

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

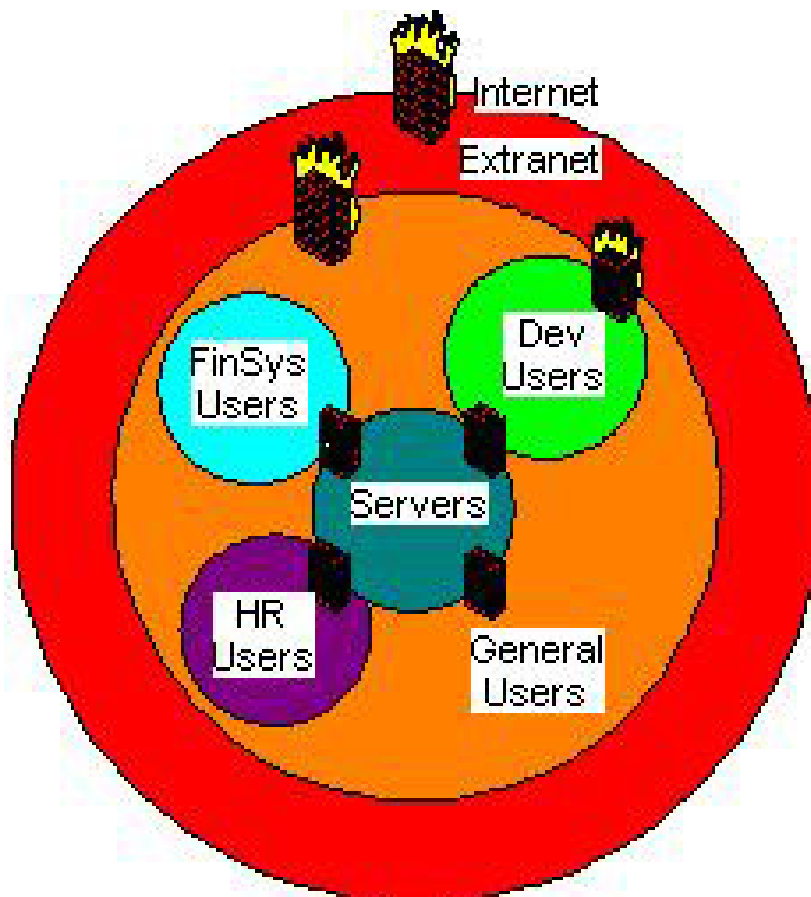
Michael J. Martin

"A firewall is a system or group of systems that enforces an access control policy between two networks." --Firewall FAQ

In Jan 2003 I gave a webcast on "Implementing stateful firewalls for IOS." The presentation briefly covered firewall concepts and implementing a Cisco IOS router as a firewall using the Context Based Access Control (CBAC). This article is the first in "The router is the firewall" series, in which we hope to expand on the concepts covered in the webcast and examine the available options in the IOS firewall for implementing dynamic access control. Since it's always a good idea to have a thorough understanding of the theory behind a product, this month we will discuss firewall theory and practice and review the features available in the IOS firewall. In the next few months, subsequent articles will discuss implementing the available features, and the final article in the series will discuss implementing a secure wireless LAN solution using the IOS firewall feature set and generic wireless access points without WEP (a great solution for those of you who need to implement wireless on the cheap). Now, let's get started with some updated firewall theory.

Layered Defense

Perhaps the most basic tenet of security is the creation of defensive perimeters. The first rule of establishing a perimeter is to establish one that can be monitored and defended. The most common way to accomplish this is by creating a layered defense. The installation doors, trenches, moats, walls, etc. are all examples of the strategy, whether used to protect property, privacy or ideas. The basic concept is the same: Erect structures to block access by an intruder. The idea is to either block, if breached, localize and contain an intruder, thus limiting their ability to inflict damage. Once you are aware of the attack, you can deal with the intrusion and repair the breach.



The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

Implementing a layered defense in a computer network involves placing firewalls between different network segments. Each firewall functions as a parameter point to actively or passively control the access of users by implementing an access policy that controls the amount and kinds of data that can be exchanged between networks. The figure above illustrates a multi-layer model that controls access between different user communities and locally and remotely located common resources. Keep in mind, though, that a firewall, like any blockade, can be overcome with enough time and effort. As a general rule, you should never expose any system that has not been adequately secured to an insecure network, so at a minimum use a firewall between your network and the Internet.

Old Wisdom, New Reality

When firewalls first became in vogue, common wisdom said that an Internet firewall provided more than adequate protection for most network environments. However, with the shift from mostly static desktop systems connected only to a single network to laptop systems that move from the office network to users home networks, client networks, and even public network access points in Starbuck's and McDonald's, security is no longer insured by a single firewall installed on your office network. This new reality makes it almost impossible for administrators to ensure that systems connecting to their networks are secure. Oddly, while the majority of enterprise networks have transitioned to this operational model as broadband and public access points have become widely available, the actual risks were not fully comprehended when making this shift. So while the newly realized gains in productivity have made this move attractive to enterprises, the security risks associated with this new access model has forced many companies to re-examine their network security postures.

Yesterday, the threats were "script kiddies" and techno-barbarians banging at the gate. Most of today's security exploits just run through the firewall. The majority of these are caused by known software bugs that individuals attack, knowing that even though patches exist they probably will not be applied. Worms and viruses embedded in e-mails, freeware "attack drone" applications, peer-to-peer, instant messaging, Internet storage clients, Web enabled java applets -- to just name a few -- all come in through permitted service ports open on the firewall. This paradigm shift in attack has made today's threat "the girl next door" who is unknowingly infecting the neighbors with SQL Slammer and Code Red. Systems transiting different networks and embedded "permitted attacks" are the new reality that security and network administrators must face. The layered defense -- to segment, control and localize events -- for many administrators is an idea that has come of age.

In addition to new threats, enterprise network and security managers must also confront an obstacle almost equally daunting -- exactly how to pay for implementing and managing the needed changes in security posture. The layered defense increases network security by enhancing access control and monitoring, but is expensive in terms of hardware, software and staffing assets. Product offerings from the firewall market leaders such as Check Point, Cisco and Net Stream and are expensive and require expertise to implement and manage. Mainline firewall products are also geared toward protecting the network perimeters, not securing within the network. Within the network, you can use routers with Layer 2 or Layer 3 static access control lists (SACLs) functioning as "choke firewalls" to restrict traffic and service flows to only known networks (see SACLs: Filtering suggestions and ideas). This will help minimize spoofing and DOS attacks.

Enter the IOS Firewall

The IOS firewall fills the gap between mainline firewalls and SACLs, developed for organizations that cannot use a mainline firewall product due to financial or technological (i.e. staffing) constraints. The IOS firewall also provides enterprises a cost-effective means to implement a layered defense utilizing

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

existing hardware and technology with which their network and security staff is already accustomed to working.

The IOS firewall was introduced in IOS v11.2p, as a collection of enhancements for implementing secure traffic processing. The IOS firewall is available for the 1xxx, 2xxx, 3xxx and 7xxx router platforms. In terms of RAM and FLASH, the IOS firewall requires more than the standard IP Plus image. In terms of performance, a typical implementation will see an increase of 10% to 15% CPU utilization. This additional load should not affect performance, provided that your router is adequately powered to handle your current network traffic level. With that said, when initially converting to the IOS firewall, adequate performance monitoring should be put in place. The current IOS v12.2x implementation provides the following functionality:

56 and 3DES IPsec VPN support

- Cisco Consolidated Client support for remote system access using "traditional" IPsec (IOS 12.2.4T or higher needed for CCC VPN client)
- Site to Site VPN support for "bridged" or "routed" connections

User session authentication

- Per user access control via http authentication interface (auth-proxy)
- Per user access control via SSH/telnet authentication interface (lock and key)

Network intrusion detection (ip audit)

- More than 50 common attack signatures, increasing with each major release
- Can log (passive IDS) or block (active IDS) suspicious events
- Integrates reporting with Cisco works, Cisco Secure Policy Manager and syslog

Stateful packet inspection for TCP and UDP sessions (ip inspect)

- Provides generic connection state tracking for any TCP or UDP connection
- Interpretable with network and port address translation (NAT/PAT)
- Provides Layer 7 protocol-specific violation inspection for:
 - CU-SeeMe (7648) peer-to-peer video conferencing
 - RTSP (544) Real Time Streaming Protocol (Multimedia and VoIP)
 - FTP (21) client-server File Transfer Protocol
 - Telnet (23) client-server virtual terminal protocol
 - H.323 (1720) packet-based multimedia communication protocol (VoIP)
 - SIP (5060) Session Initiation Protocol (VoIP)
 - HTTP (80) and Java applet blocking and inspection

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

- Net Show (1755) Microsoft's streaming media platform
- Real Audio (7070) Real Networks' streaming media platform
- Stream Works (1558) streaming media platform (now owned by Real Networks)
- TFTP (69) Trivial File Transfer Protocol
- SMTP (25) Simple Mail Transfer Protocol
- SQLnet (1521) client-server middleware for client-to-database and database-to-database communication
- VDOLive (7000) streaming media protocol
- SUNrpc (111) Sun Microsystems remote procedure call protocol (NFS)
- R-EXEC (512) Berkeley remote command protocol (Unix)
- R-SHELL (514) Berkeley remote shell protocol (Unix)
- MSRPC (135) Microsoft Remote Procedure Call Protocol (provides system-to-system process communication)
- MGCP (2427) Multimedia Control Gateway Protocol (VoIP)

Although the IOS firewall does not offer the expansive feature set of products like Check Point's Firewall 1, it does provide a viable middle-ground solution and is better than the standard packet filtering ACL "choke firewall" implementation.

Firewall Architecture Overview

The IOS firewall's CBAC filtering is a stateful packet inspection engine that extends the router's filtering capability to the application layer (Layer 7). But before we get into the CBAC architecture, let's review the operational characters of the three types of firewalls: choke, proxy, and stateful.

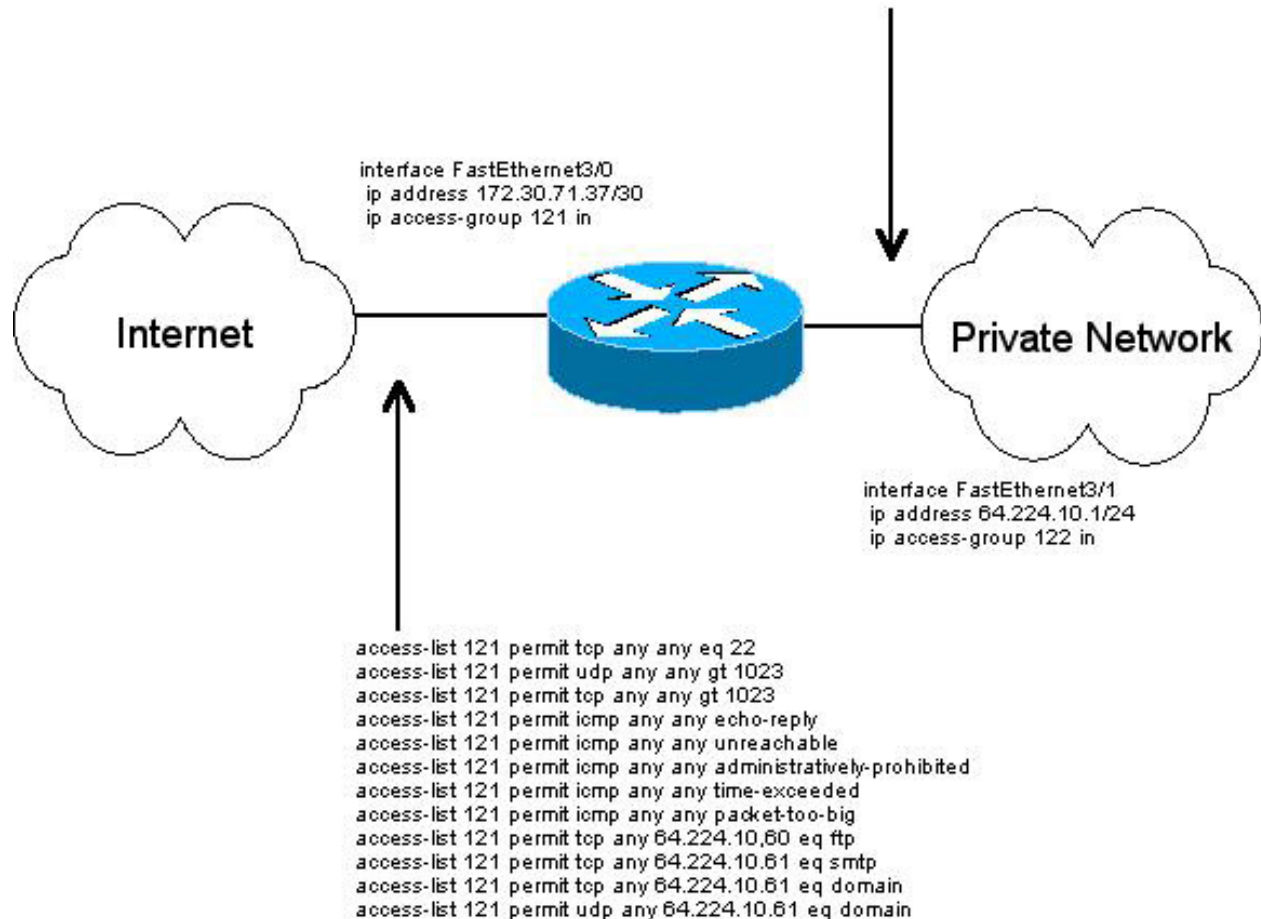
Choke firewalls operate by restricting the flow of data between networks using Layer 2, 3 or 4 static filtering. Chokes are typically implemented at network edge points. Internet gateway routers and LAN segment gateways with static access control lists are the most common choke firewall points. The value of chokes is that they can prevent access to specific devices and applications in a performance friendly way, allowing the administrator a great degree of control over inbound and outbound network access. Where they are ineffective is in providing any level of defense for hosts and services that are permitted, because they only look at Layer 3 and 4 packet headers when making forwarding decisions. Filtering beyond Layer 4 requires a level of intelligent filtering at layers 5 through 7, hence the reason for creating proxy and stateful firewalls. Here is an example of a choke firewall implementation:

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

```
access-list 122 permit tcp 64.224.10.0 0.0.0.255 any eq 22
access-list 122 permit udp 64.224.10.0 0.0.0.255 any eq domain
access-list 122 permit icmp 64.224.10.0 0.0.0.255 any echo
access-list 122 permit icmp 64.224.10.0 0.0.0.255 any echo-reply
access-list 122 permit tcp 64.224.10.0 0.0.0.255 any eq ftp
access-list 122 permit tcp 64.224.10.0 0.0.0.255 any eq http
access-list 122 permit tcp 64.224.10.0 0.0.0.255 any gt 1023 established
access-list 122 permit udp 64.224.10.0 0.0.0.255 any gt 1023
```



This example permits inbound mail delivery and FTP, DNS (zone transfers and lookup) to specific servers, and TCP and UDP traffic above port 1023 to allow outbound connections from the private network to function. Only local segment users are permitted to establish SSH, FTP, or HTTP. While this approach limits access (both in and out) for known services (below 1023), it leaves the network largely exposed. Since the majority of today's applications utilize ports above 1023 and not all IP stack and application implementations follow the 49152 through 65535 dynamic/private port guidelines, filtering above 1023 can affect the operation of applications that you want to function.

While chokes do not address permitted protocol and application security concerns, they are quite valuable for implementing broad network and service access policies. A successful network security implementation is based on understanding what is going on in the network. This baseline knowledge is the screen used to filter network activity so that inappropriate activity can be identified. Network activity should be restricted to permit acceptable service only. Chokes provide a great way to implement a coarse level of control and monitoring that can be fine-tuned using intelligent filters, such as proxy and stateful firewalls. One note of caution: Creating static access control lists require some

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

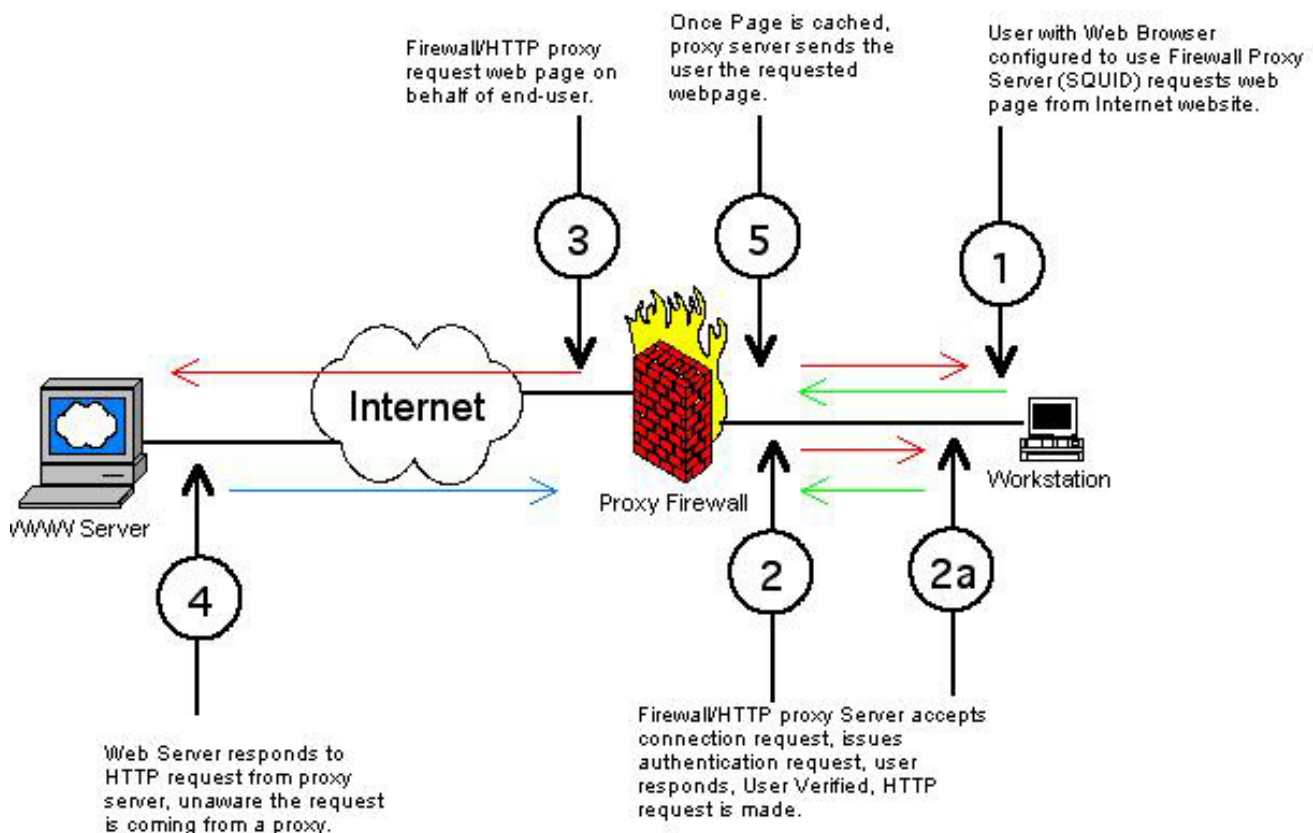
Michael J. Martin

thought and a lot of testing. A poorly written SACL can have adverse effects on the network in terms of performance and service availability.

Proxy firewalls provide intelligent filtering by operating essentially as application relays, using discrete application proxies to establish sessions between internal and external hosts (or vice versa). Proxy firewalls do qualify as "filters" since they make forwarding decisions based on an access rule base. Of the three types of firewalls, proxy firewalls are considered the most secure, because they utilize a two-connection model between the internal host and proxy, and proxy and external host. The firewall's use of discrete application-specific proxies to service network requests allows the firewall to inspect every layer of the packet exchange, providing the ability to analyze for known protocol security violations and attacks. Users connect to the application proxy (authenticated or unauthenticated), which then services the request (using the IP address of the proxy server as the source address of the request). The server then relays the session data back to the origin host.

However, proxy firewalls also have some disadvantages. First, proxy firewalls suffer from performance problems (since two sessions are needed for each transaction). This problem has largely been addressed with the major advances in computing power over the last few years. Second, since all communication is dependent on application-specific proxies, as new protocols and applications are developed new proxies must also be developed. Third, while the methodology is secure, the real risk with proxies is the operating system on which they run. Proxies run on servers, which for the most part are quite insecure out of the box. A great deal of effort must be expended to harden the server OS. What use is a secure proxy if the server has big security holes?

Here is a diagram showing the operational process between a user and a proxy firewall utilizing a web proxy/cache:



The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

Aside from the inherent security provided by the architecture (everything is blocked unless proxies are up; how much more secure could you get?), proxy firewalls also have the added ability to audit user access as they use the proxy services by implementing authentication as shown in the process diagram above. While this option enhances an administrator's ability to monitor and control user access to proxy services, it has two down sides. First, it can drive your users crazy; authentication can be in one of two ways, per session or per transaction (by transaction, we mean per Web site, based on destination IP). The per transaction method strictly controls access but can be quite burdensome. The per session (Web access for a fixed time window or until a inactivity timer has expired) method is more user friendly, but you can count on getting some user feedback.

Another, often unrealized downside, is that once you have all of the auditing data, it can be subpoenaed. Since all user requests are made using the same IP address, it's not too hard to find out where the requests are coming from. So if you log this transactional data and someone does something they should not be doing...well, you get the picture. In addition, logging and auditing this kind of data raises some issues of privacy. Whatever you decide to do with auditing, be sure you have it clearly defined in a policy that your users know about (putting the policy in the authentication window is always a good idea), and have your legal department's approval. If you are interested in looking at some application proxies, the following links offer details:

- SQUID HTTP proxy
- SOCKS generic application proxy
- DPROXY DNS proxy
- POUND SSL reverse HTTP proxy
- OBTUSE SMTP proxy

Stateful firewalls provide intelligent filtering by dynamically restricting the flow of data between networks using Layer 3, 4, or 7 filtering. Stateful filtering technology was first introduced by Check Point Software Technologies in the 1990s. Soon after its introduction, other firewall vendors developed their own interpretations of stateful filtering. The "state" in "stateful" refers to information pertaining to the sessions traversing the firewall. Choke-based firewalls are limited because they only inspect packet headers, so Layer 5 through 7 protocol-based attacks are undetected. Proxy-based firewalls inspect both packet headers and payload, but each packet is processed by an application-specific proxy that needs to be updated whenever a new exploit is discovered.

Stateful firewalls operate using a hybrid of the choke and proxy functional models. Similar to choke firewall filtering, stateful firewalls operate passively -- packets are forwarded either directly or via the router with the firewall serving the gateway of last resort. Once the packets reach the firewall, they are pre-processed prior to forwarding through a security rule base that functions essentially like a SACL on a router or Layer 3 switch to screen packets. Those that meet the requirements of the rule base are processed and forwarded; those that don't are dropped with or without end-user notification, depending on the handling rules for disqualified traffic. Most stateful firewall implementations follow two packet handling policy models:

- Explicit permit: By default all traffic is denied both inbound and outbound. For each application, the firewall uses a custom stateful filter, which examines the application specific commands, data structures and options. Any traffic that the firewall does not understand, it discards.
- Implicit permit: By default all traffic is permitted outbound and denied inbound. The firewall inspects the traffic flow only for protocol violations and known exploits.

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

While inspection is performed on packets, stateful firewalls, like proxy firewalls, operate by controlling and monitoring sessions between hosts. Proxy firewalls accomplish this by functioning as the "man in the middle," relaying packets between the local host to proxy and proxy to remote host connections, like water buckets on a fire line. Stateful firewalls achieve the same result without the second connection by performing inspection in the OS kernel or through application-specific integrated circuits (ASICs). Packets are examined at Layers 3 through 7 for policy and protocol violations, and then the firewall opens the needed ports (for only the source and destination hosts) on the external interface for the session to occur. The bi-directional packet flow between the internal and external hosts is then tracked for the lifetime of the session (determined by the user or enforced by the firewall based on inactivity timers). Once ended, the openings on the external interface are removed.

It's All In The Packet

The firewall tracks the state of sessions based on the control messages set within the transport layer header. The IP protocol, like most networking protocol suites, supports connection-oriented (TCP) and connectionless (UDP) transport protocols. In the "TCP Trinity" the foundational protocol of the stateful firewall architecture lends itself perfectly to passive state tracking. There are three operational states of a TCP session:

- ESTABLISHMENT -- during which the SYN, SYNACK, ACK handshake occurs between the client and the server.
- OPEN -- where the client and server exchange ACK for each data exchange, packets are tracked by unique sequence numbers.
- CLOSE -- during which the FIN, ACKFIN, ACK handshake occurs between the client and the server.

The defined session setup and tear-down (which is also based on a trinity) connection structure of TCP makes it easy to monitor the operational state of a session. UDP, however, along with the other stateless protocols of the IP suite (ICMP, OSPF, etc.), do not lend themselves nicely to session tracking by state. Connectionless protocols blindly transmit messages with the expectation that the destination hosts are active. Without a connection setup or tear-down, there is nothing to track that reveals the status of the data interchange. In the case of UDP, connections are tracked using pseudo-state connections. UDP connections are tracked using `source_addr:src_port -> destination_addr:destination_port` coupled with activity and inactivity timers to determine the session life. ICMP represents another set of challenges; because ICMP provides control messages in most cases there is no exchange, just a notification.

Each firewall vendor handles ICMP messages differently. Some monitor the type and frequency of messages (DDOD monitoring), and others block or pass the traffic without inspection. In fact, every firewall vendor handles the inspection of all protocols in its own unique way. The following illustration provides a generic view of UDP and TCP stateful filtering:

The Router Is The Firewall Part 1

An Overview Of The IOS Firewall Feature

Michael J. Martin

UDP Packet Handling

1a. Host makes UDP DNS request to external Server.

2a. Firewall receives packet, compares it to the security policy, Checks to see if it belongs to an existing connection in the firewall state table.

3a. Firewall creates UDP virtual connection entry in firewall state table:

CON-ID	SRC_IP:SRC_P	DST_IP:DST_P	IP_PRO	SES_TOUT	STATE
000001a	64.128.245.98:49003	68.44.36.254:53	17	60	SIS_OPEN

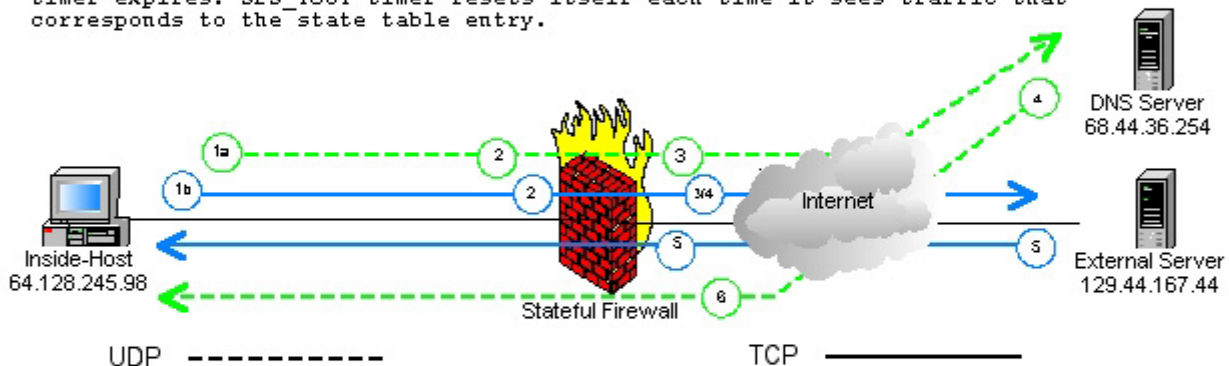
Firewall starts SES_TOUT timer, 60sec, and ads dynamic ACL entries to accept UDP traffic from remote_host.

4a. DNS server responds, within SES_TOUT timer.

5a. Firewall accepts DNS reply, Resets SES_TOUT timer to 120 sec:

CON-ID	SRC_IP:SRC_P	DST_IP:DST_P	IP_PRO	SES_TOUT	STATE
000001a	64.128.245.98:49003	68.44.36.254:53	17	120	SIS_OPEN

All future queries matching the original flow will be forwarded until SES_TOUT timer expires. SES_TOUT timer resets itself each time it sees traffic that corresponds to the state table entry.



TCP Packet Handling

1b. Host opens a TCP SSH connection to external server (SYN)

2b. Firewall receives packet, compares it to the security policy, Checks to see if it belongs to an existing connection in the firewall state table.

3b. Firewall creates a TCP-Established state table entry:

CON-ID	SRC_IP:SRC_P	DST_IP:DST_P	IP_PRO	SES_TOUT	STATE
000001b	64.128.245.98:50032	129.44.167.44:22	6	3000	SIS_EST

4b. Firewall starts SES_TOUT timer, 60sec, and ads dynamic ACL entries to accept TCP traffic from remote host.

5b. Remote SSH server responds with SYNACK, Firewall accepts SYNACK, Resets SES_TOUT timer to 300sec.

6b. Client completes the connection establishment, sending an ACK packet, Firewall updates state table from Established to OPEN:

CON-ID	SRC_IP:SRC_P	DST_IP:DST_P	IP_PRO	SES_TOUT	STATE
000001b	64.128.245.98:50032	129.44.167.44:22	6	3000	SIS_OPEN

Firewall resets SES_TOUT timer to 300sec, any future queries matching the original flow will be forwarded until SES_TOUT timer expires or the firewall detects FIN,FINACK,FIN TCP session close. SES_TOUT timer resets itself each time it sees traffic that corresponds to the state table entry.

That's the basics of firewall theory. Tune in next month for our next episode of the "The router is the firewall," when we will find out just exactly what kind of firewall the IOS firewall is and how to implement CBAC filtering.