# routeflapper

# Packages

The various source and binary packages are available at http://www.five-ten-sg.com/routeflapper/packages/ The most recent documentation is available at http://www.five-ten-sg.com/routeflapper/

A Mercurial [http://www.selenic.com/mercurial/wiki/] source code repository for this project is available at http://hg.five-ten-sg.com/routeflapper/.

# Name

routeflapper -- detects suspicious routes

routeflapper

# Synopsis

`routeflapper [-c][-d n]`

# Description

**routeflapper** is a daemon that monitors BGP updates and SMTP connections to discover whether SMTP connections are coming from ip addresses whose best route is suspicious.

The routeflapper.conf(5) file specifies the syslog files to be monitored, and the regular expressions (regex(7)) to be applied to new lines in those files.

The discussion has focused on syslog files, but any ascii text file can be used, so long as some other process appends lines to that file, and those lines containing bgp updates can be matched with some regular expression.

Considering syslog files in particular, these are normally rotated via logrotate. **routeflapper** properly detects and handles this case by closing the old file, and reopening the newly created file.

# Options

-c
: Load the configuration file, print a cannonical form of the configuration on stdout, and exit.

-d *n*
: Set the debug level to *n*.

# Usage

**routeflapper** -d 2

# Configuration

The configuration file is documented in routeflapper.conf(5). Any change to the config file will cause it to be reloaded within three minutes.

# Introduction

Consider the hypothetical case of a spammer who is connected via a provider that does not filter BGP routing announcements. The spammer then has some options to announce ip address space to be used for sending spam. Note that we only consider cases where the spammer simply wants to anonymously use some ip address space. This is very different from the case where the attacker wants to use some specific address space belonging to another organization in order to impersonate some service provided by that other organization.

They can announce a more specific route, for example a /24, inside a larger block. For example, consider 169.232.0.0/16. If the spammer pokes around, they can probably find an unused /24 in there. So they announce

169.232.240.0/24 and then send spam from that block. There are two problems with this scheme. First, the announcement of such a smaller block may be filtered out by many BGP routers, reducing their reachability to their spam targets. Second, they may have made a mistake, and that /24 is actually in use by some UCLA service that will notice their hijack.

They can announce a less specific route, for example a /16, covering some individual smaller blocks. For example, they could announce 52.129.0.0/16. The spammer could then avoid the four existing announcements inside that block, and instead spam from 52.129.128.0/17. That gives them 32K ip addresses to work with. The advantage here is that their announcement of a large block won't be filtered out by as many (if any) BGP routers, giving them better reachability to their spam targets. And they know they won't interfere with any existing use of that address space, since there was no previous BGP announcement of that /17 or any subset of it.

Or they can simply announce a prefix that is not assigned to anyone. For example, they could simply start announcing 185.10.0.0/16. This has many of the same advantages as the previous scheme, but some BGP routers may be configured to drop such bogon announcements.

In each of these cases, the spammer can use BGP to announce some address space, then send spam from those addresses, and then withdraw the route annoucement. This would make it difficult for the recipient of such spam to determine who actually sent it.

In a paper from 2006 published at http://www-static.cc.gatech.edu/~feamster/publications/p396-ramachandran.pdf [http://www-static.cc.gatech.edu/~feamster/publications/p396-ramachandran.pdf], Ramachandran and Feamster claim evidence for the statement that spammers are using such short-lived bogus BGP route announcements to send spam from hijacked parts of the IPv4 address space.

The question is, are spammers actually doing this today, or is this just a hypothetical spam tactic that they could use in the future? To help answer that question, this package monitors BGP annoucements, classifies some of them as suspicious, and logs instances of SMTP connections from suspicious prefixes.

We track the history of the AS adjacency graph, by computing the union of all AS adjacent pairs over all the announced prefixes. For example, 137.169.0.0/16 is currently announced here with an AS path of '22298 19080 3549 6517 14981', so we add (22298,19080) (19080,3549) (3549,6517) and (6517,14981) as valid adjacent AS pairs.

We track the history of the origin AS for each announced prefix. Both the origin AS and AS adjacency pairs are tracked over a timescale of 100 hours, with an exponential decay half-life of 100 hours.

A prefix announcement is suspicious if the origin AS is not in the historical AS set for that prefix at least 20% of the time, or if the AS path contains any adjacent AS pair that is not in the historical AS adjacency graph at least 40% of the time.

PHAS [http://phas.netsec.colostate.edu/] is another system that attempts to detect address space hijacking, but it is not correlated with SMTP connections or spam attempts.

IAR [http://cs.unm.edu/~karlinjf/IAR/index.php] is another system that attempts to detect address space hijacking, but it is not correlated with SMTP connections or spam attempts. IAR uses methods detailed in PGBGP [http://www.cs.unm.edu/~treport/tr/06-06/pgbgp3.pdf] to detect suspicious routes. One problem with PGBGP as applied to our hypothetical spammer problem, is that PGBGP is primarily looking for hijacks where the attacker actually wants some specific ip address space, either for a denial of service, or to impersonate the actual owner. Our hypothetical spammer does not care about that - they only care about sending spam anonymously. In particular, PGBGP ignores super-prefix hijacks, but it seems likely that that is the preferred method for our hypothetical spammer. However, the PGBGP paper does provide useful data on the required timescale to avoid most of the normal AS origin changes.

# TODO

None.

# Copyright



# Version

1.0.1

# Name

routeflapper.conf -- configuration file for routeflapper

routeflapper.conf

# Synopsis

`routeflapper.conf`

# Description

The **routeflapper.conf** configuration file is specified by this partial bnf description. The entire config file is case sensitive. All the keywords are lower case.

```
CONFIG   := {FILE}+
FILE     := "file" FILENAME "{" PATTERN+ "};"
PATTERN  := PATH | ANNOUNCE | WITHDRAW | IP
PATH     := "path" REGEX "{" INDEXPATH      '}' ";"
ANNOUNCE  := "path" REGEX "{" INDEXVAL INDEXLEN '}' ";"
WITHDRAW  := "path" REGEX "{" INDEXVAL INDEXLEN '}' ";"
IP       := "path" REGEX "{" INDEXIP        '}' ";"
INDEXPATH := "index_path"  REGEX-INTEGER-VALUE ";"
INDEXVAL  := "index_value"  REGEX-INTEGER-VALUE ";"
INDEXLEN  := "index_length" REGEX-INTEGER-VALUE ";"
INDEXIP   := "index_ip"    REGEX-INTEGER-VALUE ";"
```

# Sample

```
file "/var/log/bgp" {
   path " rcvd UPDATE w.* path (([0-9]| )*[0-9])" {
      index_path 1;
   };
   announce " rcvd (([0-9]|\.)*)/([0-9]*)$" {
      index_value  1;
      index_length 3;
   };
   withdraw " rcvd UPDATE about (([0-9]|\.)*)/([0-9]*) -- withdrawn" {
      index_value  1;
      index_length 3;
   };
};

file "/var/log/maillog" {
   ip "NOQUEUE: connect from.* \[(.*)\]" {
      index_ip 1;
   };
};
```

# Version

1.0.1