Subject: [NETFILTER]: Unable to delete a SAME rule (Using SAME target problems)
Posted by khorenko on Tue, 13 Nov 2007 12:40:24 GMT
View Forum Message <> Reply to Message

Dear all,

The problem description: unable to delete a SAME target rule.

The problem has been already raised some time ago - at least here:
http://marc.info/?l=netfilter&m=117246219803862&w=2

The problem was originally found using 2.6.18-8.1.15.el5 x86_64 kernel and iptables v1.3.5 (stock RHEL5) but it seems to me that the problem is still not fixed in newer kernel/iptables versions.

```
[root@dhcp0-204 ~]# iptables -N foo -t nat
[root@dhcp0-204 ~]# iptables -t nat -A foo -j SAME --to 1.2.3.4
[root@dhcp0-204 ~]# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain foo (0 references)
target     prot opt source               destination
SAME       all  --  anywhere             anywhere            same:1.2.3.4
[root@dhcp0-204 ~]# iptables -t nat -D foo -j SAME --to 1.2.3.4
iptables: No chain/target/match by that name
```

The root of the problem - the structure ipt_same_info:
```
struct ipt_same_info
{
        unsigned char info;
        u_int32_t rangesize;
        u_int32_t ipnum;
        u_int32_t *iparray;

        /* hangs off end. */
        struct ip_nat_range range[IPT_SAME_MAX_RANGE];
};
```

ipnum & iparray is filled/used in kernel space only.

Userspace (iptables) tries to delete the rule:
1) it asks the kernel for the existing table

2) kernel provides the table.
Note: due to generic copy code 'ipt_same_info' structure is completely filled up like any other entry structure, i mean - 'ipnum' and 'iparray' are non-zero!

3) iptables generates the ipt_same_info structure for the rule which it tries to delete.
ipnum and iparray are zeroed.

4) iptables searches the table provided by kernel for the rule to be deleted. It compares many things and at the end it compares the module dependent structures (ipt_same_info).
Ok, iptables also uses the generic code for comparison module dependent structures, so it tries not to compare the complete structure, but only first (struct iptables_target).userspacesize bytes of it.

extensions/libipt_SAME.c:
...
```
static struct iptables_target same_target = {
        .name           = "SAME",
        .version        = IPTABLES_VERSION,
        .size           = IPT_ALIGN(sizeof(struct ipt_same_info)),
        .userspacesize  = IPT_ALIGN(sizeof(struct ipt_same_info)),
...
```

But it has to set '.userspacesize' to sizeof(struct ipt_same_info) because it must compare the 'range' array of the 'ipt_same_info' cause it contains range descriptions.

5) Trying to compare complete 'ipt_same_info' iptables is unable to find the requested rule for deletion because 'ipnum' and 'iparray' fields always differ (zero in userspace-generated structure and non-zero in the tables provided by kernel).

6) So the deletion fails.

At the moment i can see only 3 ways of fixing this:
* reassemble struct ipt_same_info - put 'ipnum' and 'iparray' at the end of the structure. This will save generic code both in kernel and userspace.

* let struct ipt_same_info be as is, teach userspace to manipulate more complex masks (not only first X bytes of the structure)

* let struct ipt_same_info be as is, teach kernel to zero pointers and all the fields which are used only in kernel.

All these ways are quite painful, but could someone please comment this - may be i just missed and some decision had been already done on this issue?

Thank you,
Konstantin Khorenko

SWsoft Virtuozzo/OpenVZ Linux kernel team