Content-Based Access Control

Content-based access control (CBAC) was originally fielded in Cisco IOS version 11.2P and designated a special firewall feature set. CBAC is capable of managing multichannel applications like FTP and H.323 sessions. Additionally, CBAC is capable of inspecting content and application layer commands, such as SMTP and several other popular application layer protocols. Additionally, CBAC can block Java applets embedded in HTTP traffic; provide Denial of Service (DoS) prevention and detection. This new feature was a remarkable enhancement to the reflexive access list.

How it works

The first step is to decide which router interface CBAC will be applied. CBAC should be configured on the interface closest to the originating host or closest to the destination host. The most common method is to use CBAC on the external interface to inspect inbound traffic and allow it to dynamically open temporary access lists on other interfaces to manage, inspect and direct that traffic.

The second step is to create your access control list. Create one list that allows the external network to connect to your web, FTP, mail, DNS servers and any other public application servers you might have on your network. Then create an internal access control list that allows your internal network to connect to external servers.

example:

```
ip access-list 101 permit any host myweb.server.ip.address eq http
ip access-list 101 permit any host mymail.server.ip.address eq smtp
```

Once applied inbound to your external interface, this allows the world to connect to your web server and send you emails.

```
ip access-list 102 permit your.network.ip.address any eq http
ip access-list 102 permit your.network.ip.address any eq https
ip access-list 102 permit host your.mail.server.ip.address any eq smtp
ip access-list 102 permit your.network.ip.address any eq ftp
```

Once applied inbound to your internal interface, this allows people on your network to browse the web, use FTP to retrieve documents and your mail server to send mail to any mail server on the Internet.

Since CBAC helps to prevent DoS, our third step is to configure global timeouts and threshold values so the CBAC can determine how long to manage session state and when to drop half open connections.

example:

```
ip inspect udp idle-time 30
```
ip inspect tcp idle-time 30

This configuration tells the CBAC to maintain session state information on an idle connection for 30 seconds.

The fourth step is to define an inspection rule. The rule defines which application layer protocol will be inspected. The following protocols are supported:

CU-SeeMe (only the White Pine version)
FTP
H.323 (such as NetMeeting, ProShare)
HTTP (Java blocking)
Microsoft NetShow
UNIX R-commands (such as rlogin, rexec, and rsh)
RealAudio
RTSP (Real Time Streaming Protocol)
RPC (Sun RPC, not DCE RPC)
SMTP (Simple Mail Transport Protocol)

example:
ip inspect name firewall ftp
Forces CBAC to manage FTP traffic for opening random inbound ports on the interface closest to the FTP server.

If you want to audit and log any of your CBAC traffic, then append the commands alert on and/or audit-trail on to the end of your inspection rule.

Final thoughts

CBAC is not a replacement for your firewall. Its content filtering is limited to the application layer protocols mentioned above. It will not protect your web server from vulnerability exploits.

CBAC is an enhancement to your network security and another member of your Defense in Depth tool set.