established

To permit return connections on ports that are based on an established connection, use the **established** command in global configuration mode. To disable the **established** feature, use the **no** form of this command.

established *est_protocol dest_port* [*source_port*] [**permitto** *protocol port* [*-port*]] [**permitfrom** *protocol port*[*-port*]]

no established *est_protocol dest_port* [*source_port*] [**permitto** *protocol port* [*-port*]] [**permitfrom** *protocol port*[*-port*]]

Syntax Description

est_protocol	Specifies the IP protocol (UDP or TCP) to use for the established connection lookup.
dest_port	Specifies the destination port to use for the established connection lookup.
permitfrom	(Optional) Allows the return protocol connection(s) originating from the specified port.
permitto	(Optional) Allows the return protocol connections destined to the specified port.
port [*-port*]	(Optional) Specifies the (UDP or TCP) destination port(s) of the return connection.
protocol	(Optional) IP protocol (UDP or TCP) used by the return connection.
source_port	(Optional) Specifies the source port to use for the established connection lookup.

Defaults

The defaults are as follows:

- *dest_port*—0 (wildcard)
- *source_port*—0 (wildcard)

Command Modes

The following table shows the modes in which you can enter the command:

	Firewall Mode		Security Context		
				Multiple	
Command Mode	Routed	Transparent	Single	Context	System
Global configuration	•	•	•	•	—

Command History

Release	Modification
7.0(1)	The keywords **to** and **from** were removed from the CLI. Use the keywords **permitto** and **permitfrom** instead.

Usage Guidelines

The **established** command lets you permit return access for outbound connections through the adaptive security appliance. This command works with an original connection that is outbound from a network and protected by the adaptive security appliance and a return connection that is inbound between the same two devices on an external host. The **established** command lets you specify the destination port

that is used for connection lookups. This addition allows more control over the command and provides support for protocols where the destination port is known, but the source port is unknown. The **permitto** and **permitfrom** keywords define the return inbound connection.

Caution We recommend that you always specify the **established** command with the **permitto** and **permitfrom** keywords. Using the **established** command without these keywords is a security risk because when connections are made to external systems, those system can make unrestricted connections to the internal host involved in the connection. This situation can be exploited for an attack of your internal systems.

Examples

The following set of examples shows potential security violations could occur if you do not use the **established** command correctly.

This example shows that if an internal system makes a TCP connection to an external host on port 4000, then the external host could come back in on any port using any protocol:

```
hostname(config)# established tcp 4000 0
```

You can specify the source and destination ports as **0** if the protocol does not specify which ports are used. Use wildcard ports (0) only when necessary.

```
hostname(config)# established tcp 0 0
```

Note To allow the **established** command to work properly, the client must listen on the port that is specified with the **permitto** keyword.

You can use the **established** command with the **nat 0** command (where there are no **global** commands).

Note You cannot use the **established** command with PAT.

The adaptive security appliance supports XDMCP with assistance from the **established** command.

Caution Using XWindows system applications through the adaptive security appliance may cause security risks.

XDMCP is on by default, but it does not complete the session unless you enter the **established** command as follows:

```
hostname(config)# established tcp 6000 0 permitto tcp 6000 permitfrom tcp 1024-65535
```

Entering the **established** command enables the internal XDMCP-equipped (UNIX or ReflectionX) hosts to access external XDMCP-equipped XWindows servers. UDP/177-based XDMCP negotiates a TCP-based XWindows session, and subsequent TCP back connections are permitted. Because the source port(s) of the return traffic is unknown, specify the *source_port* field as 0 (wildcard). The *dest_port* should be 6000 + *n*, where *n* represents the local display number. Use this UNIX command to change this value:

```
hostname(config)# setenv DISPLAY hostname:displaynumber.screennumber
```

The **established** command is needed because many TCP connections are generated (based on user interaction) and the source port for these connections is unknown. Only the destination port is static. The adaptive security appliance performs XDMCP fixups transparently. No configuration is required, but you must enter the **established** command to accommodate the TCP session.

The following example shows a connection between two hosts using protocol A destined for port B from source port C. To permit return connections through the adaptive security appliance and protocol D (protocol D can be different from protocol A), the source port(s) must correspond to port F and the destination port(s) must correspond to port E.

```
hostname(config)# established A B C permitto D E permitfrom D F
```

The following example shows how a connection is started by an internal host to an external host using TCP destination port 6060 and any source port. The adaptive security appliance permits return traffic between the hosts through TCP destination port 6061 and any TCP source port.

```
hostname(config)# established tcp 6060 0 permitto tcp 6061 permitfrom tcp 0
```

The following example shows how a connection is started by an internal host to an external host using UDP destination port 6060 and any source port. The adaptive security appliance permits return traffic between the hosts through TCP destination port 6061 and TCP source port 1024-65535.

```
hostname(config)# established udp 6060 0 permitto tcp 6061 permitfrom tcp 1024-65535
```

The following example shows how a local host starts a TCP connection on port 9999 to a foreign host. The example allows packets from the foreign host on port 4242 back to local host on port 5454.

```
hostname(config)# established tcp 9999 permitto tcp 5454 permitfrom tcp 4242
```

Related Commands

Command	Description
clear configure established	Removes all established commands.
show running-config established	Displays the allowed inbound connections that are based on established connections.