Nftables and IPS

Éric Leblond

Stamus Networks

July 8, 2014

STAMUS NETWORKS

1 Introduction

2 Iptables

3 Nftables

4 Conclusion

STAMUS NETWORKS

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

STAMUS NETWORKS

IPS implementation

Network Intrusion Detection System

- Analyse traffic to detect event and anomaly
- Detect known alert
- Generate events after protocol analysis

Intrusion Prevention System

- A NIDS blocking attacks before they reach targets
- Made with Netfilter using NFQUEUE target
  - NFQUEUE must send all needed packets

STAMUS NETWORKS

Constraints of IPS implementation

Constraints

1. Get all packets for a studied flow
2. Get only accepted packets
   - Dropped packets are not a threat
   - Treating them will result in heavy ressource usage

The challenge

- Get an IPS implementation without impact on ruleset management solution
- Changing firewall rules management software should be possible
  - Without modification of IPS ruleset
  - Without change on ruleset management

STAMUS NETWORKS

1 Introduction

2 Iptables

3 Nftables

4 Conclusion

STAMUS NETWORKS

How to fail miserably

First try

```
iptables -A FORWARD -j NFQUEUE
iptables -A FORWARD \
         -m conntrack --ctstate ESTABLISHED \
         -j ACCEPT
# your firewall rules here
```

NFQUEUE is a terminal target

- NFQUEUE rule send packet to IDS
- IDS decide if packet is accepted and dropped
- packet leave current primary chain and go for next one

STAMUS NETWORKS

Simple iptables implementation

Using user chain

- Transform ACCEPT decision in a function

Implementation

```
iptables -N ips-accept
iptables -A ips-accept -j NFQUEUE --queue-balance 1-4

iptables -A FORWARD \
         -m conntrack --ctstate ESTABLISHED \
         -j ips-accept
# your firewall rules here
iptables -A FORWARD \
         -m conntrack --ctstate NEW \
         -p tcp --dport 22 \
         -j ips-accept
```

Iptables implementation using mangle

Use a chain different from FORWARD

- Let's find a unused table to dedicate to IPS
- We can use PREROUTING mangle

Implementation

```
iptables -A PREROUTING -t mangle -j NFQUEUE
iptables -A FORWARD \
         -m conntrack --ctstate ESTABLISHED
         -j ACCEPT
# your firewall rules here
```

Contra

- We don't see NAT transformation in IPS
- Mangle can be used by other software

Complex integration: repeat mode (1/2)

NFQUEUE hardcore

- Alternate decision
  - NF_REPEAT : send the packet back to the start of the table
  - NF_QUEUE : send the packet to another queue (chain software using NFQUEUE)
- Packet marking: userspace can take decision and mark packet

STAMUS NETWORKS

Complex integration: repeat mode (1/2)

NFQUEUE hardcore

- Alternate decision
  - NF_REPEAT : send the packet back to the start of the table
  - NF_QUEUE : send the packet to another queue (chain software using NFQUEUE)
- Packet marking: userspace can take decision and mark packet

Algorithm

1. NFQUEUE packet as first rule if they have no mark
2. IPS mark packet (take one bit of the mark)
3. IPS verdict with with NF_REPEAT
4. If accepted packet get back to queuing rule but marked

STAMUS NETWORKS

Complex integration: repeat mode (2/2)

Ruleset

```
iptables -I FORWARD -m mark ! --mark $MARK/$MASK -j NFQUEUE
```

STAMUS NETWORKS

Complex integration: repeat mode (2/2)

Ruleset

```
iptables -I FORWARD -m mark ! --mark $MARK/$MASK -j NFQUEUE
```

Suricata configuration

```
nfq:
  mode: repeat
  repeat_mark: 1
  repeat_mask: 1
```

STAMUS NETWORKS

Complex integration: repeat mode (2/2)

Ruleset

```
iptables -I FORWARD -m mark ! --mark $MARK/$MASK -j NFQUEUE
```

Suricata configuration

```
nfq:
  mode: repeat
  repeat_mark: 1
  repeat_mask: 1
```

Snort cookbook

You need to stop that!

STAMUS NETWORKS

1 Introduction

2 Iptables

3 Nftables

4 Conclusion

STAMUS NETWORKS

What's new with nftables ?

Nftables and queue

- No change needed on userspace software
- No new features in kernel
- Just a syntax update

STAMUS NETWORKS

What's new with nftables ?

Nftables and queue

- No change needed on userspace software
- No new features in kernel
- Just a syntax update

What's new with nftables ?

Nftables and queue

- No change needed on userspace software
- No new features in kernel
- Just a syntax update

Nftables and chain

- No chain by default
- All chains have to be created by user

STAMUS NETWORKS

Nftables chain

Principle

- User chain
  - accessible via jump
- Primary chain
  - Type: filtering, nat, route
  - Hook: point of attachement
  - Priority: ordering of chain

STAMUS NETWORKS

Nftables chain

Principle

- User chain
  - accessible via jump
- Primary chain
  - Type: filtering, nat, route
  - Hook: point of attachement
  - Priority: ordering of chain

Syntax

```
nft> add chain filter input { type filter hook input priority 0;}
```

STAMUS NETWORKS

Solution: dedicate a nftables chain to IPS

- Create one chain for filtering
- Create on IPS chain
- Put IPS chain after the filtering chain

STAMUS NETWORKS

Building the perfect IPS ruleset (1/2)

Implementation

- We name chains accordingly to function
- We set a higher priority to IPS chain

Creating the chains

```
nft -i
nft> add table filter
nft> add chain filter firewall { type filter hook forward priority 0;}
nft> add chain filter IPS { type filter hook forward priority 10;}
```

STAMUS NETWORKS

Building the perfect IPS ruleset (2/2)

Firewall rules

```
nft> add rule filter firewall ct state established accept
nft> add rule filter firewall tcp dport ssh counter accept
nft> add rule filter firewall tcp dport 443 accept
nft> add rule filter firewall counter log drop
```

STAMUS NETWORKS

Building the perfect IPS ruleset (2/2)

Firewall rules

```
nft> add rule filter firewall ct state established accept
nft> add rule filter firewall tcp dport ssh counter accept
nft> add rule filter firewall tcp dport 443 accept
nft> add rule filter firewall counter log drop
```

IPS rule

```
nft> add rule filter IPS queue
```

STAMUS NETWORKS

queue target detail

Syntax

```
nft add rule test input queue num 1-3 bypass fanout
```

Options

- num: queue num to use, interval is load balancing
- bypass: accept packet if noone is listening
- fanout: accept new packets if userspace queue length is too long

STAMUS NETWORKS

Nftables brings a fully compliant solution

Nftables solution

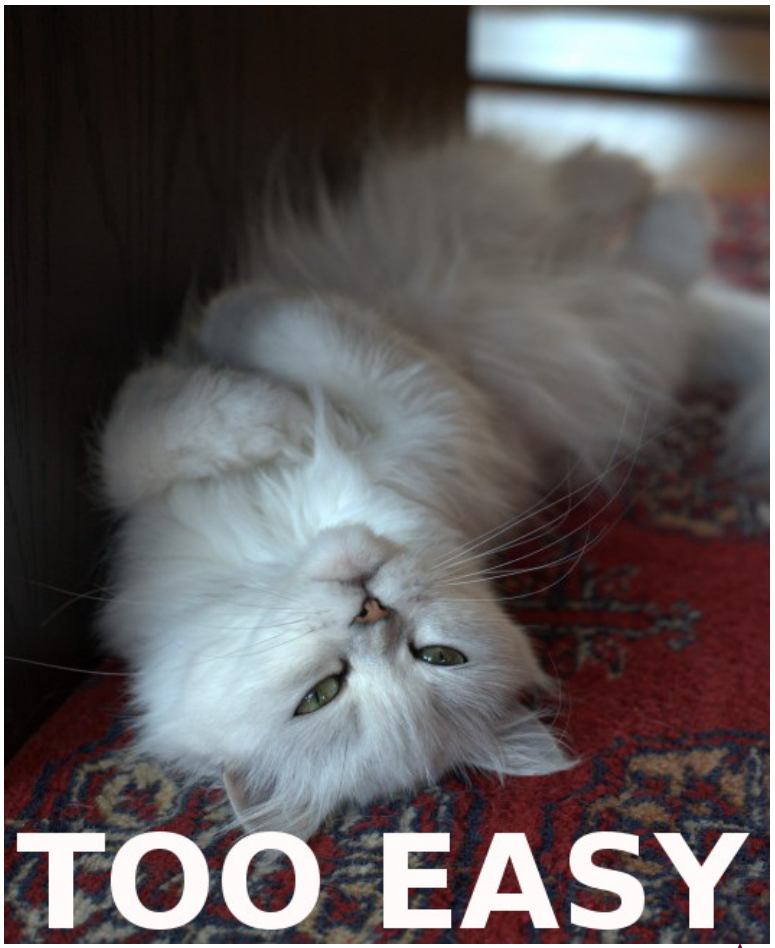
- IPS filtering are fully independant of filtering
- Need only to fix priority If ruleset is managed by a software

STAMUS NETWORKS

Nftables brings a fully compliant solution

Nftables solution

- IPS filtering are fully independant of filtering
- Need only to fix priority If ruleset is managed by a software



STAMUS NETWORKS

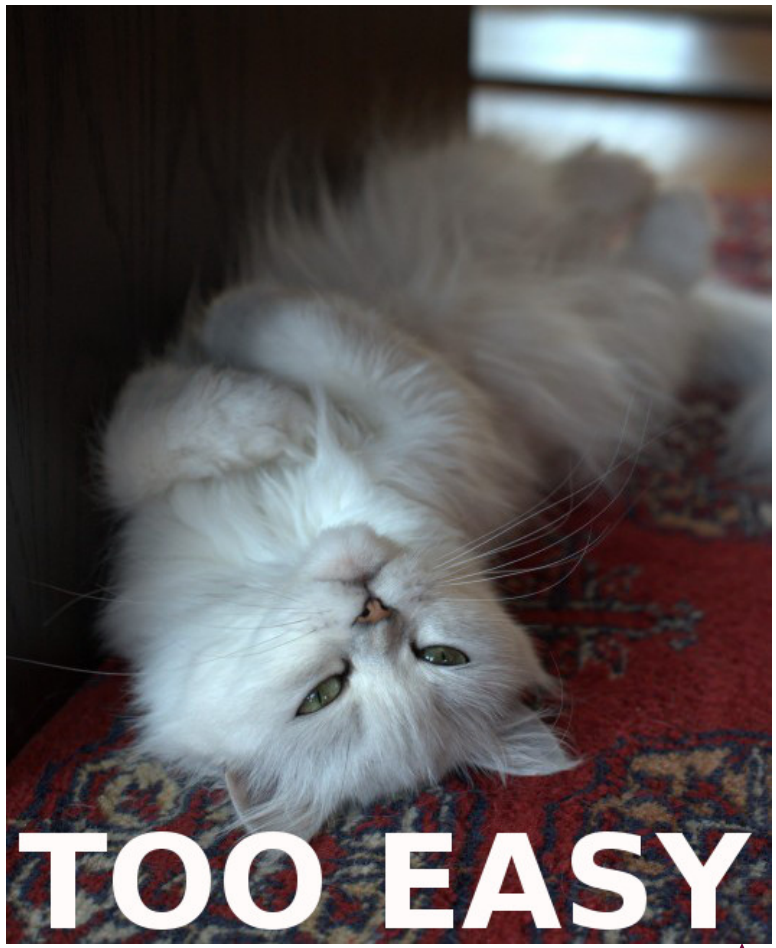
Nftables brings a fully compliant solution

Nftables solution

- IPS filtering are fully independant of filtering
- Need only to fix priority If ruleset is managed by a software

Nftables flexibility

- Powerful and brings solution
- Tools will have to be well designed



STAMUS NETWORKS

nftables chain priority

Priority and iptables

- Iptables compatibility predefined chain
- Priority has to be chosen accordingly

List of chain priorities

```
NF_IP_PRI_CONNTRACK_DEFRAG (-400): priority of defragmentation
NF_IP_PRI_RAW (-300): traditional priority of the raw table placed before connection tracking oper
NF_IP_PRI_SELINUX_FIRST (-225): SELinux operations
NF_IP_PRI_CONNTRACK (-200): Connection tracking operations
NF_IP_PRI_MANGLE (-150): mangle operation
NF_IP_PRI_NAT_DST (-100): destination NAT
NF_IP_PRI_FILTER (0): filtering operation, the filter table
NF_IP_PRI_SECURITY (50): Place of security table where secmark can be set for example
NF_IP_PRI_NAT_SRC (100): source NAT
NF_IP_PRI_SELINUX_LAST (225): SELInux at packet exit
NF_IP_PRI_CONNTRACK_HELPER (300): connection tracking at exit
```

STAMUS NETWORKS

1 Introduction

2 Iptables

3 Nftables

4 Conclusion

STAMUS NETWORKS

Questions ?

Simple is beautiful

- From over complex to a natural way of doing things

Thanks to

- Patrick McHardy to have started nftables
- Pablo Neira Ayuso and the others to have continued the work

More information

- Netfilter: `http://www.netfilter.org/`
- My blog: `https://home.regit.org/`
- Stamus Networks: `https://www.stamus-networks.com/`

STAMUS NETWORKS