and target.

- Match
  Specifies which communication attempts are included in this rule. It includes the source of the communication, the destination, and the services attempted. A match might say: “All communication initiated from the internal-network going to the Internet”. Another match can be: “Internet mail delivery (SMTP) going to the mail servers”.
- Action
  Specifies the handling of this communication attempt. Communication attempts can be accepted (passing the packets transparently), or blocked (rejecting the attempt and generating a negative-acknowledgment connection-refused message, or simply dropping the packet).
- Track
  The tracking mechanism specifies the type of auditing or alerting required:
  - No logging
  - Short format record
  - Long format log record.
  - Alert generation, e.g. pop-up window at the system manager's workstation
  - Sending a Mail message
  - Generating a network management trap
  - Activating any user-defined procedure

- Target
  incorporates the enterprise-wide security rules by enforcing each rule in its appropriate Packet Filter Module or router location. Rules can be enforced on all the gateways, on the destination servers and gateways, on the source gateways and hosts. Multiple selections and exact specification are supported.

Each communication attempt is matched against the rule-base. Rules are matched in the order specified, top to bottom, enabling predictable behavior of the system. A default, last rule always drops communication attempts leaving no place for communication leaks, thus implementing the phrase, That which is not expressly permitted is prohibited."

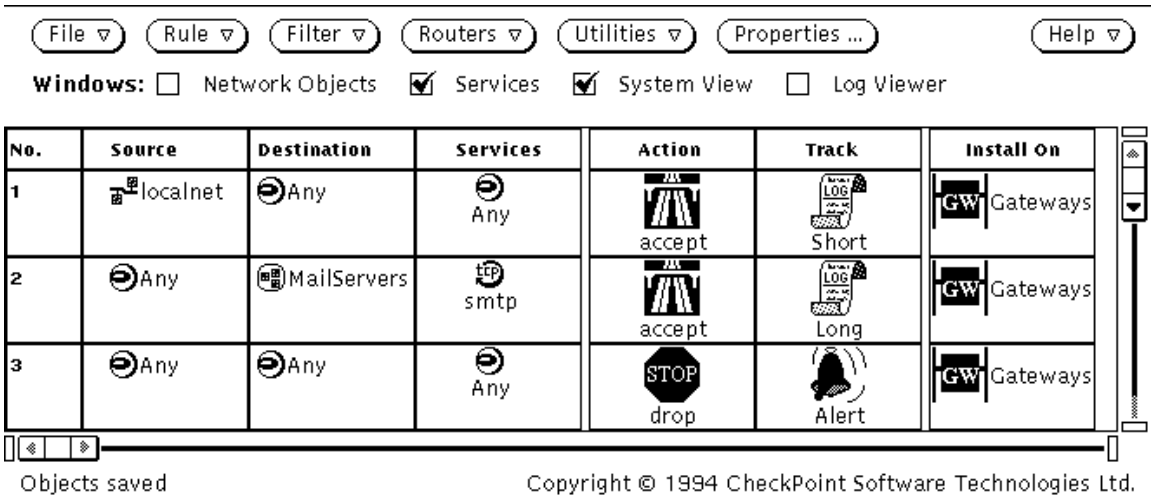
A verification mechanism checks the consistency of the rule base. It includes:

- Heuristic tests
- Definition inconsistencies
- Redundancy and order checking

From the complete rule-base a Filter Script is generated. The Filter Script describes the objects and rules using CheckPoint's filter definition language. The script definition is easy to understand, enabling simple verification of the rule-base. Using the scripts, more sophisticated security policies are easily implemented.

4.4 Example: A True Diode Internet Security Policy

The screen capture below maps basic Internet rule base that allows free and secure outbound connections from an organization's internal network to the Internet; prevents any attacks and communication attempts from the Internet; and provides only incoming mail.



4.5 Routers Access Lists

Routers access lists are generated in the same way: the rule base is verified, and an access list is generated for each router specified. The access list is then distributed to the routers. The tracking mechanism is ignored due to the lack of logging and alerting mechanisms in Cisco routers.

Note: Routers do not implement many of the essential capabilities required to provide secure packet filtering (e.g., they are stateless and look only at TCP/UDP port numbers). See previous sections for details. The limited capabilities of routers access lists prevent many rules from being implemented, and makes other rules insecure or insufficient (UDP, FTP, RPC, and others).

4.6 System Status

The System Status window displays a snapshot of all the FireWall-1 filter modules at any time interval. Status includes filter-module status as well as packet statistics through the module (accepted, blocked, logged, etc.). An SNMP agent is also supported by the packet-filter modules, exporting information and it integrating to other network management platforms.

4.7 Log Viewer

The Log Viewer displays every logged event, including communication attempts, filter installations, system shutdowns, etc. For every event, the relevant information is displayed, including the date and time, originating machine, source and destination of communication, services attempted, action taken, log and alert types, and other specific information. Fields can be displayed or hidden. Color and icons are attached to events and fields yielding a rapid visual representation.

Searching capabilities enable the location of events of interest and the tracking of communication events. Reports are generated by applying selection criteria to chosen fields, providing compound and comprehensive views. Reports can be viewed, exported in an ASCII format, or printed in PostScript. Log files management capabilities are integrated.

On-line viewing features enable real-time monitoring of communication activities and alerts. Nodes that appear in the log can be probed instantly from within the Log Viewer for SNMP information.

5. *Conclusions*

FireWall-1 provides a comprehensive system for controlling and securing an organization's inter-network. Innovative packet-filtering technology brings application-level security capabilities into an efficient packet-filtering engine. FireWall-1 delivers both application and network-level security advantages into a single comprehensive scheme.

Enterprise-wide management tools enable an organization to use advanced technology in a simple and effective way, creating an unmatched security framework for today's and future network security needs.

6. *Specifications*

Platforms:	Sun SPARC or X86 based systems
Operating System:	SunOS 4.1.3 or Solaris 2.3 and above
Window System:	X11R5/OPEN LOOK (OpenWindows 3)
Disk space:	5 Mbytes
Memory:	16MB (Control Module), no special requirements for Packet Filter Module
Network Interface:	All standard Sun Network Interfaces
Routers:	Cisco release 9.1
Media:	3.5″ diskette

United States
2550 Garcia Avenue
Mountain View, CA 94043
TEL 1-800-SUNSOFT
TEL 510-460-3267

Northern Europe
St. John's Court
Easton Street
High Wycombe
Buckinghamshire, HP11 1JX
TEL +44-494-472900
FAX +44-494-472383

Central Europe
c/o Sun Microsystems
85630 Grasbrunn
Germany
TEL +49-89-46008-551
FAX +49-89-46008-555

Southern Europe
13 Avenue Morane-Saulnier
BP53
78142 Velizy Cedex
France
TEL +33-1-3067-5477
FAX +33-1-3067-5068

Japan
SBS Tower
4-10-1, Yoga
Setagaya-ku Tokyo, 158
TEL +81-3-5717-5017
FAX +81-3-5717-2570

FireWall-1, developed by CheckPoint Software Technologies Ltd., is now available from, SunSoft.

Specifications are subject to change without notice.

 PN 94320-001