NAME

ematch − extended matches for use with "basic" or "flow" filters

SYNOPSIS

tc filter add .. basic match EXPR **.. flowid ..**

EXPR := *TERM* [{ **and | or** } *EXPR*]

TERM := [**not**] { *MATCH* | '(' *EXPR* ')' }

MATCH := *module* '(' *ARGS* ')'

ARGS := *ARG1 ARG2* ..

MATCHES

cmp

Simple comparison ematch: arithmetic compare of packet data to a given value.

cmp(*ALIGN* at *OFFSET* [*ATTRS*] { *eq* | *lt* | *gt* } *VALUE*)

ALIGN := { *u8* | *u16* | *u32* }

ATTRS := [layer *LAYER*] [mask *MASK*] [trans]

LAYER := { *link* | *network* | *transport* | *0..2* }

meta

Metadata ematch

meta(*OBJECT* { *eq* | *lt* |*gt* } *OBJECT*)

OBJECT := { *META_ID* | *VALUE* }

META_ID := *id* [shift *SHIFT*] [mask *MASK*]

meta attributes:

random 32 bit random value

loadavg_1 Load average in last 5 minutes

nf_mark Netfilter mark

vlan Vlan tag

sk_rcvbuf Receive buffer size

sk_snd_queue Send queue length

A full list of meta attributes can be obtained via

# tc filter add dev eth1 basic match 'meta(list)'

nbyte

match packet data byte sequence

nbyte(*NEEDLE* at *OFFSET* [layer *LAYER*])

NEEDLE := { *string* | *c-escape-sequence* }

OFFSET := *int*

LAYER := { *link* | *network* | *transport* | *0..2* }

u32

u32 ematch

u32(*ALIGN VALUE MASK* at [nexthdr+] *OFFSET*)

ALIGN := { *u8* | *u16* | *u32* }

ipset

test packet against ipset membership

ipset(*SETNAME FLAGS*)

SETNAME := *string*

FLAGS := { *FLAG* [, *FLAGS*] }

The flag options are the same as those used by the iptables "set" match.

When using the ipset ematch with the "ip_set_hash:net,iface" set type, the interface can be queried using "src,dst (source ip address, outgoing interface) or "src,src" (source ip address, incoming interface) syntax.

ipt

test packet against xtables matches

ipt(*[-6] -m MATCH_NAME FLAGS*)

MATCH_NAME := *string*

FLAGS := { *FLAG* [, *FLAGS*] }

The flag options are the same as those used by the xtable match used.

CAVEATS

The ematch syntax uses '(' and ')' to group expressions. All braces need to be escaped properly to prevent shell commandline from interpreting these directly.

When using the ipset ematch with the "ifb" device, the outgoing device will be the ifb device itself, e.g. "ifb0". The original interface (i.e. the device the packet arrived on) is treated as the incoming interface.

EXAMPLE & USAGE

```
# tc filter add .. basic match ...
```

```
# ’cmp(u16 at 3 layer 2 mask 0xff00 gt 20)’
```

```
# ’meta(nfmark gt 24)’ and ’meta(tcindex mask 0xf0 eq 0xf0)’
```

```
# ’nbyte("ababa" at 12 layer 1)’
```

```
# ’u32(u16 0x1122 0xffff at nexthdr+4)’
```

Check if packet source ip address is member of set named **bulk**:

```
# ’ipset(bulk src)’
```

Check if packet source ip and the interface the packet arrived on is member of "hash:net,iface" set named **interactive**:

```
# ’ipset(interactive src,src)’
```

Check if packet matches an IPSec state with reqid 1:

```
# ’ipt(-m policy --dir in --pol ipsec --reqid 1)’
```

AUTHOR

The extended match infrastructure was added by Thomas Graf.