

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

Last month, we kicked off the "Router is the firewall" series by touting the virtues of the Cisco IOS firewall feature set (FFS). We covered some of the current theory behind firewall implementation and discussed the architectures of the three types of firewalls: choke, proxy and stateful. This month we are going to delve into the foundational elements of the IOS FFS Context Based Access Control (CBAC). Most of the material on CBAC covers the basic implementation: Everything outbound, nothing inbound. In this scenario, users on the inside of the firewall can access remote (i.e. Internet) hosts. Remote hosts, however, are not permitted to originate connections to any of the inside hosts. While this example covers the basics, it is not really a practical application of CBAC, except for perhaps small or home office users. So to allow us to give CBAC its proper due, this month we will examine CBAC's operation and architecture.

Say What You Mean, Hear What I Say

The use of jargon seems to be a base requirement for any discussion about networking. (I think it's a law, actually.) When doing the research for this series I was driven crazy by the interchange and redefinition of terms. So to make sure we're all on the same page, I've listed some definitions of the key terms:

- A conversation is an application layer (L7) protocol interaction between two hosts, comprised of one or more sessions.
- A session is a unidirectional packet flow between two hosts. A control or data connection between two hosts is bidirectional. Actual data exchange between two hosts is handled by two unique session packet flows: local --> remote and remote --> local.
- A connection refers to the transport layer (L4) control channel session between two hosts. Some L7 protocols, such as telnet or HTTP, use only a single channel for both control and data exchange. Other protocols, such as FTP, use a control channel, which remains "up" for the duration of the conversation. Separate data channels are used for each file transfer.
- The term state is used to describe the operational status of a connection. The idea of "connection state" refers to TCP, the only connection-oriented protocol in the IP suite. TCP connections have three operational states: establishment, open and closed.
- A channel is a dynamic opening a firewall's rule base (or interface packet filter, to be specific). Stateful firewalls add and remove access channels as conversations between hosts are established and terminated. The primary channel is open when the protected host first establishes the connection; this is possible because the host originating the connection defines the service port to which the destination host will reply.
- A local, internal, private or protected host is a host located on a network being screened behind a firewall.
- A remote, external, public or unprotected host is a host located on a network in front of a firewall.

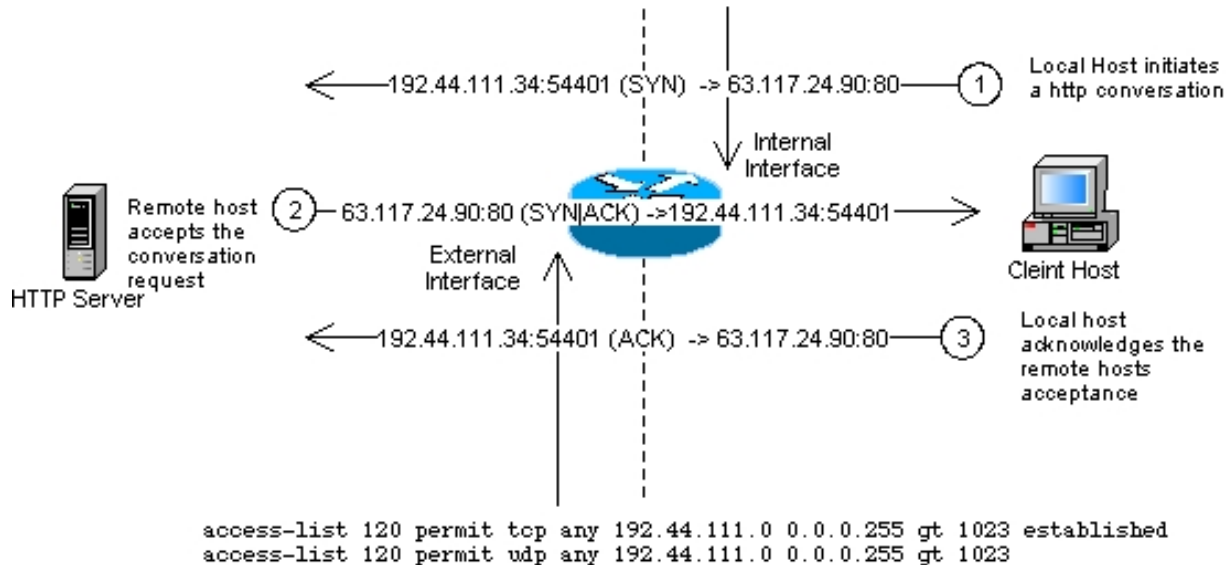
What is CBAC? At the heart of the FFS is Context Based Access Control. CBAC is a stateful packet inspection engine that tracks ICMP (as of 12.2.15T) TCP, and UDP-based application packet flows between hosts on either side of the firewall (router). While not as advanced as Check Point's Inspect packet-inspection system, CBAC qualifies the FFS IOS as a "stateful firewall" product. Functionally similar to CheckPoint and PIX, CBAC is hybrid of architectures rather than a single pure implementation. This hybrid architecture gives the IOS FFS the flexibility to function as both a router

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

```
access-list 121 permit tcp 192.44.111.0 0.0.0.255 any eq 80
access-list 121 permit tcp 192.44.111.0 0.0.0.255 any eq 443
access-list 121 permit udp 192.44.111.0 0.0.0.255 any eq 53
```



The example illustrates a basic explicit access filtering policy. The access groups installed on the router interfaces enforce the network access policy by explicitly permitting only site-acceptable network traffic. Extended access list 121 permits the outbound HTTP, SSL, and DNS lookups. Extended access lists 120 permits the return traffic from the external hosts to the local hosts that originate the application sessions. The idea behind this approach is to limit exposure by opening only explicit services from specific networks, which provides security against the following threats:

- The Outbound Use Of Unapproved TCP Network Services
- The inbound use of any TCP service through the use of limited TCP session tracking
- Inbound and outbound service connections utilizing IP spoofing or BOGON addresses.

While this approach provides protection against many inbound and outbound network level exploits, a number of security threats are left unchecked.

For starters, network access has only been limited. For network sessions to function, return traffic needs to be permitted into the protected network. Using SACLS alone, more traffic is allowed to pass than is actually needed. There are a number of UDP-based exploits and a number of common network services (SMB/NetBIOS, NFS) that are based on UDP. In order for DNS to function properly, UDP (which you should recall is connectionless) needs to be open above port 1023. While Unix systems originate DNS requests using service ports above 49152, Windows-based systems do not. So traffic above 1023 needs to be permitted inbound from untrusted hosts.

Because TCP is connection-oriented, single session TCP services can be protected using approximated session filtering (ASF). ASF is an enhanced filtering option available only to ESACL TCP match rules. ASF is applied by appending the established keyword to the end of the match rule. Once added to a TCP filter rule, the router inspects the TCP header information to see if the ACK or

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

RST flag is set. Packets with the SYN (by itself) flag indicate the start of a session and are dropped, packets with the ACK or RST are indicative of traffic belonging to already established TCP sessions.

However, application protocols that multiplex sessions, such as NetMeeting and FTP, are unable to function if the established option is used, because these applications need to open multiple TCP sessions using random source ports. So if your environment utilizes protocols that multiplex, then all TCP ports above 1023 must also be left open to allow for connection multiplexing.

In addition to "registered services" exposure, systems are also unprotected against TCP fragmentation and TCP "flag" attacks. While not exclusive to Linux, certain versions (prior to v2.2.9) of the Linux kernel are known to be susceptible to TCP fragmentation attacks that can be used to access the system. A number of other attacks, such as the LAND and Xmas tree attacks utilize TCP flag combinations to disable or gain access to systems. For example, even with the "established" filtering option in place, it is still possible to bypass the established flag by crafting spoofed packets with the correct flags set. Many exploits only need one packet to work; with good recon any system can be exploited.

That's why application layer inspection is so important. Unfortunately, SACL filters are not capable of such logic. In both "access closed" environments (like our example), and "open access" environments where inbound and outbound traffic exchanges are permitted, attackers exploit known system and application vulnerabilities to gain access or disable systems. SACLs can only perform inspection on L3 and L4 header information. Additionally, this information is also not examined within the context of a complete conversation, so attacks within the data portion of a packet are undetected.

Of course, a vulnerability list is not complete without mentioning the IP protocol suite's ultimate bugaboo: ICMP. ICMP is a problem for all firewall implementations. While a valuable tool, it can also be exploited as both a reconnaissance and attack tool. ICMP, like UDP, is connectionless. The problem, however, is not about tracking state. UDP sessions can be tracked statefully using approximation timers to determine how long a firewall should wait for a response from a UDP transaction originated by an inside host. In cases where ICMP traffic is originated from the private network, this approach can be applied. However, inbound ICMP traffic from external hosts must either be trusted or dropped because there is really no way of knowing what the true nature of the message is. So now that we're all upset about how ineffective our router filters have become, let's get started with CBAC.

Life with CBAC: You're still using access lists, baby!

CBAC is not a type of access list, but rather it a stateful inspect engine that works in conjunction with ESACLs. ESACLs enforce the global access policy, and CBAC inspects the traffic and provides the following five key security enhancements to the IOS:

Dynamic filtering exceptions -- CBAC provides dynamic return-path filtering for ICMP, UDP, and TCP for single-session and multi-session conversations. This function is very similar in operation to reflexive access lists. CBAC creates dynamic entries based on the bi-directional session flows in the filtering of access lists when a conversation is first established. This allows session traffic to pass between only hosts involved with conversations. Since permit access entries are unique, this eliminates the need (in theory) to leave any statically open ports. The service ports that are opened dynamically are limited in lifespan (the duration of the conversation) and only to specific to hosts, thus limiting the opportunity for external attacks.

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

Tracking of session states -- CBAC tracks half-open, open and closed TCP sessions to provide defense against TCP-SYN DOS attacks. Session number and rate per minute are tracked using high and low, administrator-defined thresholds.

TCP sequence numbers tracking -- CBAC monitors in --> out and out --> in sequence numbers. Packets with out-of-range numbers are discarded. This provides protection against man-in-the middle and other session hijacking attacks.

UDP and ICMP virtual connections -- UDP/ICMP sessions are regulated as stateful sessions, the same way as TCP based sessions. However, since these two protocols are connectionless, the tracking (and dynamic rule management) is achieved through the use of timer-based virtual connections to approximate conversation status.

Application-specific monitoring -- CBAC examines packet flows for a number of application-specific protocol violations along with single-path TCP, UDP and ICMP flows. If CBAC detects a violation, the conversation is blocked and the relevant channels are closed. Application packet inspection for multiplex is available for the following protocols:

- CU-SeeMe (7648) Peer-to-peer videoconferencing
- 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)
- 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)

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

- MSRPC (135) Microsoft Remote Procedure Call Protocol (provides system-to-system process communication)
- MGCP (2427) Multimedia Control Gateway Protocol (VoIP)

Session-level logging -- As a security device, a firewall needs to protect first and foremost. However, trailing ever so slightly behind protection is the ability to log and provide an audit trail. CBAC provides support for per-application auditing. Reporting is provided through the standard IOS logging mechanism with support for local as well as remote reporting to a syslog server.

CBAC Operation

The hybrid of stateful inspection (CBAC) and traditional SACLs results in two unique operational traits that need to be considered when implementing the IOS firewall:

- Inspection operates based on the packet flow of the conversation initiation.
- SACL rules have precedence over dynamic rules.

Before we get into how this impacts CBAC and implementing the FFS, we need to understand the perspective from which traditional firewalls operate. PIX and Check Point firewall implementations operate from the perspective of interface type. There are three interface designations, each having their own "default" security posture.

Outside -- The interface on the "insecure" network, such as the Internet. Most implementations do not permit inbound connections by default. With this model, the assumption made is that no hosts exist alongside or upstream to the firewall, except for a gateway device (i.e., router).

Inside -- The interface is on the "secure" network. Depending on the implementation, by default hosts on the inside network are allowed to originate conversations either explicitly or implicitly. Under the explicit (more secure) model, no sessions are permitted by default. The network administrator defines the types of conversations permitted to outbound hosts. The implicit model allows any conversation to be originated by an internal host.

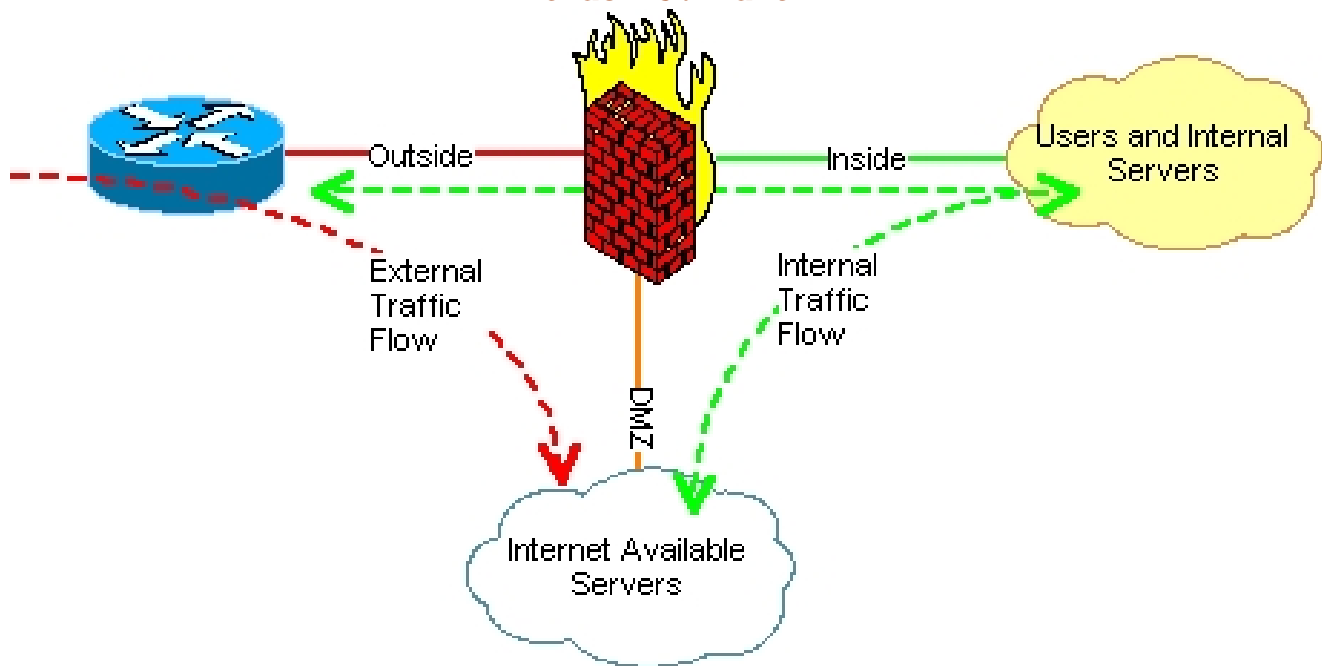
Demilitarized zone (DMZ) -- The interface is on a "partially-secure" network. Explicitly defined inbound connections are permitted from hosts on both secure and unsecured networks. The idea behind the DMZ network is to have a network with only hardened hosts that support services that require access from untrusted hosts. This reduces the both the risk and complexity of the public/private network security posture. Hardened hosts that require publicly addressed interfaces can be accessed using explicit inbound and outbound filtering rules.

"Private hosts" with questionable security can be managed by a completely different rule set. This minimizes the potential impact from rule base mistakes (hosts that are accessible from the Internet tend to require more complex rule sets) and malicious actions from private users. DMZ interfaces also often utilized by networks that use RFC 1918 addressing on their private networks. In these cases, key service hosts are "Internet available," with users behind a Port Address Translation (PAT) firewall interface. Ironically, PAT does provide some additional protection by obscuring private hosts from being identified with unique public addresses. Networks that utilize this model are allocated the address space from their network provider. The allocations are often /29 or /28 segments, making the networks easy to map since there are so few hosts. A quick lookup in DNS will tell most attackers running a simple surveillance which hosts provide what services.

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin



The figure above describes the packet flow between the different firewall interface types. Now that we have gone over firewall interfaces designations, keep the network designations in mind but forget the part about the interface designations. Let me tell you why.

The IOS FFS is built upon a router, which above everything else is designed to forward packets. It is the router's reason for being. Actually, when you think about it, filtering anything on a router is contrary to this nature. But, then again, can you think of a better place to filter packets than the device that was built specifically to forward packets in the first place? Anyway, because the FFS is based on the router, the concept of designated interfaces does not easily apply because the router's job is to forward packets.

In a divinely inspired effort to maintain the proper chi of the router's inherent purpose, or perhaps just a genius moment within the development team, the IOS FFS was developed with no preconceived notions about what interfaces should be defined to do. It has no concept of and no dependency on interface type. This idea is why the FFS functions more like a hybrid than a stateful firewall. The problem with SACLs is that they require large holes in order to permit return traffic. The problem with stateful firewalls is that that are PCs with NICs and fancy filtering software that are geared more toward processing packets than forwarding them.

The hybrid balance between CBAC and IP access groups (and by dependency ESACLs) is based on a simple concept: inspection precedes explicit access. Operationally, CBAC works using inspection points and filtering points. Here is an illustration of the basic CBAC inspection operation:

Beginning at the point where the conversation to be inspected is initiated, there can be one or more inspection points configured on the router. Corresponding to these inspection points are IP access groups that function as access-filtering points. The only requirement for CBAC to function properly is that an access-filtering point be upstream to the inspection point, in context to the conversation origin. Having the FFS determine access based on conversation direction maintains the ability for the router to still function primarily as a router. ESACLs provide the ability to explicitly control the flow of traffic into the inspection interface. Inbound and outbound access can be explicitly defined using SACLs,

The Router Is The Firewall Part 2

Implementing CBAC

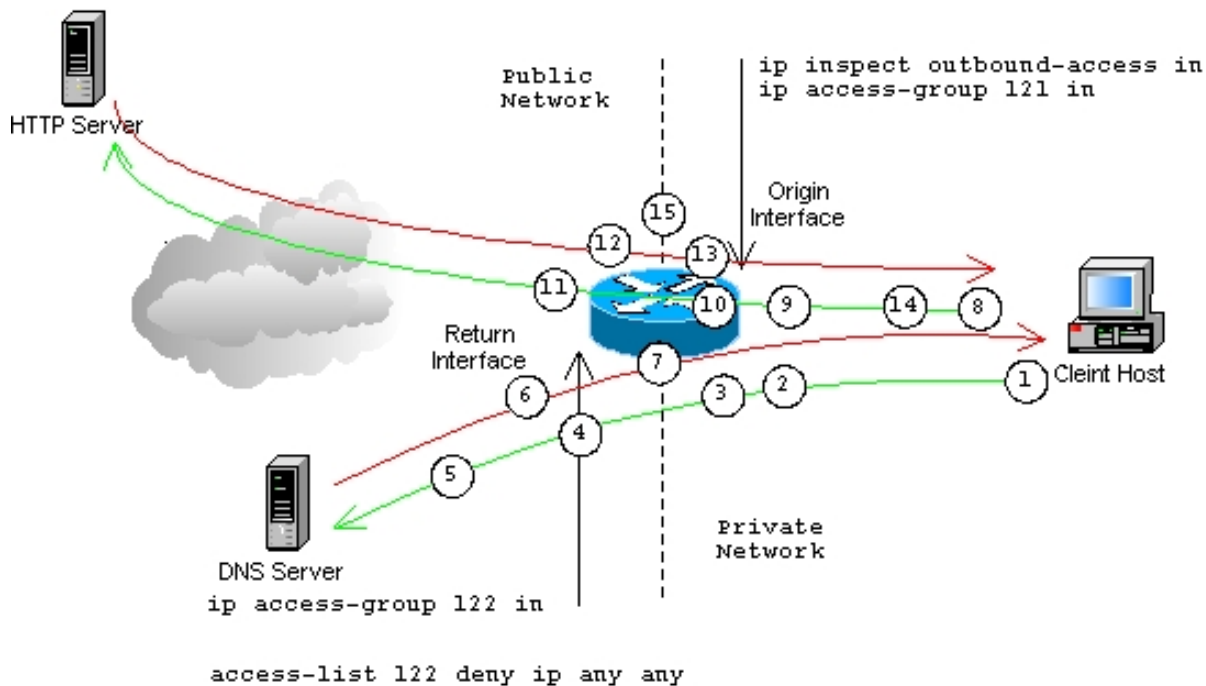
Michael J. Martin

with traffic permitted to only the known service ports. CBAC will create the return traffic "permit" statements and prepend them to the SACL, provided that the conversation establishment paths inbound and outbound are permitted by the SACLs used by the IP access groups. If a conversation path is explicitly denied before the session can be established, CBAC cannot manage the connection. From a design standpoint, careful thought and some traffic analysis needs to be done before implementing CBAC, particularly in environments requiring the use of an "explicit permit" access model. Now let's take a look at the process followed by CBAC inspection.

CBAC Inspection

In last month's article we reviewed the generic operation of a stateful firewall. Here we will take a look at the specific process used by CBAC when inspecting conversations. In this example, a host connected to the protected network will connect to an Internet Web site. The operation involves two distinct conversations. The first conversation is between the host and a remote DNS server. The second, once the Web site name has been resolved, is to the Web site. The example uses an explicit permit access model; only DNS, HTTP, and HTTPS are permitted outbound. Management access to the router via SSH and telnet is also permitted from hosts on the protected segment. The CBAC inspection model followed in this example performs inspection as traffic enters the private side router interface. An inbound IP access group is also installed to qualify the types of traffic permitted out of the private network. An inbound IP access group is also installed on the public interface. The public interface access group ACL is modified by CBAC to permit the return traffic.

```
access-list 121 permit tcp 88.67.191.0 0.0.0.255 any eq 80
access-list 121 permit tcp 88.67.191.0 0.0.0.255 any eq 443
access-list 121 permit udp 88.67.191.0 0.0.0.255 any eq domain
access-list 121 permit tcp 88.67.191.0 0.0.0.255 host 88.67.191.1 eq telnet
access-list 121 permit tcp 88.67.191.0 0.0.0.255 host 88.67.191.1 eq 22
access-list 121 permit icmp 88.67.191.0 0.0.0.255 any
```



- Here is the detail on the inspection processes:

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

- Client connects to www.yahoo.com. Before the conversation with the Web site can be established, the host issues a DNS domain name query.
- The DNS query establishment packet arrives at the router's inside interface. The UDP datagram is evaluated against the ACL associated with IP access-group 121. If the packet meets the acceptable use criteria, it is forwarded. (If the packet failed the criteria, it would be discarded and no notice would be issued to the user.
- Once qualified, the packet is inspected by CBAC and the state information about the connection (SRC_ADDR:SRC_PORT->DST_ADDR:DST_PORT:PROTOCOL) is stored in the router's state table. Each entry consumes about 6 Kb of DRAM.
- Based on the state table data, a temporary entry is made to access list 122. The new rule permits only the return traffic from the destination host to the origin host utilizing the same protocol and port numbers. The rule is prepended to the static rules already in the access list.
- The packet is forwarded out of the router interface. The dynamic UDP DNS entry will remain in the 122 access list for 5 seconds. If no reply is received within that period, the rule will be retracted.
- When the DNS reply packet arrives at the router's public interface, it is evaluated against the inbound IP access group and permitted because it belongs to an established session.
- Once accepted, the DNS reply is inspected by CBAC and then forwarded to the origin host. Because the session is UDP-based, the state entry will remain for the default timer and then be removed, along with the dynamic ACL entry.
- Now that the client host knows the IP address of the Web site, the initial HTTP session is initiated.
- The packet arrives at the router's inside interface. It is evaluated against the outbound network access policy, and the packet is accepted and forwarded.
- The packet is now evaluated by CBAC, and the session state information is added to the router's state table.
- The packet is forwarded on to the destination host. Based on the state table information, a temporary return-traffic permit ACL entry is prepended to static access list 122. The entry will remain for up to 30 seconds for a SYN-ACK response from the destination host. If no reply is received within the time limit, the dynamic entry will be retracted and the state table entry purged.
- When the reply packet arrives at the router's public interface, it is evaluated against the inbound access group's access list (122) and accepted
- CBAC inspects the return inbound payload for protocol-specific violations. If the header or data field information contain a known violation, CBAC discards the packet and closes the session. If the packet is error-free, it is forwarded onto the origin host. (This is true of all ICMP/UDP and TCP single session and multi-session application specific CBAC filters.)

In the case of HTTP and multi-session protocols, such as FTP and H.323, additional sessions will be established between the origin and destination hosts. CBAC will update the state table and inbound access group's access list accordingly. In terms of packet processing, the initial session packet will be process-switched. Additional packets belonging to the session flow will be fast-switched.

The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

Temporary access-list entries will be removed at the end of each session. ICMP and UDP sessions will be removed based on inactivity timers. TCP sessions will be removed 5 seconds after the exchange of FIN packets. In the event of an RST packet, the session will be terminated and corresponding ACL entries will be removed immediately.

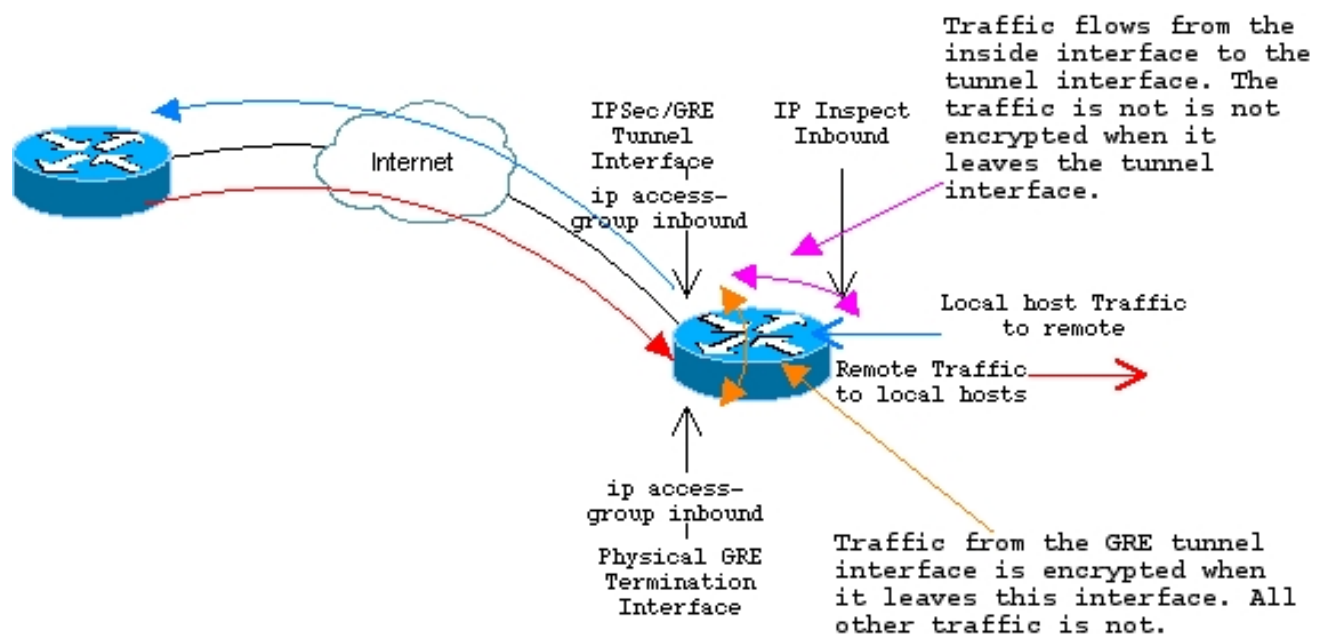
CBAC Limitations And Known Issues

CBAC enhances the effectiveness of IOS routers as security devices. Used in conjunction with other available security enhancements, IOS routers can be utilized for more than packet forwarding, increasing their ROI and allowing administrators to cost-effectively implement more secure networks. Of course, there is no perfect security device. That said, there are some operational issues and limitations to CBAC that administrators need to be aware of.

Limited protocol support: CBAC has evolved since its introduction. Most versions only support TCP and UDP single-session inspection and a fixed list of applications-inspection filters (which varies between versions). ICMP has not been supported historically, but was recently added in 12.2.(15)T. OSPF, BGP (uses TCP), RIP (uses UDP/Multicast) and IGRP and EGRP are also not supported. For these protocols to operate properly, SRC/DST and PORT permit entries must be made in the SACLs used by IP access groups installed on the appropriate router interfaces.

CBAC inspection is not performed on packets with the source or destination address of the firewall interfaces. This impacts the router's operation two different ways. First, VTY sessions between administrators and the firewall are not inspected. Second, management, authorization and accounting (TACACS/RADIUS) traffic is not inspected. Since traffic to and from the router is not inspected, dynamic permit rules are not created when session are opened (VTY) or originated (RADIUS/TACACS). For these functions to operate properly, SRC/DST and PORT permit entries must be made in the router's management interface "imp access-group" ACL.

Encrypted packet payloads are not inspected unless the router is the encrypted link endpoint. Under these conditions, CBAC inspection must be performed on an upstream interface (on the router) from the encrypted link termination endpoint. The diagram below provides an example.



The Router Is The Firewall Part 2

Implementing CBAC

Michael J. Martin

Asymmetric routing and forwarding is not supported. CBAC inspection and the resulting DACL entries are based on the direction of the conversation establishment (inspection is performed on the ingress interface). Consequently, the resulting dynamic SACL entries are created on the conversation origin's egress interface, with the expectation that all future session exchanges will utilize the same symmetric path. CBAC will drop session traffic that does not utilize the origin path. This can cause major problems in networks that utilize equal-cost load balancing, redundant or multiple-redundant paths.

The SMTP command EXPN and VRFY are permitted as part of the SMTP inspection rules. The VRFY command allows a user to telnet (over the listener port: telnet mailhost.someware.com 25) to the Sendmail server and verifies that a Sendmail server IP address is valid before sending mail to the server. The VRFY command is utilized by mail spammers as a way of finding open mail relays and spam targets. The EXPN command allows a user to telnet to the Sendmail server and provides a mail alias to the server for verification. The server then expands the mail alias into a list of all of the users belonging to the alias. This allows hackers to recon your mail server for usernames. These commands are considered by many mail administrators to be very dangerous and access to them should be disabled. Alternatively, they are also useful commands, and part of the Sendmail specification (which is why they are permitted).

CBAC does not permit third-party FTP connections and requires successful client-server authentication before opening a FTP data channel. Additionally, CBAC will only permit FTP data channels with destination ports between the range of 1024 and 66536.

In addition to CBAC's architectural limitations, there are also two documented bugs. The first is with the CBAC implementation on IOS versions 11.2 through 12.0(2)T. CBAC does not handle fragmented IP packets properly, leaving the router and other hosts susceptible to IP fragmentation attacks. The second bug affects IOS versions 11.2P to 12.1. In the affected IOS versions, CBAC fails to inspect the protocol type on return traffic. This is a major bug, leaving router and network open to spoofed IP and port attacks that could result in compromising the router and hosts with open sessions. Cisco's recommended workaround for both of these issues is an upgrade to IOS FFS 12.2 or higher.

Whew! We have crawled into the belly of CBAC and come out its blowhole. All that is left to do now is implement the CBAC beast, which we will do next month. Until then, cheers.