Rate Filter

Rate Filter Overview

Type	Module (basic)
Usage	Context
Instance Type	Singleton
Enabled	`false`

Rate-based attacks attempt to overwhelm a network or host by sending excessive traffic to a network or host, causing it to slow down or deny legitimate requests. You can use rate-based prevention to change the action of an intrusion rule in response to excessive matches on that rule.

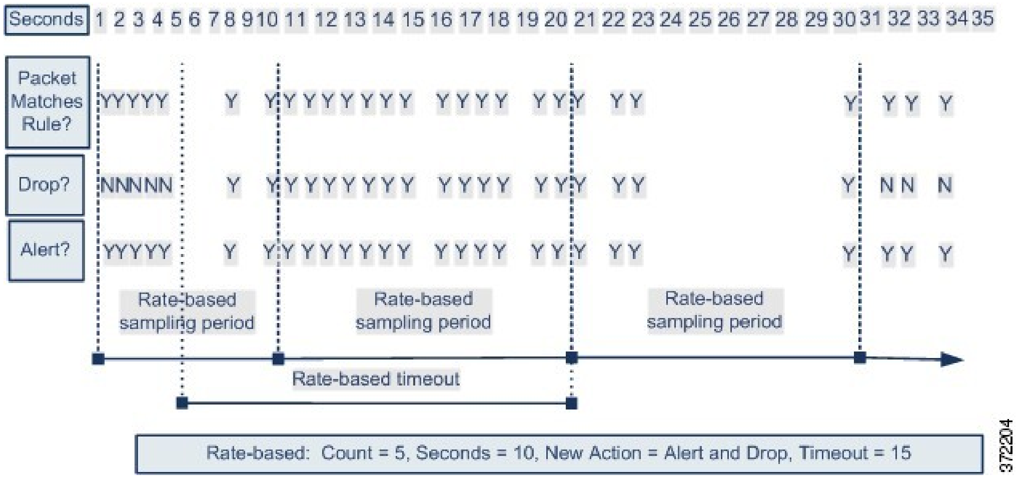
The `rate_filter` detects when too many matches for a rule occur within a given interval. You can use this feature on managed devices deployed inline to block rate-based attacks for a specified time, then revert to a rule state where rule matches only generate events and do not drop traffic.

You can configure the `rate_filter` to respond to any intrusion rule, but the rule you specify must be enabled for `rate_filter` to detect an attack and respond. For example, to establish a defense against a DDOS/SYN flood attack, enable rule 135:1 (TCP SYN received), and configure the `rate_filter` to alert on excessive triggers of rule 135:1.

Rate-based attack prevention identifies abnormal traffic patterns and attempts to minimize the impact of that traffic on legitimate requests. You can identify excessive rule matches in traffic going to a particular destination IP address or addresses or coming from a particular source IP address or addresses. You can also respond to excessive matches for a particular rule across all detected traffic.

The following diagram shows an example where an attacker is attempting to access a host. Repeated attempts to find a password trigger a rule which has rate-based attack prevention configured. The rate-based settings change the rule attribute to Drop and Generate Events after rule matches occur five times in a 10-second span. The new rule attribute times out after 15 seconds.

After the timeout, note that packets are still dropped in the rate-based sampling period that follows. If the sampled rate is above the threshold in the current or previous sampling period, the new action continues. The new action reverts to Generate Events only after a sampling period completes where the sampled rate was below the threshold rate.


Seconds
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
Packet Matches Rule?
YYYYY Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Drop?
NNNNN Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y N N N
Alert?
YYYYY Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Rate-based sampling period
Rate-based sampling period
Rate-based sampling period
Rate-based timeout
Rate-based: Count = 5, Seconds = 10, New Action = Alert and Drop, Timeout = 15
372204

You can define multiple rate-based filters on the same rule as well as on different rules. In an intrusion policy with multiple rate-based filters defined, the first filter listed in the policy has the highest priority. When two rate-based filter actions conflict, the action of the first rate-based filter is carried out.

The configuration parameters you set for the `rate_filter` apply to all traffic throughout your deployment. However, the system maintains a separate counter for the number of matches within the sampling period for each unique connection your system monitors. The system also applies changes to an action on a per-connection basis.

Note Rate-based actions cannot enable disabled rules or drop traffic that matches disabled rules.

Rate Filter Parameters

rate_filter[]

Specifies an array of `rate_filter` information. Each `rate_filter` includes a set of fields that can alter a rule action if the traffic contains a rate-based attack.

Type: array (object)

Example:

```
{
    "rate_filter": {
        "data": [
            {
                "apply_to": "[10.1.2.100, 10.1.2.101]",
                "count": 5,
                "gid": 135,
                "new_action": "alert",
                "seconds": 1,
                "sid": 1,
                "timeout": 5,
                "track": "by_src"
            }
        ],
        "enabled": true,
        "type": "singleton"
    }
}
```

rate_filter[].gid

Specifies a generator ID (GID) which identifies the rule to match.

Type: integer

Valid range: `0` to `4,294,967,295 (max32)`

Default value: `1`

rate_filter[].sid

Specifies a signature ID (SID) which identifies the rule to match.

Type: integer

Valid range: `0` to `4,294,967,295 (max32)`

Default value: `1`

rate_filter[].track

Specifies a filter to match source or destination addresses.

Type: enum

Valid values:

- `by_src`: Filter only traffic that matches the rule specified by `rate_filter[].gid` and `rate_filter[].sid`, and where source address matches `rate_filter[].apply_to`.
- `by_dst`: Filter only traffic that matches the rule specified by `gid` and `sid`, and where destination address matches `rate_filter[].apply_to`.
- `by_rule`: Filter all traffic that matches the rule specified by `rate_filter[].gid` and `rate_filter[].sid`.

Default value: `by_src`

rate_filter[].count

Specifies the number of rule matches to allow in the sampling period (`rate_filter[].seconds`) before applying the alternative action (`rate_filter[].new_action`).

Type: integer

Valid range: `0` to `4,294,967,295 (max32)`

Default value: `1`

rate_filter[].seconds

Specifies the number of seconds in the sampling period to match traffic. `rate_filter[].seconds` represents the amount of time to elapse before resetting the internal counter of matches to zero.

Type: integer

Valid range: `0` to `4,294,967,295 (max32)`

Default value: `1`

rate_filter[].new_action

Specifies the action to take in response to matches in traffic that exceed the limitation specified by `rate_filter[].seconds` and `rate_filter[].count`.

Type: string

Valid values: One of the following strings: `alert`, `block`, `drop`, `log`, `pass`, `react`, `reject`, `rewrite`.

Default value: `alert`

rate_filter[].timeout

Specifies the number of seconds to perform the action specified by `rate_filter[].new_action` in response to matching traffic.

Type: integer

Valid range: `0` to `4,294,967,295 (max32)`

Default value: `0`

rate_filter[].apply_to

Specifies the list of network addresses to match against traffic source or destination address depending on the value of `rate_filter[].track`.

Type: string

Valid values: A valid IPv4 address, or an IPv4 address block in CIDR format.

Default value: None

Rate Filter Rules

The `rate_filter` does not have any associated rules.

You can confgure the `rate_filter` to respond to any intrusion rules. Enable the `rate_filter` for a rule to detect an attack and respond.

Rate Filter Intrusion Rule Options

The `rate_filter` does not have any intrusion rule options.