



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"  
at <http://www.giac.org/registration/gsec>

## Cisco Reflexive Access Lists

Paul Lindsay  
May 10, 2001

### Description

Cisco Reflexive Access Lists have been an IOS feature since 11.3 and provide a finer degree of session filtering when compared to an extended access list relying on the **established** keyword to control traffic flow.

Reflexive access lists build a dynamic temporary access list entry that permits traffic based on the IP protocols, source & destination ports and addresses. This provides better protection from spoofing type attacks.

Reflexive access lists do not replace extended access lists but increase their functionality and effectiveness. They are relatively simple to implement and integrate into existing configurations.

The **established** keyword is only applicable to TCP traffic, other upper layer protocols such as UDP require separate access list statements to permit or deny access. Reflexive access lists are not limited to TCP and can also be configured for UDP & ICMP amongst others.

### Potential security problems with the established keyword

Cisco-ack-proof-of-concept is a small piece of code written by Codex.

The purpose of the code is to show that it is possible to communicate with a machine behind an access list only permitting established sessions. (Note: Cisco is not the only vendor that uses the established session concept).

Two small programs need to be compiled; one of which needs to be running on a machine behind an access list containing an **established** statement.

Running the program demonstrates that a text-based message can be sent from the external to the internal host passing through the access list.

This is possible due to the fact that any packet containing an ACK or any combination of ACK, PSH, URG, RST and FIN bit will be permitted access by the **established** statement.

No other functionality comes with the tool other than to show that it is possible, however the author has a number of suggestions on what it could be used for;

- Password sniffing
- File grabbing
- DoS attack amplifier
- Remote administration tool

It may currently be proof of concept but it has potential if somebody takes time and effort to develop it further.

### Reflexive access list operation

Reflexive access lists operate by using the **reflect** and **evaluate** keywords in two separate access lists.

The **reflect** keyword inspects traffic leaving the trusted network and the **evaluate** keyword traffic attempting to enter the trusted network.

Traffic matching a **reflect** statement triggers the generation of a temporary access list with an entry to permit traffic specific to that session back into the trusted network. This temporary entry contains the protocol, source & destination address's and ports but with the direction reversed to allow return traffic to enter the trusted network.

Traffic attempting to enter the trusted network that matches an **evaluate** statement is directed to the temporary access list to determine if it is permitted.

The **reflect** and **evaluate** keywords can be placed anywhere within an extended access list containing normal permit and deny statements.

The temporary access list entry for a TCP session is removed 5 seconds after detecting 2 FINs, or a RST bit being set in the packet. The entry will also be removed if no traffic passes before the timeout period.

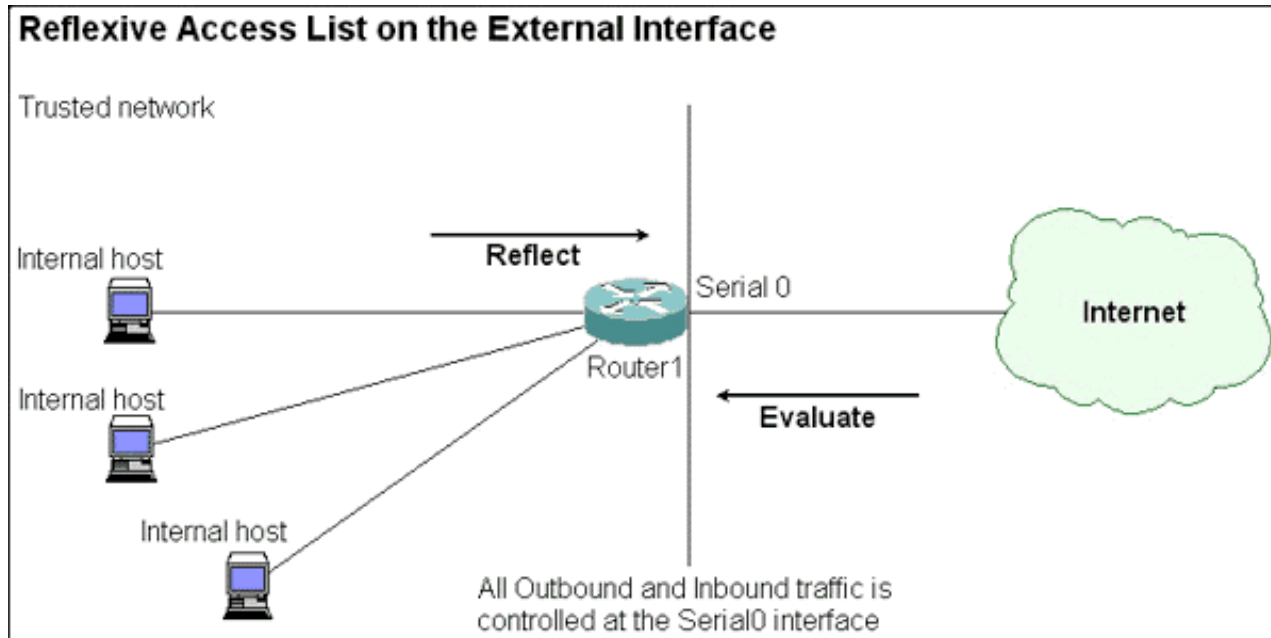
Other connectionless protocols containing no session tracking information will be timed out when no traffic is seen within a defined time limit.

Note: Source and destination ports are a characteristic of TCP & UDP, for ICMP type numbers are used.

### Where to place Reflexive access lists.

There are two basic types of configuration to utilise Reflexive access lists, the first example shown below assumes that the external interface connects directly to the internet and no additional services are provided. In this configuration the access lists are applied to the Serial 0 interface. Use this type of configuration when you have no DMZ with services that need to run outside of the trusted internal network.

Fig.1

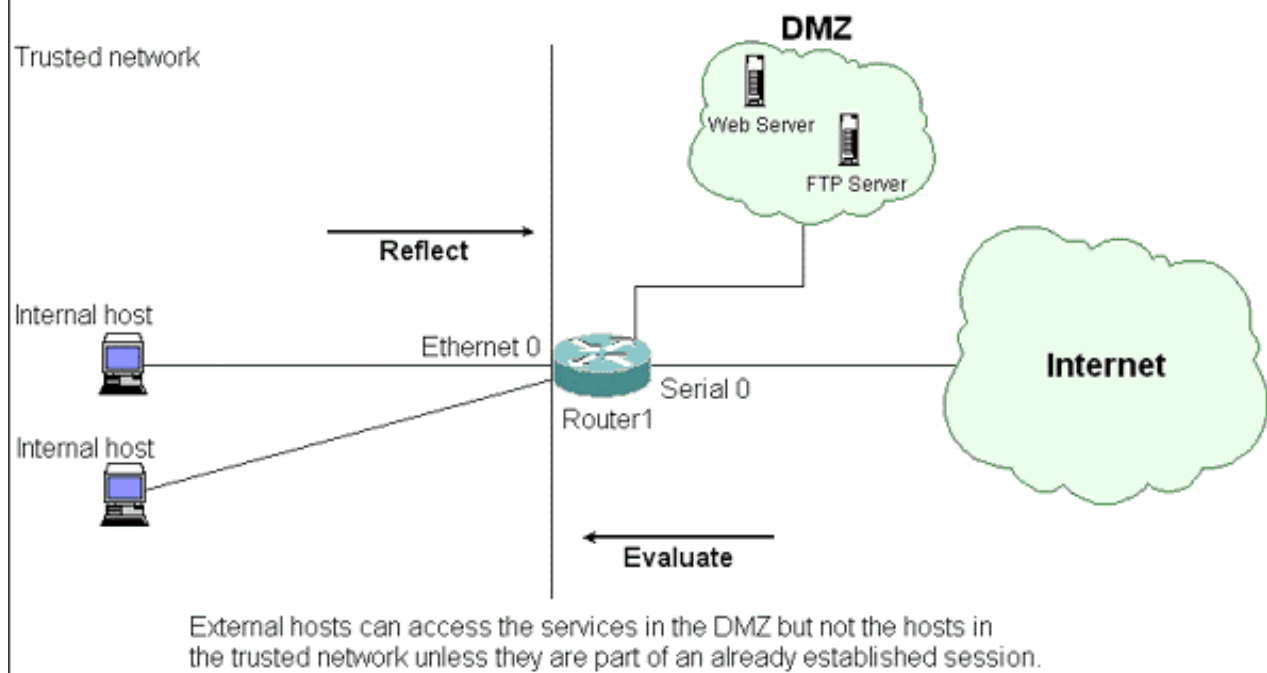


The second type of configuration should be used when there is a requirement for a DMZ where services such as Web or FTP servers need to be placed outside of the trusted network. The difference here lies in where the access lists need to be placed. To provide the type of design shown below the access lists need to be configured on the Ethernet 0 interface. This allows for Internet users to access the services in the DMZ but not have any access to the trusted internal network.

Fig.2

© SA

## Reflexive Access List on the Internal Interface



### Configuring Reflexive access lists

Configuration requires the following steps.

1. Decide on which interface to configure to the access lists.
2. Define an extended named access list to filter traffic leaving the trusted network. Include a **reflect** statement with a reference to a temporary named access list.
3. Define an extended named access list that will filter traffic attempting to enter the trusted network. Include an **evaluate** statement with a reference to the temporary named access list.
4. Apply the access lists to the selected interface in the correct direction.
5. Set a global timeout value. (if required, default is 300 seconds)

Note. Extended named IP access lists must be used when configuring Reflexive access lists, other access list types are not supported.

### External Interface configuration.

In the following example the external interface (Serial 0 in Fig.1) is configured. This filters all traffic at the Serial 0 interface and provides no possibility of running DMZ services.

1. Define the outbound extended access list and include at least one **reflect** statement. The temporary access list referenced by the **reflect** statement can be given any name, for this example **templist** is used. No further configuration is required for the temporary access list other than making a reference to it.

```
ip access-list extended trusted
permit tcp any any reflect templist
```

2. Define the inbound extended access list and include at least one **evaluate** statement.

```
ip access-list extended external
evaluate templist
```

Note: As with any access list care should be taken as to the placement of the entry to achieve the desired results.

3. Apply the access lists to Serial0 in the correct direction.

```
interface Serial0
```

```
ip access-group trusted out
```

```
ip access-group external in
```

4. Set a global timeout value (optional). The default global value is 300 seconds.

```
ip reflexive-list timeout 60
```

A timeout value can be configured per reflect statement if more control is needed over a particular session type. Add **timeout nn** to the end of the reflect statement. For example;

```
permit tcp any any eq telnet reflect templist timeout 120
```

5. Check the router configuration to ensure everything is correct and in the right place,

```
Router1#show config
```

```
interface Serial0
```

```
ip address 10.0.0.1 255.0.0.0
```

```
ip access-group trusted out
```

```
ip access-group external in
```

```
no ip mroute-cache
```

```
bandwidth 56
```

```
clockrate 56000
```

```
ip reflexive-list timeout 60
```

```
ip access-list extended trusted
```

```
permit tcp any any reflect templist
```

```
ip access-list extended external
```

```
evaluate templist
```

The command **show access-lists** should present the following.

```
Router1#sh access-lists
```

```
Extended IP access list trusted
```

```
permit tcp any any reflect templist
```

```
Extended IP access list external
```

```
evaluate templist
```

```
Reflexive IP access list templist
```

A passive FTP session started from the trusted network will give a good example of the type of entry generated by the reflect statement. There are two entries under the "**Reflexive IP access list templist**" statement. The first entry is the result of an **ls** command and shows the FTP data channel, the second entry shows the FTP control channel. The **time left nn** field shows the time remaining before the entry is removed if no further traffic is seen for the session.

```
Router1#sh access-lists
```

```
Extended IP access list trusted
```

```
permit tcp any any reflect templist
```

Extended IP access list external

evaluate templist

Reflexive IP access list templist

permit tcp host 192.168.0.1 eq 49169 host 192.168.1.1 eq

1053 (8 matches) (time left 2)

permit tcp host 192.168.0.1 eq ftp host 192.168.1.1 eq

1052 (22 matches) (time left 57)

### Internal Interface configuration.

Configuration of an internal interface to enable users to access a DMZ is similar to the external interface configuration.

In Fig.2 traffic exiting and entering the trusted network is filtered at the ethernet 0 interface.

This requires that the access lists are placed on the Ethernet 0 interface with the **reflect** and **evaluate** statements correctly positioned in relation to trusted and external traffic flow.

Looking at the configuration below the **ip access-group in & out** statements are reversed in comparison to the external interface configuration. This matches the flow of traffic relative to the router.

```
Router1#show config
```

```
interface Ethernet0
```

```
ip address 192.168.0.254 255.255.255.0
```

```
ip access-group external out
```

```
ip access-group trusted in
```

```
ip reflexive-list timeout 60
```

```
ip access-list extended external
```

```
evaluate templist
```

```
ip access-list extended trusted
```

```
permit tcp any any reflect templist
```

### Removing the configuration

If the configuration needs to be removed use the **no** form of the command;

```
no ip access group trusted out
```

```
no ip access-group external in
```

The no ip reflexive-list timeout command resets the Global timeout to the default 300 seconds.

### Limitations

#### FTP & remote ports

Reflexive access lists require that the remote ports do not change during the session, since active FTP attempts to open a data connection from the server side using a port other than the one defined in the temporary access list it will fail.

Reflexive access lists will only work with passive FTP.

## Router & performance considerations

Before proceeding to configure Reflexive access lists on a production router check the Cisco Website for any known problems with your current hardware and IOS image. The following is a known problem with IOS 11.3

Routers using reflexive access lists in Cisco IOS Release 11.3 may crash with the following stack decode:

```
mgd_timer_set_exptime
mgd_timer_start
ip_maketemp_acl
ip_accesscheck_wrapper
ip_accesscheck_snpa
ip_acc_ck_count_violations
ip_forward
ip_process_pak
```

The problem seems to be more prevalent under high traffic load. Increasing the IP reflexive-list timeout may reduce the likelihood of a crash but will not prevent it entirely. [CSCdj85302]

It is worth checking the number of reflexive access list sessions that the router is capable of supporting. For instance the Catalyst 6000 family has a limit of 512 concurrent entries. Possible effects on MLS and route processor overhead should also be considered.

## Summary

Cisco Reflexive access lists are a useful enhancement to extended access lists, they provide for a greater level of control over sessions and what traffic can explicitly enter your network. They are not too difficult to implement and add another layer of complexity to hinder the potential hacker.

## References

1. Cisco Systems

URL <http://www.cisco.com/>

2. The Internet Protocol Journal. Is your FTP Active or Passive?

URL [http://www.cisco.com/warp/public/759/ipj\\_2-3/ipj\\_2-3\\_oneb.html](http://www.cisco.com/warp/public/759/ipj_2-3/ipj_2-3_oneb.html)

3. Reflexive Access List Command reference

URL

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed\\_cr/secur\\_r/srprt3/srreflex.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_r/srprt3/srreflex.htm)

4. CERT. CERT® Advisory CA-1992-20 Cisco Access List Vulnerability

URL <http://www.cert.org/advisories/CA-1992-20.html>

5. Cisco IOS Software established Access List Keyword Error

URL <http://www.cisco.com/warp/public/770/iosgsracl-pub.shtml>

6. Cisco-ack-proof-concept

URL <http://www.phate.net/docs/security/cisco-ack-proof-concept.txt>

7. Thoughts on Extended Access Lists

URL <http://www.phate.net/docs/security/cisco-acl-thoughts.txt>

8. Firewalking. A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists

URL <http://www.packetfactory.net/projects/firewalk/firewalk-final.html>

9. "Centralized firewall" problems

URL <http://www.phate.net/docs/security/shared-firewall.txt>

10. W.Richard Stephens. TCP/IP Illustrated. Volume 1. The Protocols

11. Syngress Media, Inc. CCNP Advanced Cisco Router Configuration Study Guide

© SANS Institute 2000 - 2005, Author retains full rights.