Hello,

I have a problem with my strongswan configuration. I want to use the VPN for my mobile/laptop. There I want to use forecast to route broadcast and multicast packets. For this I have to enable the mark=%unique option within the ipsec configuration. If I don't enable the mark option I have access to the Internet through the VPN connection. If I enable the mark-option I don't have internet access.

As a gateway I use the openwrt-19.07.02 image.

This is my configuration:

```
# cat /etc/ipsec.conf
config setup

conn %default
  ikelifetime=60m
  keylife=20m
  rekeymargin=3m
  keyingtries=1
  keyexchange=ikev2

conn rw-eap
  leftid=@[public FQDN]
  left=kerberos
  leftsubnet=0.0.0.0/0
  leftcert=kerberos-strongswan.crt
  leftauth=pubkey
  leftfirewall=yes
  lefthostaccess=yes
  rightauth=eap-radius
  rightsendcert=never
  right=%any
  rightsourceip=%dhcp
  rightsubnet=%dynamic,255.255.255.255
  rightdns=192.168.165.200
  righthostaccess=yes
  auto=add
  mark=%unique
```

Connection log on the gateway side:

```
# /etc/init.d/ipsec restart ; logread -f
Mon Mar 30 22:21:03 2020 authpriv.info ipsec_starter[30988]: charon stopped after 200 ms
Mon Mar 30 22:21:03 2020 authpriv.info ipsec_starter[30988]: ipsec starter stopped
Mon Mar 30 22:21:03 2020 authpriv.info ipsec_starter[31105]: Starting strongSwan 5.8.2 IPsec [starter]...
Mon Mar 30 22:21:03 2020 daemon.info : 00[NET] using forecast interface br-lan
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] joining forecast multicast groups: 224.0.0.1,224.0.0.251,224.0.0.252,239.255.255.250
```
Mon Mar 30 22:21:03 2020 daemon.info : 00[NET] forwarding multicast group 224.0.0.1
Mon Mar 30 22:21:03 2020 daemon.info : 00[NET] forwarding multicast group 224.0.0.22
Mon Mar 30 22:21:03 2020 daemon.info : 00[NET] forwarding multicast group 224.0.0.251
Mon Mar 30 22:21:03 2020 daemon.info : 00[NET] forwarding multicast group 224.0.0.252
Mon Mar 30 22:21:03 2020 daemon.info : 00[NET] forwarding multicast group 239.255.255.250
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loading ca certificates from '/etc/ipsec.d/cacerts'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loaded ca certificate "C=DE, ST=Berlin, L=Berlin, O=example, CN=example Root CA, E=user@example.org" from '/etc/ipsec.d/cacerts/ca.crt'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loading aa certificates from '/etc/ipsec.d/aacerts'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loading ocsp signer certificates from '/etc/ipsec.d/ocspcerts'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loading attribute certificates from '/etc/ipsec.d/aacerts'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loading crls from '/etc/ipsec.d/crls'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loading secrets from '/etc/ipsec.secrets'
Mon Mar 30 22:21:03 2020 daemon.info : 00[CFG] loaded RSA private key from '/etc/ipsec.d/private/example-kerberos-strongswan.key'
Mon Mar 30 22:21:03 2020 daemon.info : 00[JOB] spawning 16 worker threads
Mon Mar 30 22:21:03 2020 authpriv.info ipsec_starter[31105]: charon (31106) started after 60 ms
Mon Mar 30 22:21:03 2020 daemon.info : 05[CFG] loaded certificate "C=DE, ST=Berlin, L=Berlin, O=example, OU=example IPSec VPN, CN=example.com" from 'example-kerberos-strongswan.crt'
Mon Mar 30 22:21:05 2020 daemon.info : 07[IKE] [public IP Client] is initiating an IKE_SA
Mon Mar 30 22:21:05 2020 daemon.info : 07[IKE] local host is behind NAT, sending keep alives
Mon Mar 30 22:21:05 2020 daemon.info : 07[IKE] remote host is behind NAT
Mon Mar 30 22:21:05 2020 daemon.info : 11[IKE] received cert request for "C=DE, ST=Berlin, L=Berlin, O=example, CN=example Root CA, E=user@example.org"
Mon Mar 30 22:21:05 2020 daemon.info : 11[IKE] initiating EAP_PEAP method (id 0x01)
Mon Mar 30 22:21:05 2020 daemon.info : 11[IKE] sending end entity cert "C=DE, ST=Berlin, L=Berlin, O=example, OU=example IPSec VPN, CN=example.com"
Mon Mar 30 22:21:05 2020 daemon.info : 08[CFG] received RADIUS Access-Challenge from server 'localhost'

ConnectionLog on the client side:

Mär 30 22:20:30 setbook charon-nm[2388]: 05[CFG] received initiate for NetworkManager connection example VPN (IKEv2)
Mär 30 22:20:30 setbook charon-nm[2388]: 05[CFG] using CA certificate, gateway identity 'example.com'
Mär 30 22:20:31 setbook charon-nm[2388]: 16[IKE] local host is behind NAT, sending keep alives
Mär 30 22:20:31 setbook charon-nm[2388]: 16[IKE] remote host is behind NAT
Mär 30 22:20:31 setbook charon-nm[2388]: 16[IKE] sending cert request for "C=DE, ST=Berlin, L=Berlin, O=example, CN=example Root CA, E=user@example.com"
Mär 30 22:20:31 setbook charon-nm[2388]: 16[IKE] establishing CHILD_SA example VPN (IKEv2)
Mär 30 22:20:31 setbook charon-nm[2388]: 17[ENC] parsed IKE_AUTH response 1 [ EF(1/2) ]
Mär 30 22:20:31 setbook charon-nm[2388]: 17[ENC] received fragment #1 of 2, waiting for complete IKE message
Mär 30 22:20:31 setbook charon-nm[2388]: 06[ENC] parsed IKE_AUTH response 1 [ EF(2/2) ]
Mär 30 22:20:31 setbook charon-nm[2388]: 06[ENC] received fragment #2 of 2, reassembled fragmented IKE message (2256 bytes)
Mär 30 22:20:31 setbook charon-nm[2388]: 06[IKE] received end entity cert "C=DE, ST=Berlin, L=Berlin, O=example, OU=example IPSec VPN, CN=example.com"
Mär 30 22:20:31 setbook charon-nm[2388]: 06[CFG] using certificate "C=DE, ST=Berlin, L=Berlin, O=example, OU=example IPSec VPN, CN=example.com"
Mär 30 22:20:31 setbook charon-nm[2388]: 06[CFG] using trusted ca certificate "C=DE, ST=Berlin, L=Berlin, O=example, CN=example Root CA, E=user@example.com"
Mär 30 22:20:31 setbook charon-nm[2388]: 06[CFG] checking certificate status of "C=DE, ST=Berlin, L=Berlin, O=example, CN=example IPSec VPN, CN=example.com"
Mär 30 22:20:31 setbook charon-nm[2388]: 06[LIB] unable to fetch from http://example.com/rv.crl, no capable fetcher found
Mär 30 22:20:31 setbook charon-nm[2388]: 06[CFG] certificate status is not available
Mär 30 22:20:31 setbook charon-nm[2388]: 06[CFG] reached self-signed root ca with a path length of 0
Mär 30 22:20:31 setbook charon-nm[2388]: 06[IKE] authentication of 'example.com' with RSA EMSA PKCS1_SHA2_384 successful
Mär 30 22:20:31 setbook charon-nm[2388]: 06[IKE] server requested EAP_PEAP authentication (id 0x01)
Mär 30 22:20:31 setbook charon-nm[2388]: 06[IKE] EAP method not supported, sending EAP_NAK
Mär 30 22:20:31 setbook charon-nm[2388]: 07[IKE] server requested EAP_GTC authentication (id 0x02)
Mär 30 22:20:35 setbook charon-nm[2388]: 11[IKE] retransmit 1 of request with message ID 4
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] authentication of 'example.com' with EAP successful
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] IKE_SA example VPN (IKEv2)[4] established between [private IP Client][user]...[public IP Server][example.com]
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] scheduling rekeying in 35961s
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] maximum IKE_SA lifetime 36561s
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] CHILD_SA example VPN (IKEv2)(5) established with SPIs caab0754_i c64e30df_o and TS 192.168.165.41/32 --- 0.0.0.0/0
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] received AUTH_LIFETIME of 3398s, scheduling reauthentication in 2798s
Mär 30 22:20:35 setbook charon-nm[2388]: 01[IKE] peer supports MOBIKE

IPTables with connected client (Gateway side)

# iptables-save
# Generated by iptables-save v1.8.3 on Mon Mar 30 22:41:32 2020
*nat
:PREROUTING ACCEPT [3235:475643]
:INPUT ACCEPT [655:49668]
:OUTPUT ACCEPT [177:13971]
:POSTROUTING ACCEPT [184:14423]
:postrouting_VPN_rule - [0:0]
:postrouting_lan_rule - [0:0]
:postrouting_rule - [0:0]
:postrouting_wan_rule - [0:0]
:prerouting_VPN_rule - [0:0]
:prerouting_lan_rule - [0:0]
:prerouting_rule - [0:0]
:prerouting_wan_rule - [0:0]
:zone_VPN_post routing - [0:0]
:zone_VPN_prerouting - [0:0]
:zone_lan_post routing - [0:0]
:zone_lan_prerouting - [0:0]
:zone_wan_post routing - [0:0]
:zone_wan_prerouting - [0:0]
-A PREROUTING -m comment --comment "!fw3: Custom prerouting rule chain" -j prerouting_rule
-A PREROUTING -i br-lan -m comment --comment "!fw3" -j zone_lan_prerouting
-A PREROUTING -i tun0 -m comment --comment "!fw3" -j zone_wan_prerouting
-A POSTROUTING -m comment --comment "!fw3: Custom postrouting rule chain" -j postrouting_rule
-A POSTROUTING -m policy --dir out --pol ipsec -m comment --comment "!fw3: IPSec Forward" -j ACCEPT
-A POSTROUTING -o br-lan -m comment --comment "!fw3" -j zone_lan_post routing
-A POSTROUTING -o tun0 -m comment --comment "!fw3" -j zone_wan_post routing
-A zone_VPN_post routing -m comment --comment "!fw3: Custom VPN post routing rule chain" -j post routing_VPN_rule
-A zone_VPN_prerouting -m comment --comment "!fw3: Custom VPN prerouting rule chain" -j prerouting _VPN_rule
-A zone_lan_post routing -m comment --comment "!fw3: Custom lan post routing rule chain" -j post routing_lan_rule
-A zone_lan_prerouting -m comment --comment "!fw3: Custom lan prerouting rule chain" -j prerouting _lan_rule
-A zone_wan_post routing -m comment --comment "!fw3: Custom wan post routing rule chain" -j post routing_wan_rule
-A zone_wan_prerouting -m comment --comment "!fw3" -j MASQUERADE
-A zone_wan_post routing -m comment --comment "!fw3: Custom wan post routing rule chain" -j post routing _wan_rule

08.05.2020 6/19
# Completed on Mon Mar 30 22:41:32 2020
# Generated by iptables-save v1.8.3 on Mon Mar 30 22:41:32 2020

*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:FORWARDING_VPN_rule - [0:0]
:FORWARDING_lan_rule - [0:0]
:FORWARDING_rule - [0:0]
:FORWARDING_wan_rule - [0:0]
:input_VPN_rule - [0:0]
:input_lan_rule - [0:0]
:input_rule - [0:0]
:output_VPN_rule - [0:0]
:output_lan_rule - [0:0]
:output_rule - [0:0]
:output_wan_rule - [0:0]
:reject - [0:0]
:syn_flood - [0:0]
:zone_VPN_dest_ACCEPT - [0:0]
:zone_VPN_dest_REJECT - [0:0]
:zone_VPN_forward - [0:0]
:zone_VPN_input - [0:0]
:zone_VPN_output - [0:0]
:zone_VPN SRC_ACCEPT - [0:0]
:zone_lan_dest_ACCEPT - [0:0]
:zone_lan_forward - [0:0]
:zone_lan_input - [0:0]
:zone_lan_output - [0:0]
:zone_lan_SRC_DROP - [0:0]
:zone_wan_dest_ACCEPT - [0:0]
:zone_wan_dest_DROP - [0:0]
:zone_wan_dest_REJECT - [0:0]
:zone_wan_forward - [0:0]
:zone_wan_input - [0:0]
:zone_wan_OUTPUT - [0:0]
:zone_wan_SRC_REJECT - [0:0]
-A INPUT -s 255.255.255.255/32 -i br-lan -m policy --dir in --pol ipsec --reqid 2 --proto esp -j ACCEPT
-A INPUT -s 192.168.165.47/32 -i br-lan -m policy --dir in --pol ipsec --reqid 2 --proto esp -j ACCEPT
-A INPUT -l o -m comment --comment "!fw3" -j ACCEPT
-A INPUT -m comment --comment "!fw3: Custom input rule chain" -j input_rule
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -m comment --comment "!fw3" -j ACCEPT
-A INPUT -m conntrack --ctstate INVALID -m comment --comment "!fw3" -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -m comment --comment "!fw3" -j syn_flood

08.05.2020
# Firewall Configuration

- Accept HTTP connections to local network
- Accept HTTPS connections to local network
- Accept DHCP renew requests
- Accept DHCP renew requests
- Accept LDAP connections to local network
- Accept DNS requests to local network
- Accept NTP connections to local network
- Accept RADIUS auth requests to local network
- Accept RADIUS acc requests to local network
- Accept IPSEC connections
- Accept port redirections
- Accept RADIUS acct requests to local network
- Accept DNS requests to local network
- Accept NTP connections to local network

# Ping Test

Ping on the client side

```
# ping -b 255.255.255.255
WARNING: pinging broadcast address
PING 255.255.255.255 (255.255.255.255) 56(84) bytes of data.
        ^C
--- 255.255.255.255 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 130ms
```

```
setbook ~ # ping devil (This is a server within the local network, not the Gateway!)
PING devil (192.168.165.201) 56(84) bytes of data.
^C
--- devil ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 8ms
```

```
setbook ~ # ping google.de
PING google.de (172.217.23.3) 56(84) bytes of data.
^C
```

# Completed on Mon Mar 30 22:41:32 2020
--- google.de ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 140ms

TCPDump while pinging on the Gateway

ping broadcast
-----------------------
22:32:50.270461 IP 192.168.165.41 > 255.255.255.255: ICMP echo request, id 20, seq 1, length 64
22:32:50.270750 IP 192.168.165.41 > 255.255.255.255: ICMP echo request, id 20, seq 1, length 64
22:32:50.298506 IP 192.168.165.5 > 192.168.165.41: ICMP echo reply, id 20, seq 1, length 64
22:32:51.225676 IP 192.168.165.41 > 255.255.255.255: ICMP echo request, id 20, seq 2, length 64
22:32:51.225930 IP 192.168.165.5 > 192.168.165.41: ICMP echo reply, id 20, seq 2, length 64
22:32:52.260087 IP 192.168.165.41 > 255.255.255.255: ICMP echo request, id 20, seq 3, length 64
22:32:52.260995 IP 192.168.165.5 > 192.168.165.41: ICMP echo reply, id 20, seq 3, length 64
22:32:53.298671 IP 192.168.165.41 > 255.255.255.255: ICMP echo request, id 20, seq 4, length 64
22:32:53.301660 IP 192.168.165.5 > 192.168.165.41: ICMP echo reply, id 20, seq 4, length 64

ping devil
----------
22:32:57.104751 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 1, length 64
22:32:57.104841 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 1, length 64
22:32:57.105161 IP 192.168.165.201 > 192.168.165.41: ICMP echo reply, id 21, seq 1, length 64
22:32:58.157595 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 2, length 64
22:32:58.157701 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 2, length 64
22:32:58.157951 IP 192.168.165.201 > 192.168.165.41: ICMP echo reply, id 21, seq 2, length 64
22:32:59.157914 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 3, length 64
22:32:59.158010 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 3, length 64
22:32:59.158275 IP 192.168.165.201 > 192.168.165.41: ICMP echo reply, id 21, seq 3, length 64
22:33:00.157329 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 4, length 64
22:33:00.157425 IP 192.168.165.41 > 192.168.165.201: ICMP echo request, id 21, seq 4, length 64
22:33:00.157719 IP 192.168.165.201 > 192.168.165.41: ICMP echo reply, id 21, seq 4, length 64

ping google.de
-----------------------
22:33:03.377973 IP 192.168.165.41 > 172.217.23.3: ICMP echo request, id 22, seq 1, length 64
22:33:03.417528 IP 172.217.23.3 > 192.168.165.41: ICMP echo reply, id 22, seq 1, length 64
22:33:04.419084 IP 192.168.165.41 > 172.217.23.3: ICMP echo request, id 22, seq 2, length 64
22:33:04.461666 IP 172.217.23.3 > 192.168.165.41: ICMP echo reply, id 22, seq 2, length 64
22:33:05.457383 IP 192.168.165.41 > 172.217.23.3: ICMP echo request, id 22, seq 3, length 64
22:33:05.497772 IP 172.217.23.3 > 192.168.165.41: ICMP echo reply, id 22, seq 3, length 64
22:33:06.497773 IP 192.168.165.41 > 172.217.23.3: ICMP echo request, id 22, seq 4, length 64
22:33:06.537516 IP 172.217.23.3 > 192.168.165.41: ICMP echo reply, id 22, seq 4, length 64

log while pinging

Mon Mar 30 22:32:50 2020 daemon.info : 02[NET] forwarding a 255.255.255.255 broadcast from peer 192.168.165.41 to internal network
Mon Mar 30 22:32:51 2020 daemon.info : 05[NET] forwarding a 255.255.255.255 broadcast from peer 192.168.165.41 to internal network
Mon Mar 30 22:32:52 2020 daemon.info : 07[NET] forwarding a 255.255.255.255 broadcast from peer 19.2.168.165.41 to internal network


History

#1 - 01.04.2020 18:05 - Tobias Brunner
- Category set to configuration
- Status changed from New to Feedback

Is the output of iptables-save from the same session? The IP address in the rules there looks wrong (192.168.165.47 instead of 192.168.165.41).

Anyway, it's definitely possible that all these rules conflict somehow. The rule in PREROUTING should mark packets addressed to the virtual IP so they match the IPsec policy. So check if that rule is applied and whether the out policy is used at all and ESP packets sent (ip -s xfrm policy|state or ipsec statusall to some degree). If available you can also check /proc/net/xfrm_stat for errors.


You probably want to avoid that by disabling sending ICMP redirects (see ForwardingAndSplitTunneling).

Also, Linux won't respond to multi-/broadcasted ICMP echo requests unless you explicitly enable that via sysctl -w net.ipv4.icmp_echo_ignore_broadcasts = 0.

#2 - 01.04.2020 20:52 - Sebastian Koschmieder
- File iptables.save added
- File xfrm.policy added
- File xfrm.state added
- File ip.route added

Hello Tobias,

thanks for our answer.

Is the output of iptables-save from the same session? The IP address in the rules there looks wrong (192.168.165.47 instead of 192.168.165.41).

If I remember correctly it was the same session. But I could be wrong.

So I've connected again and saved the outputs. Also the firewall rules.

But could this be a mistake? (The second line?)

-A OUTPUT -d 192.168.165.35/32 -j MARK --set-xmark 0x2/0xffffffff
-A OUTPUT -p esp -j MARK --set-xmark 0x55/0xffffffff

I use a VPN connection for my complete network via openvpn.

To access the ipsec connection from the internet I mark the ESP packets with the 0x55 and route them via a novpn routing table. Does the second line overwrite the mark set by the first line?

# grep novpn /etc/iproute2/rt_tables
85 novpn
# ip route list table novpn
default via 192.168.165.1 dev br-lan proto static

Thanks for the advice with the redirects. ^^

The broadcasts I used to see if they are correctly transmitted into my lan.

#3 - 02.04.2020 11:19 - Tobias Brunner
If I remember correctly it was the same session. But I could be wrong. So I’ve connected again and saved the outputs. Also the firewall rules.

OK, now at least the IP address matches. What about /proc/net/xfrm_stat? Also, the output of ip rule would be helpful if you use multiple tables.

To access the ipsec connection from the internet I mark the ESP packets with the 0x55 and route them via a novpn routing table. Does the second line overwrite the mark set by the first line?

It definitely does change the mark, but only after encryption with ESP and encapsulation with UDP (i.e., after the policy match for which the other mark is important), so this should not have any negative effects. But you should perhaps still check if the packets don’t end up in the OpenVPN tunnel (or if they really are dropped by the gateway).

Maybe try to debug the rules e.g., by sampling iptables-save -c while you ping or via TRACE target (the latter can be done quite targeted e.g., only affecting return packets from a specific IP address).

#4 - 02.04.2020 21:06 - Sebastian Koschmieder
- File iptables.save added
- File iptables.trace added
- File ip.routeall added

Hello,

oh, I’m sorry. The file /proc/net/xfrm_stat does not exist on my openwrt gateway. But I have an iptable trace while I ping the google-server. It is attached.

I think, there is no routing decision. Is it correct?

Here is an ip rule list

```
# ip rule list
0: from all lookup local
1: from all fwmark 0x55 lookup novpn
220: from all lookup 220
32766: from all lookup main
32767: from all lookup default
```

and an ip route list table all is also attached...

#5 - 03.04.2020 09:03 - Tobias Brunner

But I have an iptable trace while I ping the google-server.

Thanks. Looks like there is some kind of NAT that causes this problem. When the return packet goes through PREROUTING in the mangle table the destination address is 10.128.204.144, so that won’t match the rule installed by the forecast plugin (-d 192.168.165.48/32), which means the mark required to match the IPsec policies is never applied. Interestingly, in the trace there is no entry that maps the virtual IP address to that address in the first place (we never see SRC=10.128.204.144), and there is no entry that shows it mapping back either, yet the last four lines show 192.168.165.48 as destination (but as you can see without mark). So where does that IP address come from?

#6 - 03.04.2020 09:12 - Sebastian Koschmieder

This is the IP Address of the openvpn device.

There is a nat MASQUERADE for this interface.

```
-t nat -A zone_wan_postrout -m comment --comment "fw3" -j MASQUERADE
```

This is to route the internet traffic through the OpenVPN connection.

But here after natting, within the mangle:FORWARD we have the correct dst address:

```
Thu Apr 2 20:37:49 2020 kern.warn kernel: [293393.828289] TRACE: mangle:FORWARD:rule:2 IN=tun0 OUT=br-lan MAC = SRC=172.217.22.99 DST=192.168.165.48 LEN=84 TOS=0x00 PREC=0x00 TTL=53 ID=0 PROTO=ICMP TYPE=0 CODE=0 ID=48 SE Q=1
```

But even if I insert there the mark it is nor routed to the client...
iptables -t mangle -I FORWARD -d 192.168.165.48 -j MARK --set-xmark 0x1

#7 - 03.04.2020 10:57 - Tobias Brunner

-t nat -A zone_wan_postrouting -m comment --comment "fw3" -j MASQUERADE
This is to route the internet traffic through the OpenVPN connection.

So IPsec traffic goes into the OpenVPN tunnel after decryption? (Why is that NAT mapping not seen in the trace, though?)

But here after natting, within the mangle:FORWARD we have the correct dst address:

Yes, but as you can see, no mark.

But even if I insert there the mark it is nor routed to the client...

Is the mark correct (it changes with each connection)? Check the trace/counters, is the rule applied?

#8 - 03.04.2020 13:00 - Sebastian Koschmieder

Tobias Brunner wrote:

-t nat -A zone_wan_postrouting -m comment --comment "fw3" -j MASQUERADE
This is to route the internet traffic through the OpenVPN connection.

So IPsec traffic goes into the OpenVPN tunnel after decryption? (Why is that NAT mapping not seen in the trace, though?)

Yes. That's correct.

For the trace I've added the trace target in the raw table, the PREROUTE and the OUTPUT chain with the IP-Address of the google server.
If there is a NAT rule, I should be traced...
From iptables-save posted before

-A PREROUTING -s 172.217.22.99/32 -j TRACE
-A PREROUTING -d 172.217.22.99/32 -j TRACE
-A PREROUTING -i br-lan -m comment --comment "$fw3: lan CT helper assignment" -j zone_lan_helper
-A OUTPUT -s 172.217.22.99/32 -j TRACE
-A OUTPUT -d 172.217.22.99/32 -j TRACE

But here after natting, within the mangle:FORWARD we have the correct dst address:

Yes, but as you can see, no mark.

But even if I insert there the mark it is nor routed to the client...

Is the mark correct (it changes with each connection)? Check the trace/counters, is the rule applied?

I've checked this. The rule was applied.
In the evening I'll test it again and post the logs for this.

#9 - 03.04.2020 20:32 - Sebastian Koschmieder

- File iptables.trace added
- File iptables.save added

I've tried it again, with no success...
The trace and the iptable-save are attached.

At this line you can see, there is a mark

Fri Apr  3 20:29:24 2020 kern.warn kernel: [379288.514966] TRACE: mangle:FORWARD:rule:3 IN=tun0 OUT=br-lan MAC = SRC=172.217.16.195 DST=192.168.165.34 LEN=84 TOS=0x00 PREC=0x00 TTL=53 ID=0 PROTO=ICMP TYPE=0 CODE=0 ID=56 SEQ=1 MARK=0x01

08.05.2020
For this mark I inserted the rule

```
iptables -t mangle -I FORWARD -d 192.168.165.34/32 --set-xmark 0x1/0xffffffff
```

On the client side, I get no answer from google.

### #10 - 04.04.2020 20:28 - Sebastian Koschmieder

- File `iptables.trace` added
- File `iptables.save` added
- File `xfrm.policy` added
- File `xfrm.state` added

Sebastian Koschmieder wrote:

> If I don’t enable the mark option I have access to the Internet through the VPN connection. If I enable the mark-option I don’t have internet access.

I've take a step back...

I comment the mark=%unique option and ping the server with tracing the connection.

The files are attached.

```
filter:FORWARD:rule:6 IN=tun0 OUT=tun0 MAC= SRC=172.217.16.195 DST=192.168.165.35 LEN=84 TOS=0x00 PREC=0x00 TTL=53 ID=0 PROTO=ICMP TYPE=0 CODE=0 ID=59 SEQ=1
```

```
mangle:OUTPUT:rule:3 IN= OUT=tun0 SRC=192.168.165.200 DST=109.41.1.139 LEN=164 TOS=0x00 PREC=0x00 TTL=64 ID=7600 PROTO=UDP SPT=4500 DPT=29152 LEN=144
```

What happens between these two rules?

It couldn’t be the routing...

Is it a policy?

I think this policy:

```
src 0.0.0.0/0 dst 192.168.165.35/32
dir out priority 383615
tmpl src 192.168.165.200 dst 109.41.1.139
proto esp spi 0xc5f66fc8 reqid 1 mode tunnel
```

With mark=%unique the policy looks a little bit different.

```
src 0.0.0.0/0 dst 192.168.165.35/32 uid 0
dir out action allow index 16545 priority 383615 share any flag (0x00000000)
lifetime config:
limit: soft (INF)(bytes), hard (INF)(bytes)
limit: soft (INF)(packets), hard (INF)(packets)
expire add: soft 0(sec), hard 0(sec)
expire use: soft 0(sec), hard 0(sec)
lifetime current:
0(bytes), 0(packets)
add 2020-04-01 20:07:03 use 2020-04-01 20:08:27
mark 0x2/0xffffffff
tmpl src 192.168.165.200 dst [public Client IP]
proto esp spi 0x21c40a66(566495846) reqid 2(0x00000002) mode tunnel
level required share any
enc-mask fffffff auth-mask fffffff comp-mask ffffffff
```

### #11 - 06.04.2020 10:27 - Tobias Brunner

On the client side, I get no answer from google.

I guess no ESP is sent to the client?

I think I tried it once and marking in FORWARD generally should work. But perhaps you could try to mark in PREROUTING already.

What happens between these two rules?
It couldn't be the routing...
Is it a policy?

Yes. However, if you compare the traces, the routing seems to differ too as the OUT interface is different (without mark it's tun0, with br-lan).

With mark=%unique the policy looks a little bit different.

Besides -s, the only difference is the mark (i.e. mark 0x2/0xffffffff).

#12 - 06.04.2020 19:36 - Sebastian Koschmieder
I'm sorry, but how do I have to check if there was a ESP packet send to the client?
Also with IPTables only client side?
In the trace of their server I don't see a ESP packet.

I think I tried it once and marking in FORWARD generally should work. But perhaps you could try to mark in PREROUTING already.

in the mangle list?

#13 - 07.04.2020 09:32 - Tobias Brunner

I'm sorry, but how do I have to check if there was a ESP packet send to the client?

Packet counter on the SA (also in the status output of strongSwan) or tcpdump on server or client.

In the trace of their server I don't see a ESP packet.

You'd see UDP-encapsulated ESP, not ESP directly. But it's definitely possible there isn't one. Or perhaps tcpdump didn't see them e.g. because they went out a different interface, or they got filtered for some reason.

I think I tried it once and marking in FORWARD generally should work. But perhaps you could try to mark in PREROUTING already.

in the mangle list?

I guess.

#14 - 07.04.2020 21:27 - Sebastian Koschmieder

- File iptables.save added
- File iptables.trace added

Hello,

First:
I think there is no packet ESP packet returned to the client.
This is a tcpdump on the client while ping.

# tcpdump -n
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlp4s0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:14:21.889387 IP 192.168.43.228.45651 > [publicServerIP].4500: UDP-encap: ESP(spi=0xc7d2e232,seq=0x7), length 136

Second:

I think I tried it once and marking in FORWARD generally should work. But perhaps you could try to mark in PREROUTING already.

in the mangle list?
Attached are the iptables-save and the trace.
I added # iptables -t mangle -I PREROUTING -s 216.58.215.67 -j MARK --set-xmark=0x3 and I got the pong on the client.
But the packed was identified by the IP address of the server I pinged...

#15 - 07.04.2020 23:08 - Alexander Sukhomlinov
I have the same situation, but without OpenVPN setup, everything the same,

if i use

    mark=%unique

with forecast, clients can reach only subnets (LAN) behind home gateway (router), but no Internet, rather that without marking and forecast settings in left/right subnet, which working as expected...

I also don't see counting return packets that should mark in mangle table PREROUTING...

If i find the solution, i will post.

#16 - 08.04.2020 13:17 - Tobias Brunner

I added # iptables -t mangle -I PREROUTING -s 216.58.215.67 -j MARK --set-xmark=0x3 and I got the pong on the client.
But the packed was identified by the IP address of the server I pinged...

OK, so marking in PREROUTING would work. I wonder if you could use the CONNMARK target for this, that is, save the mark (--save-mark) for forwarded packets from IPsec tunnels (e.g. via policy module) and then restore the mark for the return packets again in PREROUTING (--restore-mark).

#17 - 08.04.2020 14:35 - Sebastian Koschmieder

Hello,

I create the following rules

```
iptables -t mangle -I FORWARD ! -d 192.168.0.0/16 -m policy --dir in --pol ipsec --proto esp -j CONNMARK --save-mark
iptables -t mangle -I FORWARD -i tun0 -j CONNMARK --restore-mark
```

With this I have internet access and i get the broad- and multicasts.
Thank you! ;)

#18 - 08.04.2020 23:38 - Alexander Sukhomlinov

Sebastian Koschmieder wrote:

```
Hello,

I create the following rules

[...]

With this I have internet access and i get the broad- and multicasts.
Thank you! ;)
```

I don't quite understand these rules ¬\`

In my setup, i don't have second tunnel as you (OpenVPN), but just "clean" ISP. I don't know where i should save/restore marks

Rules like:

```
iptables -t mangle -I FORWARD -m policy --dir in --pol ipsec -j CONNMARK --save-mark
```

OR

```
iptables -t mangle -I FORWARD -m policy --dir out--pol ipsec -j CONNMARK --save-mark
```
iptables -t mangle -I PREROUTING -j CONNMARK --restore-mark

Seems doesn't work, --restore-mark rules always counting bytes, but --save-mark newer!

#19 - 09.04.2020 00:16 - Alexander Sukhomlinov

Ok, root of the problem is return packet that is not yet DNAT-ed back after SNAT-ing (MASQUERADING). When return packet comes back, it goes to mangle PREROUTING, and there destination address is still MASQUERADED, that's why rule installed by %unique is not work! Return packet becomes Original destination address (Virtual IP address given by gateway for roadwarrior) after going through the table nat PREROUTING, and we see it later in mangle FORWARD or POSTROUTING.

For me it looks like (mangle):

PREROUTING:
IN=eth0.2 OUT= MAC=00:00:00:00:00:01:4c:f2:bf:2c:85:00:08:00:45:20:00:3c SRC=87.250.250.242 DST=192.168.101.2 LEN=60 TOS=0x00 PREC=0x20 TTL=51 ID=44554 PROTO=TCP SPT=443 DPT=46814 WINDOW=43338 RES=0x00 ACK SYN URG=0

FORWARD:
IN=eth0.2 OUT=eth0.2 MAC=00:00:00:00:00:01:4c:f2:bf:2c:85:00:08:00:45:20:00:3c SRC=87.250.250.242 DST=192.168.100.129 LEN=60 TOS=0x00 PREC=0x20 TTL=50 ID=44554 PROTO=TCP SPT=443 DPT=46814 WINDOW=43338 RES=0x00 ACK SYN URG=0

POSTROUTING:
IN= OUT=eth0.2 SRC=87.250.250.242 DST=192.168.100.129 LEN=60 TOS=0x00 PREC=0x20 TTL=50 ID=44554 PROTO=TCP SPT=443 DPT=46814 WINDOW=43338 RES=0x00 ACK SYN URG=0 MARK=0x1

Where
87.250.250.242 - Internet address of some site,
192.168.101.2 - WAN address (MASQUERADE outgoing to Internet),
192.168.100.129 - Original address (Virtual IP address given by gateway for roadwarrior)

Now i don't understand why the rule manually installed (which we see mark MARK=0x1 above):

iptables -t mangle -I POSTROUTING -d 192.168.100.129 -j MARK --set-mark 1
doesn't let packet routed back.. dah... =(

#20 - 09.04.2020 10:55 - Tobias Brunner

With this I have internet access and i get the broad- and multicasts.

That's great!

Now i don't understand why the rule manually installed (which we see mark MARK=0x1 above):

[...]
doesn't let packet routed back.. dah... =(

POSTROUTING is too late, IPsec policies are matched way earlier (i.e. the mark has to be set/restored in PREROUTING, as seen in Sebastian's scenario, and it's also what the rules installed by the plugin try to do).

#21 - 09.04.2020 13:56 - Alexander Sukhomlinov

Tobias Brunner wrote:

With this I have internet access and i get the broad- and multicasts.

That's great!

Now i don't understand why the rule manually installed (which we see mark MARK=0x1 above):

[...]
doesn't let packet routed back.. dah... =(

POSTROUTING is too late, IPsec policies are matched way earlier (i.e. the mark has to be set/restored in PREROUTING, as seen in Sebastian's scenario, and it's also what the rules installed by the plugin try to do).
The cut from iptables-save (mangle):

- A PREROUTING -s 192.168.100.129/32 -j CONNMARK --save-mark
- A PREROUTING -j CONNMARK --restore-mark
- A PREROUTING -d 192.168.100.129/32 -j MARK --set-xmark 0x1
- A PREROUTING -s 2.73.21.214/32 -d 192.168.101.2/32 -p udp -m udp --sport 44257 --dport 4500 -j MARK --set-xmark 0x1
- A OUTPUT -d 192.168.100.129/32 -j MARK --set-xmark 0x1

I did next:

Put

```
PREROUTING -s 192.168.100.129/32 -j CONNMARK --save-mark

near old
```

- A PREROUTING -d 192.168.100.129/32 -j MARK --set-xmark 0x1

and after "save mark"

```
PREROUTING -j CONNMARK --restore-mark
```

Since we’re saving mark for Subnet/Virtual IP from roadwarriors and restoring everything later, I think it will never broke other situations in any setup.

There is a little overhead of rule -j CONNMARK --restore-mark because it restores everything, but we cannot put there any -m policy or src/dst addresses

#22 - 14.04.2020 22:05 - Alexander Sukhomlinov

Ok! To generalize this, I would say:

The Rules

```
-I PREROUTING -m connmark --mark <unique mark> -j CONNMARK --restore-mark
-I PREROUTING -s <Virtual IP/Subnet given for roadwarriors> -j CONNMARK --save-mark
```

Would better next to

```
-I PREROUTING -d <Virtual IP/Subnet given for roadwarriors> -j MARK --set-mark <unique mark>
```

What do you think, Tobias Brunner?

It would be awesome to see such small changes in next release, for plugin.

#23 - 16.04.2020 11:39 - Tobias Brunner

The Rules

```
[…]
```

Do these actually work? You match packets with the mark to restore the mark? Shouldn’t it be the other way around?

The problem is that the CONNMARK target is provided by an optional kernel/netfilter module, plus such rules are only necessary in certain situations (I guess a NAT on the gateway is one of them). And depending on the situation the rules might have to be adapted too. So I’d rather just document it on the plugin's wiki page for now.

#24 - 16.04.2020 19:19 - Alexander Sukhomlinov

Tobias Brunner wrote:

```
The Rules

[…]
```

Do these actually work? You match packets with the mark to restore the mark? Shouldn’t it be the other way around?

The problem is that the CONNMARK target is provided by an optional kernel/netfilter module, plus such rules are only necessary in certain situations (I guess a NAT on the gateway is one of them). And depending on the situation the rules might have to be adapted too. So I’d rather just document it on the plugin's wiki page for now.
Yes, only these two work for most common situations (with/without NAT, and others), we have already marked packets when they come as ESP|UDP by rule (plugin) -A PREROUTING -s <src> -d <dst> <match esp or udp> <--dport 4500> -j MARK --set-mark <unique> <-- this mark also saves for decrypted/de-encapsulated packets later. So, we just save that mark for connection -I PREROUTING -s <Virtual IP/Subnet given for roadwarriors> -j CONNMARK --save-mark and later (return packets), we match by connmark (coz we saved before!), and restore back from connection to packets -m connmark --mark <unique mark> -j CONNMARK --restore-mark

I don't think it could break something, because we actually just save mark for given <Virtual IP/Subnet given for roadwarriors>

In PREROUTING, we cannot use src|dst addresses or policy match to capture return packets, or something else, nothing works, only CONNMARK works properly... (CONNMARK and MARK are different, we cannot mark packet with MARK and later match with CONNMARK and vise versa)

Yes, i know that the CONNMARK target is provided by an optional kernel/netfilter module, but for plugins like connmark plugin provided by strongswan have the same requirement.

I do not force to include those rules inside the code, but i think it's more generalized solution.

I tested it for my roadwarriors (iOS/Android/Windows/etc.), they connect to my Gateway(NAT-ed), they have internet, local home-access to home-services, including Gateway (Web page, ssh, etc.) itself, networks behind, multicasts, broadcasts and so on...

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>iptables.save</td>
<td>11.4 KB</td>
<td>01.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>xfrm.policy</td>
<td>7.76 KB</td>
<td>01.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>xfrm.state</td>
<td>1.83 KB</td>
<td>01.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>ip.route</td>
<td>2.89 KB</td>
<td>01.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.save</td>
<td>10.4 KB</td>
<td>02.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.trace</td>
<td>6.25 KB</td>
<td>02.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>ip.routeall</td>
<td>2.83 KB</td>
<td>02.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.trace</td>
<td>6.55 KB</td>
<td>03.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.save</td>
<td>10.4 KB</td>
<td>03.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.save</td>
<td>10 KB</td>
<td>04.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.trace</td>
<td>4.92 KB</td>
<td>04.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>xfrm.policy</td>
<td>1.44 KB</td>
<td>04.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>xfrm.state</td>
<td>978 Bytes</td>
<td>04.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.save</td>
<td>10.3 KB</td>
<td>07.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
<tr>
<td>iptables.trace</td>
<td>7.63 KB</td>
<td>07.04.2020</td>
<td>Sebastian Koschmieder</td>
</tr>
</tbody>
</table>