

# Vuurmuur - iptables manager

Victor Julien

June 27, 2016

## Vuurmuur

- 'founder' and lead developer of Vuurmuur

## Open Source

- Suricata IDS/IPS
- ModSecurity, libhttp, modsec2sguil, sgul, snort\_inline

## Contact

- @inliniac
- <http://blog.inliniac.net/>

## Powerful, but complex

- Packet processing happens in several tables: mangle, filter, nat, raw
- Default chains: INPUT, OUTPUT, FORWARD and several others
- Also, define your own chains
- Don't get me started on traffic shaping

# Rule Example

Example of a rule:

```
iptables -t filter -A FORWARD -i eth1 -o ppp0 \
  -p tcp -m tcp --syn \
  -s 192.168.0.33/255.255.255.255 --sport 1024:65535 \
  -d 0.0.0.0/0.0.0.0 --dport 4070 \
  -m limit --limit 5/sec --limit-burst 10 \
  -m conntrack --ctstate NEW \
  -j NFLOG --nflog-prefix "ACCEPT " --nflog-group 9
```

Rather complex, right?

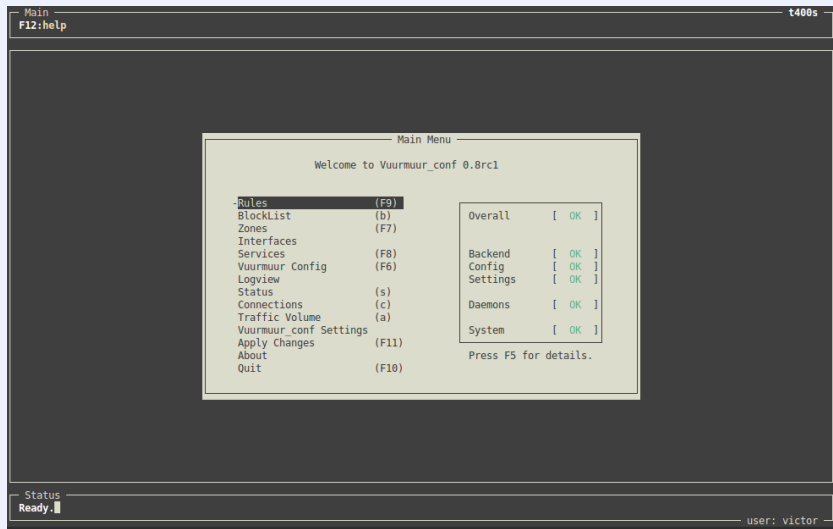
- Started in 2002 as a project to learn programming
- Born out of frustration with managing iptables scripts
- Mature, free-time project
- Therefore, slow moving project :)

## Goal

*Allow users to easily setup and manage a secure and efficient firewall, without needing iptables specific knowledge.*

## Features

- Ncurses GUI – manage over SSH
- Target is gateway firewalls
- Log viewer, connection viewer
- Easy way to setup NAT, portforwarding
- NFQUEUE support for integrating with Suricata IPS
- Basic traffic shaping and prioritization support
- Basic IPv6 support
- Keeps an 'audit log' of all changes

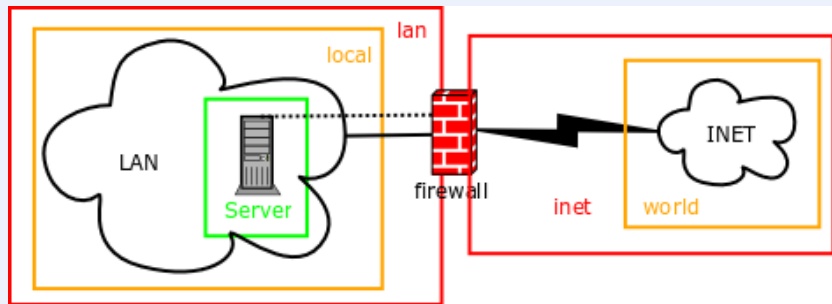




## Concepts

- One or more 'zones': in/out, lan/wan, red/green
- Within each zone: one or more networks
- Within each network: one or more hosts (optional)
- Interface mapping with local interfaces
- Interfaces are connected to a network
- Services define protocols and ports
- Consistent use of named objects in rules, log viewer, connection viewer

# Concepts



## Rule Example

- accept service http from local.lan to world
- snat service http from local.lan to world

## About the names

- Zone names have a fixed structure
- "local.lan" means: zone "lan" and within that network "local"
- In "server.local.lan", "server" is the host
- This way it's always clear what part of your network a rule applies to

## Port forwarding rule Example

- portfw service ssh from world to myserver.servers.dmz

## Port forwarding rule example, with NFQUEUE

- nfqueue service smtp from world to mailserver.servers.dmz
- dnat service smtp from world to mailserver.servers.dmz

# Rules

Rules t400s

F12:help INS:new DEL:del RET:edit m:move f:filter F10:back

Rules Section					
Nr.	Action	Service	Source	Destination	Options
[ ] 1	Accept	ping	firewall	inet	log,loglimit="20"
[ ] 2	Accept	any	lan	inet	log,loglimit="20"
[ ] 3	NFQueue	any	inet	firewall(any)	-
[ ] 4	Accept	any	inet	firewall(any)	-
[ ] 5	NFQueue	any	firewall(any)	inet	-
[ ] 6	Accept	any	firewall(any)	inet	-
[ ] 7	NFQueue	any	inet	firewall	-
[ ] 8	Accept	any	inet	firewall	-
[ ] 9	NFQueue	any	firewall	inet	-
[ ] 10	Accept	any	firewall	inet	-
[ ] 11	Accept	any	any	any	-
12	-----[ IPS stuff ]-----				
[x] 13	NFQueue	any	any	any	nfqueueenum="1"
[ ] 14	NFQueue	any	any	any	-
[ ] 15	Accept	any	lan	lan	log,loglimit="20"
[ ] 16	NFQueue	any	lan	inet	nfqueueenum="0"
[ ] 17	NFQueue	any	lan	inet	nfqueueenum="1"
[ ] 18	Queue	any	lan	inet	log
[ ] 19	Queue	any	inet	firewall	log
[ ] 20	Accept	any	firewall	inet	-
21	-----[ Firewall outside connectivity ]-----				
[x] 22	Accept	dns	firewall	world.inet	log,loglimit="20"
[x] 23	Accept	http	firewall	world.inet	log,loglimit="20"
[x] 24	Accept	https	firewall	world.inet	log,loglimit="20"
[x] 25	Accept	ssh	firewall	world.inet	log,loglimit="20"

Status

Ready.

user: victor

## Traffic Shaping Rule Example

```
accept service any from voip.local.lan to world.inet \
options log,loglimit="1", \
in_min="50kbps",out_min="50kbps",prio="1"
```

## How it works

- Read rules, zones, etc
- Turn into iptables and 'tc' rulesets
- Feeds ruleset to iptables-restore
- Enable/disable ip forwarding if necessary
- Helpful command: `vuurmuur -b` (bash out)

# Log Viewer

```
Logview t400s
F12:help m:manage s:search f:filter p:pause c:clear 1-7:hide F10:back

Jul 4 15:36:11: DROP 5353->5353(udp) fe80::221:6aff:fe87:d4dc -> ff02::fb 'out policy' (out: wlan0 fe80::221:6aff:fe87:d4dc:5353 ->
Jul 4 15:36:11: DROP 5353->5353(udp) firewall(wireless) -> 224.0.0.251 'out policy' (out: wlan0 192.168.0.31:5353 -> 224.0.0.251:53
Jul 4 15:36:14: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:36:17: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:36:25: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:36:35: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:36:54: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:37:07: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:37:21: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:37:35: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:37:42: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:37:49: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:37:58: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l
Jul 4 15:38:08: DROP 68->67(udp) firewall(wireless) -> 192.168.0.1 'out policy' (out: wlan0 192.168.0.31:68 -> 192.168.0.1:67 UDP l

Status
Loading loglines into memory... loaded 13 lines.
user: victor
```



# Connection Viewer

Connections

t400s

F12:help m:manage i:in/out/fw c:connect g:grp u:unknown ip f:filter a:account d:details F10:back

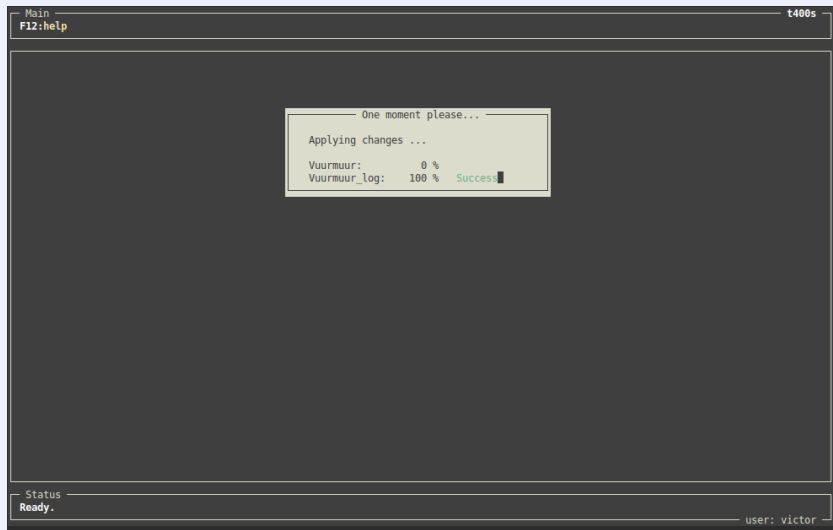
31: dns	firewall(wireless)	-> dmz.lan	ESTA OUT <-	9 k	2 k ->	192.168.0.31 ->	192.168.0.1:53 (17)
10: https	firewall(wireless)	-> world.inet	ESTA OUT <-	605 k	51 k ->	192.168.0.31 ->	82.94.234.57:443 (6)
6: http	firewall(wireless)	-> world.inet	ESTA OUT <-	8 k	6 k ->	192.168.0.31 ->	74.125.136.94:80 (6)
4: http	firewall(wireless)	-> world.inet	CONN OUT <-	0 b	0 b ->	192.168.0.31 ->	80.101.90.58:80 (6)
3: http	firewall(wireless)	-> world.inet	DISC OUT <-	11 k	3 k ->	192.168.0.31 ->	83.145.197.2:80 (6)
1: https	firewall(wireless)	-> world.inet	DISC OUT <-	5 k	1 k ->	192.168.0.31:50470 ->	82.103.140.40:443 (6)
1: proto 2	firewall(wireless)	-> world.inet	- OUT <-	n/a	n/a ->	192.168.0.31 ->	224.0.0.22 (2)
1: proto 2	world.inet	-> world.inet	- FWD <-	n/a	n/a ->	192.168.122.1 ->	224.0.0.22 (2)
1: proto 2	local.lan	-> world.inet	- FWD <-	n/a	n/a ->	192.168.1.48 ->	224.0.0.22 (2)
1: 41514 -> 631 ::1		-> ::1	ESTA FWD <-	0 b	0 b ->	::1:41514 ->	::1:631 (6)

Status

Ready.

user: victor

# Applying Changes





**KEEP  
CALM  
AND  
PRAY THE DEMO  
WORKS**

## Vuurmuur to JSON logging

```
stack=log1:NFLOG,base1:BASE,ifi1:IFINDEX, \
    ip2str1:IP2STR,mac2str1:HWHDR,json1:JSON
[log1]
group=9
[json1]
sync=1
file="/var/log/ulogd.json"
```

## PCAP Logging

```
stack=log2:NFLOG,base1:BASE,pcap1:PCAP  
[log2]  
group=7  
[pcap1]  
file="/var/log/vuurmuur.pcap"  
sync=1
```

## In (Slow) Development

- Use Ulogd2 to replace vuurmuur\_log

## nftables support

- nftables unifies ip4/ip6 => "inet"
- built-in traffic shaping features. No more 'tc' hell.
- sets and other data types will be very helpful
- BUT! A major feature -> time constraints

## Get Involved!

- Open Source: GPLv2+
- <https://www.vuurmuur.org/>
- #vuurmuur on freenode
- <https://github.com/inliniac/vuurmuur>