Latest advance in Suricata $I_P^D S$

Éric Leblond

OISF

July 9th 2012

Suricata ?

(C) Jean-Marie Hullot, CC BY 3.0

Suricata ?

(C) Jean-Marie Hullot, CC BY 3.0

OISF

Introduction

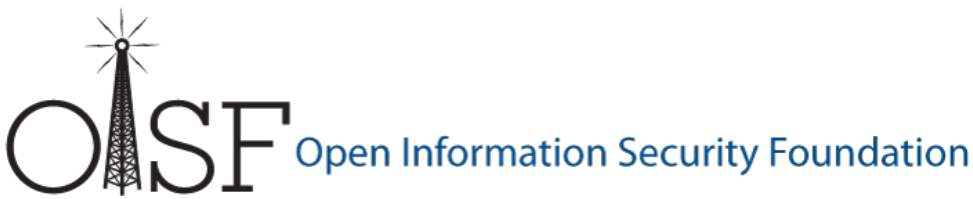
Éric Leblond

- Initial and lead developer of NuFW
- Netfilter Contributor (mainly ulogd2 and userpace interaction)
- Suricata core developer (IPS, multicore optimisation, . . .)
- Independant Open Source et security consultant
- . . .

About OISF

Open Information Security Foundation

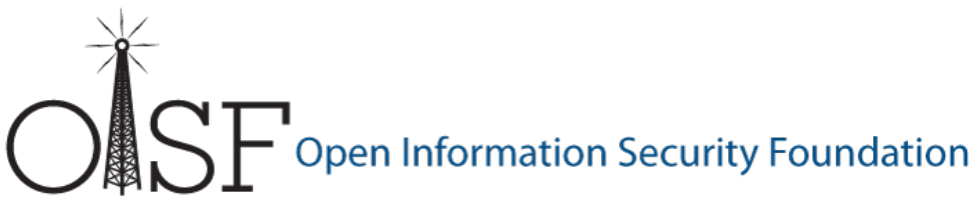
- `http://www.openinfosecfoundation.org`
- Non-profit foundation organized to build a next generation IDS/IPS engine
- Funded by US Governement (DHS, Navy)
- Development of an Open Source IDS/IPS:



About OISF

Open Information Security Foundation

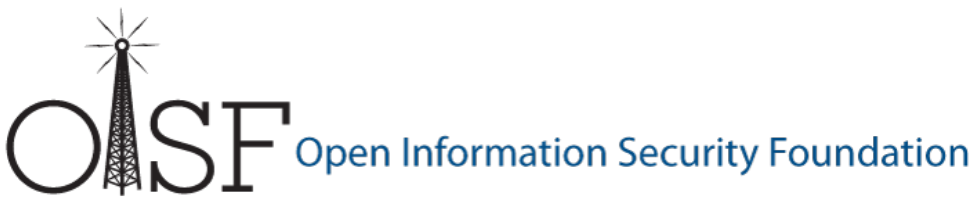
- `http://www.openinfosecfoundation.org`
- Non-profit foundation organized to build a next generation IDS/IPS engine
- Funded by US Governement (DHS, Navy)
- Development of an Open Source IDS/IPS:
  - Developers financement



About OISF

Open Information Security Foundation

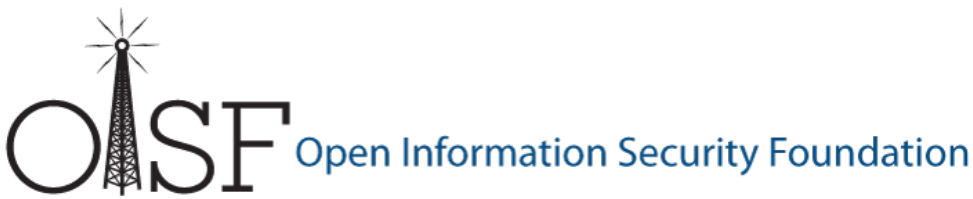
- `http://www.openinfosecfoundation.org`
- Non-profit foundation organized to build a next generation IDS/IPS engine
- Funded by US Governement (DHS, Navy)
- Development of an Open Source IDS/IPS:
  - Developers financement
  - Financial support of related projects (barnyard2)



About OISF

Open Information Security Foundation

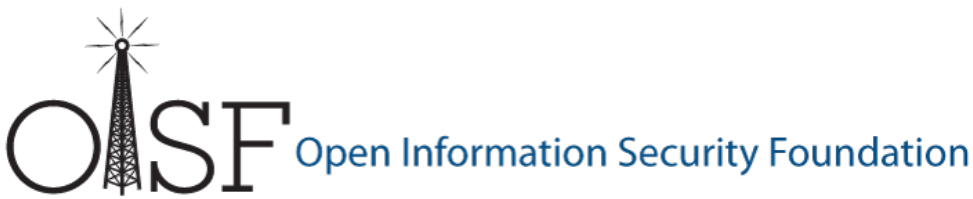
- `http://www.openinfosecfoundation.org`
- Non-profit foundation organized to build a next generation IDS/IPS engine
- Funded by US Governement (DHS, Navy)
- Development of an Open Source IDS/IPS:
  - Developers financement
  - Financial support of related projects (barnyard2)
  - Board who defines big orientation



About OISF

Open Information Security Foundation

- `http://www.openinfosecfoundation.org`
- Non-profit foundation organized to build a next generation IDS/IPS engine
- Funded by US Governement (DHS, Navy)
- Development of an Open Source IDS/IPS:
  - Developers financement
  - Financial support of related projects (barnyard2)
  - Board who defines big orientation
  - Roadmap is defined in public reunion



About OISF

- Consortium members
  - HOST program: Homeland Open Security Technology
  - Platinium level: BAE systems
  - Gold level: Npulse, Endace, Emerging Threats
  - Bronze level: SRC, Everis, Bivio networks, Nitro Security, Mara systems, ...

About OISF

- Consortium members
  - HOST program: Homeland Open Security Technology
  - Platinium level: BAE systems
  - Gold level: Npulse, Endace, Emerging Threats
  - Bronze level: SRC, Everis, Bivio networks, Nitro Security, Mara systems, ...
  - Technology partner: Napatech, Nvidia

About OISF

- Consortium members
  - HOST program: Homeland Open Security Technology
  - Platinium level: BAE systems
  - Gold level: Npulse, Endace, Emerging Threats
  - Bronze level: SRC, Everis, Bivio networks, Nitro Security, Mara systems, ...
  - Technology partner: Napatech, Nvidia
- Developers
  - Leader : Victor Julien

About OISF

- Consortium members
  - HOST program: Homeland Open Security Technology
  - Platinium level: BAE systems
  - Gold level: Npulse, Endace, Emerging Threats
  - Bronze level: SRC, Everis, Bivio networks, Nitro Security, Mara systems, ...
  - Technology partner: Napatech, Nvidia
- Developers
  - Leader : Victor Julien
  - Developers: Anoop Saldanha, Gurvinder Singh, Pablo Rincon, William Metcalf, Eric Leblond, ...

About OISF

- Consortium members
  - HOST program: Homeland Open Security Technology
  - Platinium level: BAE systems
  - Gold level: Npulse, Endace, Emerging Threats
  - Bronze level: SRC, Everis, Bivio networks, Nitro Security, Mara systems, ...
  - Technology partner: Napatech, Nvidia
- Developers
  - Leader : Victor Julien
  - Developers: Anoop Saldanha, Gurvinder Singh, Pablo Rincon, William Metcalf, Eric Leblond, ...
- Board
  - Matt Jonkmann
  - Richard Bejtlich, Dr. Jose Nazario, Joel Ebrahimi, Marc Norton, Stuart Wilson
  - ...

Goals

- Bring new technologies to IDS
- Performance
  - Multi-threads
  - Hardware acceleration
  - `http://packetchaser.org/index.php/opensource/suricata-10gbps`
- Open source
- Support of Linux / *BSD / Mac OSX / Windows

Similar projects

Bro

- Different technology (capture oriented)
- Statistical study

Snort

- Equivalent
- Compatible
- Frontal concurrence
- Sourcefire has felt endangered and has been aggressive
- `http://www.informationweek.com/news/software/enterprise_apps/226400079`

Suricata vs Snort

Suricata

- Drived by a foundation
- Multi-threaded
- Native IPS
- Advanced functions (flowint, libHTP)
- PF_RING support, CUDA support
- Modern and modular code
- Young but dynamic

Snort

- Developed by Sourcefire
- Multi-process
- IPS support
- SO ruleset (advanced logic + perf but closed)
- No hardware acceleration
- Old code
- 10 years of experience

Independant study:

`http://www.aldeid.com/index.php/Suricata-vs-snort`

Suricata with snort ruleset

- Not optimised
- Don’t use any advanced feature

Suricata with dedicated ruleset

- Use Suricata optimised matchs
- Use Suricata advanced keywords
- Can get one from `http://www.emergingthreats.net/`

Fonctionnalities

- Ipv6 native support

Fonctionnalities

- Ipv6 native support
- Multi-threaded

Fonctionnalities

- Ipv6 native support
- Multi-threaded
- Native hardware acceleration (PF_RING, Napatech, Endace, Myricom)

Fonctionnalities

- Ipv6 native support
- Multi-threaded
- Native hardware acceleration (PF_RING, Napatech, Endace, Myricom)
- Numerous options for performance optimisation

Fonctionnalities

- Ipv6 native support
- Multi-threaded
- Native hardware acceleration (PF_RING, Napatech, Endace, Myricom)
- Numerous options for performance optimisation
- Optimized support of IP only tests

Fonctionnalities

- Ipv6 native support
- Multi-threaded
- Native hardware acceleration (PF_RING, Napatech, Endace, Myricom)
- Numerous options for performance optimisation
- Optimized support of IP only tests
- IPS is native (inline mode)

Fonctionnalities

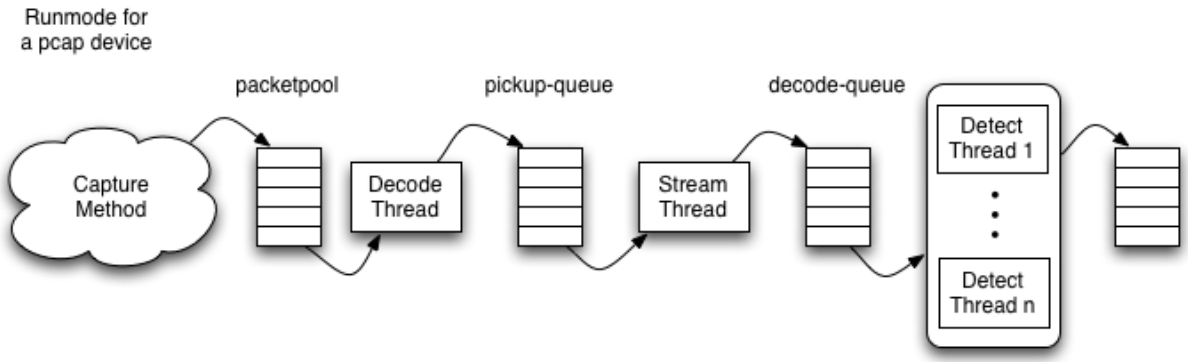
- Ipv6 native support
- Multi-threaded
- Native hardware acceleration (PF_RING, Napatech, Endace, Myricom)
- Numerous options for performance optimisation
- Optimized support of IP only tests
- IPS is native (inline mode)
- Protocol detection

Global architecture

- Chained treatment modules
- Each *running mode* can have its own architecture

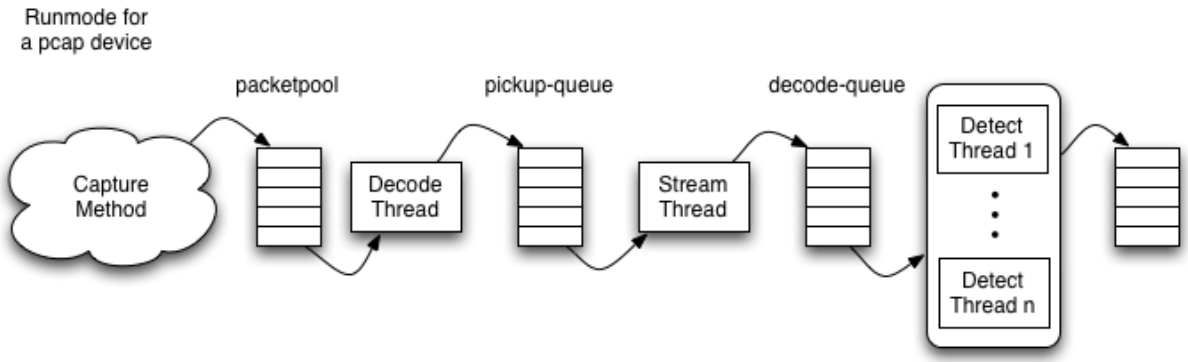
Global architecture

- Chained treatment modules
- Each *running mode* can have its own architecture
- Architecture of mode "pcap auto v1":



Global architecture

- Chained treatment modules
- Each *running mode* can have its own architecture
- Architecture of mode "pcap auto v1":



- Fine setting of CPU preferences
  - Attach a thread to a CPU
  - Attach a threads family to a CPU set
  - Allow IRQs based optimisation

Entry modules

IDS

- PCAP
  - live, multi interface
  - offline support
- AF_PACKET
- PF_RING: mutltithread

  `http://www.ntop.org/PF_RING.html`
- Capture card support: Napatech, Myricom, Endace

Entry modules

IDS

- PCAP
  - live, multi interface
  - offline support
- AF_PACKET
- PF_RING: mutltithread
  `http://www.ntop.org/PF_RING.html`
- Capture card support: Napatech, Myricom, Endace

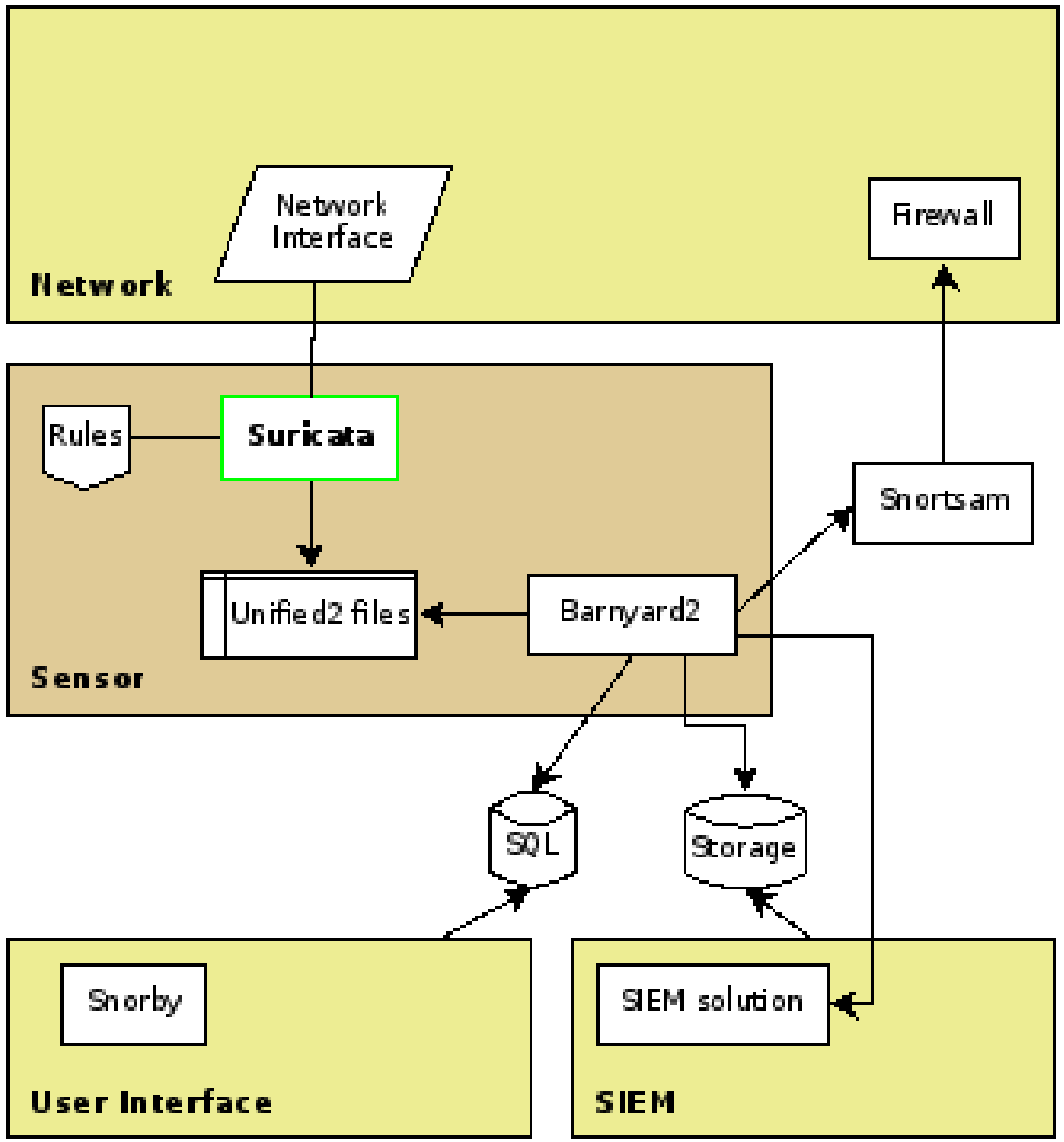
IPS

- NFQueue:
  - Linux: multi-queue, advanced support
  - Windows
- ipfw :
  - FreeBSD
  - NetBSD

Output modules

- Fastlog
- Unified log (Barnyard 1 & 2)
- HTTP log (log in apache-style format)
- Prelude (IDMEF)

Suricata Ecosystem



Signatures

- Support almost all snort ruleset features
- Exclusive features used by VRT ou Emerging Threats rulesets

---

alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; msg: "FTP root login";)

Signatures

- Support almost all snort ruleset features
- Exclusive features used by VRT ou Emerging Threats rulesets

---

alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; msg: "FTP root login";)

Action: alert / drop / pass

Signatures

- Support almost all snort ruleset features
- Exclusive features used by VRT ou Emerging Threats rulesets

---

alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; msg: "FTP root login";)

IP parameters

Signatures

- Support almost all snort ruleset features
- Exclusive features used by VRT ou Emerging Threats rulesets

---

alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; msg: "FTP root login";)

Motif

Signatures

- Support almost all snort ruleset features
- Exclusive features used by VRT ou Emerging Threats rulesets

---

alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; msg: "FTP root login";)

Other parameters

libHTP

- Security oriented HTTP parser
- Written by Ivan Ristić (ModSecurity, IronBee)
- Flow tracking
- Support of keywords
  - http_body
  - http_raw_uri
  - http_header
  - http_cookie
  - ...
- Able to decode gzip compressed flows

Using HTTP features in signature

Signature example: Chat facebook

```
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; \
 classtype:policy-violation; reference:url,doc.emergingthreats.net/2010784; \
 reference:url,www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/POLICY/POLICY_Facebook_Chat; \
 sid:2010784; rev:4; \
)
```

This signature tests:

- The HTTP method: *POST*
- The page: */ajax/chat/send.php*
- The domain: *facebook.com*

Flow variables

Objectives

- Detection of in-multiple-step attack
- Verify condition on a flow
- Modify alert treatment
- State machine inside each flow

Flow variables

Objectives

- Detection of in-multiple-step attack
- Verify condition on a flow
- Modify alert treatment
- State machine inside each flow

Flowbits

- boolean condition
- Set a flag

Flow variables

Objectives

- Detection of in-multiple-step attack
- Verify condition on a flow
- Modify alert treatment
- State machine inside each flow

Flowbits

- boolean condition
- Set a flag

Flowint

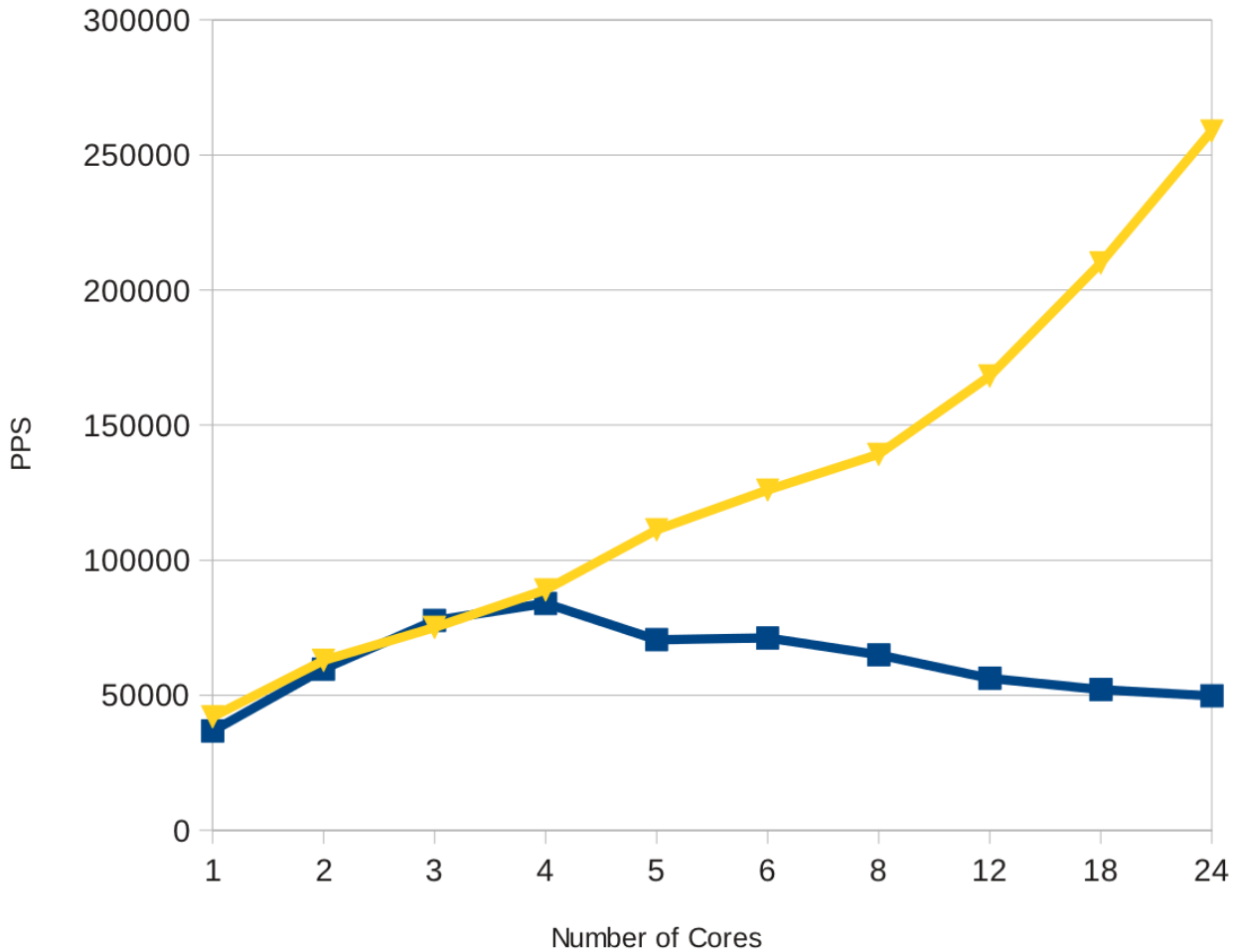
- Define counter
- Arithmetic operation

Suricata 1.3

- New hardware support: Myricom, Endace, Napatech
- TLS/SSL handshake parser
- Better performances
- On the fly MD5 calculation and matching for files in HTTP streams
- Scripts for looking up files/file md5s at Virus Total and others (contributed by Martin Holste)
- http_user_agent keyword for matching on the HTTP User-Agent header

More scability

- Flow engine: removal of a contention point and better hash function.
- Thresholding and Tag engines: fine locking instead of global one.



In the engine

Less memory

- New ac-bs algorithm
- From 35G with ac-full to less than 4G for ac-bs
- Handling 1Gb/s with 7000 rules

Rule engine improvement

- Reload ruleset without breaking the flow analysis
- Rule analyser

Extraction et inspection of files

- Get files from HTTP downloads and uploads
- Detect information about the file using libmagic
  - Type of file
  - Other details
  - ...
- A dedicated extension of signature language

Dedicated keywords

- *filemagic* : description of content

```
alert http any any -> any any (msg:"windows exec"; \
                               filemagic:"executable for MS Windows"; sid:1; rev:1;)
```

- *filestore* : store file for inspection

```
alert http any any -> any any (msg:"windows exec";
                               filemagic:"executable for MS Windows"; \
                               filestore; sid:1; rev:1;)
```

- *fileext* : file extension

```
alert http any any -> any any (msg:"jpg claimed, but not jpg file"; \
                               fileext:"jpg"; \
                               filemagic:!"JPEG image data"; sid:1; rev:1;)
```

- *filename* : file name

```
alert http any any -> any any (msg:"sensitive file leak";
                               filename:"secret"; sid:1; rev:1;)
```

Examples

- Files sending on a server only accepting PDF

```
alert http $EXTERNAL_NET -> $WEBSERVER any (msg:"suspicious upload"; \
        flow:established,to_server; content:"POST" http_method; \
        content:"/upload.php"; http_uri; \
        filemagic:!"PDF document"; \
        filestore; sid:1; rev:1;)
```

- Private keys in the wild

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"outgoing private key"; \
        filemagic:"RSA private key"; sid:1; rev:1;)
```

Disk storage

- Every file is stored on disk
- with a metadata file

```
TIME:              10/02/2009-21:34:53.796083
PCAP PKT NUM:      5678
SRC IP:            61.191.61.40
DST IP:            192.168.2.7
PROTO:             6
SRC PORT:          80
DST PORT:          1091
FILENAME:          /ww/aa5.exe
MAGIC:             PE32 executable for MS Windows (GUI)
                   Intel 80386 32-bit
STATE:             CLOSED
SIZE:              30855
```

- Disk usage limit can be set
- Scripts for looking up files / file md5's at Virus Total and others

Actual limit of files extraction

- Limited to the HTTP protocol
- Storage limit are suboptimal
- MS Office files are not decoded

TLS Handshake parser

- TLS is an application in Suricata way
- Automatic detection of protocol
  - Independent of port
  - Made by pattern matching
- Dedicated keywords
- Usable in the signatures

Other supported applications

- *HTTP* :
  - keywords: http_uri, http_body, http_user_agent, . . .
- *SMTP*
- *FTP*
  - keyword: ftpbounce
- *SSH*
  - keywords: ssh.softwareversion, ssh.protoversion
- *DCERPC*
- *SMB*

A TLS handshake parser

- No traffic decryption
- Method
  - Analyse of TLS handshake
  - Parsing of TLS messages
- A security-oriented parser
  - Coded from scratch
    - Provide a hackable code-base for the feature
    - No external dependency
  - Contributed by Pierre Chifflier (ANSSI)
  - With security in mind:
    - Resistance to attacks (audit, fuzzing)
    - Anomaly detection

OISF

A handshake parser

- The syntax

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 443
```

- becomes

```
alert tls $HOME_NET any -> $EXTERNAL_NET any
```

- Interest:
  - No dependency to IP params
  - Pattern matching is limited to identified protocol
    - Less false positive
    - More performance

TLS keywords

- *TLS.version*: Match protocol version number
- *TLS.subject*: Match certificate subject
- *TLS.issuerdn*: Match the name of the CA which has signed the key
- More to come

TLS keywords

- *TLS.version*: Match protocol version number
- *TLS.subject*: Match certificate subject
- *TLS.issuerdn*: Match the name of the CA which has signed the key
- More to come
- *TLS.fingerprint*: Match the fingerprint of the certificate

Example: verify security policy (1/2)

- Environnement:
  - A company with servers
  - With an official PKI

Example: verify security policy (1/2)

- Environnement:
  - A company with servers
  - With an official PKI
- The goal:
  - Verify that the PKI is used

Example: verify security policy (1/2)

- Environnement:
  - A company with servers
  - With an official PKI
- The goal:
  - Verify that the PKI is used
  - Without working too much

Example: verify security policy (2/2)

- Let's check that the certificates used when a client negotiate a connection to one of our servers are the good one

Example: verify security policy (2/2)

- Let's check that the certificates used when a client negotiate a connection to one of our servers are the good one
- The signature:

```
alert tls any any -> $SERVERS any ( tls.issuerdn:!"C=NL, O=Staat der Nederlanden, \
        CN=Staat der Nederlanden Root CA";)
```

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA (Diginotar by example)

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA (Diginotar by example)
- If it is the case, this is bad!

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA (Diginotar by example)
- If it is the case, this is bad!
- Let's block that!

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA (Diginotar by example)
- If it is the case, this is bad!
- Let's block that!

- Signature:

```
drop tls $CLIENT any -> any any ( \
  tls.subject="C=US, ST=California,  L=Mountain View, O=Google Inc,  CN=*.google.com"; \
  tls.issuerdn=!"C=US, O=Google Inc, CN=Google Internet Authority";)
```

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA (Diginotar by example)
- If it is the case, this is bad!
- Let's block that!

- Signature:

```
drop tls $CLIENT any -> any any ( \
  tls.subject="C=US, ST=California,  L=Mountain View, O=Google Inc,  CN=*.google.com"; \
  tls.issuerdn=!"C=US, O=Google Inc, CN=Google Internet Authority";)
```

- What! KPN has been hacked too!

Example: detect certificate anomaly

- Google.com is signed by Google Internet Authority
- Not by an other CA (Diginotar by example)
- If it is the case, this is bad!
- Let's block that!

- Signature:

```
drop tls $CLIENT any -> any any ( \
  tls.subject="C=US, ST=California, L=Mountain View, O=Google Inc, CN=*.google.com"; \
  tls.issuerdn=!"C=US, O=Google Inc, CN=Google Internet Authority";)
```

- What! KPN has been hacked too!
- Let's get rid of the Dutch!

```
drop tls $CLIENT any -> any any (tls.issuerdn="C=NL");
```

Actual limit

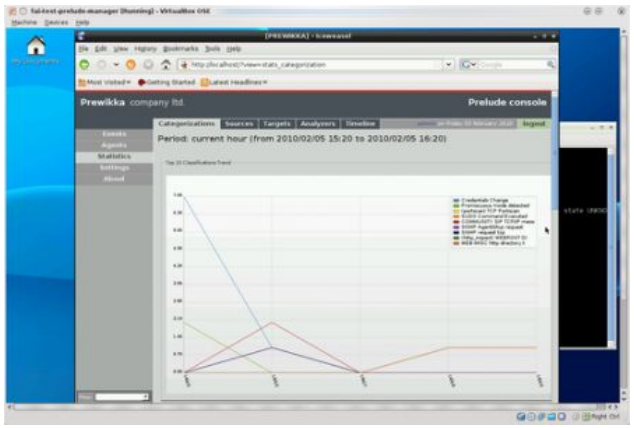
- Keywords apply only to first certificate of the chain.
  - Impossible to do check on chained certificates
  - Supported by the parser but not by the keywords.
- Some keyword are missing and will be added
  - used cryptographic algorithm
  - Key size
  - Diffie-Hellman parameters
- Statistical study and certificate storage

Roadmap

- IP and DNS reputation
- SCADA Preprocessor (thanks to Digital Bond)
- Keyword *geoip*

How to test it fast and easy?

- Already available in Debian, Ubuntu, Gentoo, Freebsd
- Live distribution:
  - SIEM live (Suricata + Prelude + Openvas) : `https://www.wzdftpd.net/redmine/projects/siem-live/wiki`



  - Smooth-Sec (Suricata + Snorby) : `http://bailey.st/blog/smooth-sec/`

Questions

Do you have questions ?

- Big thanks:
  - Pierre Chifflier : `http://www.wzdftpd.net/blog/`
  - The whole OISF team and especially Victor Julien
- Related read:
  - OISF website: `http://www.openinfosecfoundation.org/`
  - Planet suricata: `http://planet.suricata-ids.org/`
  - Suricata devel site: `https://redmine.openinfosecfoundation.org/`
- Join me:
  - Mail: `eric@regit.org`
  - Twitter: Regiteric
  - Blog: `https://home.regit.org`