Access Control Lists and IP Fragments

Document ID: 8014

Contents

Introduction

This White Paper explains the different kinds of Access Control List (ACL) entries and what happens when different kinds of packets encounter these various entries. ACLs are used to block IP packets from being forwarded by a router.

RFC 1858 covers security considerations for IP fragment filtering and highlights two attacks on hosts that involve IP fragments of TCP packets, the Tiny Fragment Attack and the Overlapping Fragment Attack. Blocking these attacks is desirable because they can compromise a host, or tie up all of its internal resources.

RFC 1858 also describes two methods of defending against these attacks, the direct and the indirect. In the direct method, initial fragments that are smaller than a minimum length are discarded. The indirect method involves discarding the second fragment of a fragment set, if it starts 8 bytes into the original IP datagram. Please see RFC 1858 for more details.

Traditionally, packet filters like ACLs are applied to the non−fragments and the initial fragment of an IP packet because they contain both Layer 3 and 4 information that the ACLs can match against for a permit or deny decision. Non−initial fragments are traditionally allowed through the ACL because they can be blocked based on Layer 3 information in the packets; however, because these packets do not contain Layer 4 information, they do not match the Layer 4 information in the ACL entry, if it exists. Allowing the non−initial fragments of an IP datagram through is acceptable because the host receiving the fragments is not able to reassemble the original IP datagram without the initial fragment.

Firewalls can also be used to block packets by maintaining a table of packet fragments indexed by source and destination IP address, protocol, and IP ID. Both the Cisco PIX Firewall and the Cisco IOS® Firewall can filter all the fragments of a particular flow by maintaining this table of information, but it is too expensive to do this on a router for basic ACL functionality. A firewall's primary job is to block packets, and its secondary role is to route packets; a router's primary job is to route packets, and its secondary role is to block them.

Two changes were made in Cisco IOS Software Releases 12.1(2) and 12.0(11) to address some security issues surrounding TCP fragments. The indirect method, as described in RFC 1858 , was implemented as part of the standard TCP/IP input packet sanity checking. Changes were also made to ACL functionality with respect to non−initial fragments.

Types of ACL Entries

There are six different types of ACL lines, and each has a consequence if a packet does or does not match. In the following list, FO = 0 indicates a non−fragment or an initial fragment in a TCP flow, FO > 0 indicates that the packet is a non−initial fragment, L3 means Layer 3, and L4 means Layer 4.

Note: When there is both Layer 3 and Layer 4 information in the ACL line and the **fragments** keyword is present, the ACL action is conservative for both permit and deny actions. The actions are conservative because you do not want to accidentally deny a fragmented portion of a flow because the fragments do not contain sufficient information to match all of the filter attributes. In the deny case, instead of denying a non−initial fragment, the next ACL entry is processed. In the permit case, it is assumed that the Layer 4 information in the packet, if available, matches the Layer 4 information in the ACL line.

Permit ACL line with L3 information only

1. If a packet's L3 information matches the L3 information in the ACL line, it is permitted.
2. If a packet's L3 information does not match the L3 information in the ACL line, the next ACL entry is processed.

Deny ACL line with L3 information only

1. If a packet's L3 information matches the L3 information in the ACL line, it is denied.
2. If a packet's L3 information does not match the L3 information in the ACL line, the next ACL entry is processed.

Permit ACL line with L3 information only, and the fragments keyword is present

If a packet's L3 information matches the L3 information in the ACL line, the packet's fragment offset is checked.

a. If a packet's FO > 0, the packet is permitted.
b. If a packet's FO = 0 , the next ACL entry is processed.

Deny ACL line with L3 information only, and the fragments keyword is present

If a packet's L3 information does match the L3 information in the ACL line, the packet's fragment offset is checked.

a. If a packet's FO > 0, the packet is denied.
b. If a packet's FO = 0, the next ACL line is processed.

Permit ACL line with L3 and L4 information

1. If a packet's L3 and L4 information matches the ACL line and FO = 0, the packet is permitted.
2. If a packet's L3 information matches the ACL line and FO > 0, the packet is permitted.

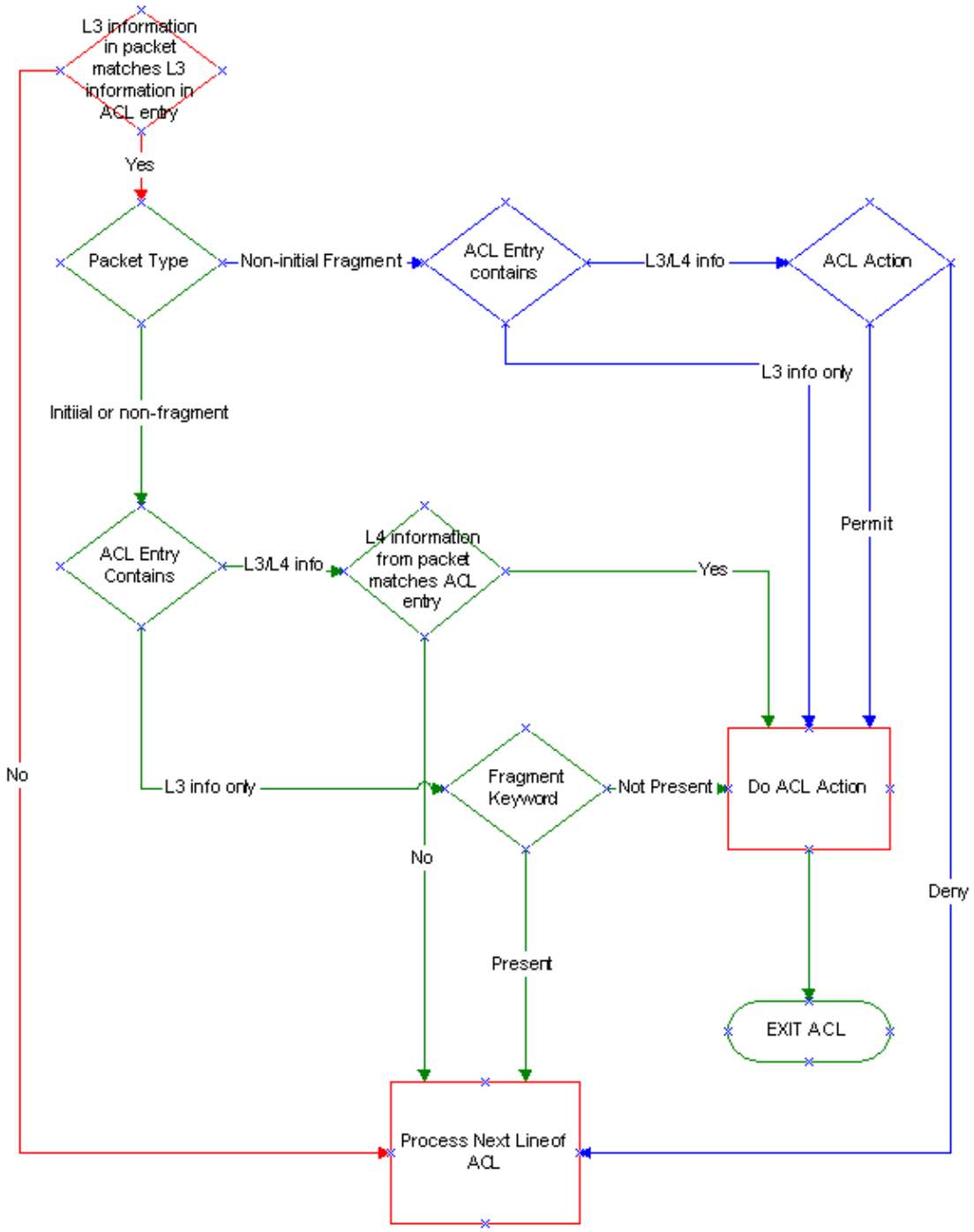
Deny ACL line with L3 and L4 information

1. If a packet's L3 and L4 information matches the ACL entry and FO = 0, the packet is denied.
2. If a packet's L3 information matches the ACL line and FO > 0, the next ACL entry is processed.

ACL Rules Flowchart

The following flowchart illustrates the ACL rules when non−fragments, initial fragments, and non−initial fragments are checked against the ACL.

Note: The non−initial fragments themselves contain only Layer 3, never Layer 4 information, although the ACL may contain both Layer 3 and Layer 4 information.



How Packets Can Match an ACL

Example 1

The following five possible scenarios involve different types of packets encountering ACL 100. Please refer to the table and flow chart as you follow what happens in each situation. The IP address of the web server is 171.16.23.1.

```
access−list 100 permit tcp any host 171.16.23.1 eq 80

access−list 100 deny ip any any
```

The packet is an initial fragment or a non−fragment destined for the server on port 80:

The first line of the ACL contains both Layer 3 and Layer 4 information, which matches the Layer 3 and Layer 4 information in the packet, so the packet is permitted.

The packet is an initial fragment or a non−fragment destined for the server on port 21:

1. The first line of the ACL contains both Layer 3 and Layer 4 information, but the Layer 4 information in the ACL does not match the packet, so the next ACL line is processed.
2. The second line of the ACL denies all packets, so the packet is denied.

The packet is a non−initial fragment to the server in a port 80 flow:

The first line of the ACL contains Layer 3 and Layer 4 information, the Layer 3 information in the ACL matches the packet, and the ACL action is to permit, so the packet is permitted.

The packet is a non−initial fragment to the server in a port 21 flow:

The first line of the ACL contains both Layer 3 and Layer 4 information. The Layer 3 information in the ACL matches the packet, there is no Layer 4 information in the packet, and the ACL action is to permit, so the packet is permitted.

The packet is an initial fragment, non−fragment or non−initial fragment to another host on the server subnet:

1. The first line of the ACL contains Layer 3 information that does not match the Layer 3 information in the packet (the destination address), so the next ACL line is processed.
2. The second line of the ACL denies all packets, so the packet is denied.

Example 2

The following same five possible scenarios involve different types of packets encountering ACL 101. Again, please refer to the table and flow chart as you follow what happens in each situation. The IP address of the web server is 171.16.23.1.

```
access−list 101 deny ip any host 171.16.23.1 fragments

access−list 101 permit tcp any host 171.16.23.1 eq 80

access−list 101 deny ip any any
```

The packet is an initial fragment or non−fragment destined for the server on port 80:

1. The first line of the ACL contains Layer 3 information which matches the Layer 3 information in the packet. The ACL action is to deny, but because the **fragments** keyword is present, the next ACL

entry is processed.
2. The second line of the ACL contains Layer 3 and Layer 4 information, which matches the packet, so the packet is permitted.

The packet is an initial fragment or non−fragment destined for the server on port 21:

1. The first line of the ACL contains Layer 3 information, which matches the packet, but the ACL entry also has the **fragments** keyword, which doesn't match the packet because FO = 0, so the next ACL entry is processed.
2. The second line of the ACL contains Layer 3 and Layer 4 information. In this case, the Layer 4 information does not match, so the next ACL entry is processed.
3. The third line of the ACL denies all packets, so the packet is denied

The packet is a non−initial fragment to the server in a port 80 flow:

The first line of the ACL contains Layer 3 information which matches the Layer 3 information in the packet. Remember that even though this is part of a port 80 flow, there is no Layer 4 information in the non−initial fragment. The packet is denied because the Layer 3 information matches.

The packet is a non−initial fragment to the server in a port 21 flow:

The first line of the ACL contains Layer 3 information only, and it matches the packet, so the packet is denied.

The packet is an initial fragment, non−fragment or non−initial fragment to another host on the server subnet:

1. The first line of the ACL contains Layer 3 information only, and it does not match the packet, so the next ACL line is processed.
2. The second line of the ACL contains Layer 3 and Layer 4 information. The Layer 4 and Layer 3 information in the packet does not match that of the ACL, so the next ACL line is processed.
3. The third line of the ACL denies this packet

fragments Keyword Scenarios

Scenario 1

Router B connects to a web server, and the network administrator does not want to allow any fragments to reach the server. This scenario shows what happens if the network administrator implements ACL 100 versus ACL 101. The ACL is applied inbound on the routers Serial0 (s0) interface and should allow only non−fragmented packets to reach the web server. See the ACL Rules Flowchart and the How Packets Can Match an ACL sections as you follow the scenario.

Consequences of Using the fragments Keyword

The following is ACL 100:

```
access-list 100 permit tcp any host 171.16.23.1 eq 80

access-list 100 deny ip any any
```

The first line of the ACL 100 allows only HTTP to the server, but it also permits non−initial fragments to any TCP port on the server. It permits these packets because non−initial fragments do not contain Layer 4 information, and the ACL logic assumes that if the Layer 3 information matches, then the Layer 4 information would also match, if it was available. The second line is implicit and denies all other traffic.

It is important to note that, as of Cisco IOS Software Releases 12.1(2) and 12.0(11), the new ACL code drops fragments that do not match any other line in the ACL. Earlier releases allow non−initial fragments through if they do not match any other line of the ACL.

The following is ACL 101:

```
access-list 101 deny ip any host 171.16.23.1 fragments

access-list 101 permit tcp any host 171.16.23.1 eq 80

access-list 101 deny ip any any
```

ACL 101 does not allow non−initial fragments through to the server because of the first line. A non−initial fragment to the server is denied when it encounters the first ACL line because the Layer 3 information in the packet matches the Layer 3 information in the ACL line.

Initial or non−fragments to port 80 on the server also match the first line of the ACL for Layer 3 information, but because the fragments keyword is present, the next ACL entry (the second line) is processed. The second line of the ACL permits the initial or non−fragments because they match the ACL line for Layer 3 and Layer 4 information.

Non−initial fragments destined to the TCP ports of other hosts on the 171.16.23.0 network are blocked by this ACL. The Layer 3 information in these packets does not match the Layer 3 information in the first ACL line, so the next ACL line is processed. The Layer 3 information in these packets does not match the Layer 3 information in the second ACL line either, so the third ACL line is processed. The third line is implicit and denies all traffic.

The network administrator in this scenario decides to implement ACL 101 because it permits only non−fragmented HTTP flows to the server.

Scenario 2

A customer has Internet connectivity at two different sites, and there is also a backdoor connection between the two sites. The network administrator's policy is to allow Group A in Site 1 to access the HTTP server at Site 2. The routers at both sites are using private addresses (RFC 1918) and Network Address Translation (NAT) to translate packets that are routed through the Internet.

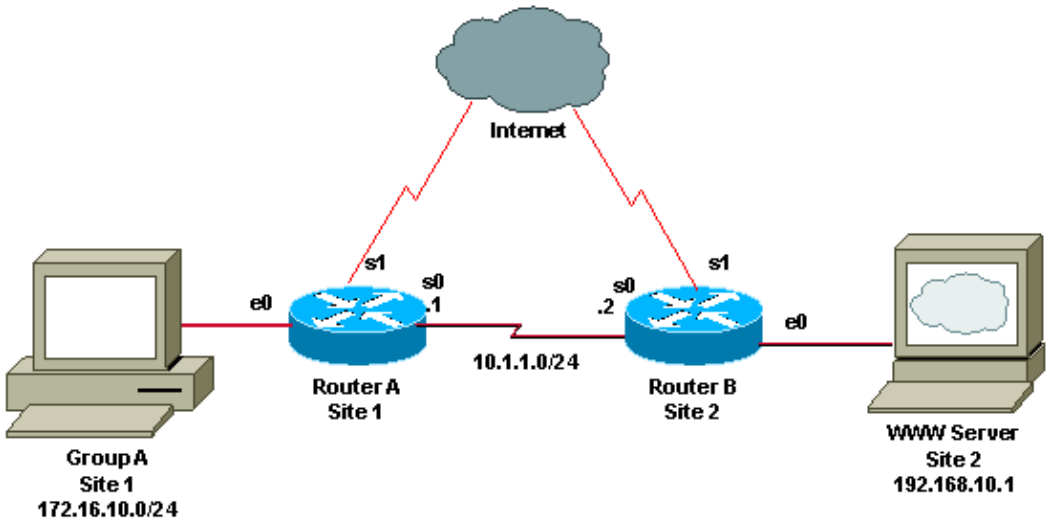
The network administrator at Site 1 is policy−routing the private addresses assigned to Group A, so that they use the backdoor through Router A's Serial0 (s0) when accessing the HTTP server at Site 2. The router at Site 2 has a static route to 172.16.10.0, so that return traffic to Group A is also routed through the backdoor. All other traffic is processed by NAT and routed through the Internet. The network administrator in this scenario has to decide which application or flow is going to work if the packets are fragmented. It is not possible to make both the HTTP and File Transfer Protocol (FTP) flows work at the same time because one or the other breaks.

See the ACL Rules Flowchart and the How Packets Can Match an ACL sections as you follow the scenario.

Explanation of the Network Administrator's Options

In the following example, the route map called FOO on Router A sends packets that match ACL 100 across to Router B through s0. All packets that do not match are processed by NAT and take the default route through the Internet.

Note: If a packet falls off the bottom of the ACL, or is denied by it, then it is not policy−routed.



The following is a partial configuration of Router A, showing that a policy route−map called FOO is applied to interface e0, where the traffic from Group A enters the router:

```
hostname Router_A
int e0
ip policy route-map FOO
route-map FOO permit 10
match ip address 100
set ip next-hop 10.1.1.2

access-list 100 permit tcp 172.16.10.0 0.0.0.255 host 192.168.10.1 eq 80
access-list 100 deny ip any any
```

ACL 100 allows policy routing on both initial, non−fragments and non−initial fragments of HTTP flows to the server. Initial and non−fragments of HTTP flows to the server are permitted by the ACL and policy routed because they match the Layer 3 and Layer 4 information in the first ACL line. Non−initial fragments are permitted by the ACL and policy routed because the Layer 3 information in the packet also matches the first ACL line; the ACL logic assumes that the Layer 4 information in the packet would also match if it was available.

Note: ACL 100 breaks other types of fragmented TCP flows between Group A and the server because the initial and non−initial fragments get to the server through different paths; the initial fragments are processed by NAT and routed through the Internet, but the non−initial fragments of the same flow are policy routed.

A fragmented FTP flow helps illustrate the problem in this scenario. The initial fragments of an FTP flow match the Layer 3 information, but not the Layer 4 information, of the first ACL line, and they are subsequently denied by the second line. These packets are processed by NAT and routed through the Internet.

The non−initial fragments of an FTP flow match the Layer 3 information in the first ACL line, and the ACL logic assumes a positive match on Layer 4 information. These packets are policy routed, and the host

reassembling these packets does not recognize the initial fragments as part of the same flow as the policy−routed non−initial fragments because NAT has changed the source address of the initial fragments.

ACL 100 in the configuration below fixes the FTP problem. The first line of ACL 100 denies both initial and non−initial FTP fragments from Group A to the server.

```
hostname Router_A

int e0
ip policy route-map FOO
route-map FOO permit 10
match ip address 100
set ip next-hop 10.1.1.2

access-list 100 deny tcp 172.16.10.0 0.0.0.255 host 192.168.10.1 fragments
access-list 100 permit tcp 172.16.10.0 0.0.0.255 host 192.168.10.1 eq 80
access-list 100 deny ip any any
```

Initial fragments match on the Layer 3 information in the first ACL line, but the presence of the **fragments** keyword causes the next ACL line to be processed. The initial fragment does not match the second ACL line for Layer 4 information, and so the next implicit line of the ACL is processed, which denies the packet. Non−initial fragments match the Layer 3 information in the first line of the ACL, so they are denied. Both initial and non−initial fragments are processed by NAT and routed through the Internet, so the server has no problem with reassembly.

Fixing FTP flows breaks fragmented HTTP flows because the initial HTTP fragments are now policy routed, but the non−initial fragments are processed by NAT and routed through the Internet.

When an initial fragment of an HTTP flow from Group A to the server encounters the first line of the ACL, it matches on the Layer 3 information in the ACL, but because of the **fragments** keyword, the next line of the ACL is processed. The second line of the ACL permits and policy routes the packet to the server.

When non−initial HTTP fragments destined from Group A to the server encounters the first line of the ACL, the Layer 3 information in the packet matches the ACL line and the packet is denied. These packets are processed by NAT and traverse the Internet to get to the server.

The first ACL in this scenario allows fragmented HTTP flows and breaks fragmented FTP flows. The second ACL allows fragmented FTP flows and breaks fragmented HTTP flows. The TCP flows break in each case because the initial and non−initial fragments take different paths to the server. Reassembly is not possible because NAT has changed the source address of the non−initial fragments.

It is not possible to construct an ACL that allows both kinds of fragmented flows to the server, so the network administrator has to choose which flow he wants to work.

Related Information

- **IP Routing Support Page**
- **Technical Support & Documentation − Cisco Systems**

Updated: Aug 10, 2005 Document ID: 8014