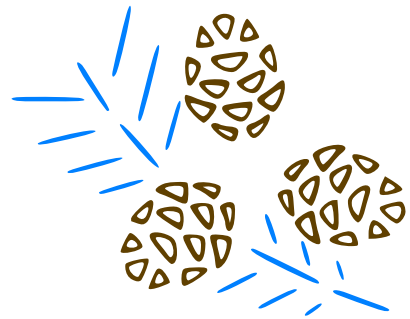
Iptables tricks: native fail2ban



LVEE Winter

Linux Vacation / Eastern Europe

Andrew Savchenko

NRNU MEPhI, Moscow, Russia

15 February 2013

Login server is under attack

Need to ban login server abusers? Options:

- fail2ban
- sshguard
- ...

But thus we have:

- extra daemons
- dependence on log parsing
- questionable stability
- performance overhead

Native approach

How about something native?

- iptables: xt_recent hash tables for per ip activity
- pam_exec.so feedback on successful login

Consider ssh as a sample:

```
iptables -P INPUT DROP
iptables -N ssh
iptables -N ssh_intrusion
iptables -N ssh_drop
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,\
RELATED -j ACCEPT
[...]
# deny ssh abusers, but without auto prolongation
iptables -A INPUT -m recent --name ssh_intrusion \
--rcheck --seconds $ban_period -g ssh_drop
# handle ssh
iptables -A INPUT -p tcp --dport 22 -g ssh
```

Native approach

How about something native?

- iptables: xt_recent hash tables for per ip activity
- pam_exec.so feedback on successful login

Consider ssh as a sample:

```
iptables -P INPUT DROP
iptables -N ssh
iptables -N ssh_intrusion
iptables -N ssh_drop
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,\
RELATED -j ACCEPT
[...]
# deny ssh abusers, but without auto prolongation
iptables -A INPUT -m recent --name ssh_intrusion \
--rcheck --seconds $ban_period -g ssh_drop
# handle ssh
iptables -A INPUT -p tcp --dport 22 -g ssh
```

Iptables: further details

```
iptables -A ssh_intrusion -m recent --name ssh_intrusion --set
iptables -A ssh_intrusion -g ssh_drop

iptables -A ssh_drop \
-m limit --limit $log_limit --limit-burst $log_burst \
-j LOG --log-prefix ”fw:ipt: ssh *intrusion*: ”

# rate per ip protection->ban
iptables -A ssh -m recent --name ssh --set
iptables -A ssh -m recent --name ssh \
--rcheck --hitcount 8 --seconds 300 -g ssh_intrusion
iptables -A ssh -m recent --name ssh \
--rcheck --hitcount 35 --seconds 86400 -g ssh_intrusion

# additional per ip rate limit which for authorized users
iptables -A ssh -m recent --name ssh_auth --set
iptables -A ssh -m recent --name ssh_auth \
--rcheck --hitcount 35 --seconds 600 -g ssh_intrusion
```

External configuration

- /etc/modprobe.d/xt_recent.conf:

```
options xt_recent ip_list_tot=2000 \
ip_pkt_list_tot=35 ip_list_hash_size=0 \
ip_list_perms=0600
```

- /etc/pam.d/sshd:

```
session optional pam_exec.so seteuid \
/usr/local/sbin/unlock-ssh-ip
```

- /etc/ssh/sshd_config:

```
UseDNS no
```

- /usr/local/sbin/unlock-ssh-ip:

```
#!/bin/bash
[[ $PAM_TYPE != ”open_session” ]] && exit 0
[[ -n $PAM_RHOST ]] && echo -$PAM_RHOST > \
/proc/net/xt_recent/ssh
```

Summary

Native, reliable fast banner for ssh abuser :)

The same approach can be used for other auth daemons

Thank you for your attention!