

ACI

Mehdi JAFARIZADEH

september 6, 2024

Abstract

tets

Introduction to ACLs on Cisco Devices

What are Access Control Lists?

Access Control Lists (ACLs) represent a vital component of network security and traffic management within Cisco devices. Essentially, an ACL consists of a series of rules that regulate the flow of network traffic. They determine whether specific packets should be allowed or denied according to various criteria. Such criteria may involve source and destination IP addresses, protocols, port numbers, and additional factors.

ACLs function like gatekeepers. They decide which types of network traffic are permitted to pass through interfaces found on routers, switches, and firewalls. Their role is critical in filtering both incoming and outgoing traffic. By protecting systems from unauthorized access or potential attacks, they assist network administrators in maintaining secure operations. Additionally, ACLs enhance network performance by effectively managing bandwidth and minimizing unnecessary traffic.

In summary, ACLs function as an effective method to control traffic flow while bolstering overall network security and efficiency [1].

Types of ACLs: Standard ACLs vs. Extended ACLs

ACLs are categorized into two major types: **Standard ACLs** and **Extended ACLs**.

1. **Standard ACLs** are the most fundamental form of these lists. They use numbers from 1 to 99 (plus extended ranges from 1300 to 1999). The primary function of Standard ACLs is to filter traffic based on the source IP address only. They disregard the destination IP address and omit consideration of the type of traffic (protocol or port). This makes them appropriate for less complex scenarios requiring basic control, such as allowing or blocking entire groups based on their source address.

Use case: Consider a scenario where you intend to restrict traffic from a certain department within an organization. A standard ACL can serve this purpose effectively. For example, if you intend to block users from a designated subnet from accessing a specific part of your network, implementing a standard ACL would fulfill that purpose by preventing any traffic that originates from that particular subnet.

2. **Extended ACLs:** These offer a more granular level of control over filtering traffic. Extended ACLs can filter data based on both the source and the destination IP addresses, as well as on protocols such as TCP, UDP, ICMP, and even particular port numbers. They are identified by numbers ranging from 100 to 199 (and extended ranges 2000 to 2699).

Use case: If an administrator needs to block all HTTP traffic (port 80) coming from a specific source IP to a certain web server while allowing other kinds of traffic, an extended ACL would be appropriate. Extended ACLs are frequently used in situations that demand precise traffic control. This includes establishing security policies for various application types or services.

The versatility of extended ACLs makes them a powerful asset for network security. They are especially useful in complex environments where multiple protocols and services are operational [2].

Why ACLs are Essential

ACLs offer significant advantages to network administrators in terms of security, performance, and traffic management.

1. **Network Security:** One primary reason for applying ACLs is to protect a network from unauthorized access. They serve as the first line of defense by blocking harmful or unwanted traffic from entering the network. By managing which users or devices can interact across network segments, administrators can noticeably mitigate the risk of attacks such as denial-of-service (DoS) or data breaches.
2. **Traffic Filtering:** ACLs are crucial for filtering and managing traffic. They allow networks to efficiently handle large volumes of data. For example, they can be used to limit bandwidth-intensive applications or services, preventing overload on the network. By implementing ACLs, administrators prioritize important business applications while blocking unnecessary or harmful traffic. This ensures more efficient operations within the network.
3. **Performance Optimization:** ACLs, or Access Control Lists, are highly beneficial. They assist in filtering unnecessary traffic, thereby enhancing the overall performance of the network. With reduced unwanted traffic, congestion decreases, resulting in faster transmission speeds for critical applications. ACLs also prevent excessive bandwidth usage, ensuring that essential services receive priority over less important or harmful traffic.
4. **Traffic Control & Compliance** In addition to enhancing security and performance, ACLs are key in complying with regulatory rules. They enforce traffic control policies that many organizations are required to follow by law. This often involves restricting access to sensitive data or ensuring that only authorized users can access certain network segments. With ACLs, it is straightforward to implement these security measures and meet standards such as PCI-DSS or HIPAA.

In summary, ACLs provide an effective approach for network administrators to improve security, optimize traffic flow, and ensure compliance with regulations. Their ability to manage traffic using simple or complex rules makes them essential for managing today's networks [3].

Basic ACL Configuration Steps

Configuring Standard ACLs

Standard ACLs provide fundamental control over filtering traffic based on the source IP address. The following outlines the process to configure a Standard ACL on a Cisco device:

1. **Access the CLI of the Cisco Device:** Start by connecting to your Cisco router or switch. You may use terminal software such as PuTTY or connect directly through the console.
2. **Enter Global Configuration Mode:** Enter Global Configuration Mode: After accessing the command-line interface (CLI), proceed to enter global configuration mode.

```
Router> enable
Router# configure terminal
```

3. **Create a Standard ACL:** To create a Standard ACL, execute the relevant command. In this example, traffic will be blocked from the IP address 192.168.1.10, while all other traffic is permitted. Standard ACLs use numbers ranging from 1 to 99.

```
Router(config)# access-list 10 deny 192.168.1.10 0.0.0.0
Router(config)# access-list 10 permit any
```

The first command denies traffic from 192.168.1.10, while the second command permits all other traffic. The 0.0.0.0 wildcard mask ensures the ACL applies only to the exact IP address specified.

4. **Applying the ACL to an Interface:** Once the ACL is created, apply it to an interface to control the traffic passing through it. In this case, the ACL will be applied to inbound traffic on interface GigabitEthernet0/0.

```
Router(config)# interface gigabitEthernet 0/0
Router(config-if)# ip access-group 10 in
```

5. **Save the Configuration:** After applying the ACL, save the configuration to ensure it remains active after a device reboot.

```
Router(config)# end
Router# write memory
```

By completing these steps, you will have successfully configured a Standard ACL that blocks traffic from one specific source IP address while permitting all other traffic [4].

Configuring Extended ACLs

Extended ACLs provide significantly more granular control over network traffic than Standard ACLs. These Extended ACLs are capable of filtering based on source and destination IP addresses, as well as protocols and ports. The following outlines the process for configuring an Extended ACL.

1. **Enter Global Configuration Mode:** Begin by entering global configuration mode, as previously described.

```
Router> enable
Router# configure terminal
```

2. **Create an Extended ACL:** Extended ACLs use numbers from 100 to 199 or 2000 to 2699. The example below demonstrates how to block HTTP traffic (port 80) from the source IP 192.168.2.0/24 to the destination 10.1.1.0/24. All other traffic will be permitted.

```
Router(config)# access-list 110 deny tcp 192.168.2.0 0.0.0.255
10.1.1.0 0.0.0.255 eq 80
Router(config)# access-list 110 permit ip any any
```

The "deny tcp" command indicates that TCP traffic from 192.168.2.0 to 10.1.1.0 on port 80 (HTTP) will be denied. The "permit ip any any" rule permits all other traffic to pass through.

3. **Apply the Extended ACL to an Interface:** Similar to Standard ACLs, it is necessary to attach the Extended ACL to a specific interface to control the flow of traffic.

```
Router(config)# interface gigabitEthernet 0/1
Router(config-if)# ip access-group 110 in
```

4. **Save the Configuration:** Always save the configuration to ensure it remains permanent.

```
Router(config)# end
Router# write memory
```

With this setup, HTTP traffic from the specified source to the specified destination will be blocked, while all other types of traffic will be allowed [5].

Named ACLs

Named ACLs assist with easier management and scalability, especially in larger networks where access control needs can become complex. The primary difference between numbered ACLs and named ACLs is that named ACLs use descriptive names that describe their functions instead of just numbers, making them simpler to identify and update.

1. **Create a Named ACL:** Rather than using a number, create your ACL with a name. For instance, create a named ACL called BLOCK_HTTP that blocks HTTP traffic from 192.168.3.0/24 to 10.1.2.0/24.

```
Router(config)# ip access-list extended BLOCK_HTTP
Router(config-ext-nacl)# deny tcp 192.168.3.0 0.0.0.255 10.1.2.0
0.0.0.255 eq 80
Router(config-ext-nacl)# permit ip any any
```

2. **Apply the Named ACL to an Interface:** After setting up the ACL, ensure you apply it to the correct interface.

```
Router(config)# interface gigabitEthernet 0/2
Router(config-if)# ip access-group BLOCK_HTTP in
```

3. **Advantages of Named ACLs:** Named ACLs make management easier in various ways:

- **Descriptive Names:** Using names such as BLOCK_HTTP allows network administrators to easily understand the ACL's intent. There's no need to remember complex numerical ranges.
- **Modifications:** Named ACLs facilitate easier editing. Rather than having to remove and recreate the entire list, administrators can simply modify specific rules within a named ACL. This process does not affect the entire access list.
- **Scalability:** Managing larger networks is simplified with named ACLs. They allow for more rules and configurations. As networks grow, ACLs will often require modifications. Named ACLs provide the necessary flexibility and transparency for such updates.

For large and complex network environments, named ACLs are considered a best practice. They make administering these lists more intuitive and scalable [6].

Best Practices for Implementing ACLs

Placement of ACLs

Proper placement of Access Control Lists (ACLs) is vital for enhancing network security, performance, and manageability. Where you place your ACLs—inbound or outbound, near sensitive areas, or at the network edge—can significantly influence how effectively they filter traffic.

1. **Inbound vs. Outbound ACLs:**

- **Inbound ACLs:** Inbound ACLs filter traffic as it enters an interface. They block unwanted or malicious packets before they enter the network itself. Often, inbound ACLs block specific types of traffic before they reach routers or switches, helping to conserve processing resources effectively.

Best practice: It is a best practice to implement inbound ACLs to block harmful

traffic as soon as possible. This approach reduces the strain on internal network resources and mitigates risks from threats such as Distributed Denial of Service (DDoS) attacks.

- **Outbound ACLs:** Outbound ACLs fulfill a different role. These ACLs filter traffic before it leaves the interface, directing it toward its destination. They are crucial in managing outgoing traffic. For example, they can prevent internal users from accessing restricted external sites or services.

Best practice: When enforcing traffic policies for outbound traffic, best practice suggests implementing outbound ACLs. This could involve limiting access to specific external services or monitoring data that exits sensitive network segments.

2. Edge of the Network vs. Locations Closer to Sensitive Resources:

- **Edge of the Network:** At the Edge of the Network: Applying ACLs at this point—such as on perimeter routers or firewalls—allows you to stop unwanted traffic immediately. This is critical when filtering external traffic, as security threats are at their highest here.

Best practice: The best practice in this case is positioning ACLs at the network edge to reinforce security policies. This action prevents unauthorized traffic from penetrating the network, ensuring that threats are addressed at the earliest point and reducing exposure risks for internal systems.

- **Closer to Sensitive Resources:** In addition, ACLs can be deployed closer to sensitive resources—such as data centers, servers, or critical applications—to provide an additional layer of security. This method is particularly advantageous when there is a need for stricter access control for internal users or specific services.

Best practice: It is advisable to position Access Control Lists (ACLs) closer to critical resources when detailed control is required. For example, an ACL can be applied to a particular switch interface to restrict access to sensitive servers based on user roles or device types.

Order of ACL Entries

ACLs are examined in sequence, from the top downwards. This means that every packet is checked against each rule one by one. Once a match is found, the ACL ceases further rule checks. Therefore, the arrangement of ACL entries plays a vital role in ensuring that filtering functions as intended.

1. **Sequential Processing:** ACLs operate using a top-down approach with their rules. When a match is found, the ACL promptly ceases processing and applies the specified action (which may be either permit or deny). If no matches are found, the packet is denied by default, referred to as implicit deny. If rules are improperly ordered, it may result in unexpected issues.

Example: Consider the following ACL entries:

```
access-list 100 deny tcp 192.168.1.0 0.0.0.255 any eq 80
access-list 100 permit ip any any
```

Initially, this ACL denies HTTP traffic from the 192.168.1.0 network while permitting all other traffic afterward. However, if you reverse the order of these entries, the "permit ip any any" rule would allow all traffic before the HTTP block is evaluated.

2. Why Ordering Matters:

- **Specific Rules First:** Specific rules should be placed before general ones. For instance, if you have rules that focus on particular IP addresses or services, they should be listed first. General rules (such as "permit all other traffic") should appear at the bottom. This ensures accurate application of the intended traffic filtering.
- **Deny Rules Before Permit Rules:** Deny rules should precede permit rules to avoid unintentionally allowing unauthorized traffic because of a following permissive rule.

Best Practice: Always carefully review and meticulously arrange your ACL entries. Ensure that critical deny rules or specific entries are positioned at the top of the ACL, followed by broader permit rules.

Performance Considerations

When using ACLs in high-traffic environments, performance becomes critical. High volumes of traffic can affect speed, especially with long or complex ACLs. Here are some key points to consider to help mitigate performance issues:

1. **Limit ACL Complexity:** Limit ACL Complexity: A large number of rules and conditions can impede packet processing. On high-speed connections, each packet must be checked against all the rules, which takes time and may reduce network speed. Simplifying the rules or reducing the number of entries can enhance performance.

Best practice: Keep ACLs simple. Use fewer rules whenever possible. For instance, rather than having multiple individual rules for similar types of traffic, consolidate them into a single rule using wildcard masks.

2. **Use Hardware-Based ACLs:** Some Cisco devices offer hardware-based ACLs, where specialized hardware (such as ASICs) performs the processing instead of the device's CPU. Using these hardware-based ACLs can significantly improve performance in high-traffic areas.

Best practice: For networks requiring high performance, ensure that your devices support hardware-based ACLs. Routers and switches equipped with ASICs can manage large ACLs without significantly affecting packet forwarding rates.

3. **Positioning of ACLs:** Position ACLs strategically to reduce the volume of traffic requiring processing. For example, filter unwanted traffic as early as possible, such as at the network edge. This reduces the traffic volume for deeper inspections later in the network and enhances overall performance.

4. **Monitoring and Optimization:** Regularly monitor ACL performance along with network traffic to identify performance slowdowns. Tools like Cisco's **NetFlow** or **SNMP** can help measure the impact of ACLs on network performance. Adjusting the rules when necessary can maintain optimal performance.

Best practice: It is important to regularly review and optimize ACL configurations based on real-time performance monitoring data. This ensures they do not negatively impact network throughput [7].

Advanced ACL Use Cases

Traffic Filtering

One of the most common uses for ACLs is traffic filtering. They can filter based on specific criteria such as IP addresses, protocols, or services. By controlling which traffic is allowed or blocked, network administrators can enhance security. They also optimize performance and enforce access policies [8].

1. **Filtering Based on IP Addresses:** You can configure ACLs to allow or deny traffic from specific source or destination IP addresses. This is useful when limiting access to certain parts of the network. For example, you might require blocking external traffic from reaching sensitive internal resources or restricting access for particular departments or subnets within a company.

Use case: Imagine you intend to block traffic from the IP range 192.168.50.0/24 from entering the corporate data center (10.10.10.0/24). You could configure an extended ACL as follows:

```
access-list 101 deny ip 192.168.50.0 0.0.0.255 10.10.10.0
0.0.0.255
access-list 101 permit ip any any
```

2. **Filtering Traffic Using Protocols:** ACLs can filter traffic based on certain protocols, including TCP, UDP, ICMP, and others. For instance, you might wish to block all ICMP (ping) traffic from external networks. This helps mitigate denial-of-service attacks or reconnaissance attempts.

Use case: To block ICMP traffic from an external network, such as 203.0.113.0/24, you could configure an ACL as follows:

```
access-list 102 deny icmp 203.0.113.0 0.0.0.255 any
access-list 102 permit ip any any
```

3. **Filtering Traffic by Services (Ports):** Extended ACLs can also control traffic based on specific ports or services. For example, it is possible to block HTTP (port 80) or FTP (port 21) traffic. This enables detailed control over which services can be accessed on the network.

Use case: If you intend to block all HTTP traffic from the IP range 192.168.1.0/24 to the outside world while allowing other types of traffic, the configuration might appear as follows:


```
access-list 103 deny tcp 192.168.1.0 0.0.0.255 any eq 80
access-list 103 permit ip any any
```

In this case, the ACL permits all other traffic but denies HTTP traffic.

Security Access Management:

ACLs are crucial in securing access to network devices such as routers and switches. This is especially important when using remote management protocols such as SSH or Telnet. By applying ACLs to management interfaces, administrators can limit who has access to the device. This reduces the attack surface and enhances security [9].

1. **Limiting SSH Access:** SSH (Secure Shell) is commonly used for securely managing Cisco devices remotely. However, if SSH access is not properly controlled, it creates a security risk, particularly when connected to the internet. By using ACLs, administrators can restrict SSH access to only trusted IP addresses.

Use case: If you intend to allow only a specific IP address (192.168.100.10) to SSH into a router, you can configure an ACL as shown below:

```
access-list 104 permit tcp host 192.168.100.10 any eq 22
access-list 104 deny ip any any
```

Next, apply the ACL to the VTY (virtual terminal) lines:

This setup ensures that only the designated IP address has access to the router using SSH, while access from all other sources will be denied.

2. **Restricting Telnet Access:** Similar to SSH, Telnet can also be restricted through ACLs. However, it is important to emphasize that Telnet is not recommended for secure access since it transmits data in plain text. If Telnet needs to be used internally, and you wish to restrict it to specific devices, an ACL can be applied in the same way as with SSH.

Use case: Permit Telnet access solely from a particular internal subnet (10.10.10.0/24):

```
access-list 105 permit tcp 10.10.10.0 0.0.0.255 any eq 23
access-list 105 deny ip any any
```

Apply this ACL to the Telnet lines:

```
line vty 0 4
access-class 105 in
```

This configuration restricts Telnet access solely to the defined subnet, thus preventing unauthorized access.

Time-Based ACLs

Time-based ACLs enable network administrators to control traffic dynamically based on defined time intervals. This feature is particularly advantageous for enforcing access policies during specific times of the day or for restricting access outside of business hours [10].

1. **Creating Time Ranges:** Time-based ACLs function by establishing a time range that defines when the ACL should be active. For example, you may wish to allow specific traffic during business hours (8:00 AM to 6:00 PM, Monday to Friday) and block it at other times.

Use case: Permit web traffic (HTTP) from a specific IP range (192.168.1.0/24) only during business hours (8:00 AM to 6:00 PM) on weekdays.

```
time-range WORK_HOURS
  periodic weekdays 8:00 to 18:00
```

2. **Implementing the Time-Based ACL:** Once the time range is defined, you can apply it to an ACL. In this example, the ACL allows HTTP traffic within the specified time frame and denies it outside of that time.

```
access-list 106 permit tcp 192.168.1.0 0.0.0.255 any eq 80 time-
  range WORK_HOURS
access-list 106 deny ip any any
```

3. ImUse Cases for Time-Based ACLs:

- **Network Access Control:** Allow staff to access certain network resources only during business hours.
- **Bandwidth Management:** Restrict the usage of bandwidth-intensive services (like streaming) to off-peak times.
- **Enhanced Security:** Automatically restrict traffic from non-essential services after business hours, which reduces the attack surface when network monitoring is reduced.

Use case:
ACL Logging & Monitoring

References

1. **Cisco.**
configure and filter ip access lists.
2. **Cisco.**
security configuration guide: access control lists.
3. **Cisco.**
implementing access control lists (acls).
4. **Cisco.**
standard access control lists.
5. **Cisco.**
extended access control lists.
6. **Cisco.**
named access control lists.
7. **twingate.**
how they work & best practices.
8. **Cisco.**
configuring access control lists.
9. **Cisco.**
security configuration guide: access control lists.
10. **Cisco.**
time-based access lists using time ranges.