Understanding Firewall Filter Match Conditions

Before you define terms for firewall filters, you must understand how the conditions that you specify in a term are handled and how to specify interface filter, numeric filter, address filter, and bit-field filter match conditions to achieve the desired filtering results.

Filter Match Conditions

In the from statement of a firewall filter term, you specify the conditions that the packet must match for the action in the then statement to be taken. All conditions in the from statement must match for the action to be taken. The order in which you specify match conditions is not important, because a packet must match all the conditions in a term for a match to occur.

If you specify no match conditions in a term, that term matches all packets.

An individual condition in a from statement cannot contain a list of values. For example, you cannot specify numeric ranges or multiple source or destination addresses.

Individual conditions in a from statement cannot be negated. A negated condition is an explicit mismatch.

Numeric Filter Match Conditions

Numeric filter conditions match packet fields that are identified by a numeric value, such as port and protocol numbers. For numeric filter match conditions, you specify a keyword that identifies the condition and a single value that a field in a packet must match.

You can specify the numeric value in one of the following ways:

- Single number—A match occurs if the value of the field matches the number. For example:

  ```
  source-port 25;
  ```

- Text synonym for a single number— A match occurs if the value of the field matches the number that corresponds to the synonym. For example:

  ```
  source-port http;
  ```

To specify more than one value in a filter term, you enter each value in its own match statement. For example, a match occurs in the following term if the value of vlan field is 10 or 30.

```
[edit firewall family family-name filter filter-name term term-name from]
vlan 10;
vlan 30;
```

The following restrictions apply to numeric filter match conditions:

- You cannot specify a range of values.
- You cannot specify a list of comma-separated values.
- You cannot exclude a specific value in a numeric filter match condition. For example, you cannot specify a condition that would match only if the match condition was not equal to a given value.

Interface Filter Match Conditions

Interface filter match conditions can match interface name values in a packet. For interface filter match conditions, you specify the name of the interface, for example:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set interface ge-0/0/1
```

Port and VLAN interfaces do not use logical unit numbers. However, a firewall filter that is applied to a router interface can specify the logical unit number in the interface filter match condition, for example:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set interface ge-0/1/0.0
```

You can include the * wildcard as part of the interface name, for example:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set interface ge-0/*/1
user@host# set interface ge-0/1/*
user@host# set interface ge-*
```

IP Address Filter Match Conditions

Address filter match conditions can match prefix values in a packet, such as IP source and destination prefixes. For address filter match conditions, you specify a keyword that identifies the field and one prefix of that type that a packet must match.

You specify the address as a single prefix. A match occurs if the value of the field matches the prefix. For example:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set destination-address 10.2.1.0/28;
```

Each prefix contains an implicit 0/0 except statement, which means that any prefix that does not match the prefix that is specified is explicitly considered not to match.

To specify the address prefix, use the notation prefix/prefix-length. If you omit prefix-length, it defaults to /32. For example:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set destination-address 10
[edit firewall family family-name filter filter-name term term-name from]
user@host# show
destination-address {
10.0.0.0/32;
}
```

To specify more than one IP address in a filter term, you enter each address in its own match statement. For example, a match occurs in the following term if the value of the source-address field matches either of the following source-address prefixes:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set source-address 10.0.0.0/8
user@host# set source-address 10.1.0.0/16
```

MAC Address Filter Match Conditions

MAC address filter match conditions can match source and destination MAC address values in a packet. For MAC address filter match conditions, you specify a keyword that identifies the field and one value of that type that a packet must match.

You can specify the MAC address as six hexadecimal bytes in the following formats:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set destination-mac-address 0011.2233.4455

[edit firewall family family-name filter filter-name term term-name from]
user@host# set destination-mac-address 00:11:22:33:44:55

[edit firewall family family-name filter filter-name term term-name from]
user@host# set destination-mac-address 001122334455
```

To specify more than one MAC address in a filter term, you enter each MAC address in its own match statement. For example, a match occurs in the following term if the value of the source-mac-address field matches either of the following addresses.

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set source-mac-address 00:11:22:33:44:55
user@host# set source-mac-address 00:11:22:33:20:15
```

Bit-Field Filter Match Conditions

Bit-field filter conditions match packet fields if particular bits in those fields are or are not set. You can match the IP options, TCP flags, and IP fragmentation fields. For bit-field filter match conditions, you specify a keyword that identifies the field and tests to determine that the option is present in the field.

To specify the bit-field value to match, enclose the value in double quotation marks. For example, a match occurs if the RST bit in the TCP flags field is set:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set tcp-flags "rst"
```

Typically, you specify the bits to be tested by using keywords. Bit-field match keywords always map to a single bit value. You also can specify bit fields as hexadecimal or decimal numbers.

To match multiple bit-field values, use the logical operators, which are described in Table 1. The operators are listed in order from highest precedence to lowest precedence. Operations are left-associative.

Table 1: Actions for Firewall Filters

Logical Operators	Description
!	Negation.
& or +	Logical AND.
| or ,	Logical OR.

To negate a match, precede the value with an exclamation point. For example, a match occurs only if the RST bit in the TCP flags field is not set:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set tcp-flags "!rst"
```

In the following example of a logical AND operation, a match occurs if the packet is the initial packet on a TCP session:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set tcp-flags "syn&!ack"
```

In the following example of a logical OR operation, a match occurs if the packet is not the initial packet on a TCP session:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set tcp-flags "syn|ack"
```

For a logical OR operation, you can specify a maximum of two match conditions in a single term. If you need to match more than two bit-field values in a logical OR operation, configure the same match condition in consecutive terms with additional bit-field values. In the following example, the two terms configured match the SYN, ACK, FIN, or RST bit in the TCP flags field:

```
[edit firewall family family-name filter filter-name term term-name1 from]
user@host# set tcp-flags "syn|ack"
[edit firewall family family-name filter filter-name term term-name2 from]
user@host# set tcp-flags "fin|rst"
```

You can use text synonyms to specify some common bit-field matches. You specify these matches as a single keyword. In the following example of a text synonym, a match occurs if the packet is the initial packet on a TCP session:

```
[edit firewall family family-name filter filter-name term term-name from]
user@host# set tcp-flags tcp-initial
```

Related Topics

- Firewall Filters for EX Series Switches Overview
- Understanding How Firewall Filters Test a Packet's Protocol
- Example: Configuring Firewall Filters for Port, VLAN, and Router Traffic on EX Series Switches
- Example: Using Filter-Based Forwarding to Route Application Traffic to a Security Device on EX Series Switches
- Firewall Filter Match Conditions and Actions for EX Series Switches

---

Published: 2009-12-18