Intrusion Prevention with Suricata and NFQUEUE

Andreas Herz
aherz@oisf.net

OISF

27.06.2016

About me

Andreas Herz

- Living near Augsburg in Germany
- Working on OpenSource, Networking and Security
- Full-time Developer at Linogate GmbH
- Part-time Developer for Suricata at OISF
- Minor Contributions to the Linuxkernel and Netfilter

About Suricata

- OpenSource (GPLv2) backed by OISF
- Cross-platform support (primarily Linux and BSD)
- Stable versions 3.1 and 3.0.2
- Multithreading and High Performance
- Protocol detection, file extraction, lua scripting
- Many supported output formats like Eve/Json
- Hardware Acceleration
- Reading PCAPs
- Emerging Threats ruleset support
- Support via IRC, Mailinglist, Redmine

About OISF

Open Information Security Foundation

- Non-profit foundation
- Support for community-driven technology like Suricata and libhtp
- Funding comes from donations
- Organizations can become Consortium members
- Organizes SuriCon and Trainings (User and Developer)

Why do you want to do IPS?

IPS can extend your existing security/firewall setup:

- Analyse traffic based on packet, connection or flow
- Detect and prevent malicious traffic
- Generate events/alerts
- Use dedicated rules

IPS within Suricata

Suricata supports several capture methods to run IPS mode

- NFQUEUE (Linux)
- AFPACKET (Linux)
- Netmap (BSD, Linux)
- ipfw (BSD)

Requirements

- modern Linux system
- Suricata built with –enable-nfqueue (check –build-info)
- libnetfilter_queue and iptables
- nftables works as well but setup is little bit different
- Ruleset (needs some customization)

Signature

```
alert ip any any -> any any (
    msg:     "GPL ATTACK_RESPONSE
             id check returned root";
    content:    "uid=0|28|root|29|";
    classtype:  bad-unknown;
    sid:     2100498;
    rev:     7;)
```

Prepare Suricata

- Check Suricata config (defaults should be fine though)
- Set .rules files you want to include
- Run Suricata: suricata -q 0 -v
- Turn off NIC-offloading
- Fix Warnings/Errors :)

Simple Setup

```
iptables -N QUEUEIPS
iptables -A QUEUEIPS -j NFQUEUE
iptables -A FORWARD ...
[...]
iptables -A FORWARD -j QUEUEIPS
```

Demo

Let's see it in action!

Improve Performance

- Find the bottleneck :)
- Try balancing into more queues (both nfq and suri)
- Try runmode workers
- Get more CPU power or RAM (depending on bottleneck)

Advanced Usage

- NF_REPEAT (send packets back)
- Use MARK on packets
- nftables instead of iptables
- Balance
- Bypass (accept packet when nothing listening on queue)
- Fanout (accept packetes when queue length got full)

Experiences from productive systems

- Performance highly depends on CPU and RAM
- Ruleset has a huge effect as well
- One bad rule can increase drop rate
- nfqueue might overflow
- Most modern system should handle 1GBit/s
- Smaller embedded system (like APU) can handle 100Mbit/s
- Some rules shouldn’t be converted

End

Questions?

- join us at #suricata in irc.freenode.net
- join oisf-users or oisf-devel mailinglist
- https://suricata-ids.org
- https://redmine.openinfosecfoundation.org
- https://oisf.net

Enjoy Netfilter User Day!