An alternate way to use IPSet to increase firewall throughput

Prepared by: Sanket Shah
sanket.shah@cyberoam.com

Contents

- Iptables chains without IPSet
- Extending power of IPSet
- PoC results
- Code snippet
- Discussion points

Iptables chains without IPSet

- **Tuple matching in all hooks:**

iptables –A PREROUTING –t nat <tuple$_1$ match> -j DNAT <a.b.c.d>

...

iptables –A PREROUTING –t nat <tuple$_n$ match> -j DNAT <p.q.r.s>

iptables –A FORWARD –t filter <tuple$_1$ match> -j ACCEPT

...

iptables –A FORWARD –t filter <tuple$_m$ match> -j DROP

iptables –A POSTROUTING –t nat <tuple$_1$ match> -j MASQUERADE

...

iptables –A POSTROUTING –t nat <tuple$_n$ match> -j ACCEPT

Iptables chains without IPSet (cont...)

- **Connection mark based matching:**

iptables –A PREROUTING –t mangle <tuple$_1$ match> -j CONNMARK <1>

...

iptables –A PREROUTING –t mangle <tuple$_n$ match> -j CONNMARK <n>

iptables –A PREROUTING –t nat –m connmark <1> -j DNAT <a.b.c.d>

...

iptables –A PREROUTING –t nat –m connmark <n> -j DNAT <p.q.r.s>

iptables –A FORWARD –t filter –m connmark <1> -j ACCEPT

...

iptables –A FORWARD –t filter –m connmark <m> -j DROP

Iptables chains without IPSet (cont...)

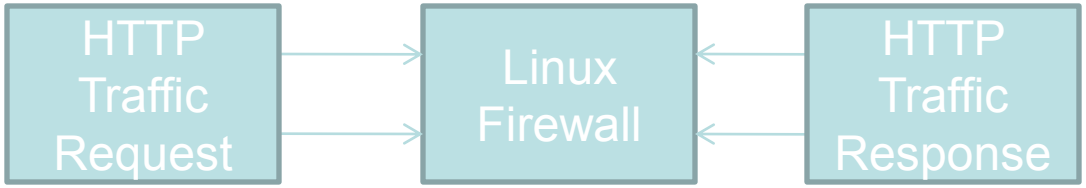
- Issue with both approaches
  - Throughput hampers a lot as number of iptables chain increases
- Can IPSet solve this issue? - NO
  - Current IPSet framework looks at tuple information only
  - Current operations of IPSet are add/delete/test only

Extending power of IPSet

- Bitmap type of IPSet is extended to array of structure pointer map
  - When entry is added into SET, it adds mark and its action whether to DROP or ACCEPT the packet into allocated entry structure
- SET target is extended to lookup and take the action on the packet (ACCEPT/DROP)
- Instead of looking at tuple information, lookup is done on connection mark

PoC results

- System information:
  - CPU: Intel dual core, 3.0 GHz
  - Memory: 2 GB
- Topology:
  - HTTP Data: Avg. 1000 bytes size packet
  - Four 1GB interfaces



PoC results (cont...)

- Configuration:

(With Iptables chain)

iptables -A PREROUTING -t mangle -m state --state NEW -j CONNMARK --set-mark 1000

iptables -A FORWARD -t filter -m connmark --mark 1 -j ACCEPT

...

iptables -A FORWARD -t filter -m connmark --mark 1000 -j ACCEPT

PoC results (cont...)

- Configuration:

(With IPSet chain)

iptables -A PREROUTING -t mangle -m state --state NEW -j CONNMARK --set-mark 1000

iptables -A FORWARD -t filter -j SET --target-set connmarkset

ipset -A connmarkset 1,DROP

ipset -A connmarkset 1000,ACCEPT

ipset -D connmarkset 1000

PoC results (cont...)

- Result:

At ~50% CPU consumption (both core)

(With Iptables chain)

~250 Mbps

(With IPSet chain)

~ 1.8 Gbps

Code snippet

ipt_SET.c

```
if (info->target_set.index != IP_SET_INVALID_ID)
        return ip_set_targetip_kernel(info->target_set.index,
                                      pskb,
                                      info->target_set.flags);

if (info->add_set.index != IP_SET_INVALID_ID)
        ip_set_addip_kernel(info->add_set.index,
                            pskb,
                            info->add_set.flags);
if (info->del_set.index != IP_SET_INVALID_ID)
        ip_set_delip_kernel(info->del_set.index,
                            pskb,
                            info->del_set.flags);

return XT_CONTINUE;
```

Code snippet (cont...)

ip_set_connmarkmap.h

```
struct ip_set_connmarkmap{
        void *members;
        u_int32_t size;
};
struct ip_set_connmarkdetail {
        u_int16_t connmark;
        u_int32_t verdict;
};
struct ip_set_connmark{
        struct ip_set_connmarkdetail *entry;
};
```

Code snippet (cont...)

ip_set_connmarkmap.c

```
targetip_kernel (...) {
        struct ip_set_connmarkmap *table = map->members;
        struct ip_set_connmarkdetail *entry = NULL;
        struct nf_conn *ct = nf_ct_get(skb, &ctinfo);

        entry = table[ct->mark].entry;
        if (entry)
                return entry->verdict;

        return XT_CONTINUE;
}
```

Discussion points

- SNAT/DNAT/REJECT actions directly does not fit in this set. It may require duplication of code into this set
- Can target set generalized for all targets?

Thank You