Présentation de l'$I_P^D S$ Suricata

Éric Leblond

OISF

10 janvier 2012

Suricata ?

(C) +marcelo math

Suricata ?

(C) +marcelo math

OISF

Introduction

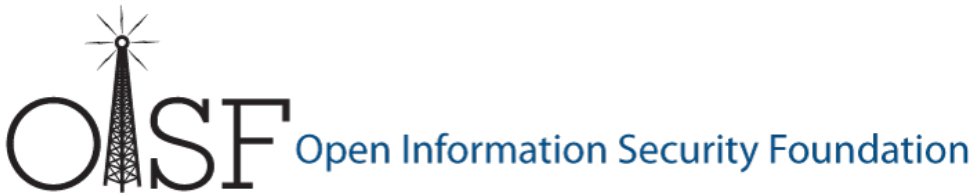
Éric Leblond

- Initiateur du projet NuFW, ancien CTO d'EdenWall
- Contributeur Netfilter notamment sur ulogd2
- Core développeur Suricata
  - Responsable de l'acquisition
  - Optimisation multicore
  - Fonctions IPS
  - ...
- Consultant indépendant Open Source et Sécurité

À propos de l'OISF

Open Information Security Foundation

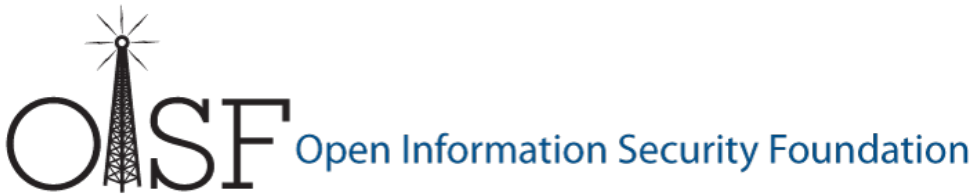
- `http://www.openinfosecfoundation.org`
- Fondation à but non lucratif dont le but est de construire un moteur IDS/IPS de nouvelle génération
- Soutenue financièrement par le gouvernement américain (DHS, Navy)
- Développement d'un IDS/IPS Open Source :



À propos de l'OISF

Open Information Security Foundation

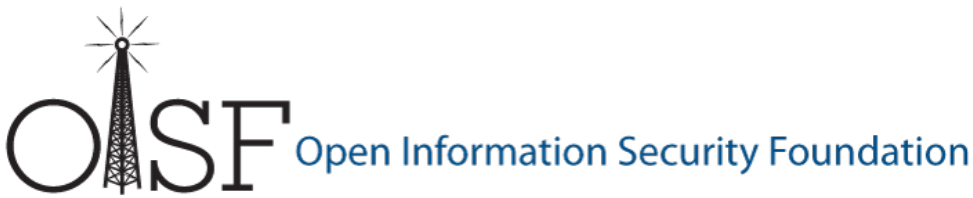
- `http://www.openinfosecfoundation.org`
- Fondation à but non lucratif dont le but est de construire un moteur IDS/IPS de nouvelle génération
- Soutenue financièrement par le gouvernement américain (DHS, Navy)
- Développement d'un IDS/IPS Open Source :
  - Financement des développeurs



À propos de l'OISF

Open Information Security Foundation

- `http://www.openinfosecfoundation.org`
- Fondation à but non lucratif dont le but est de construire un moteur IDS/IPS de nouvelle génération
- Soutenue financièrement par le gouvernement américain (DHS, Navy)
- Développement d'un IDS/IPS Open Source :
  - Financement des développeurs
  - Board chargé de définir les orientations



À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : Nitro Security, Mara systems, . . .

À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : Nitro Security, Mara systems, . . .
  - Partenaire technologique : Napatech, Nvidia

À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : Nitro Security, Mara systems, . . .
  - Partenaire technologique : Napatech, Nvidia
- Développeurs
  - Leader : Victor Julien

À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : Nitro Security, Mara systems, . . .
  - Partenaire technologique : Napatech, Nvidia
- Développeurs
  - Leader : Victor Julien
  - Développeurs : Anoop Saldanha, Eileen Doilon, Eric Leblond, . . .

À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : Nitro Security, Mara systems, ...
  - Partenaire technologique : Napatech, Nvidia
- Développeurs
  - Leader : Victor Julien
  - Développeurs : Anoop Saldanha, Eileen Doilon, Eric Leblond, ...
- Board
  - Matt Jonkmann
  - Richard Bejtlich, Dr. Jose Nazario, Joel Ebrahimi, Marc Norton, Stuart Wilson
  - ...

Buts

- Apporter de nouvelles technologies aux IDS
- Performance
  - Multi-threadé
  - Accélération matérielle
  - `http://packetchaser.org/index.php/opensource/suricata-10gbps`
- Open source
- Support de Linux / *BSD / Mac OSX / Windows

Projets similaires

Bro

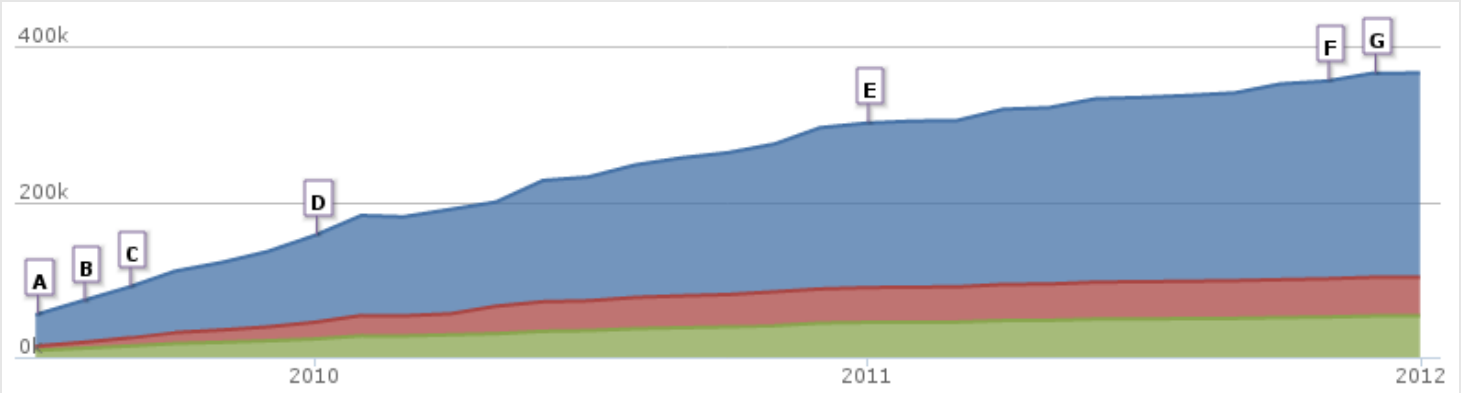
- Positionnement différent (orientation capture)
- Études statistiques/anomalies

Snort

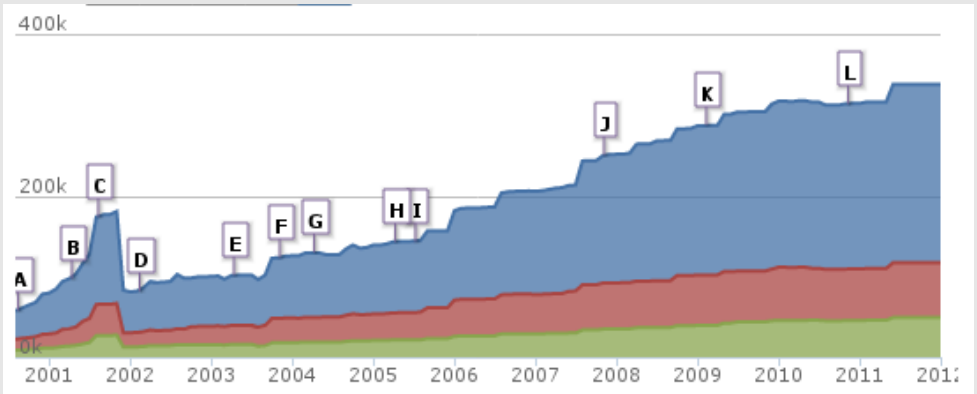
- Fonctionnellement équivalent
- Compatibilité
- Concurrence frontale
- Sourcefire se sent menacé et est agressif
- `http://www.informationweek.com/news/software/enterprise_apps/226400079`

Volume de code

Suricata



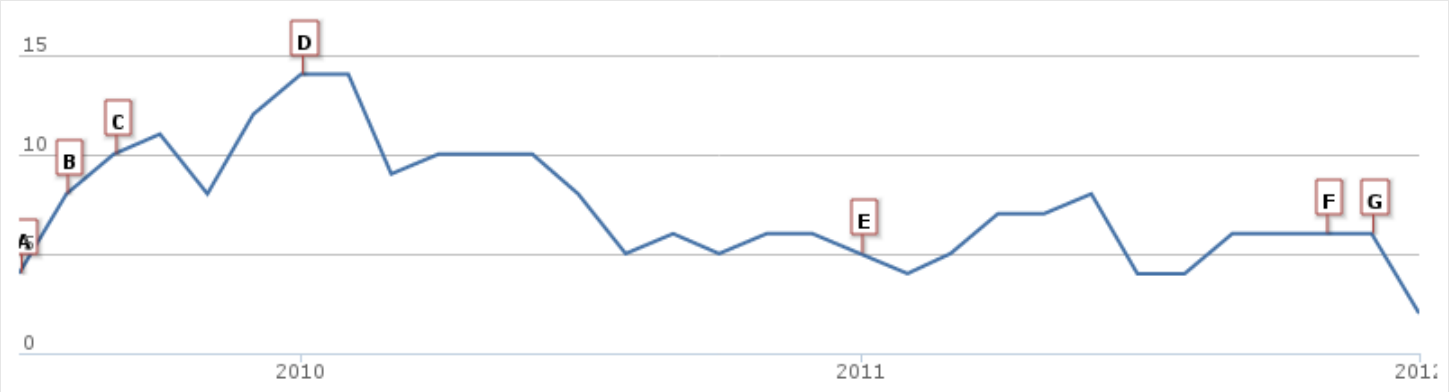
Snort



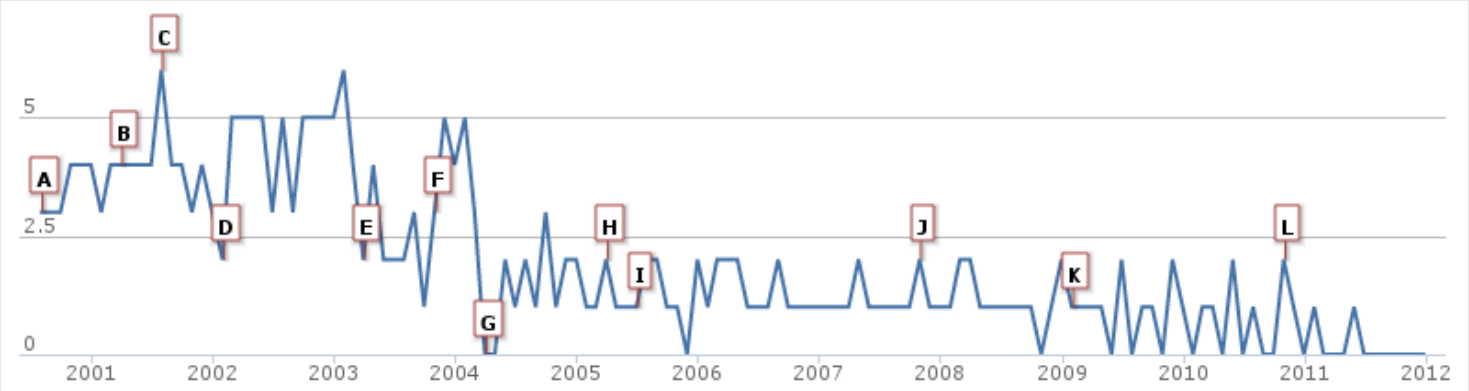
Source : ohloh.net

Committeurs

Suricata



Snort



Source : ohloh.net

Suricata vs Snort

Suricata

- Soutenu par une fondation
- Multi-threadé
- IPS natif
- Fonctions avancées (flowint, libHTP)
- Support de PF_RING
- Code moderne et modulaire
- Jeune mais dynamique

Snort

- Développé par Sourcefire
- Multi-process
- IPS supporté
- Jeu de règles SO (logique avancée + perf mais fermé)
- Pas d'accélération matérielle
- Code vieillissant
- 10 ans d'expérience

Étude intéressante :

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

Fonctionnalités

- Support Ipv6 natif

Fonctionnalités

- Support Ipv6 natif
- Multi-threadée

Fonctionnalités

- Support Ipv6 natif
- Multi-threadée
- Accélération matérielle native (Accélération par GPU, PF_RING)

Fonctionnalités

- Support Ipv6 natif
- Multi-threadée
- Accélération matérielle native (Accélération par GPU, PF_RING)
- De nombreuses options pour optimiser les performances

Fonctionnalités

- Support Ipv6 natif
- Multi-threadée
- Accélération matérielle native (Accélération par GPU, PF_RING)
- De nombreuses options pour optimiser les performances
- Support optimisé des tests sur IP seules

Fonctionnalités

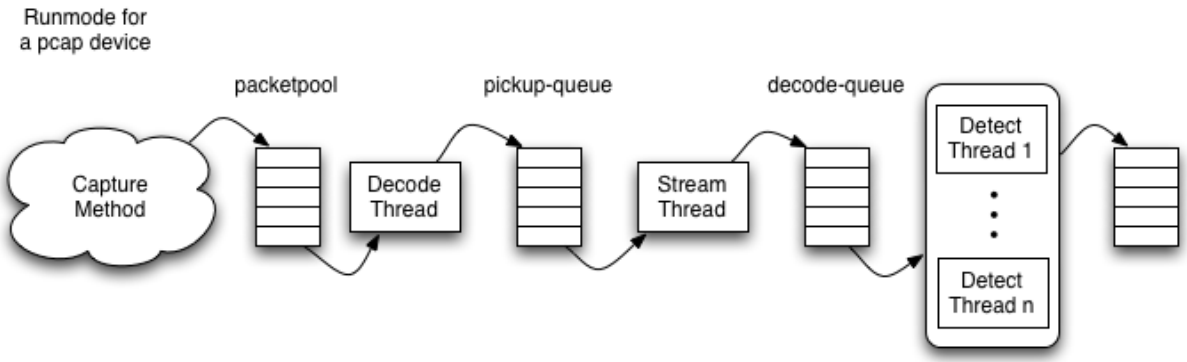
- Support Ipv6 natif
- Multi-threadée
- Accélération matérielle native (Accélération par GPU, PF_RING)
- De nombreuses options pour optimiser les performances
- Support optimisé des tests sur IP seules
- IPS (mode inline) natif

Architecture globale

- Enchainement des modules de traitements
- Chaque *running mode* peut avoir sa propre architecture

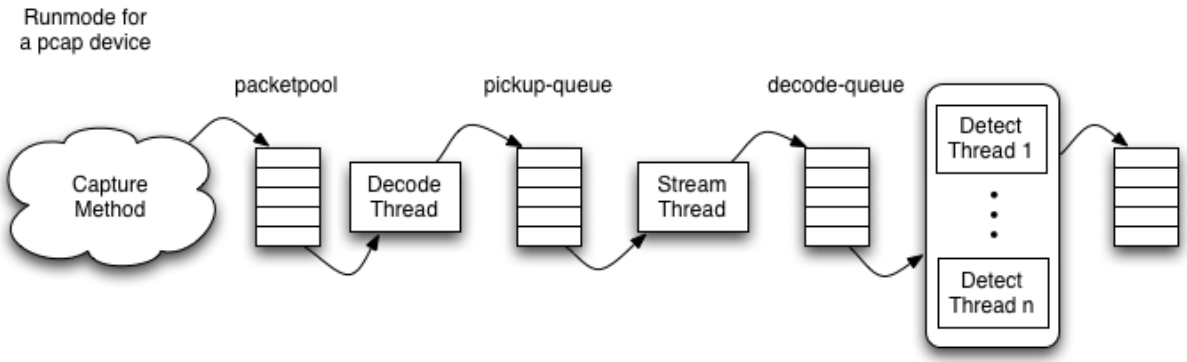
Architecture globale

- Enchainement des modules de traitements
- Chaque *running mode* peut avoir sa propre architecture
- Architecture du mode pcap auto v1 :



Architecture globale

- Enchainement des modules de traitements
- Chaque *running mode* peut avoir sa propre architecture
- Architecture du mode pcap auto v1 :



- Paramétrage fin des préférences CPU
  - Affectation d'un thread à un CPU
  - D'une famille de threads à un ensemble de CPU
  - Permet la prise en compte des IRQs

Modules d'entrées

IDS

- PCAP
  - live, multi interface
  - hors ligne
- PF_RING
  - `http://www.ntop.org/PF_RING.html`
- AF_PACKET

Modules d'entrées

IDS

- PCAP
  - live, multi interface
  - hors ligne
- PF_RING
  - `http://www.ntop.org/PF_RING.html`
- AF_PACKET

IPS

- NFQueue :
  - Linux : multi-queue
  - Windows
- ipfw :
  - FreeBSD
  - NetBSD

Modules de sortie

- Fastlog
- Unified log (Barnyard 1 & 2)
- HTTP log (log dans un format de type apache)
- Prelude (IDMEF)

Signatures

- Support de presque toutes les signatures de snort
- Support de fonctionnalités exclusives utilisées par les rulesets comme VRT ou Emerging Threats

---

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

Signatures

- Support de presque toutes les signatures de snort
- Support de fonctionnalités exclusives utilisées par les rulesets comme VRT ou Emerging Threats

---

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

Action : alert / drop / pass

Signatures

- Support de presque toutes les signatures de snort
- Support de fonctionnalités exclusives utilisées par les rulesets comme VRT ou Emerging Threats

---

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

Paramètres IP

Signatures

- Support de presque toutes les signatures de snort
- Support de fonctionnalités exclusives utilisées par les rulesets comme VRT ou Emerging Threats

---

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

Motif

Signatures

- Support de presque toutes les signatures de snort
- Support de fonctionnalités exclusives utilisées par les rulesets comme VRT ou Emerging Threats

---

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

Autres paramètres

Stream inline

- L'analyse au niveau applicatif travaille sur un flux de données
- Les données envoyés dans une session TCP peuvent être désordonnées
  - Perte de paquets
  - Réemission de paquets
  - Paquets arrivant dans le désordre
- l'$I_P^D S$ doit donc reconstruire les flux TCP avant de les livrer à l'analyse applicative

Problématique

- Prise en compte des différences entre IDS et IPS
- L'IDS doit être au plus proche de ce qui est reçu par la cible
  - Analyse des paquets quand leur réception est établie
  - La réception d'un ACK déclenche l'analyse des données
- l'IPS doit bloquer les paquets avant qu'ils atteignent leur cible
  - La technique de l'IDS bloquerait les paquets après passage
  - Il faut donc envisager une autre solution

IPS comme point de contrôle

- l'IPS agit comme un point de bloquage
  - Il est donc représentatif de ce qui traverse
  - Il peut donc reconstruire les flux avant de les envoyer

IPS comme point de contrôle

- l'IPS agit comme un point de bloquage
  - Il est donc représentatif de ce qui traverse
  - Il peut donc reconstruire les flux avant de les envoyer
- Solution retenue
  - Reconstruction des segments de données à la réception
  - Passage de données reconstruites à la couche applicative
  - Décision prise sur la donnée
  - Réécriture des paquets si nécessaire
  - Transmission

IPS comme point de contrôle

- l'IPS agit comme un point de bloquage
  - Il est donc représentatif de ce qui traverse
  - Il peut donc reconstruire les flux avant de les envoyer
- Solution retenue
  - Reconstruction des segments de données à la réception
  - Passage de données reconstruites à la couche applicative
  - Décision prise sur la donnée
  - Réécriture des paquets si nécessaire
  - Transmission
- Détails : `http://www.inliniac.net/blog/2011/01/31/suricata-ips-improvements.html`

CUDA

- Utilisation de CUDA (architecture de calcul parrallèle dévéloppée par NVIDIA)
- Actuellement : implémentation d'un algorithme de matching en CUDA
- Travail en cours, Nvidia est partenaire technologique de l'OISF

CUDA

- Utilisation de CUDA (architecture de calcul parrallèle dévéloppée par NVIDIA)
- Actuellement : implémentation d'un algorithme de matching en CUDA
- Travail en cours, Nvidia est partenaire technologique de l'OISF
- Difficulté d'utiliser le pipeline du GPU de manière efficace

CUDA

- Utilisation de CUDA (architecture de calcul parrallèle dévéloppée par NVIDIA)
- Actuellement : implémentation d'un algorithme de matching en CUDA
- Travail en cours, Nvidia est partenaire technologique de l'OISF
- Difficulté d'utiliser le pipeline du GPU de manière efficace
- ... Performance similaire avec ou sans (avec des CPUs décents)

libHTP

- Parseur orienté sécurité du protocole HTTP
- Écrit par Ivan Ristić (ModSecurity, IronBee)
- Suivi des flux
- Support des mots clés
  - http_body
  - http_raw_uri
  - http_header
  - http_cookie
  - ...
- Capable de décoder des flux compressés par Gzip

Utilisation de HTTP dans les règles

Exemple de règles : 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; \
)
```

Cette signature teste donc :

- La méthode HTTP : *POST*
- La page : */ajax/chat/send.php*
- Le domaine : *facebook.com*

Variables de flux

Objectifs

- Détection des attaques en étapes
- Vérification de conditions sur un flux
- Modification du traitement sur l'alerte
- Machine à état au sein du flux

Variables de flux

Objectifs

- Détection des attaques en étapes
- Vérification de conditions sur un flux
- Modification du traitement sur l'alerte
- Machine à état au sein du flux

Flowbits

- Condition booléenne
- Positionnement d'un drapeau

Variables de flux

Objectifs

- Détection des attaques en étapes
- Vérification de conditions sur un flux
- Modification du traitement sur l'alerte
- Machine à état au sein du flux

Flowbits

- Condition booléenne
- Positionnement d'un drapeau

Flowint

- Définition de compteur
- Opération arithmétique

Variables de type Flowint

- Permet la capture, le stockage et la comparaison de données dans une variable
- Stockage et opérations mathématiques
- Variable lié à un flux donné

Ex : montre une alerte si et seulement si *usernamecount* est plus grand que 5 :

```
alert tcp any any -> any any (msg:"Counting Usernames"; content:"jonkman"; \
flowint: usernamecount, +, 1; flowint:usernamecount, >, 5;)
```

Variables de type Flowint (2)

Ex : Suivi des logins
Mise en place d'un compteur des échecs de login :

```
alert tcp any any -> any any (msg:"Start a login count"; content:"login failed"; \
flowint:loginfail, notset; flowint:loginfail, =, 1; flowint:noalert;)
alert tcp any any -> any any (msg:"Counting Logins"; content:"login failed"; \
flowint:loginfail, isset; flowint:loginfail, +, 1; flowint:noalert;)
```

Alerte si il y a un login réussi après 5 échecs :

```
alert tcp any any -> any any (msg:"Login success after file failures"; \
                content:"login successful"; \
flowint:loginfailed, isset; flowint:loginfailed, =, 5;)
```

Extraction et inspection de fichiers

- Récupération des fichiers des downloads et uploads HTTP
- Détection des information sur le fichier (libmagic)
  - Type de fichiers
  - ...
- Une extension dédiée du langage des signatures

Mots clés dédiés

- *filemagic* : description du contenu

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

- *filestore* : stockage du fichier pour inspection

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

- *fileext* : extension du fichier

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

- *filename* : nom du fichier

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

Exemples

- Envoi de fichier sur un serveur n'acceptant que les 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;)
```

- Clefs privées circulant en clair

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

Stockage sur disque

- Chaque fichier est stocké sur disque
- Accompagné par un fichier de metadonnées

```
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
```

- Des limites d'utilisation globale sont positionnables

Limites actuelles de l'extraction de fichiers

- Limité au seul protocole HTTP
- Limite de stockage
- Fichier MS Office non décodé

Étude de la négociation TLS

- TLS est une application au sens de Suricata
- Détection automatique du protocole
  - Indépendamment du port
  - Grâce à un système de pattern matching
- Des mots clés dédiés
- Utilisables dans les signatures

Autres protocoles supportés

- *HTTP* :
  - Mots clés : http_uri, http_body, . . .
- *SMTP*
- *FTP*
  - Mots clés : ftpbounce
- *SSH*
  - Mots clés : ssh.softwareversion, ssh.protoversion
- *DCERPC*
- *SMB*

Un parseur de handshake

- pas de déchiffrage du trafic
- Méthode
  - Analyse du handshake TLS
  - Parsing des messages TLS
- Un parseur orienté sécurité
  - Code developpé depuis zéro
    - Provide a hackable code-base for the feature
    - No external dependency
  - Contribution de Pierre Chifflier (ANSSI)
  - Objectif sécurité
    - Résistance aux attaques (audité, fuzzé)
    - Detection d'anomalies

Un parseur de handshake

- La syntaxe

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

- Devient

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

- Intérêts
  - Pas de dependances sur les paramètres IP
  - Limite les recherches à des protocoles identifiés
    - Moins de faux positifs
    - Plus de performances

Mots clés TLS

- *TLS.version* : Correspondance avec le numéro de version du protocole
- *TLS.subject* : Correspondance sur le sujet du certificat
- *TLS.issuerdn* : Correspondance sur le nom du générateur de certificat
- Plus à venir

Exemple : vérification politique de sécurité (1/2)

- L'environnement :
  - Une entreprise avec des serveurs
  - Qui a une PKI officielle

Exemple : vérification politique de sécurité (1/2)

- L'environnement :
  - Une entreprise avec des serveurs
  - Qui a une PKI officielle
- L'objectif :
  - S'assurer que la PKI est utilisée

Exemple : vérification politique de sécurité (1/2)

- L'environnement :
  - Une entreprise avec des serveurs
  - Qui a une PKI officielle
- L'objectif :
  - S'assurer que la PKI est utilisée
  - Sans trop se fatiguer

OISF

Exemple : vérification politique de sécurité (2/2)

- Asssurons nous que les certificats utilisés dans les négotiations clients vers nos serveurs sont les notres

Exemple : vérification politique de sécurité (2/2)

- Asssurons nous que les certificats utilisés dans les négotiations clients vers nos serveurs sont les notres
- La signature :

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

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité (Diginotar par exemple)

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité (Diginotar par exemple)
- Si c'est le cas, c'est mal !

OISF

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité (Diginotar par exemple)
- Si c'est le cas, c'est mal !
- Bloquons ça !

OISF

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité (Diginotar par exemple)
- Si c'est le cas, c'est mal !
- Bloquons ça !

- 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";)
```

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité (Diginotar par exemple)
- Si c'est le cas, c'est mal !
- Bloquons ça !

- 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";)
```

- Ah, Quoi ! KPN a aussi été hacké !

Exemple : détection d'anomalies de certificats

- Google.com est signé par Google Internet Authority
- Il n'est pas signé par une autre autorité (Diginotar par exemple)
- Si c'est le cas, c'est mal !
- Bloquons ça !

- 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";)
```

- Ah, Quoi ! KPN a aussi été hacké !
- Sus aux hollandais !

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

Limitation actuelle

- Vient juste d'être commité dans le git
- Les mots clés ne portent que sur le premier certificat de la chaine.
  - Impossible actuellement de faire des vérifications sur les certificats chainés
  - Le parseur les supporte mais pas les mots clés.
- Certains mots clés manquent et vont être ajoutés
  - Algorithme cryptographique utilisé/proposé
  - Taille de clefs
  - Paramètre Diffie-Hellman
- Études statistiques

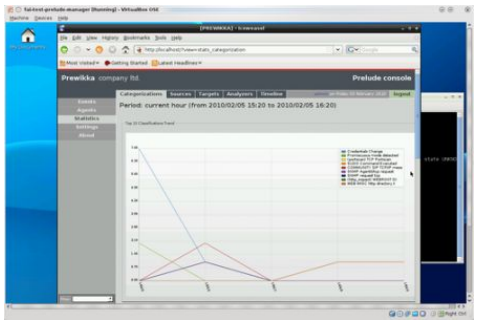
Fonctionnalités planifiées

- Réputation IP et DNS
- Compteur de répétition temporelle
- Mot clé *geoip*
- Rechargement du jeu de signatures sans rupture de l'analyse des flux

Détails : `http://www.openinfosecfoundation.org/index.php/component/content/article/1-latest-news/136-brainstorming-meeting-summary-and-phase-three-development-roadmap-draft`

Comment tester rapidement

- Déjà disponible dans Debian, Ubuntu, Gentoo, Freebsd
- Distribution live :
  - SIEM live (Suricata + Prelude + Openvas) : `https://www.wzdftpd.net/redmine/projects/siem-live/wiki`



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

  - Security Onion : `http://securityonion.blogspot.com/`

Questions

Avez-vous des questions ?

- Merci à :
  - Pierre Chifflier : `http://www.wzdftpd.net/blog/`
  - Toute l'équipe de l'OISF et Victor Julien en particulier
- Pour aller plus loin :
  - Site de l'OISF : `http://www.openinfosecfoundation.org/`
  - Site développeurs de Suricata : `https://redmine.openinfosecfoundation.org/`
  - Blog de Victor Julien : `http://www.inliniac.net/blog/`
  - Mon blog : `http://home.regit.org`
- Me joindre :
  - Courriel : `eric@regit.org`
  - Twitter : Regiteric