- CBAC-

Context Based Access Control (CBAC)

CBAC allows for the stateful inspection and filtering of TCP or UDP packets, based on where those connections are *sourced*. **Stateful inspection** does not just scrutinize the header of a packet to *send* to a destination, but additionally *tracks* TCP or UDP sessions between devices.

For example, CBAC can be configured to track connections that originate *within* the local network. This session information is kept in a **state table** by CBAC. CBAC will open temporary holes in the firewall to allow those connections to come back in to the local network.

This ability allows CBAC to both monitor and prevent DoS and other network attacks. If CBAC detects an attack, it can be configured to either drop the session (plus block the source), or sent an alert message indicating an attack is occurring.

To configure CBAC, we must accomplish the following:

- Configuring Auditing
- Set timeouts and thresholds
- Identify the type of traffic we want to inspect, such as HTTP, FTP, SMTPetc.
- Apply CBAC to an interface

Timeouts and thresholds help CBAC determine when a DoS or network attack is occurring. These thresholds include:

- Total number of half-opened TCP/UDP sessions
- Number of half-opened sessions over a given time period
- Number of half-opened session from a specific host

A **half-opened TCP** session indicates that the three-way handshake has not yet completed. A **half-opened UDP** session indicates that no return UDP traffic has been sent. A large number of half-opened sessions on a router will chew up resources, while preventing legitimate connections from being established.

Configuring CBAC Auditing

The first step to configuring CBAC is to enable auditing of traffic permitted or denied through the router:

```
Router(config)#  logging on
Router(config)#  logging 10.1.1.1
Router(config)#  ip inspect audit-trail
```

The first command enables *logging*, while the second points to a syslog server. The third command will send CBAC specific auditing information to the syslog server.

Configuring CBAC Thresholds

Next, the CBAC **thresholds** must be defined.

To adjust how long CBAC will wait before dropping an unestablished connection (default is **30 seconds**):

```
Router(config)#  ip inspect tcp synwait-time 15
```

To adjust how long CBAC will wait to disconnect a TCP (default is **3600 seconds)** or UDP (default is **30 seconds)** session after no activity:

```
Router(config)#  ip inspect tcp idle-time 300
Router(config)#  ip inspect udp idle-time 15
```

To configure *high* (default is **500 sessions)** and *low* (default is **400 sessions)** thresholds for **total** "half-open" sessions:

```
Router(config)#  ip inspect max-incomplete low 150
Router(config)#  ip inspect max-incomplete high 250
```

Once the *max-incomplete high* threshold has breached, CBAC will begin actively deleting half-open sessions. Once the threshold reaches the *max-incomplete low*, CBAC will stop deleting half-open sessions.

Configuring CBAC Thresholds (continued)

To configure *high* (default is **500 sessions)** and *low* (default is **400 sessions)** thresholds for "half-open" sessions **over the course of one minute**:

```
Router(config)#  ip inspect one-minute low 200
Router(config)#  ip inspect one-minute high 300
```

Again, once the *one-minute high* threshold has breached, CBAC will begin actively deleting half-open sessions. Once the threshold reaches the *one-minute low*, CBAC will stop deleting half-open sessions.

To configure the threshold of TCP half-open sessions **from a single host** (default is **50 sessions**):

```
Router(config)#  ip inspect tcp max-incomplete host 25 block-time 2
```

The *block-time* argument indicates how long CBAC will continue deleting any new connections from that host (default is **0 minutes**).

Identifying Traffic for CBAC to Monitor

To configure what traffic CBAC should monitor:

```
Router(config)#  ip inspect name MYINSPECTION http
Router(config)#  ip inspect name MYINSPECTION smtp
Router(config)#  ip inspect name MYINSPECTION ftp
```

A CBAC profile called *MYINSPECTION* was created, that is monitoring *http, smtp,* and *ftp* traffic. To force CBAC to monitor all TCP and UDP traffic:

```
Router(config)#  ip inspect name MYINSPECTION tcp
Router(config)#  ip inspect name MYINSPECTION udp
```

To have CBAC auditing information sent to a SYSLOG server:

```
Router(config)#  ip inspect name MYINSPECTION http audit-trail on
```

Defining a Custom Application for CBAC

CBAC supports several standard protocols and applications by default, including FTP, HTTP, SIP, etc. There are circumstances when organizations run these standard protocols over non-standard port numbers.

Using **Port-to-Application Mapping (PAM)**, standard protocols (such as HTTP) can be mapped to non-standard port numbers**:**

```
Router(config)#  ip port-map http port 3000 list 10
Router(config)#  access-list 10 permit 10.5.0.0 0.0.255.255
Router(config)#  ip inspect name MYINSPECTION http
```

CBAC will now monitor *http* traffic across port *3000* for all hosts on the 10.5.x.x/16 network.

Applying CBAC to an Interface

Finally, CBAC must be applied to an interface. On an internal interface, it should be applied inbound. On an external interface, it should be applied outbound:

```
Router(config)#  interface e0
Router(config-if)#  ip inspect MYINSPECTION in
```