

This information originally appeared in TechRepublic's Cisco Routers and Switches newsletter. [Automatically sign up for our free Cisco Routers and Switches newsletter](#), delivered each Friday!

By David Davis

If you work with Cisco routers, you're more than likely familiar with Cisco IOS access control lists (ACLs). But that doesn't mean you know all there is to know about these important gatekeepers. Access lists are an integral part of working with routers, and they're vital to security.

Test your knowledge of the Cisco IOS with this short, [15 question quiz!](#)

Cisco CCNA's or CCNA candidates should be able to answer these questions in a snap. [This quiz](#) will cover routing protocols, IOS modes, common administrative tasks, and access-list basics.

Because ACLs are a fundamental part of router administration, I want to address 10 things you should know about working with these lists. If you're new to working with Cisco routers, this list offers a good foundation to get you started. But even if you've worked with Cisco routers for a while, it never hurts to review the basics—you might even learn something new.

Here are 10 things you need to know about Cisco IOS access lists, beginning with the basic definition of an ACL.

1

**What is an access control list (ACL)?** – In the Cisco IOS, an access control list is a record that identifies and manages traffic. After identifying that traffic, an administrator can specify various events that can happen to that traffic.

2

**What's the most common type of ACL?** – IP ACLs are the most popular type of access lists because IP is the most common type of traffic. There are two types of IP ACLs: standard and extended. Standard IP ACLs can only control traffic based on the SOURCE IP address. Extended IP ACLs are far more powerful; they can identify traffic based on source IP, source port, destination IP, and destination port.

3

**What are the most common numbers for IP ACLs?** – The most common numbers used for IP ACLs are 1 to 99 for standard lists and 100 to 199 for extended lists. However, many other ranges are also possible.

- Standard IP ACLs: 1 to 99 and 1300 to 1999
- Extended IP ACLs: 100 to 199 and 2000 to 2699

4

**How can you filter traffic using ACLs?** – You can use ACLs to filter traffic according to the "three P's"—per protocol, per interface, and per direction. You can only have one ACL per protocol (e.g., IP or IPX), one ACL per interface (e.g., FastEthernet0/0), and one ACL per direction (i.e., IN or OUT).

5

**How can an ACL help protect my network from viruses?** – You can use an ACL as a packet sniffer to list packets that meet a certain requirement. For example, if there's a virus on your network that's sending out traffic over IRC port 194, you could create an extended ACL (such as number 101) to identify that traffic. You could then use the "debug ip packet 101 detail" command on your Internet-facing router to list all of the source IP addresses that are sending packets on port 194.

6

**What's the order of operations in an ACL?** – Routers process ACLs from top to bottom. When the router evaluates traffic against the list, it starts at the beginning of the list and moves down, either permitting or denying traffic as it goes. When it has worked its way through the list, the processing stops.

That means whichever rule comes first takes precedence. If the first part of the ACL denies traffic, but a lower part of the ACL allows it, the router will still deny the traffic. Let's look at an example:

```
Access-list 1 permit any
Access-list 1 deny host 10.1.1.1
Access-list 1 deny any
```

What does this ACL permit? The first line permits anything. Therefore, all traffic meets this requirement, so the router will permit all traffic, and processing will then stop.

7

**What about traffic you don't specifically address in an ACL?** – At the end of an ACL is an implicit deny statement. Whether you see the statement or not, the router denies all traffic that doesn't meet a condition in the ACL. Here's an example:

```
Access-list 1 deny host 10.1.1.1
Access-list 1 deny 192.168.1.0 0.0.0.255
```

What traffic does this ACL permit? None: The router denies all traffic because of the implicit deny statement. In other words, the ACL really looks like this:

```
Access-list 1 deny host 10.1.1.1
Access-list 1 deny 192.168.1.0 0.0.0.255
Access-list 1 deny ANY
```

8

**Can I name an ACL?** – Numbers—who needs numbers? You can also name your ACLs so you can more easily identify their purpose. You can name both standard and extended ACLs. Here's an example of using a named ACL:

```
router(config)# ip access-list ?
  extended      Extended Access List
  log-update     Control access list log updates
  logging        Control access list logging
  resequence     Resequence Access List
  standard       Standard Access List

router(config)# ip access-list extended test
router(config-ext-nacl)#
router(config-ext-nacl)# 10 deny ip any host 192.168.1.1
router(config-ext-nacl)# exit
router(config)# exit
router# show ip access-list
Extended IP access list test
  10 deny ip any host 192.168.1.1
```

9

**What's a numbering sequence?** – In the "old days," you couldn't edit an ACL—you could only copy it to a text editor (such as Notepad), remove it, edit it in notepad, and then re-create it. In fact, this is still a good way to edit some Cisco configurations.

However, this approach can also create a security risk. During the time you've removed the ACL to modify it, the router isn't controlling traffic as needed. But it's possible to edit a numbered ACL with commands. Here's an example:

```
router(config)# access-list 75 permit host 10.1.1.1
router(config)^Z
router# conf t
Enter configuration commands, one per line. End with CNTL/Z.

router(config)# ip access-list standard 75

router(config-std-nacl)# 20 permit any
router(config-std-nacl)# no 10 permit 10.1.1.1
router(config-std-nacl)^Z

router# show ip access-lists 75
Standard IP access list 75
    20 permit any
router#
```

10

**How else can I use an ACL?** – ACLs aren't just for filtering traffic. You can also use them for a variety of operations. Let's look at some of their possible other uses:

- To control debug output: You can use the debug list X command to control debug output. By using this command before another debug command, the command only applies to what you've defined in the list.
- To control route access: You can use a routing distribute-list ACL to only permit or deny certain routes either into or out of your routing protocol.
- As a BGP AS-path ACL: You can use regular expressions to permit or deny BGP routes.
- For router management: You can use an ACL to control which workstation or network manages your router with an ACL and an access-class statement to your VTY lines.
- For encryption: You can use ACLs to determine how to encrypt traffic. When encrypting traffic between two routers or a router and a firewall, you must tell the router what traffic to encrypt, what traffic to send unencrypted, and what traffic to drop.

**To wrap up this review, I'll leave you with one last tip:** Don't forget to use remark statements in your ACLs. They'll come in handy when you have to troubleshoot something later.



*David Davis has worked in the IT industry for 12 years and holds several certifications, including CCIE, MCSE+I, CISSP, CCNA, CCDA, and CCNP. He currently manages a group of systems/network administrators for a privately owned retail company and performs networking/systems consulting on a part-time basis.*

## Additional resources

- **Subscribe to TechRepublic's [Downloads RSS Feed](#) **
- Sign up for TechRepublic's [Downloads Weekly Update newsletter](#)
- Sign up for TechRepublic's [Cisco Routers and Switches newsletter](#)
- Check out all of TechRepublic's [free newsletters](#)
- [Learn how to troubleshoot slowdowns and crashes on Cisco routers](#)
- [CCNA Command Quick Reference: Configuring a Router](#)
- [Cisco IOS router: Lock it down in 10 steps](#)

## Version history

**Version:** 1.0

**Published:** August 12, 2005

## Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to [drop us a line](#) and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Downloads Team