Logging in the nftables age

Éric Leblond
Stamus Networks
July 5, 2014
Eric Leblond a.k.a Regit

- French
- Network security expert
- Free Software enthousiast
- NuFW project creator (Now ufwi), EdenWall co-founder
- Netfilter developer:
  - Maintainer of ulogd2: Netfilter logging daemon
  - Misc contributions:
    - NFQUEUE library and associates
    - Port of some features iptables to nftables
- Currently:
  - co-founder of Stamus Networks, a company providing Suricata based network probe appliances.
  - Suricata IDS/IPS funded developer
A history of Netfilter logging

Nftables logging

Latest evolution of ulogd2

The future of ulogd2

Conclusion
Packet logging

Goal
- Keep trace of an activity
- Create a message when a rule match

Syntax
```
iptables -A INPUT -p tcp --dport 25 --syn \ 
-j LOG --log-prefix "SMTP access "
```
Packet logging

Syslog logging

- Flat packet logging
- One line per packet
- Use printk kernel facility

Not sexy

```plaintext
INPUT DROP IN=eth0 OUT= MAC=00:1a:92:05:ee:68:00:b0:8e:83:3b:f0:08:00 \ 
  SRC=62.212.121.211 DST=91.121.73.151 LEN=60 TOS=0x00 PREC=0x00 \ 
  TTL=58 ID=35342 DF PROTO=TCP SPT=59261 DPT=113 WINDOW=5440 RES=0x00 SYN URGP=0
```

```plaintext
IN IN=eth0 OUT= MAC=d4:be:d9:69:d1:51:00:11:95:63:c7:5e:08:00 \ 
  SRC=31.13.80.7 DST=192.168.11.3 LEN=40 TOS=0x00 PREC=0x00 TTL=244 \ 
  ID=37732 DF PROTO=TCP SPT=443 DPT=48875 WINDOW=0 RES=0x00 ACK RST URGP=0
```

```plaintext
IN IN=eth0 OUT= MAC=d4:be:d9:69:d1:51:00:11:95:63:c7:5e:08:00 \ 
  SRC=31.13.80.23 DST=192.168.11.3 LEN=86 TOS=0x00 PREC=0x00 TTL=243 \ 
  ID=33964 DF PROTO=TCP SPT=80 DPT=49617 WINDOW=0 RES=0x00 ACK RST URGP=0
```
Socket base messaging

- Netlink based communication
- Different groups
- Batching system
- IPv4 only

Syntax

```bash
iptables -A INPUT -p tcp --dport 25 --syn \ 
-j ULOG --ulog-prefix "SMTP access" \ 
--ulog-nlgroup 2 \ 
--ulog-qthreshold 10
```
A logging daemon

- Listen to event
- Store event in various formats
  - Flat file
  - Databases

Ulogd outputs

- LOGEMU
- OPRINT
- MySQL
- Postgresql
- sqlite3
- pcap
- Using SQL backend
- Providing a dashboard
- Nulog released 14 Apr 2000
2.6.14: the nfnetlink revolution

Nfnetlink
- First major evolution of Netfilter (Linux 2.6.14, 2005)
- Netfilter dedicated configuration and message passing mechanism

New interactions
- NFLOG: enhanced logging system
- NFQUEUE: improved userspace decision system
- NFCT: get information and update connection tracking entries

Based on Netlink
- datagram-oriented messaging system
- passing messages from kernel to user-space and vice-versa
Interaction via libnetfilter_log

- Library to get messages from
- Same kernel parameters as ULOG

Syntax

```
iptables -A INPUT -p tcp --dport 25 --syn \ 
  -j NFLOG --nflog-prefix "SMTP access" \ 
  --nflog-group 2 \ 
  --nflog-threshold 10
```
Interaction via libnetfilter_conntrack
- Dump connection tracking info
- Update/Delete connection tracking entries
- Event mode

Used by conntrack-tools
- conntrackd
  - connection tracking replication daemon
  - provide high availability
  - developed by Pablo Neira Ayuso
- conntrack: command line tool to update and query connection tracking
Ulold reloaded

- Interact with the post 2.6.14 libraries
- First release on 01 Feb 2006
- Multiple output and input through the use of stacks

Stack example

```plaintext
stack=log2:NFLOG,mark1:MARK,base1:BASE,ifi1:IFINDEX,ip2bin1:IP2BIN,\
   mac2str1:HWHDR,mysql1:MYSQL
stack=log2:NFLOG,base1:BASE,ifi1:IFINDEX,ip2str1:IP2STR,\
   mac2str1:HWHDR,pgsql1:PGSQL
```
Nothing really new

- One ulogd2 can handle multiple logging input
- Multiple output is also supported

But improved databases

- Magical schema discovery
- Better schema
- Insertion via SQL procedure
  - It is possible to create custom logging in SQL
  - No need to know C
ulogd2: connection logging

Interests
- Log volume of exchange data
- Log NAT transformation

Ulogd2 support
- File and database output
  
  stack=ct2:NFCT,ip2str1:IP2STR,pgsql2:PGSQL

More info https://home.regit.org/2014/02/logging-connection-tracking-event-with-ulogd/
nfacct

- Efficient accounting system
- Appeared in 2012

Usage

nfacct add https
nfacct add http
iptables -I INPUT -p tcp --sport 80 -m nfacct --nfacct name http
ip6tables -I INPUT -p tcp --sport 80 -m nfacct --nfacct name http
iptables -I INPUT -p tcp --sport 443 -m nfacct --nfacct name https
ip6tables -I INPUT -p tcp --sport 443 -m nfacct --nfacct name https
nfacct list
Dump nfacct counter at regular interval

Realize storage

- XML
- Postgresql
- Graphite

Ulogd stacks

stack=acct1:NFACCT,xml1:XML
stack=acct1:NFACCT,pgsql4:PGSQL
Graphite

- Scalable Realtime Graphing
- Based on rrdtools
- Allow to combine data
- http://graphite.wikidot.com/start

Ulogd2 configuration

```
stack=acct1:NFACCT,\
    graphite1:GRAPHITE

[acct1]
pollinterval = 2

[graphite1]
host="127.0.0.1"
port="2003"
```
Proposed removal

- Pablo Neira has sent patches to remove ULOG target
- Nearly 9 years after NFLOG introduction
- Ulogd will be flagged End Of Life.

Two targets remaining

- LOG: logging possible without logging daemon
- NFLOG: require a running ulogd2 or similar
1. A history of Netfilter logging

2. Nftables logging

3. Latest evolution of ulogd2

4. The future of ulogd2

5. Conclusion
Packet logging

Features

- Two log mechanisms
  - Syslog
  - Via nfnetlink
- One single keyword: `log`

Syntax

```
nft add rule filter input tcp dport 22 \n  ct state new \n  log prefix "SSH" group 2
```
2in1 syntax

Use nftable syntax improvement
- Bytecode allow flexibility in rules
- Use multiple actions in one rule

log and accept rule

```nft add rule filter input tcp dport 22 \  
  ct state new log prefix \"SSH for ever\" \  
  accept```
System configuration

Global configuration

- First module loaded get the log
- **Use** `/proc/sys` **to setup logging**
- Set value by choosing from loaded modules

Configuration method

```bash
# cat /proc/net/nf_log
0 NONE (nfnetlink_log)
1 NONE (nfnetlink_log)
2 nfnetlink_log (nfnetlink_log, ipt_LOG)
...
# echo "nfnetlink_log" >/proc/sys/net/nf_log/2
```
Natural selection

- If `group` keyword is used, logging is done nfnetlink.
- If `level` keyword is used, logging is done via syslog.

Coming soon

- Should be available in Linux 3.17
Nftables notification

Event mode
- Listen to netlink socket
- Wait for update events

Syntax

```
# nft monitor
add table ip test
add chain ip test example
add rule ip test example tcp dport ssh counter packets 0 bytes 0
```
1. A history of Netfilter logging
2. Nftables logging
3. Latest evolution of ulogd2
4. The future of ulogd2
5. Conclusion
DB improvements

Backlog mode

- Allocate memory
- In case of db problem, backlog store waiting requests
DB improvements

Backlog mode

- Allocate memory
- In case of db problem, backlog store waiting requests

Ring mode

- Start a thread dedicated to insertion task
- Store a given amount of requests
- Event treatment time is not dependant of output module
- Avoid kernel side buffer overrun
### JSON output

#### JSON format
- Formatted message
- Schema less
- Easy to use in code and tools
- Integration with Splunk or Elasticsearch

#### JSON plugin
- Use ulogd key, value system
- Translation to text of key is enough
- Usable for all input plugins
Ulogd + Kibana

Netfilter Logs

Timeline

Netfilter Source IPs

Destination Port

Drop and Accept

Trends

Blocked Sources

Netfilter Packets

IPv4 & IPv6

Eric Leblond (Stamus Networks)

Logging in the nftables age

July 5, 2014

29 / 37
Perfect logging configuration (1/2)

Objectives

- Log blocked packets
- Log accepted packets
- Store everything and distinguish decision in JSON

Method

- Use two netlink groups
  - One for accepted packets
  - One for dropped packets
- Setup uLOGD2 to separate logging
  - Use numeric_label option of NFLOG plugin
  - Use boolean_label option of JSON plugin
Perfect logging configuration (2/2)

```
stack=log2:NFLOG,base1:BASE,ifl1:IFINDEX,ip2str1:IP2STR,\
  mac2str1:HWHDR,json1:JSON
stack=log3:NFLOG,base1:BASE,ifl1:IFINDEX,ip2str1:IP2STR,\
  mac2str1:HWHDR,json1:JSON

[log2]
group=1
numeric_label=1

[log3]
group=2
numeric_label=0

[json1]
sync=1
device="My awesome FW"
boolean_label=1
```
Am I missing something?

Why log2 and log3 input? Where is log1?
group 0

- Group 0 is dedicated to system log
- Using module activated via /proc

System log

- Logging on invalid packets following connection tracking
- Dropped if packet in invalid state are dropped

Extended configuration

```
stack=log1: NFLOG, base1: BASE, ifil: IFINDEX, ip2str1: IP2STR, \mac2str1: HWHDR, json1: JSON

[log1]
group=0
numeric_label=0
```
1. A history of Netfilter logging
2. Nftables logging
3. Latest evolution of ulogd2
4. The future of ulogd2
5. Conclusion
Future of ulogd2

Nftables notification
- New input plugin
- Store all information

Other improvements
- Ipfix support
- Multithreading
1. A history of Netfilter logging
2. Nftables logging
3. Latest evolution of ulogd2
4. The future of ulogd2
5. Conclusion
### Conclusion

Nftables brings complete logging
- Packets logging
- Connection tracking logging
- Ruleset modifications logging

### More information
- **My blog**: [https://home.regit.org/](https://home.regit.org/)
- **Stamus Networks**: [https://www.stamus-networks.com/](https://www.stamus-networks.com/)