1. **calico iptables详解**

转载 2017年06月15日 19:45:06

**报文处理过程**

报文处理过程中使用的标记位：

一共使用了3个标记位，0x7000000对应的标记位

0x1000000: 报文的处理动作，置1表示放行，默认0表示拒绝。

0x2000000: 是否已经经过了policy规则检测，置1表示已经过。

0x4000000: 报文来源，置0，表示来自host-endpoint

流入报文来源:

1. 以cali+命名的网卡收到的报文，这部分报文是本node上的endpoint发出的

2. 其他网卡接收的报文，这部分报文可能是其它node发送过来

, 也可能是node上本地进程发出的

流入的报文去向：

1. 访问本node的host endpoint，通过INPUT过程处理

2. 访问本node的workload endpoint，通过INPUT过程处理

3. 访问其它node的host endpoint，通过FORWARD过程处理。

4. 访问其它node的workload endpoint，通过FORWARD过程处理。

流入的报文在路由决策之前的处理过程相同的，路由决策之后，分别进入INPUT规则链和FORWARD链。

raw.PREROUTING -> mangle.PREROUTING -> nat.PREROUTING -> mangle.INPUT -> filter.INPUT

raw.PREROUTING -> mangle.PREROUTING -> nat.PREROUTING -> mangle.FORWARD -> filter.FORWARD -> mangle.POSTROUTING -> nat.POSTROUTING

报文处理流程（全):

from-XXX: XXX发出的报文 tw: 简写，to wordkoad endpoint

to-XXX: 发送到XXX的报文 po: 简写，policy outbound

cali-: 前缀，calico的规则链 pi: 简写，policy inbound

wl: 简写，workload endpoint pro: 简写，profile outbound

fw: 简写，from workload endpoint pri: 简写，profile inbound

(receive pkt)

cali-PREOUTING@raw -> cali-from-host-endpoint@raw -> cali-PREROUTING@nat

| ^ |

| (-i cali+) | |

+--- (from workload endpoint) ----+ |

|

cali-fip-dnat@nat

|

(rotuer decision)

|

+--------------------------------------------+

| |

cali-INPUT@filter cali-FORWARD@filter

(-i cali+) | (-i cali+) | (-o cali+)

+----------------------------+ +------------+-------------+

| | | | |

cali-wl-to-host cali-from-host-endpoint | cali-from-host-endpoint |

@filter @filter | @filter |

| < END > | | |

| | cali-to-host-endpoint |

| | @filter |

| | < END > |

| | |

cali-from-wl-dispatch@filter <---------------------+ cali-to-wl-dispatch@filter

(-i cali+) | ----------------+ (-o cali+) |

+-----------------------+ | +----------------------+

| | | | |

cali-fw-cali0ef24b1 cali-fw-cali0ef24b2 | cali-tw-cali03f24b1 cali-tw-cali03f24b2

@filter @filter | @filter @filter

(-i cali0ef24b1) (-i cali0ef24b2) | (-o cali0ef24b1) (-o cali0ef24b2)

| | | | |

+-----------------------+ | +----------------------+

| | |

cali-po-[POLICY] | cali-pi-[POLICY]

@filter | @filter

| | |

cali-pro-[PROFILE] | cali-pri-[PROFILE]

@filter | @filter

| | |

< END > +-----> |----> cali-POSTROUTING@nat

+-----> | |

| cali-fip-snat@nat

| |

| cali-nat-outgoing@nat

| |

| (if dip is local: send to lookup)

+---------+--------+ (else: send to nic's qdisc)

| | < END >

cali-to-host-endpoint@filter |

| |

+------------------+

^ (-o cali+)

|

cali-OUTPUT@filter

^

(send pkt) |

(router descition) -> cali-OUTPUT@nat -> cali-fip-dnat@nat

node本地发出的报文，经过路由决策之后，直接进入raw,OUTPUT规则链:

raw.OUTPUT -> mangle.OUTPUT -> nat.OUTPUT -> filter.OUTPUT -> mangle.POSTROUTING -> nat.POSTROUTING

**路由决策之前：流入node的报文的处理**

**进入raw表**

PREROUTING@raw:

-A PREROUTING -m comment --comment "cali:6gwbT8clXdHdC1b1" -j cali-PREROUTING

cali-PREROUTING@RAW:

-A cali-PREROUTING -m comment --comment "cali:x4XbVMc5P\_kNXnTy" -j MARK --set-xmark 0x0/0x7000000

-A cali-PREROUTING -i cali+ -m comment --comment "cali:fQeZek80kVOPa0xO" -j MARK --set-xmark 0x4000000/0x4000000

-A cali-PREROUTING -m comment --comment "cali:xp3NolkIpulCQL\_G" -m mark --mark 0x0/0x4000000 -j cali-from-host-endpoint

-A cali-PREROUTING -m comment --comment "cali:fbdE50A0BiINbNiA" -m mark --mark 0x1000000/0x1000000 -j ACCEPT

规则1，清空所有标记

规则2，从cali+网卡进入的报文，设置mark: 0x4000000/0x4000000

规则3，非cali+网卡收到的报文，即从host-endpoint进入的报文，进入cali-from-host-endpoints规则链条

这里没有设置host-endpoint的策略，所有cali-from-host-endpoint规则链是空的。

**进入nat表**

PREROUTING@nat:

-A PREROUTING -m comment --comment "cali:6gwbT8clXdHdC1b1" -j cali-PREROUTING

-A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER

直接进入cali-PREROUTING

cali-PREROUTING@nat:

-A cali-PREROUTING -m comment --comment "cali:r6XmIziWUJsdOK6Z" -j cali-fip-dnat

如果目标地址是fip(floating IP)，会在cali-fip-dnat中做dnat转换

nat表中做目的IP转换，这里没有设置，所以cali-fip-dnat是空的。

经过nat表之后，会进行路由决策:

1. 如果是发送给slave1的报文，经过规则链: INPUT@mangle、INPUT@filter

2. 如果不是发送给slave1报文，经过规则链: FORWARD@mangle、FORWARD@filer、POSTROUTING@mangle、POSTROUTING@nat

**路由决策之后：发送到本node的host endpoint 和 workload endpoint**

**进入filter表**

INPUT@filter:

-A INPUT -m comment --comment "cali:Cz\_u1IQiXIMmKD4c" -j cali-INPUT

直接进入cali-INPUT

cali-INPUT@filter:

-A cali-INPUT -m comment --comment "cali:46gVAqzWLjH8U4O2" -m mark --mark 0x1000000/0x1000000 -m conntrack --ctstate UNTRACKED -j ACCEPT

-A cali-INPUT -m comment --comment "cali:5M2EkEm-RVlDLAfE" -m conntrack --ctstate INVALID -j DROP

-A cali-INPUT -m comment --comment "cali:8ggYjLbFRX5Ap9Zj" -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A cali-INPUT -i cali+ -m comment --comment "cali:mA3ZJKi9nadUmYVF" -g cali-wl-to-host

-A cali-INPUT -m comment --comment "cali:hI4IjifGj0fegLPE" -j MARK --set-xmark 0x0/0x7000000

-A cali-INPUT -m comment --comment "cali:wdegoKfPlcmsZTOM" -j cali-from-host-endpoint

-A cali-INPUT -m comment --comment "cali:r875VVc8vFk1f-ZA" -m comment --comment "Host endpoint policy accepted packet." -m mark --mark 0x1000000/0x1000000 -j ACCEPT

规则4，从cali+网卡进入的报文，进入wl-to-host的规则链，wl是workload的缩

规则6，非cali+网卡收到的报文，host-endpoint的规则链

**来自其它node的报文**

这里没有对host endpoint设置规则，所以规则链时空

cali-from-host-endpoint@filter:

空

**来自本node上workload endpoint的报文**

检察一下是否允许workload enpoint发出这些报文。

cali-wl-to-host@filter:

-A cali-wl-to-host -p udp -m comment --comment "cali:aEOMPPLgak2S0Lxs" -m multiport --sports 68 -m multiport --dports 67 -j ACCEPT

-A cali-wl-to-host -p udp -m comment --comment "cali:SzR8ejPiuXtFMS8B" -m multiport --dports 53 -j ACCEPT

-A cali-wl-to-host -m comment --comment "cali:MEmlbCdco0Fefcrw" -j cali-from-wl-dispatch

-A cali-wl-to-host -m comment --comment "cali:Q2b2iY2M-vmds5iY" -m comment --comment "Configured DefaultEndpointToHostAction" -j RETURN

规则1，允许请求DHCP

规则2，允许请求DNS

规则3，匹配workload endpoint各自的规则，将会依次检察policy的egress、各自绑定的profile的egress。

根据接收报文的网卡做区分，cali-from-wl-dispatch@filter:

-A cali-from-wl-dispatch -i cali0ef24b1 -m comment --comment "cali:RkM6MKQgU0OTxwKU" -g cali-fw-cali0ef24b1

-A cali-from-wl-dispatch -i cali0ef24b2 -m comment --comment "cali:7hIahXYNmY9JDfKG" -g cali-fw-cali0ef24b2

-A cali-from-wl-dispatch -m comment --comment "cali:YKcphdGNZ1PwfGvt" -m comment --comment "Unknown interface" -j DROP

规则1，cali0ef24b1是slave1-frontend1

规则2，cali0ef24b2是slave1-frontend2

只查看其中一个，cali-fw-cali0ef24b1@filter:

-A cali-fw-cali0ef24b1 -m comment --comment "cali:KOIFJxkWqvpSMSzk" -j MARK --set-xmark 0x0/0x1000000

-A cali-fw-cali0ef24b1 -m comment --comment "cali:Mm\_GAikGLiINmRQh" -m comment --comment "Start of policies" -j MARK --set-xmark 0x0/0x2000000

-A cali-fw-cali0ef24b1 -m comment --comment "cali:c6bGtQzwKsoipZq6" -m mark --mark 0x0/0x2000000 -j cali-po-namespace-default

-A cali-fw-cali0ef24b1 -m comment --comment "cali:46b6gNjtXYDXasAi" -m comment --comment "Return if policy accepted" -m mark --mark 0x1000000/0x1000000 -j RETURN

-A cali-fw-cali0ef24b1 -m comment --comment "cali:6kNf2\_vqiCYkwInx" -m comment --comment "Drop if no policies passed packet" -m mark --mark 0x0/0x2000000 -j DROP

-A cali-fw-cali0ef24b1 -m comment --comment "cali:GWdesho87l08Srht" -m comment --comment "Drop if no profiles matched" -j DROP

这个endpoint没有绑定profile，所以只做了policy的egress规则检测

规则4，cali-po-namespace-default，policy“namespace-default”的egress规则，po表示policy outbound。

slave2上用于service”database”的endpoint绑定了profile，cali-fw-cali0ef24b3@filter:

-A cali-fw-cali0ef24b3 -m comment --comment "cali:CxOkDjFlTZaT70VP" -j MARK --set-xmark 0x0/0x1000000

-A cali-fw-cali0ef24b3 -m comment --comment "cali:2QQMYVCQs\_pXjuNx" -m comment --comment "Start of policies" -j MARK --set-xmark 0x0/0x2000000

-A cali-fw-cali0ef24b3 -m comment --comment "cali:DyV6lV76WK8YZaJX" -m mark --mark 0x0/0x2000000 -j cali-po-namespace-default

-A cali-fw-cali0ef24b3 -m comment --comment "cali:TvuIyAsPjYsOd6oG" -m comment --comment "Return if policy accepted" -m mark --mark 0x1000000/0x1000000 -j RETURN

-A cali-fw-cali0ef24b3 -m comment --comment "cali:TXGkGvhZNM8gWSFv" -m comment --comment "Drop if no policies passed packet" -m mark --mark 0x0/0x2000000 -j DROP

-A cali-fw-cali0ef24b3 -m comment --comment "cali:sc2HAyx9fn5\_mw0k" -j cali-pro-profile-database

-A cali-fw-cali0ef24b3 -m comment --comment "cali:LxL3UEOyLww7VztW" -m comment --comment "Return if profile accepted" -m mark --mark 0x1000000/0x1000000 -j RETURN

-A cali-fw-cali0ef24b3 -m comment --comment "cali:PMXWen2JRtHBNBVn" -m comment --comment "Drop if no profiles matched" -j DROP

可以看到，多了一个cali-pro-profile-database的检测

规则6，cali-pro-profile-database, profile"profile-database"的egress规则，pro表示profile outbound。

policy的egress规则，cali-po-namespace-default@filter:

-A cali-po-namespace-default -m comment --comment "cali:uT-hMQk\_SRgHsKxT" -j MARK --set-xmark 0x1000000/0x1000000

-A cali-po-namespace-default -m comment --comment "cali:KDa-ASKrRQu4eYZs" -m mark --mark 0x1000000/0x1000000 -j RETURN

policy“namespace-default”的egress规则是allow，所以规则1直接打了标记"0x1000000/0x1000000"。

slave2上的endpoint绑定的profile规则的egress规则，cali-pro-profile-database@filter:

-A cali-pro-profile-database -m comment --comment "cali:laSwzk9Ihy5ArWJB" -j MARK --set-xmark 0x1000000/0x1000000

-A cali-pro-profile-database -m comment --comment "cali:BpvFNyMPRLC0lDtu" -m mark --mark 0x1000000/0x1000000 -j RETURN

profile-database的egress是allow，直接打标记0x1000000/0x1000000。

**路由决策之后：需要转发的报文**

filter.FORWARD:

-A FORWARD -m comment --comment "cali:wUHhoiAYhphO9Mso" -j cali-FORWARD

直接进入cali-FROWARD

filter.cali-FORWARD，根据接收网卡做egress规则匹配，根据目标网卡做ingress规则匹配:

-A cali-FORWARD -m comment --comment "cali:jxvuJjmmRV135nVu" -m mark --mark 0x1000000/0x1000000 -m conntrack --ctstate UNTRACKED -j ACCEPT

-A cali-FORWARD -m comment --comment "cali:8YeDX9Z0tXyO0Sp8" -m conntrack --ctstate INVALID -j DROP

-A cali-FORWARD -m comment --comment "cali:1GMSV-PhhZ8QbJg4" -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A cali-FORWARD -i cali+ -m comment --comment "cali:36TkoGXj9EF7Plkv" -j cali-from-wl-dispatch

-A cali-FORWARD -o cali+ -m comment --comment "cali:URMhBRo8ugd8J8Yx" -j cali-to-wl-dispatch

-A cali-FORWARD -i cali+ -m comment --comment "cali:FyhWsW08U3a5niLK" -j ACCEPT

-A cali-FORWARD -o cali+ -m comment --comment "cali:G655uIfZuidj1gAw" -j ACCEPT

-A cali-FORWARD -m comment --comment "cali:4GbueNC2iWajKnxO" -j MARK --set-xmark 0x0/0x7000000

-A cali-FORWARD -m comment --comment "cali:bq3wVY3mkXk96NQP" -j cali-from-host-endpoint

-A cali-FORWARD -m comment --comment "cali:G8sjbYXH5\_QiYnBl" -j cali-to-host-endpoint

-A cali-FORWARD -m comment --comment "cali:wYFYRdMhtSYCqKNm" -m comment --comment "Host endpoint policy accepted packet." -m mark --mark 0x1000000/0x1000000 -j ACCEPT

规则4，报文是workload endpoint发出的，过对应endpoint的规则的egress规则。

规则5，报文要转发给本地的workload endpoint的，过对应endpoint的ingress规则。

规则6，规则7，默认允许转发。

规则9，报文是其它node发送过来的，过host endpoint的ingress规则。

规则10，报文要转发给host endpoint，过host endpoint的egress规则。

filter.cali-from-wl-dispatch，过对应endpoint的egress规则:

-A cali-from-wl-dispatch -i cali0ef24b1 -m comment --comment "cali:RkM6MKQgU0OTxwKU" -g cali-fw-cali0ef24b1

-A cali-from-wl-dispatch -i cali0ef24b2 -m comment --comment "cali:7hIahXYNmY9JDfKG" -g cali-fw-cali0ef24b2

-A cali-from-wl-dispatch -m comment --comment "cali:YKcphdGNZ1PwfGvt" -m comment --comment "Unknown interface" -j DROP

规则1, 过对应endpoint的inbound规则， fw表示from workload

filter.cali-to-wl-dispatch，过对应endpoint的ingress规则:

-A cali-to-wl-dispatch -o cali0ef24b1 -m comment --comment "cali:ofrbQ8PhcrIR6rgF" -g cali-tw-cali0ef24b1

-A cali-to-wl-dispatch -o cali0ef24b2 -m comment --comment "cali:l9Rs20XXIl4D5AVE" -g cali-tw-cali0ef24b2

-A cali-to-wl-dispatch -m comment --comment "cali:dxGyc\_mZA\_GT16Wb" -m comment --comment "Unknown interface" -j DROP

规则1，过对应endpoint的规则链，tw表示to workload

workload endpoint的outbound规则，在前面已经看过了，这里省略，只看inbound。

查看一个workload-endpoint的inbound规则，filter.cali-tw-cali0ef24b1

-A cali-tw-cali0ef24b1 -m comment --comment "cali:v-IVzQuOaLDTvlKQ" -j MARK --set-xmark 0x0/0x1000000

-A cali-tw-cali0ef24b1 -m comment --comment "cali:vE8JWROTKOuSK0cA" -m comment --comment "Start of policies" -j MARK --set-xmark 0x0/0x2000000

-A cali-tw-cali0ef24b1 -m comment --comment "cali:fVy5z1nXaCLhF0EQ" -m mark --mark 0x0/0x2000000 -j cali-pi-namespace-default

-A cali-tw-cali0ef24b1 -m comment --comment "cali:\_B9yiomhSoQTzhKL" -m comment --comment "Return if policy accepted" -m mark --mark 0x1000000/0x1000000 -j RETURN

-A cali-tw-cali0ef24b1 -m comment --comment "cali:uNPReN9\_BghUJj7S" -m comment --comment "Drop if no policies passed packet" -m mark --mark 0x0/0x2000000 -j DROP

首先过policy的ingress规则，然后过绑定的profile的ingress规则:

规则3: cali-pi-namespace-default，pi表示policy inbound。

filter.cali-pi-namespace-default，policy inbound规则:

-A cali-pi-namespace-default -m comment --comment "cali:K4jTheFcVvdYaw0q" -j DROP

-A cali-pi-namespace-default -m comment --comment "cali:VTQ78plyA8u\_8\_YC" -m set --match-set cali4-s:CEmFgJFwDvohR01JKvOkO8D src -j MARK --set-xmark 0x1000000/0x1000000

-A cali-pi-namespace-default -m comment --comment "cali:OAWI2ts9a8YpVP2b" -m mark --mark 0x1000000/0x1000000 -j RETURN

注意，规则1直接丢弃了报文，但是规则2又在设置标记，这是因为这里policy的egress规则设置是有问题的:

ingress:

- action: deny

- action: allow

source:

selector: namespace == 'default'

配置了两条ingress规则，第一条直接deny，第二条则是对指定的source设置为allwo。这样的规则配置是有问题的。

从上面的iptables规则中也可以看到，iptables规则是按照ingress中的规则顺序设定的。

如果第一条规则直接deny，那么后续的规则就不会发生作用了。

所以结果就是allow规则不生效。

salve1上的workload endpoint没有绑定profile，所有没有profile的inbound规则。

slave2上的endpoint设置了profile，允许访问TCP 3306端口，可以看到profile的inbound规则，filter.cali-tw-cali0ef24b3：

-A cali-tw-cali0ef24b3 -m comment --comment "cali:-l47AwgMbB6upZ-7" -j MARK --set-xmark 0x0/0x1000000

-A cali-tw-cali0ef24b3 -m comment --comment "cali:3qLl7L7-k49jf6Eu" -m comment --comment "Start of policies" -j MARK --set-xmark 0x0/0x2000000

-A cali-tw-cali0ef24b3 -m comment --comment "cali:Q6ycGZQm9W9l4KiJ" -m mark --mark 0x0/0x2000000 -j cali-pi-namespace-default

-A cali-tw-cali0ef24b3 -m comment --comment "cali:\_ILnIsDpaSEGOULc" -m comment --comment "Return if policy accepted" -m mark --mark 0x1000000/0x1000000 -j RETURN

-A cali-tw-cali0ef24b3 -m comment --comment "cali:CtKcOQPXG9FZiCN-" -m comment --comment "Drop if no policies passed packet" -m mark --mark 0x0/0x2000000 -j DROP

-A cali-tw-cali0ef24b3 -m comment --comment "cali:NR6mgOGAOw90NLpp" -j cali-pri-profile-database

-A cali-tw-cali0ef24b3 -m comment --comment "cali:\_OapaK4JADerp4Fv" -m comment --comment "Return if profile accepted" -m mark --mark 0x1000000/0x1000000 -j RETURN

-A cali-tw-cali0ef24b3 -m comment --comment "cali:ZVuAf3Bzin6dOKSX" -m comment --comment "Drop if no profiles matched" -j DROP

规则6，多出的profile inboud规则。

salve2上的profile的inbound规则，filter.cali-pri-profile-database:

-A cali-pri-profile-database -m comment --comment "cali:viAiQwvuZPt5-44a" -j DROP

-A cali-pri-profile-database -p tcp -m comment --comment "cali:Vcuflyj-wUF-f\_Mo" -m set --match-set cali4-s:i357Nlxxj3AMBTQ4WyOllNt src -m multiport --dports 3306 -j MARK --set-xmark 0x1000000/0x1000000

-A cali-pri-profile-database -m comment --comment "cali:JWP\_zDo3JNywNc0V" -m mark --mark 0x1000000/0x1000000 -j RETURN

同样也是因为profile的ingress第一条是deny的原因，规则1直接全部drop。

规则2，允许访问tcp 3306。

nat.POSTROUTING:

-A cali-POSTROUTING -m comment --comment "cali:Z-c7XtVd2Bq7s\_hA" -j cali-fip-snat

-A cali-POSTROUTING -m comment --comment "cali:nYKhEzDlr11Jccal" -j cali-nat-outgoing

这里没有设置fip，所以cali-fip-snat和cali-nat-outging都是空的

**node发送本地发出的报文**

OUTPUT@nat:

-A OUTPUT -m comment --comment "cali:tVnHkvAo15HuiPy0" -j cali-OUTPUT

-A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER

cali-OUTPUT@nat:

-A cali-OUTPUT -m comment --comment "cali:GBTAv2p5CwevEyJm" -j cali-fip-dnat

OUTPUT@filter:

-A OUTPUT -m comment --comment "cali:tVnHkvAo15HuiPy0" -j cali-OUTPUT

cali-OUTPUT@filter:

-A cali-OUTPUT -m comment --comment "cali:FwFFCT8uDthhfgS7" -m mark --mark 0x1000000/0x1000000 -m conntrack --ctstate UNTRACKED -j ACCEPT

-A cali-OUTPUT -m comment --comment "cali:KQN1p6BZgCGuApYk" -m conntrack --ctstate INVALID -j DROP

-A cali-OUTPUT -m comment --comment "cali:ThMSEAwgeF4nAqRa" -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A cali-OUTPUT -o cali+ -m comment --comment "cali:0YpIH4BWIJL90PfX" -j RETURN

-A cali-OUTPUT -m comment --comment "cali:sUIDpoFnawuqGYyG" -j MARK --set-xmark 0x0/0x7000000

-A cali-OUTPUT -m comment --comment "cali:vQVzNX-dNxUnYjUT" -j cali-to-host-endpoint

-A cali-OUTPUT -m comment --comment "cali:Ry2SAIVyda14xWHB" -m comment --comment "Host endpoint policy accepted packet." -m mark --mark 0x1000000/0x1000000 -j ACCEPT

规则4，如果是发送到cali网卡的，报文不出node，没有必要继续匹配了

规则6，过host-endpoint的outbond规则。

POSTROUTING@nat:

-A POSTROUTING -m comment --comment "cali:O3lYWMrLQYEMJtB5" -j cali-POSTROUTING

-A POSTROUTING -s 172.16.163.0/24 ! -o docker0 -j MASQUERADE

nat.cali-POSTROUTING:

-A cali-POSTROUTING -m comment --comment "cali:Z-c7XtVd2Bq7s\_hA" -j cali-fip-snat

-A cali-POSTROUTING -m comment --comment "cali:nYKhEzDlr11Jccal" -j cali-nat-outgoing