TECH NOTE

Stateful Inspection Technology

Security Requirements

In order to provide robust security, a firewall must track and control the flow of communication passing through it. To reach control decisions for TCP/IP based services (e.g., whether to accept, reject, authenticate, encrypt and/or log communication attempts), a firewall must obtain, store, retrieve and manipulate information derived from all communication layers and from other applications.

It is not sufficient to examine packets in isolation. State information—derived from past communications and other applications—is an essential factor in making the control decision for new communication attempts. Depending upon the communication attempt, both the communication state (derived from past communications) and the application state (derived from other applications) may be critical in the control decision.

Thus, to ensure the highest level of security, a firewall must be capable of accessing, analyzing and utilizing the following:

- Communication Information
  Information from all seven layers in the packet

- Communication-derived State
  The state derived from previous communications. *For example, the outgoing PORT command of an FTP session could be saved so that an incoming FTP data connection can be verified against it.*

- Application-derived State
  The state information derived from other applications. *For example, a previously authenticated user would be allowed access through the firewall for authorized services only.*

- Information Manipulation
  The ability to perform logical or arithmetic functions on data in any part of the packet

Stateful Inspection Technology

Stateful Inspection, invented by Check Point Software Technologies, has emerged as the industry standard for enterprise-class network security solutions. Stateful Inspection is able to meet all the security requirements defined above while traditional firewall technologies, such as packet filters and application-layer gateways, each fall short in some areas. (See Table 1.)

With Stateful Inspection, packets are intercepted at the network layer for best performance (as in packet filters), but then data derived from all communication layers is accessed and analyzed for improved security (compared to layers 4–7 in application-layer gateways). Stateful Inspection then introduces a higher level of security by incorporating communication- and application-derived state and context information which is stored and updated dynamically. This provides cumulative data against which subsequent communication attempts can be evaluated. It also delivers the ability to create virtual session information for tracking connectionless protocols (e.g. RPC and UDP-based applications), something no other firewall technology can accomplish.

CHECK POINT™
Software Technologies Ltd.

Check Point FireWall-1: Extensible Stateful Inspection

Check Point FireWall-1's Stateful Inspection architecture utilizes a unique, patented INSPECT Engine which enforces the security policy on the gateway on which it resides. The INSPECT Engine looks at all communication layers and extracts only the relevant data, enabling highly efficient operation, support for a large number of protocols and applications, and easy extensibility to new applications and services.

The INSPECT Engine is programmable using Check Point's powerful INSPECT Language. This provides important system extensibility, allowing Check Point, as well as its technology partners and end-users, to incorporate new applications, services, and protocols, without requiring new software to be loaded. For most new applications, including most custom applications developed by end users, the communication-related behavior of the new application can be incorporated simply by modifying one of FireWall-1's built-in script templates via the graphical user interface. Even the most complex applications can be added quickly and easily via the INSPECT Language. Check Point provides an open application programming interface (API) for third-party developers and regularly posts INSPECT Scripts to support new applications on the Check Point Web site at http://www.checkpoint.com.

The INSPECT™ Engine

When installed on a gateway, the FireWall-1 INSPECT Engine controls traffic passing between networks. The INSPECT Engine is dynamically loaded into the operating system kernel, between

INSPECT Virtual Machine

FireWall-1's patented INSPECT Virtual Machine intercepts, analyzes, and takes action on all communications before they enter the operating system of the gateway machine, ensuring the full security and integrity of the network. Cumulative data from the communication and application states, network configuration and security rules are used to enforce the enterprise security policy.

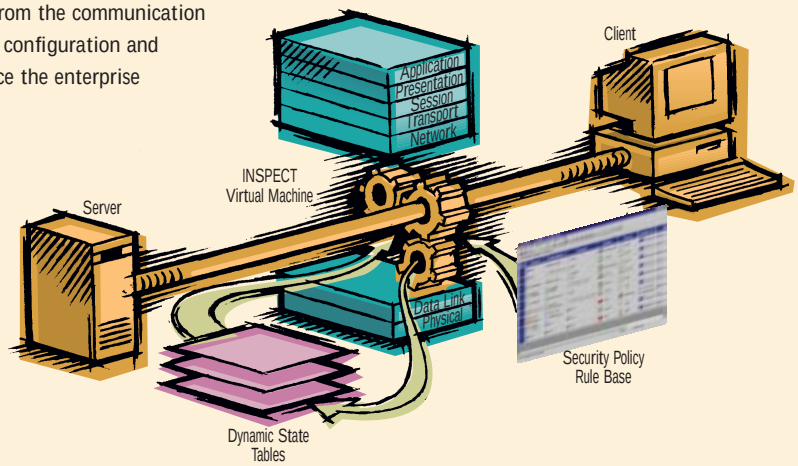


Table1: Comparison of Firewall Technologies

Firewall Capability	Packet Filters	Application-layer Gateways	Stateful Inspection
Communication Information	Partial	Partial	**Yes**
Communication-derived State	No	Partial	**Yes**
Application-derived State	No	Yes	**Yes**
Information Manipulation	Partial	Yes	**Yes**

the Data Link and the Network layers (layers 2 and 3). Since the data link is the actual network interface card (NIC) and the network link is the first layer of the protocol stack (for example, IP), FireWall-1 is positioned at the lowest software layer. By inspecting at this layer, FireWall-1 ensures that the INSPECT Engine intercepts and inspects all inbound and outbound packets on all interfaces. No packet is processed by any of the higher protocol stack layers, no matter what protocol or application the packet uses, unless the INSPECT Engine first verifies that the packet complies with the security policy.

Because the INSPECT Engine has access to the 'raw message', it can inspect all the information in the message, including information relating to all the higher communication layers, as well as the message data itself (the communication- and application-derived state and context). The INSPECT Engine examines IP addresses, port numbers, and any other information required in order to determine whether packets should be accepted, in accordance with the defined security policy.

FireWall-1's INSPECT Engine understands the internal structures of the IP protocol family and applications built on top of them. For stateless protocols such as UDP and RPC, the INSPECT Engine creates and stores context data, maintaining a virtual connection on top of the UDP communication. The INSPECT Engine is able to extract data from the packet's application content and store it to provide context in those cases where the application does not provide it. Moreover, the INSPECT Engine is able to dynamically allow and disallow connections as necessary. These dynamic capabilities are designed to provide the highest level of security for complex protocols, but the user may disable them if they are not required.

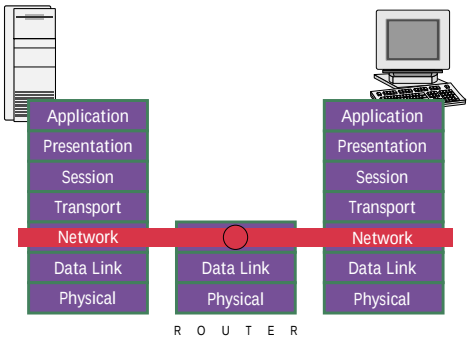
The INSPECT Engine's ability to look inside a packet enables it to allow certain commands within an application while disallowing others. For example, the INSPECT Engine can allow an ICMP ping while disallowing redirects, or allow SNMP gets while disallowing sets, and so on. The INSPECT Engine can store and retrieve values in tables (providing dynamic context) and perform logical or arithmetic operations on data in any part of the packet. In addition to the operations compiled from the security policy, the user can write his or her own expressions.

Stateful Inspection vs. Traditional Firewall Architectures

Firewall Technologies

Packet Filters

Packet filters, historically implemented on routers, filter on user defined content, such as IP addresses. They examine a packet at the network layer and are application independent, which allows them to deliver good performance and scalability. They are the least secure type of firewall, however. The reason is that they are not application aware—that is, they cannot understand the context of a given communication, making them easier for hackers to break.

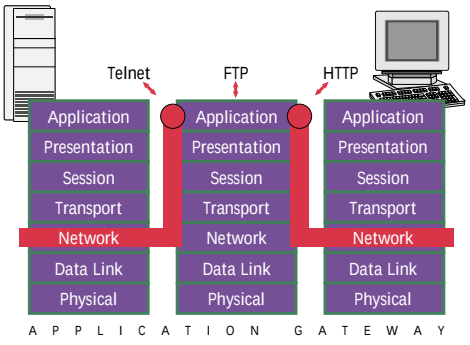


PROS
- Application Independence
- High Performance
- Scalability

CONS
- Low Security
- No Screening Above Network Layer (No 'state' or application-context information)

Application-Layer Gateways

Application gateways improve on security by examining all application layers, bringing context information into the decision process. However, they do this by breaking the client/server model. Every client/server communication requires two connections: one from the client to the firewall and one from the firewall to the server. In addition, each proxy requires a different application process, or daemon, making scalability and support for new applications a problem.

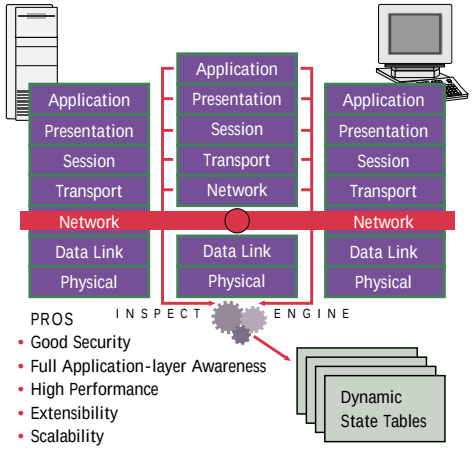


PROS
- Good Security
- Full Application-layer Awareness

CONS
- Poor Performance
- Limited Application Support
- Poor Scalability (Breaks client/server model)

Stateful Inspection

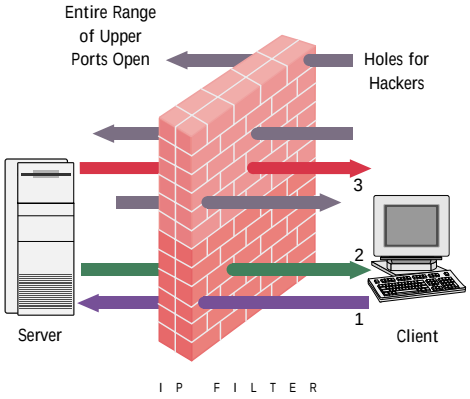
Check Point FireWall-1's Stateful Inspection overcomes the limitations of the previous two approaches by providing full application-layer awareness without breaking the client/server model. With Stateful Inspection, the packet is intercepted at the network layer, but then the INSPECT Engine takes over. It extracts state-related information required for the security decision from all application layers and maintains this information in dynamic state tables for evaluating subsequent connection attempts. This provides a solution which is highly secure and offers maximum performance, scalability, and extensibility.



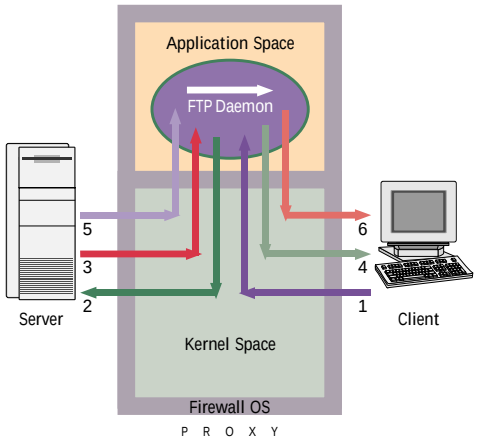
PROS
- Good Security
- Full Application-layer Awareness
- High Performance
- Extensibility
- Scalability
- Transparency

FTP Examples

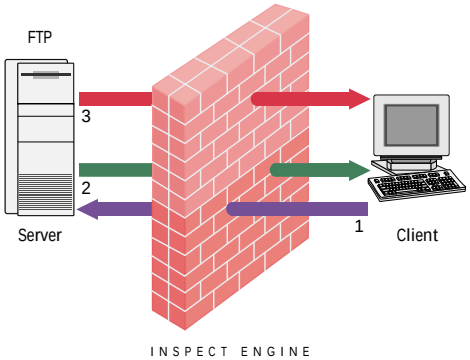
Packet filters have two choices with regard to outbound FTP connections. They can either leave the entire upper range (greater than 1023) of ports open which allows the file transfer session to take place over the dynamically allocated port, but exposes the internal network, or they can shut down the entire upper range of ports to secure the internal network which blocks other services. This trade-off between application support and security is not acceptable to users today.



In using an FTP proxy, the application gateway duplicates the number of sessions, acting as a proxied broker between the client and the server. Although this approach overcomes the limitation of IP filtering by bringing application-layer awareness to the decision process, it does so with an unacceptable performance penalty. In addition, each service needs its own proxy, so the number of available services and their scalability is limited. Finally, this approach exposes the operating system to external threats.



Check Point FireWall-1's Stateful Inspection tracks the FTP session, examining FTP application-layer data. When the client requests that the server generate the back-connection (an FTP PORT command), FireWall-1 extracts the port number from the request. Both client and server IP addresses and both port numbers are recorded in an FTP-data pending request list. When the FTP data connection is attempted, FireWall-1 examines the list and verifies that the attempt is in response to a valid request. The list of connections is maintained dynamically, so that only the required FTP ports are opened. As soon as the session is closed the ports are locked, ensuring maximum security.



Unlike other security solutions, FireWall-1's Stateful Inspection architecture intercepts, analyzes, and takes action on all communications before they enter the operating system of the gateway machine, ensuring the full security and integrity of the network. Cumulative data from the communication and application states, network configuration and security rules, are used to generate an appropriate action, either accepting, rejecting, authenticating, or encrypting the communication. Any traffic not explicitly allowed by the security rules is dropped by default and real-time security alerts and logs are generated, providing the system manager with complete network status.

Broad Application Support

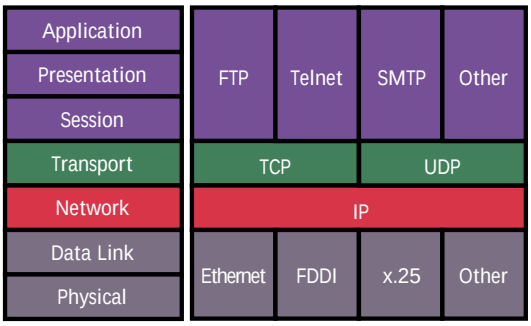
Check Point FireWall-1's Stateful Inspection implementation supports hundreds of pre-defined applications, services, and protocols—more than any other firewall vendor. Support is provided for all major Internet services, including secure Web browsers, the traditional set of Internet applications (e.g. mail, FTP, Telnet, etc.), the entire TCP family, and connectionless protocols such as RPC and UDP-based applications. In addition, only FireWall-1's Stateful Inspection offers support for critical business applications such as Oracle SQL*Net database access and emerging multimedia applications such as RealAudio, VDOLive, and Internet Phone.

Some of the complex protocols uniquely secured by Check Point FireWall-1's Stateful Inspection implementation are described below and in the diagrams on Pages 4 and 5.

Securing Connectionless Protocols such as UDP

UDP (User Datagram Protocol)-based applications (DNS, WAIS, Archie, etc.) are difficult to filter with simplistic packet-filtering techniques because in UDP, there is no distinction between a request and a response. In the past, the choice has been to either eliminate UDP sessions entirely or to open a large portion of the UDP range to bi-directional communication, and thus to expose the internal network.

FireWall-1's Stateful Inspection implementation secures UDP-based applications by maintaining a virtual connection on top of UDP communications. FireWall-1's INSPECT Engine maintains state information for each session through the gateway. Each UDP request packet permitted to cross the firewall is recorded, and UDP packets traveling in the opposite direction are verified against the list of pending sessions to ensure that each UDP packet is in an authorized context. A packet that is a genuine response to a request is delivered and all others are dropped. If a response does not arrive within the specified time period, the connection times out. In this way, all attacks are blocked, while UDP applications can be utilized securely.



TCP/IP services mapped to 7-layer OSI model

Securing Dynamically Allocated Port Connections such as RPC

Simple tracking of port numbers fails for RPC (Remote Procedure Call) because RPC-based services (NFS, NIS) do not use pre-defined port numbers. Port allocation is dynamic and often changes over time. FireWall-1's INSPECT Engine dynamically and transparently tracks RPC port numbers using the port mappers in the system. The INSPECT Engine tracks initial portmapper requests and maintains a cache that maps RPC program numbers to their associated port numbers and servers. Whenever the INSPECT Engine examines a rule in which an RPC-based service is involved, it consults the cache, comparing the port numbers in the packet and cache and verifying that the program number bound to the port is the one specified in the rule. If the port number in the packet is not in the cache (this can occur when an application relies on prior knowledge of port numbers and initiates communication without first issuing a portmapper request) the INSPECT Engine issues its own request to portmapper and verifies the program number found to the port.

Performance

The simple and effective design of FireWall-1's INSPECT Engine achieves optimum performance as follows:

- Running inside the operating-system kernel imposes negligible overhead in processing. No context switching is required, and low-latency operation is achieved.

- Advanced memory management techniques, such as caching and hash tables, are used to unify multiple object instances and to efficiently access data.

- Generic and simple inspection mechanisms are combined with a packet inspection optimizer to ensure optimal utilization of modern CPU and OS designs.

Independent test results indicate that FireWall-1 imposes negligible performance degradation on network traffic and can support data throughput rates exceeding 100 Mbps. In addition, the platform flexibility of FireWall-1 enables customers to scale their security infrastructure to meet the increasing demands of enterprise networks.

How to Contact Us

For product information, visit us at
http://www.checkpoint.com.

Check Point Software Technologies, Inc.
Three Lagoon Drive, Suite 400
Redwood City, CA 94063

Check Point Software Technologies Ltd.
3A Jabotinsky Street, 24th Floor
Ramat-Gan 52520, Israel

P/N 30202200100