How to detect abnormal traffic using Pike

Applies to VoipNow 4.x and higher!

VoipNow comes packed with an application layer firewall at the SIP level called Pike. Pike is a module implemented in Kamailio that keeps track of all incoming requests, logging the source IP address for requests exceeding limits.

This module was implemented for the purpose of blocking IP addresses when limits are exceeded. It's better not to rely only on Kamailio to block such IP addresses.

Pike also reports abnormal traffic coming from different sources, allowing the system administrator to decide what measures to take using a script.

Step-by-step guide

Pike is disabled by default, but you can easily enable it by switching SIP_ANTIABUSE 1 in /etc/voipnow/local.conf and then restarting Kamailio.

```
# Disable/Enable SIP antiabuse (0/1)
SIP_ANTIABUSE 1
```

Pike has three different trees and each of them tries to detect signs of abnormal activity within a certain period of time.

1. Level 1 IP tree detects more than 300 auth requests per 10-second sampling unit.

   ```
   modparam("pike", "ip_tree", "l1_tree=>sampling_time_unit=10;reqs_density_per_unit=300;
   remove_latency=120")
   ```

2. Level 2 IP tree detects more than 5 failed auth requests per 30-second sampling unit.

   ```
   modparam("pike", "ip_tree", "l2_tree=>sampling_time_unit=30;reqs_density_per_unit=5;remove_latency=240")
   ```

3. Level 3 IP tree detects more than 30 failed auth requests per 10-minute sampling unit.

   ```
   modparam("pike", "ip_tree", "l3_tree=>sampling_time_unit=600;reqs_density_per_unit=30;
   remove_latency=1800")
   ```

4. Level 4 IP tree detects more than 20 failed auth requests per 5-minute sampling unit.

   ```
   modparam("pike", "ip_tree", "l4_tree=>sampling_time_unit=300;reqs_density_per_unit=20;
   remove_latency=1200")
   ```

Here's what each parameter means:

- **sampling_time_unit**: time period used for sampling to detect peaks; small values should be used.
- **reqs_density_per_unit**: the number of requests that should be allowed per sampling_time_unit before all incoming requests from that IP are blocked.
- **remove_latency*:*** for how long an IP address is stored since the last request sent from that IP address.

Whitelisting

In the database used for whitelisting IP addresses or IP classes, there is also a ser_address table. To whitelist an IP address, the following query should be used:

```
insert into ser_address values ('',1,'10.150.5.194','32',0,'');
```

The IP address 10.150.5.194 will be considered clean and all requests sent from that IP address will no longer be analyzed by Pike.

This is the query for whitelisting a /24 IP class:

```
insert into ser_address values ('',1,'10.150.5.0','24',0,'');
```

Logging

When one of the limits is exceeded, a new line will be added to /var/log/kamailio/abuse.log.

The syntax of the log file is the following:

```
<timestamp>: L<1|2|3> Pike block from <source_ip>:<source_port>
```

Related articles

- How to detect abnormal traffic using Pike
- Troubleshooting calls and debug steps
- How to prioritize VoIP traffic in the network
- How to debug Asterisk and Kamailio
- How to debug incoming calls