author	Jozsef Kadlecsik <kadlec@netfilter.org>	2022-11-02 10:40:47 +0100
committer	Pablo Neira Ayuso <pablo@netfilter.org>	2022-11-02 19:22:23 +0100
commit	510841da1fcc16f702440ab58ef0b4d82a9056b7 (patch)	
tree	0ae9ae1a55f8b0b01e3783c9caa4b306ddf7f396	
parent	cbc1dd5b659f5a2c3cba88b197b7443679bb35a0 (diff)	
download	linux-510841da1fcc16f702440ab58ef0b4d82a9056b7.tar.gz	

diff options

context: 3

space: include

mode: unified

netfilter: ipset: enforce documented limit to prevent allocating huge memory

```
Daniel Xu reported that the hash:net,iface type of the ipset subsystem does
not limit adding the same network with different interfaces to a set, which
can lead to huge memory usage or allocation failure.

The quick reproducer is

$ ipset create ACL.IN.ALL_PERMIT hash:net,iface hashsize 1048576 timeout 0
$ for i in $(seq 0 100); do /sbin/ipset add ACL.IN.ALL_PERMIT 0.0.0.0/0,kaf_$i timeout 0 -exist; done

The backtrace when vmalloc fails:

        [Tue Oct 25 00:13:08 2022] ipset: vmalloc error: size 1073741848, exceeds total pages
        <...>
        [Tue Oct 25 00:13:08 2022] Call Trace:
        [Tue Oct 25 00:13:08 2022]  <TASK>
        [Tue Oct 25 00:13:08 2022]  dump_stack_lvl+0x48/0x60
        [Tue Oct 25 00:13:08 2022]  warn_alloc+0x155/0x180
        [Tue Oct 25 00:13:08 2022]  __vmalloc_node_range+0x72a/0x760
        [Tue Oct 25 00:13:08 2022]  ? hash_netiface4_add+0x7c0/0xb20
        [Tue Oct 25 00:13:08 2022]  ? __kmalloc_large_node+0x4a/0x90
        [Tue Oct 25 00:13:08 2022]  kvmalloc_node+0xa6/0xd0
        [Tue Oct 25 00:13:08 2022]  ? hash_netiface4_resize+0x99/0x710
        <...>

The fix is to enforce the limit documented in the ipset(8) manpage:

>  The internal restriction of the hash:net,iface set type is that the same
>  network prefix cannot be stored with more than 64 different interfaces
>  in a single set.

Fixes: ccf0a4b7fc68 ("netfilter: ipset: Add bucketsize parameter to all hash types")
Reported-by: Daniel Xu <dxu@dxuuu.xyz>
Signed-off-by: Jozsef Kadlecsik <kadlec@netfilter.org>
Signed-off-by: Pablo Neira Ayuso <pablo@netfilter.org>
```

Diffstat

-rw-r--r-- net/netfilter/ipset/ip_set_hash_gen.h 30

1 files changed, 6 insertions, 24 deletions

```
diff --git a/net/netfilter/ipset/ip_set_hash_gen.h b/net/netfilter/ipset/ip_set_hash_gen.h
index 6e391308431da0..3adc291d9ce189 100644
--- a/net/netfilter/ipset/ip_set_hash_gen.h
+++ b/net/netfilter/ipset/ip_set_hash_gen.h
```

```
@@ -42,31 +42,8 @@
 #define AHASH_MAX_SIZE                 (6 * AHASH_INIT_SIZE)
 /* Max muber of elements in the array block when tuned */
 #define AHASH_MAX_TUNED                        64
-
 #define AHASH_MAX(h)                   ((h)->bucketsize)

-/* Max number of elements can be tuned */
-#ifdef IP_SET_HASH_WITH_MULTI
-static u8
-tune_bucketsize(u8 curr, u32 multi)
-{
-       u32 n;
-
-       if (multi < curr)
-               return curr;
-
-       n = curr + AHASH_INIT_SIZE;
-       /* Currently, at listing one hash bucket must fit into a message.
-        * Therefore we have a hard limit here.
-        */
-       return n > curr && n <= AHASH_MAX_TUNED ? n : curr;
-}
-#define TUNE_BUCKETSIZE(h, multi)      \
-       ((h)->bucketsize = tune_bucketsize((h)->bucketsize, multi))
-#else
-#define TUNE_BUCKETSIZE(h, multi)
-#endif
-
 /* A hash bucket */
 struct hbucket {
        struct rcu_head rcu;    /* for call_rcu */
@@ -936,7 +913,12 @@ mtype_add(struct ip_set *set, void *value, const struct ip_set_ext *ext,
                goto set_full;
        /* Create a new slot */
        if (n->pos >= n->size) {
-               TUNE_BUCKETSIZE(h, multi);
+#ifdef IP_SET_HASH_WITH_MULTI
+               if (h->bucketsize >= AHASH_MAX_TUNED)
+                       goto set_full;
+               else if (h->bucketsize < multi)
+                       h->bucketsize += AHASH_INIT_SIZE;
+#endif
                if (n->size >= AHASH_MAX(h)) {
                        /* Trigger rehashing */
                        mtype_data_next(&h->next, d);
```