netfilter/iptables/conntrack debugging

Florian Westphal

Networking Services Team, Red Hat

June 2015

packets disappearing

e.g. added some dnat-based port forwarding, but it doesn't seem to work

How to debug that?

netfilter tracing can be used to find wich rules match a packet

```
iptables -t raw -A PREROUTING -j TRACE
```

```
nft add rule ip raw prerouting nftrace set 1
```

its good idea to only trace packets that you're interested in, e.g. `-p tcp --dport 22 --syn`

```
TRACE: raw:PREROUTING:policy:2 IN=eth0 SRC=192.168.0.8 \
       DST=192.168.0.10 SPT=7627 DPT=22 SYN
TRACE: mangle:PREROUTING:policy:1 IN=eth0 SRC=192.168.0.8 \
       DST=192.168.0.10 SPT=7627 DPT=22 SYN
TRACE: nat:PREROUTING:rule:1 IN=eth0 SRC=192.168.0.8 \
       DST=192.168.0.10 SPT=7627 DPT=22 SYN \
TRACE: mangle:INPUT:policy:1 IN=eth0 SRC=192.168.0.8 \
       DST=192.168.0.10 SPT=7627 DPT=2222 SYN
TRACE: filter:INPUT:rule:11 IN=eth0 SRC=192.168.0.8 \
       DST=192.168.0.10 SPT=7627 DPT=2222 SYN
```

- there is DNAT rewrite active (nat rule 1 rewrites dport to 2222)
- last match is in filter table, rule number 11

logs match events `rule`, `policy`, `return`

- keeps a record of what packets have passed through machine
- in order to figure out how they are related
- decides if packet x and y are part of same connection
- NAT is built on top of this, configuration via `nat` table in iptables
- kernel api for optional extensions, e.g. accounting, event, ...

Kernel provides a full featured userspace interface

- dump list of connections/expectations
- change metadata associated with a connection, e.g. connmark
- delete or insert new connections
- provides access to expectation table too

conntrack(8): command line interface

```
# conntrack -L
tcp      6 431985 ESTABLISHED src=192.168.0.3 \
 dst=10.45.5.39 sport=45579 dport=443 .. [ASSURED] mark=0
```

connect to closed port:

```
# conntrack -E
[NEW] tcp      6 120 SYN_SENT src=192.168.0.10 \
      dst=192.168.0.7 sport=40607 dport=12345 [UNREPLIED]
[DESTROY] tcp 6 src=192.168.0.10 \
      dst=192.168.0.7 sport=40607 dport=12345 [UNREPLIED]
```

no output:

- no conntrack or ctnetlink support
- first packet got dropped by iptables rule

conntrack sysctls:

- max connection size (default 64k)
  `net.netfilter.nf_conntrack_max`
  - exceeding this is announced in dmesg ('table full, dropping packet')
- `net.netfilter.nf_conntrack_log_invalid` (set to 6 for tcp, see `/etc/protocols`)
- `nf_conntrack_tcp_be_liberal` (no strict in-window check)
- `net.netfilter.nf_conntrack_tcp_loose` (mid-stream pickup)

Full list:
`Documentation/networking/nf_conntrack-sysctl.txt`

- can exclude packets from being tracked via
  `-t raw -A PREROUTING -j CT --notrack`
- can tune conntrack timeouts also using `CT` target in addition to sysctls

connection tracking helpers

- some protocols are more complex
  - FTP data
  - audio/video traffic
- helper sniffs traffic, e.g. tcp port 21 for ftp
  - if 'magic' found, e.g. `PORT a.b.c.d.e.f\r\n` → add entry in expect table
  - 'new' DATA connection is then RELATED rather than NEW (see `-m conntrack --ctstate` match)
- unfortunately, best effort only – don't want tcp stream reassembly in kernel
- often problematic, e.g. can allow internal clients to open arbitrary ports
- thus moving over to explict configuration via
  `-t raw -p tcp --dport 21 -j CT --helper ftp`

Summary

- tcpdump
- nft/iptables trace to determine 'problematic' rule
- make conntrack more verbose via `nf_conntrack_log_invalid` sysyctl
- conntrack tool to look at connections, state, ...