

# Intrusion Detection/Prevention mit Suricata

Andreas Herz

andi@geekosphere.org

Open Information Security Foundation

16.04.2016



# Über mich

- Diplom Informatiker (Uni Augsburg)
- OpenSource und Linux Enthusiast
- Hauptberuflich Entwickler bei Linogate GmbH
- Freiberuflich Entwickler bei Suricata (OISF)
- **Twitter:** @shad0whunter

# Wozu überhaupt ein ID/IP System?

Ein ID/IP System entdeckt gefährliche und ungewünschte Aktivitäten im Netzwerk:

- Analyse auf Paket-, Verbindungs- und Flussebene
- Erkennung von Anomalien und verdächtigem Traffic
- NIDS vs. HIDS
- Heuristiken und Patterns
- Umfangreichere Auswertung

Sinnvolle Ergänzung im Sicherheitskonzept

# Unterschied zwischen IDS/IPS

## IDS

- Detection (Erkennung)
- Passiv, untersucht und alarmiert
- Mit Videokamera vergleichbar

## IPS

- Prevention (Verhinderung)
- Aktive, greift in den Netzwerkverkehr ein
- Mit Sicherheitskontrolle vergleichbar

# Signaturen

```
alert http $HOME_NET any -> any any (  
  msg:          "Outgoing Basic Auth Base64 HTTP  
                Password detected unencrypted";  
  flow:         established,to_server;  
  content:      "|0d 0a|Authorization|3a 20|Basic";  
  content:      !"YW5vbmltb3VzOg==" ; within:32;  
  threshold:   type both, count 1, seconds 300,  
                track by_src;  
  reference:   url, *snip*  
  classtype:   policy-violation;  
  sid:         2006380; rev:13;)
```

# Über Suricata

- Betrieben von OISF (Open Information Security Foundation)
- Open Source (GPLv2)
- Aktuelle Version 3.0.1 Stable und 3.1 Roadmap
- Komplette Neuentwicklung (kein Fork)
- Dedizierte HTTP Library für HTTP

# Über OISF

## Open Information Security Foundation

- <https://www.openinfosecfoundation.org>
- Non-profit foundation
- Ehemals von US Regierung finanziert, mittlerweile viele kommerzielle Sponsoren
- Entwickler und Events werden bezahlt
- Finanzierung durch Consortium Member, Trainings und Spenden

# Features

- Multithreading
- Protokollerkennung
- Dateien extrahieren
- Umfangreiche Logging Optionen (EVE, JSON, Unified2)
- Lua Skripts
- Inline (IPS) und Monitor (IDS) Modus
- High Performnce (PF\_RING, AF\_PACKET, NETMAP)
- pcap Support
- Support für Linux, BSD aber auch Mac OSX/Windows
- und vieles mehr, siehe <https://suricata-ids.org/features/all-features/>



# Vergleich mit Snort

Feature	Suricata	Snort
Threading	Multithreaded	Singlethreaded
Regelsatz	VRT, ET	VRT, ET, SO
Performance	u.a. AF_Packet, Netmap	libpcap
Sniffer	Kein Live Modus	Live Modus
Code	Cross-platform C	Cross-platform C

# Vergleich mit Bro

- Grundlegend andere Technologie
- Statistiken untersuchen und auswerten
- Skriptbasiert
- Eher Netzwerkmonitoring

# Verarbeitungsmodulare

## IDS

- PCAP
- AF\_PACKET (Linux)
- PF\_RING (ntop)
- Hardware Karten

## IPS

- NFQUEUE (u.a. Multiqueue, balancing)
- AF\_PACKET (Linux)
- ipfw
- Netmap (Neu!)

# Ausgabe

- Fastlog (simpel)
- Eve log, JSON alert
- HTTP log
- Unified2
- Debug log
- Drop log
- Syslog
- Stats
- File log usw.

# HTTP signature example

```
alert http
  $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (
  msg:      "ET CHAT Facebook Chat (send message)";
  flow:     established,to_server;
  content:   "POST"; http_method;
  content:   "/ajax/chat/send.php"; http_uri;
  content:   "facebook.com"; http_header;
  *snip*
```

Diese Signatur testet:

- HTTP method: POST
- Seite: /ajax/chat/send.php
- Domain: facebook.com

# Dateien untersuchen

- Dateien aus HTTP Downloads und Uploads extrahieren
- Mit libmagic lassen sich Informationen auslesen
- Werden dediziert geloggt und gespeichert (inkl. Metadaten)
- Keywords z.B. fileext oder filename

# Regeln/Signaturen

- Angepasste EmergingThreats Regeln (Open und Pro)
- Regelmanagement mit Skripten oder Tools wie Pulledpork/Oinkmaster
- Entwicklung von neuen Keywords/Decodern geht stetig voran

Mit Lua sind auch noch anspruchsvollere Regeln möglich

# IPS Einsatz unter Linux

Kompiliert mit libnfqueue!

- Suricata muss alle Pakete einer Verbindung sehen
- iptables -A FORWARD -j NFQUEUE
- Regeln von *alert* zu *drop* umwandeln
- Suricata mit -q 0 starten
- Log beobachten :)

Komplexere Implementierungen sind natürlich möglich



# Community

- Livesystem SELKS von Stamus Networks
- Kibana/Elasticsearch/Logstash für Eyecandy
- Wachsende Zahl an Distributionen mit Suricata in den Repos

# Support

- IRC Support in #suricata (irc.freenode.net)
- Mailinglist (oisy-users und oisy-devel)
- Redmine Issue Tracker
- Github (Pull Requests)
- Redmine Wiki (wird durch Sphinx Doc nach und nach abgelöst)
- Kommerzielle Anfragen leite ich gerne weiter :)