When iptables != iptables!

St. Louis Unix Users Group
13 November 2024

Lee Lammert

OMNITEC Corporation

The problem

- LE sent alert 10/1, .. cert expires 10/30!
- Renewal fails, ..
  - Debian 10
  - iptables shows port open rule, netstat shows port listening, certbot pre & post hooks appear correct
  - Updating to Debian 11 changed nothing
- Current system, ..

certbot pre & post

```
# renew_before_expiry = 30 days
version = 1.12.0
...
# Options used in the renewal process
[renewalparams]
authenticator = standalone
rsa_key_size = 4096
server = https://acme-v02.api.letsencrypt.org/directory
account = 2dfa79b8024ac9c1f400cbd43f6756ff
pre_hook = /usr/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT
post_hook = /usr/sbin/iptables -D INPUT -p tcp --dport 80 -j ACCEPT
```

netstat

```
# netstat -lntp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address     Foreign Address   State   PID/Program name

tcp        0      0 0.0.0.0:10025     0.0.0.0:*         LISTEN      11165/master
tcp        0      0 127.0.0.1:783     0.0.0.0:*         LISTEN       6620/spamd child
tcp        0      0 0.0.0.0:25        0.0.0.0:*         LISTEN      9190/smtpd
tcp        0      0 0.0.0.0:2206      0.0.0.0:*         LISTEN      5193/sshd: /usr/sbi
tcp6       0      0 :::10025          :::*              LISTEN      11165/master
tcp6       0      0 :::80             :::*              LISTEN      9255/python3
tcp6       0      0 :::25             :::*              LISTEN      9190/smtpd
tcp6       0      0 :::2206           :::*              LISTEN      5193/sshd: /usr/sbi
```

iptables

```
# iptables -vnL
# Warning: iptables-legacy tables present, use iptables-legacy to see them
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target  prot opt in  out source      destination
 1728 80654 ACCEPT  tcp  --  *    *    0.0.0.0/0  0.0.0.0/0    tcp dpt:80
```

iptables-save

```
# iptables-save
# Generated by iptables-save v1.8.7 on Sat Oct 26 14:26:11 2024
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
COMMIT
# Completed on Sat Oct 26 14:26:11 2024
```

So, .. what’s happening?

- certbot pre/post look OK!
- netstat shows port open!
- iptables shows ACCEPT rule for 80!
- Iptables-save confirms!

Verify – no packets to 80!

```
# tcpdump -nni eth0 port 80

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:24:31.603683 IP 206.197.251.202.51958 > 23.64.114.213.80: Flags [S], seq
1124559809, win 29200, options [mss 1460,sackOK,TS val 1516055522 ecr
0,nop,wscale 7], length 0
14:24:31.612908 IP 23.64.114.213.80 > 206.197.251.202.51958: Flags [S.], seq
884228395, ack 1124559810, win 65160, options [mss 1460,sackOK,TS val
3212205822 ecr 1516055522,
nop,wscale 7], length 0
```

Recall for a moment systemd?

- Are there other ways to manage rules?
- iptables is actually a symlink to nftables!
- But, netfilter is also available, managed with iptables-legacy!

In reality:

```
# which iptables
iptables                iptables-legacy-restore   iptables-nft-restore      iptables-restore-
translate  iptables-xml           iptables-apply            iptables-legacy-save      iptables-
nft-save          iptables-save
iptables-legacy            iptables-nft              iptables-restore          iptables-translate

# ls -l /usr/sbin/iptables
lrwxrwxrwx 1 root root 26 Aug  9 16:57 /usr/sbin/iptables -> /etc/alternatives/iptables

# ls -l /etc/alternatives/iptables
lrwxrwxrwx 1 root root 22 Aug  9 16:57 /etc/alternatives/iptables -> /usr/sbin/iptables-nft
```

iptables-legacy

```
# iptables-legacy-save
# Generated by iptables-save v1.8.7 on Sat Oct 26 14:27:48 2024
*filter
:INPUT DROP [1263584:63598901]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [3789060:51432263927]
:f2b-sshd - [0:0]
-A INPUT -p tcp -m multiport --dports 2206 -j f2b-sshd
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 206.197.251.0/24 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A f2b-sshd -j RETURN
COMMIT
# Completed on Sat Oct 26 14:27:48 2024
```

Where is port 80??

Iptables & certbot

- iptables on ***this*** system is nft, ..
- certbot appears to not be listenting to nftables, ..
- So, .. let’s add a netfilter rule!

```
# iptables-legacy -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

legacy-save

```
# iptables-legacy-save
# Generated by iptables-save v1.8.7 on Sat Oct 26 14:29:02 2024
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [5:640]
:f2b-sshd - [0:0]
-A INPUT -p tcp -m multiport --dports 2206 -j f2b-sshd
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 206.197.251.0/24 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A f2b-sshd -j RETURN
COMMIT
# Completed on Sat Oct 26 14:29:02 2024
```

Test again, ..

```
# tcpdump -nni eth0 port 80

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:29:31.882364 IP 23.178.112.214.61223 > 206.197.251.202.80: Flags [S], seq
1721951872, win 64240, options [mss 1436,sackOK,TS val 4190404002 ecr
0,nop,wscale 7], length 0
14:29:31.882509 IP 206.197.251.202.80 > 23.178.112.214.61223: Flags [S.], seq
1249112636, ack 1721951873, win 28960, options [mss 1460,sackOK,TS val
728361296 ecr 4190404002
,nop,wscale 7], length 0
14:29:31.929108 IP 23.178.112.214.61223 > 206.197.251.202.80: Flags [.], ack 1,
win 502, options [nop,nop,TS val 4190404049 ecr 728361296], length 0
```

Success!

```
# ./check_cert
Saving debug log to /var/log/letsencrypt/letsencrypt.log
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Processing /etc/letsencrypt/renewal/mx3.omnitec.net.conf
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Cert is due for renewal, auto-renewing...
Plugins selected: Authenticator standalone, Installer None
Renewing an existing certificate for mx3.omnitec.net
Performing the following challenges:
http-01 challenge for mx3.omnitec.net
Waiting for verification...
Cleaning up challenges
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
new certificate deployed without reload, fullchain is
/etc/letsencrypt/live/mx3.omnitec.net/fullchain.pem
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

But wait!!

- iptables
- nftables
- iptables-legacy
- netfilter
- iptables-nft

iptables, ..

- Originally, `iptables` was the standard tool for managing firewall rules on Linux with netfilter.
- Over time, `iptables` evolved, with multiple versions available to enhance performance, compatibility, and manageability.
- `iptables` is now `iptables-legacy`.

nftables

- A new packet filtering framework introduced in Linux kernel 3.13.
- Designed to replace `iptables` with a simpler, more efficient structure and better performance.
- In recent Linux distributions, `nftables` is the default backend, and `iptables` uses `nft` for compatibility.

iptables flavors

- iptables-legacy: The original `iptables` command and syntax, using the traditional iptables backend.
- iptables-nft: A newer version that uses the `nftables` framework under the hood, while keeping the `iptables` syntax.
- iptables = symlink to current

Summary

- *Legacy:* Uses the original `iptables` backend with the classic iptables rules framework.
- *Nft:* Uses the newer `nftables` backend, designed to replace `iptables` with improved efficiency, flexibility, and simpler rule management.
- Both are accessed via the same `iptables` command; the active one depends on which version (`iptables-legacy` or `iptables-nft`) is set as the default on your system.

Conclusions

- Both backends are accessed via `iptables`, with either (`iptables-legacy` or `iptables-nft`), per the default.
- Rules are **not** shared!
- In this case, certbot used legacy, .. so a different rule was required.

Prevention, ..

- To prevent future problems, certbot will now add an allow rule to both nftables AND netfilter:

```
pre_hook = /usr/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT
pre_hook = /usr/sbin/iptables-legacy -I INPUT -p tcp --dport 80 -j ACCEPT
post_hook = /usr/sbin/iptables -D INPUT -p tcp --dport 80 -j ACCEPT
post_hook = /usr/sbin/iptables-legacy -D INPUT -p tcp --dport 80 -j ACCEPT
```

Questions?

Thank you!!

Lee Lammert
lvl@omnitec.net

Credits: Grant Taylor
gtaylor@tnetconsulting.net