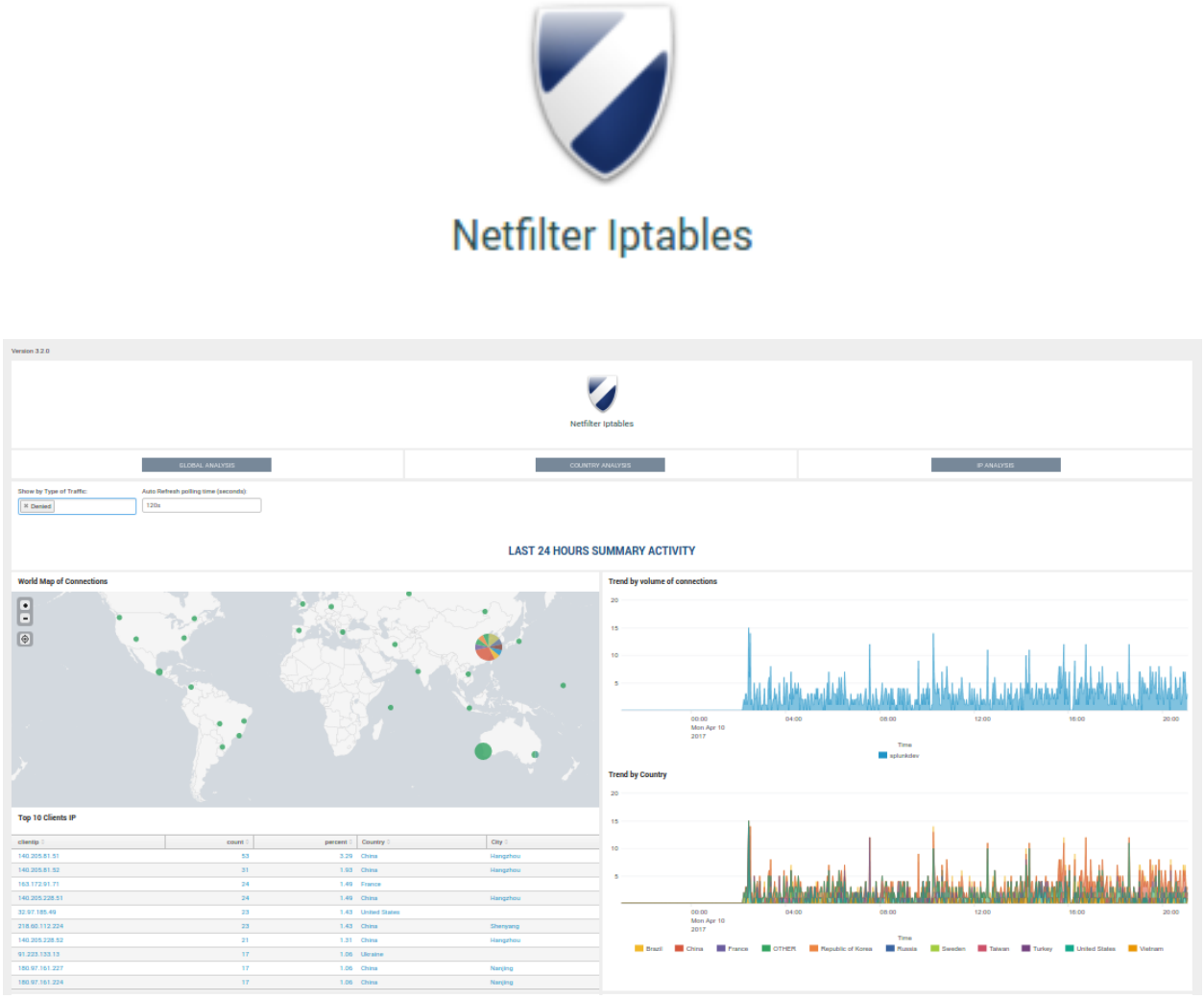
Netfilter Iptables for Splunk Documentation

Release 0

Guilhem Marchand

Oct 06, 2017

Contents


Netfilter Iptables
GLOBAL ANALYSIS
COUNTRY ANALYSIS
IP ANALYSIS
LAST 24 HOURS SUMMARY ACTIVITY
World Map of Connections
Trend by volume of connections
Trend by Country
Top 10 Clients IP

Contents:

CHAPTER 1

Overview:

About the Netfilter Iptables application for Splunk

- Author: Guilhem Marchand
- First release was published on starting 2014
- Purposes:

The Netfilter Iptables application for Splunk manage Linux iptables based firewall logs (iptables, ufw...) generated to provide easy and accessible information about the firewall activity of your servers.

It is a very simple and lightweight application.

Splunk versions

The application is compatible with any version of Splunk 6.4 and later

Index time operations

The Netfilter application relies on the installation of the "Linux Netfilter (iptables)" technology add-on:

- https://splunkbase.splunk.com/app/3089/

Index creation

The application **does not** create any index at installation time.

Summarization implementation

The application **does not** currently implement any piece of summarization, accelerated reports or data models.

Release notes

Requirements

- Splunk 6.x and later Only

What has been fixed by release

V3.2.1:

- Minors improvements in app home page with bootstrap window providing shortcut access links to raw data

V3.2.0:

- Global review of the application for last Splunk version compatibility
- Read the docs documentation
- Removal of useless items

V3.1.0:

- Totally rewritten version of the App in the Splunk 6.x fashion !
- All views are now completely designed in Simple XML and Django / Javascript
- Both Accepted and Denied Traffic can be analysed with no changes within views
- Application setup page to allow main settings to be customized within Splunk UI
- Migrating from Google Maps views to Splunk Map vizualization
- Migrating to Splunk iplocation command and geoip db
- The App does not requires anymore any third party Apps to be fully usable, Splunk is the only requirement as for now

V2.04:

- Corrected Event Search interface

V2.03:

- Corrected span definition error for timerange and realtime views for peak load identification
- Deleted redundant information in Activity Summary sections

V2.02:

- Code cleaning
- Views improvement
- Hide info message when subsearches running over realtime

V2.01:

- Added Dashboard view about Index Activity (System view)

V2.0:

Fully rewritten version of this apps, release notes:

- Added Networking Services translation with associated Charts and Stats
- New centralized home page
- Realtime stats in home page, top offsenser stats...
- New Realtime and Timerange Charts and stats view
- New event search interface
- Various other corrections

V2.0:

Fully rewritten version of this apps, release notes:

- Added Networking Services translation with associated Charts and Stats
- New centralized home page
- Realtime stats in home page, top offsenser stats...
- New Realtime and Timerange Charts and stats view
- New event search interface
- Various other corrections

V2.01:

- Added Dashboard view about Index Activity (System view)

V2.0:

Fully rewritten version of this apps, release notes:

- Added Networking Services translation with associated Charts and Stats
- New centralized home page
- Realtime stats in home page, top offsenser stats...
- New Realtime and Timerange Charts and stats view
- New event search interface
- Various other corrections

Version 2.01:

Fully rewritten version of this apps, release notes:

- Added Networking Services translation with associated Charts and Stats
- New centralized home page
- Realtime stats in home page, top offsenser stats...
- New Realtime and Timerange Charts and stats view
- New event search interface
- Various other corrections

Version 1.0

- Initial version

Known Issues

Major or minor bug, enhancement requests will always be linked to an opened issue on the github project issue page:

https://github.com/guilhemmarchand/iptables-for-splunk/issues

Please note that once issues are considered as solved, by a new release or support exchanges, the issue will be closed. (but closed issues can still be reviewed)

There are no issues currently referenced

Support

Community support

This application and all of its components are provided under the Creative Commons BY 3.0 licence, please remember that it comes with no warranty even if i intend to do my best in helping any people interested in using the App.

DISCLAIMER:

Community support comes in "best effort" with absolutely no warranties.

Github

https://github.com/guilhemmarchand/iptables-for-splunk

Use Github to open an issue for errors and bugs to be reported, or to ask for enhancements requests.

You can even provide your own improvements by submitting a pull request.

Splunk Answers

Splunk has a strong community of active users and Splunk Answers is an important source of information.

Access previous messages of users or open your own discussion:

https://splunkbase.splunk.com/app/1353

http://answers.splunk.com/answers/app/1353

Issues and enhancement requests

For any bug reporting, or enhancement request about the Netfilter Iptables application, you can:

- Open a question on Splunk Answers related to the app: http://answers.splunk.com/answers/app/1353.html
- Open an issue on the Git project home page: https://github.com/guilhemmarchand/iptables-for-splunk/issues
- Get in touch by mail: guilhem.marchand@gmail.com

Pre-requisites

Linux Netfilter (iptables) technology add-on

Since the release 3.2.0, the application relies on the very good quality add-on:

- https://splunkbase.splunk.com/app/3089/

As such, please refer to the application documentation to every step related to the indexing configuration for Splunk:

- https://github.com/doksu/TA_netfilter/wiki

Basically, the application globally use the following default query to retrieve the Iptables events::

```
(index=* eventtype=linux_netfilter)
```

However, this is very easily customizable via the settings page of the application. (available at first startup or in the "Help & Settings" menu.

Download

Official Splunk certified release

The official and Splunk certified release can be downloaded from Splunk Base: https://splunkbase.splunk.com/app/3089

Github releases

The application is hosted on a Github project, you can freely download the application from the Github project page: https://github.com/guilhemmarchand/iptables-for-splunk

Downloading and installing from Github:

You can download and install the application directly from git using the git command:

```
git clone https://github.com/guilhemmarchand/iptables-for-splunk.git
mv iptables-for-splunk iptables
```

Deployment Matrix

What goes where ?

Very simple, the application lives only on search head:

Splunk Instance (role)	Install ?
Standalone server	X
Search head (single instance or clustered)	X
Indexer (single instance or clustered)	
Master node	
Deployment servers	
Heavy Forwarder	
Universal Forwarder	

CHAPTER 2

Installation and configuration:

Deployment and configuration

The deployment of the application has to respect Splunk good practices depending on the topology of your deployment:

Deployment

The deployment of the application is very simple and relies on your Splunk installation:

Standalone server

- Download the application as a tgz archive from Splunk base: https://splunkbase.splunk.com/app/1353
- Use the application manager to install the application, or uncompress using CLI:

example::

```
cd /opt/splunk/etc/apps/
tar -xvf <path to archive>
```

Distributed deployment

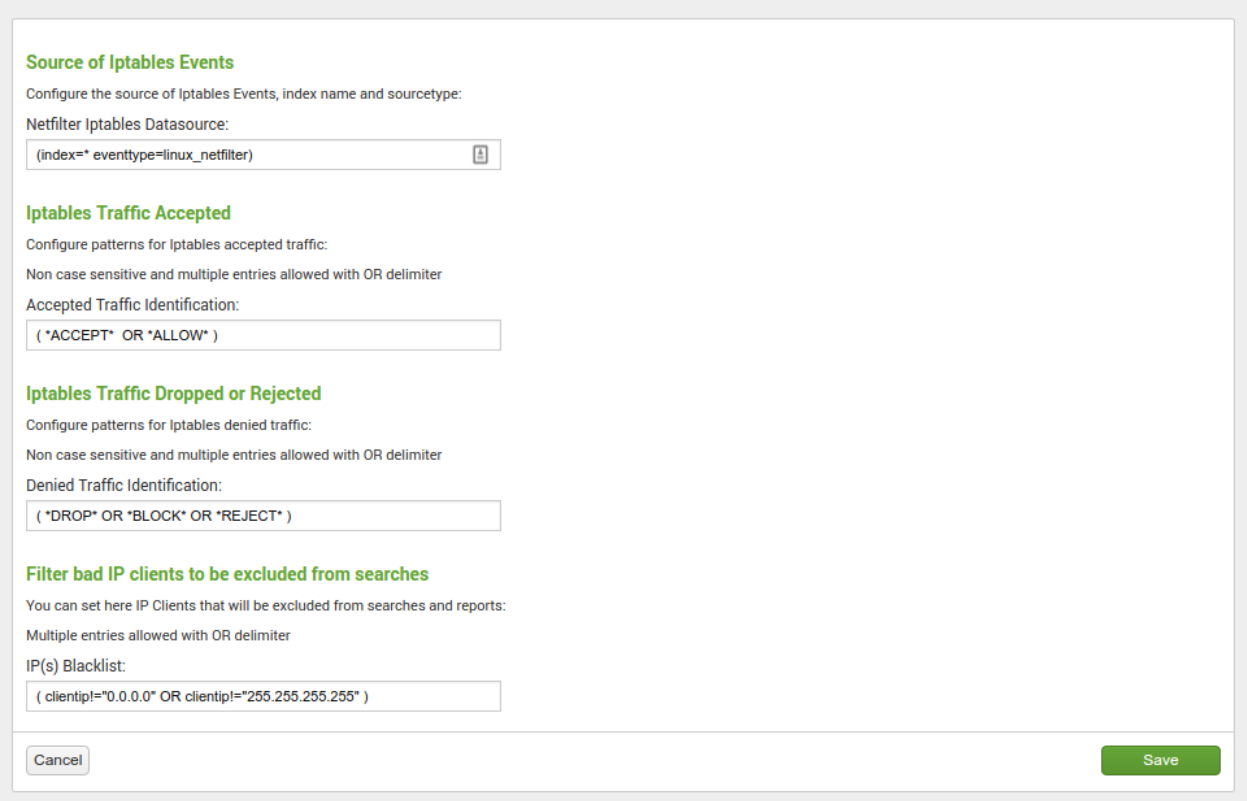
- If you are not using Search Head Clustering (SHC), deploy the application in every search head needed (same procedure than for standalone servers)
- If you are using an SHC, uncompress the content of the tgz archive in your SHC deployer node, and apply the SHC cluster bundle, refer to:

http://docs.splunk.com/Documentation/Splunk/latest/DistSearch/PropagateSHCconfigurationchanges

Configuration

Application configuration

After the installation and when you open the application, the setup screen of the application will automatically be displayed:



This screen allows you to:

- Configure the main macro that is used to retrieve your firewall events
- Configure the patterns that must be used to identify the accepted connections
- Configure the patterns that must be used to identity refused and dropped connections
- Blacklist specific IP addresses or host to avoid pollution

Recommendations:

- Once you have planned the name of the index(es) for your deployment, it is recommended for best performance to configure them in the screen above

Getting data in

pre-requisites:

You need to install the TA for iptables:

https://splunkbase.splunk.com/app/3089

Please review the documentation of the THA:

https://github.com/doksu/TA_netfilter/wiki

Once you have deployed the TA:

The TA documentation provides sample configuration for iptables, in addition of this documentation, you will find above some configuration examples.

Ubuntu based servers

1. Install ufw

Run:

```
sudo apt-get install ufw
```

2. Configure ufw according to your needs, and activate logging

https://help.ubuntu.com/community/UFW

Activating the logging will enable logging dropped or refused packets:

```
sudo ufw logging on
```

The activity of ufw is logged on the server in:

```
/var/log/ufw.log
```

3. Make sure the Splunk instance can access this file with read permissions (you can use extended acl) and create a very basic and simple file monitor

Example: (customize the name of the index according to your deloyment):

inputs.conf:

```
[monitor:/var/log/ufw.log]
index = security_firewall_os
sourcetype = syslog
```

This inputs.conf can be configured in the TA local directory, or wherever you like.

Centos "like" servers

1. Configure iptables and activate logging

CentOS doc: https://wiki.centos.org/HowTos/Network/IPTables

example configuration: /etc/sysconfig/iptables:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:LOGGING - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -j LOGGING
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A LOGGING -m limit --limit 2/min -j LOG --log-prefix "iptables: DROP: " --log-level 7
```

```
-A LOGGING -j DROP
COMMIT
```

2. Configure rsyslog to log iptables events in a separated log file

example configuration: /etc/rsyslog.d/iptables.conf:

```
:msg, contains, "iptables:" -/var/log/iptables.log
& ~
```

Restart rsyslog:

```
service rsyslog restart
```

3. Configure logrotate.d

example configuration: /etc/logrotate.d/iptables:

```
/var/log/iptables.log
{
        rotate 7
        daily
        missingok
        notifempty
        delaycompress
        compress
        create 0664 root root
        postrotate
                invoke-rc.d rsyslog rotate > /dev/null
        endscript
}
```

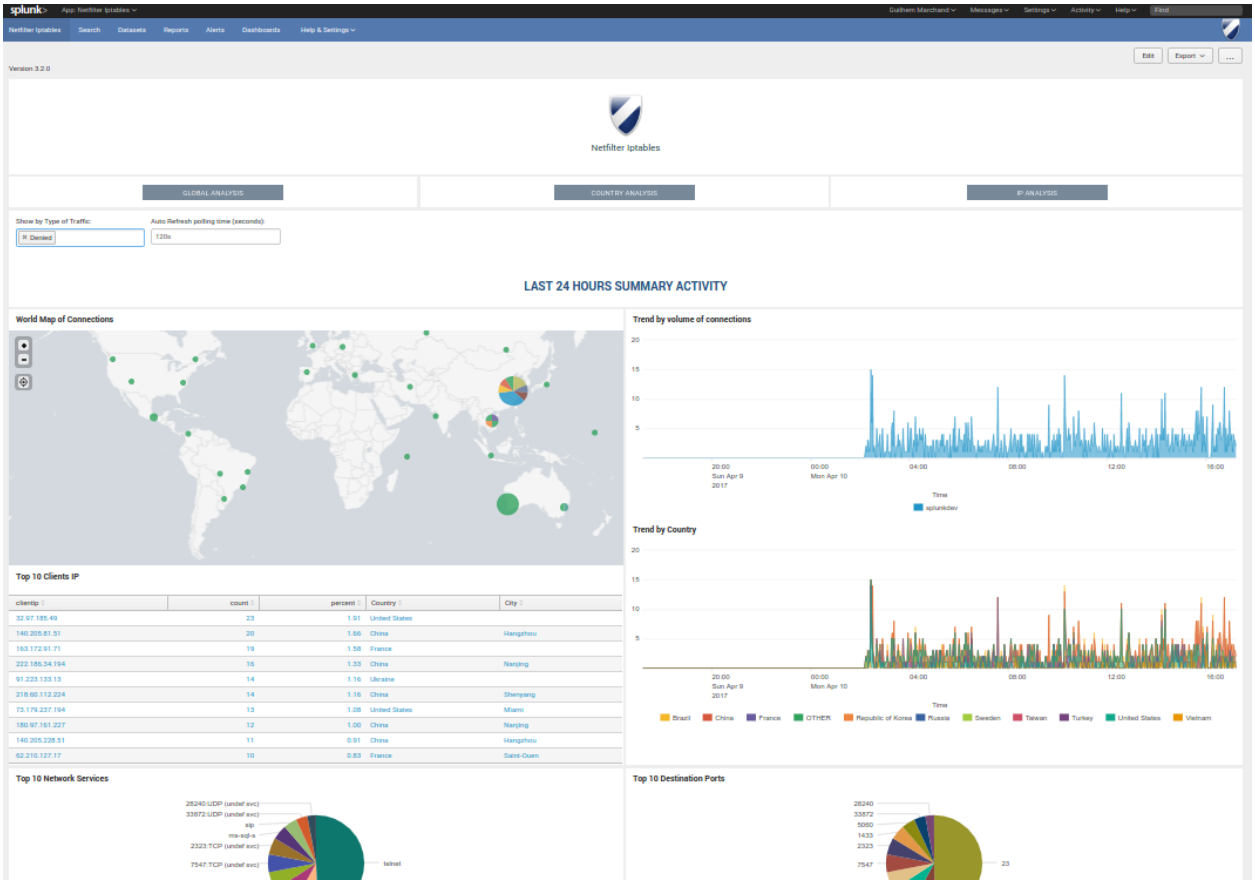
4. Make sure the Splunk instance can access this file with read permissions (you can use extended acl) and create a very basic and simple file monitor

Example: (customize the name of the index according to your deloyment):

inputs.conf:

```
[monitor:/var/log/iptables.log]
index = security_firewall_os
sourcetype = syslog
```

Ensure Splunk is restarted after the deployment of this inputs.conf, et voila!



Upgrade

Upgrading the application is entirely similar to the initial installation.

Please refer to the installation procedure: *Deployment*

Good practices for upgrade resiliency:

- do not modify any file out of the "local" directory (create it yourself if you need any modification)

Upgrade from version 3.1.x and previous

Since the release 3.2.0, the application relies on the "Linux Netfilter (iptables)".

As such, if you have previously indexes data, the following steps have to be done:

- Deploy and configure the TA for iptables
- Replace and/or update the existing file monitor
- Deploy the Netfilter application new release
- Use the setup page of the application to search for both the old index/sourcetype and the standard index/eventtype

CHAPTER 3

Guidance:

Userguide

Key concepts

The Iptables application for Splunk is a simple frontend application for Netfilter Iptables based firewall.

Indexing data in Splunk relies on the TA for iptables available in Splunk base: https://splunkbase.splunk.com/app/3089

This is a pretty simple application that comes with a few reports and dashboards, as documented below.

Data Types (sourcetype)

The application uses the data sourcetype created by the TA for iptables:

```
sourcetype=linux:netfilter
```

Those events can be retrieved using eventtypes, which is what is done by the application:

```
eventtype=linux_netfilter
```

By default, the application will search for data using the following Splunk search:

```
index=* eventtype="linux_netfilter"
```

Notes: This is configurable in the application setting page, it is recommended to customize the name of the index(es) where you will decide to ingest your data.

To do this, all the searches implemented in the application will use a root macro (potentially followed by other search criterias):

```
`iptables_datasource`
```

The iptables events are basically syslog type events, where the TA for iptables will rewrite the sourcetype to its definitive value.

Lookups and KV Store

The application does not currently implement any KVstore based lookup.

It does however contains a file based lookup table:

```
network_services
```

Which is based on the lookup csv file:

```
lookups/network_services.csv
```

This lookup table is being used to provide an auto lookup mapping with the network destination port and protocol to generate a more human friendly service name.

This mapping operates in the "props.conf" configuration file:

```
# Translation of port number to service name
# Lookups
lookup_network_services = network_services DPT, PROTO OUTPUTNEW Service As Service
```

The lookup file content is based on standard network services. (/etc/services in any Linux OS)

Main configuration files

props.conf

The props.conf file implements search time extractions over the following "source" field:

```
[source::.../(iptables|ufw).log]
```

This a simple stanza using regular expression, and this will match any location for the following files:

example for standard log file locations:

- /var/log/iptables.log
- /var/log/ufw.log

If for some reasons the "source" value in your deployment differs from this standard, you will need to customized your configuration:

- create a copy of default/props.conf to local/props.conf
- customize the stanza definition to match your source
- deploy and apply your new configuration

transforms.conf

The transforms.conf provided by default configures the file based lookup:

```
# Translation of port number to service name (IANA source)

[network_services]
filename = network_services.csv
```

savedsearches.conf

The savedsearches.conf file contains a few pre-built reports used in the application's dashboards:

- Iptables Activity Summary
- Iptables Top Offenser
- Iptables Top 10 denied client by count
- Iptables Top 10 denied client by Country
- Iptables Top 10 denied Networking Services
- Iptables Top 10 denied Destination Ports
- Iptables Top 10 Reporting Hosts

None of these reports implement by default any acceleration or scheduling.

macros.conf

The macros.conf configuration contains several macros globally used in the dashboards of the application.

The main macro defines the root search for retrieving iptables events:

```
# Datasource of Iptables Events

[iptables_datasource]
definition = (index=* eventtype=linux_netfilter)
iseval = 0
```

Notes: This macro can (and should) be customized using the application setup page to match your index(es)

The next interest macros will define the recognition of accepted and denied / dropped packets:

```
# Accepted Traffic - Patterns used to filter allowed traffic

[traffic_accepted]
definition = ( *ACCEPT*  OR *ALLOW* )
iseval = 0

[traffic_denied]
definition = ( *DROP* OR *BLOCK* OR *REJECT* )
iseval = 0
```

Notes: These macros can be customized as well with the setup page

The following macro is used to filter some unwanted pollution:

```
# Some filter for traffic pollution

[filter_badclients]
```

```
definition = ( clientip!="0.0.0.0" OR clientip!="255.255.255.255" )
iseval = 0
```

Finally, the following macro "iptables_span" is used to improve Splunk charts