Présentation de l'$IDPS$ Suricata

Éric Leblond

OISF

11 mai 2011

Suricata ?

(C) Jean-Marie Hullot, CC BY 3.0

Suricata ?

(C) Jean-Marie Hullot, CC BY 3.0

SURICATA

OISF

Introduction

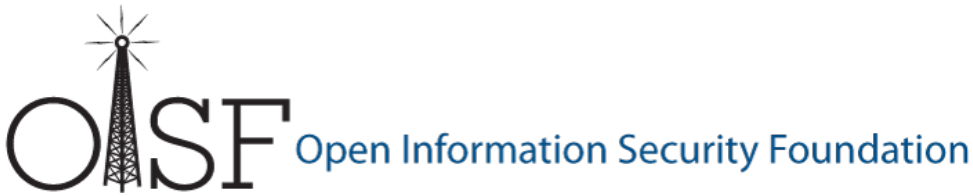
Éric Leblond

- Initiateur du projet NuFW
- Contributeur Netfilter notamment sur ulogd2
- Core développeur Suricata (IPS, optimisation multicore, . . .)
- Consultant indépendant Open Source et Sécurité
- . . .

À propos de l'OISF

Open Information Security Foundation

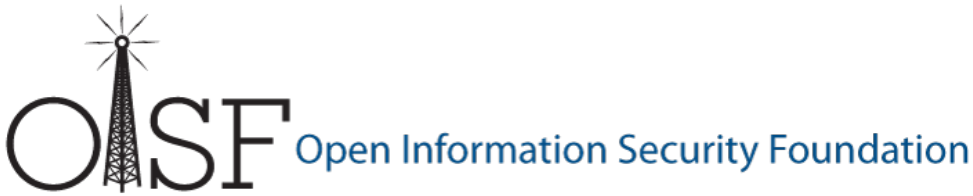
- `http://www.openinfosecfoundation.org`
- Fondation à but non lucratif dont le but est de construire un moteur IDS/IPS de nouvelle génération
- Soutenue financièrement par le gouvernement américain (DHS, Navy)
- Développement d'un IDS/IPS Open Source :



À propos de l'OISF

Open Information Security Foundation

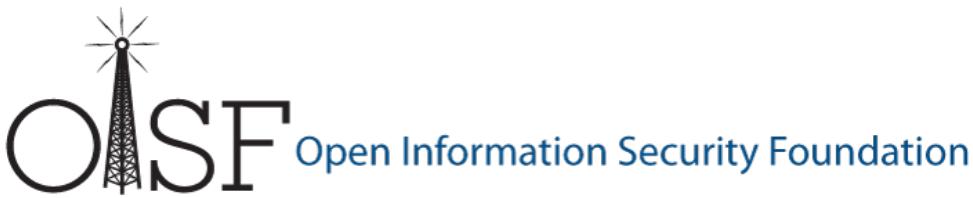
- `http://www.openinfosecfoundation.org`
- Fondation à but non lucratif dont le but est de construire un moteur IDS/IPS de nouvelle génération
- Soutenue financièrement par le gouvernement américain (DHS, Navy)
- Développement d'un IDS/IPS Open Source :
  - Financement des développeurs



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 : EdenWall, Nitro Security, Mara systems, ...

À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : EdenWall, 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 : EdenWall, 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 : EdenWall, Nitro Security, Mara systems, ...
  - Partenaire technologique : Napatech, Nvidia
- Développeurs
  - Leader : Victor Julien
  - Développeurs : Anoop Saldanha, Gurvinder Singh, Pablo Rincon, William Metcalf, Eric Leblond, ...

À propos de l'OISF

- Membres du consortium
  - Programme HOST : Homeland Open Security Technology
  - Niveau or : Npulse, Endace
  - Niveau bronze : EdenWall, Nitro Security, Mara systems, ...
  - Partenaire technologique : Napatech, Nvidia
- Développeurs
  - Leader : Victor Julien
  - Développeurs : Anoop Saldanha, Gurvinder Singh, Pablo Rincon, William Metcalf, 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

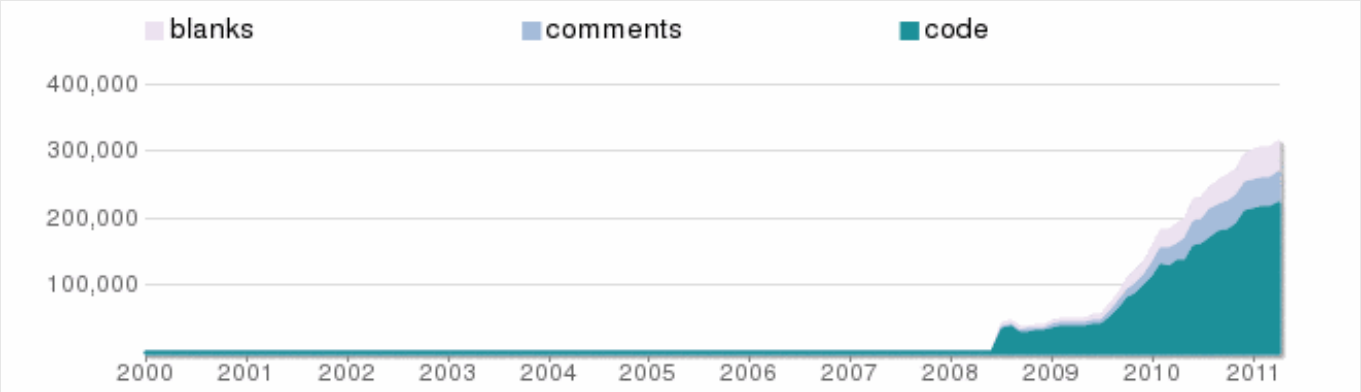
- Positionnement différent (orientation capture)
- Études statistiques

Snort

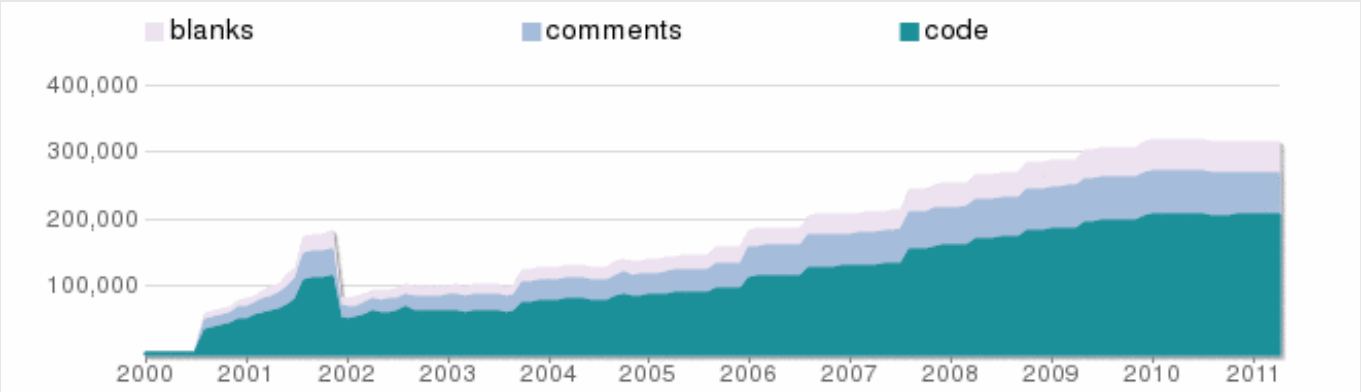
- Fonctionnellement équivalent
- Compatibilité
- Concurrence frontale
- Sourcefire se sent menacé et est agressif
- `http://www.informationweek.com/news/software/enterprise_apps/226400079`

Volume de code

Suricata



Snort



Source : ohloh.net

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

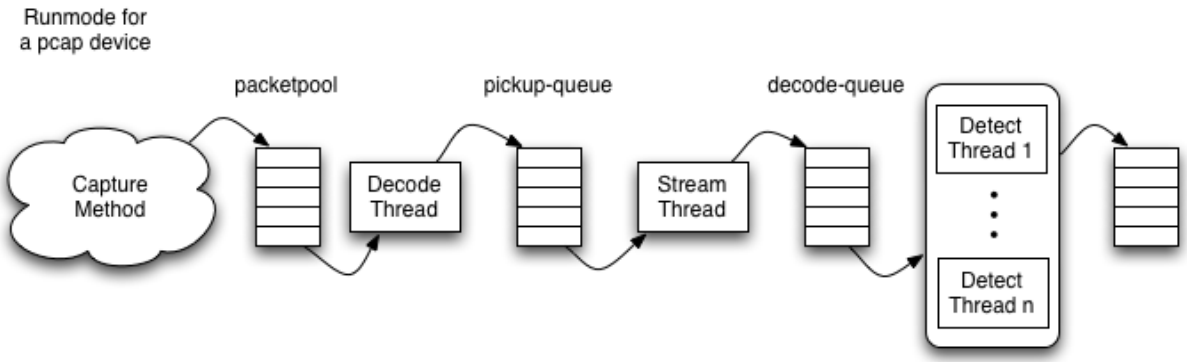
- Support Ipv6 natif
- Multi-threadée
- Accélération matérielle native (Accélération par GPU, PF_RING)
- De nombreuses options pour optimiser les performances
- Support optimisé des tests sur IP seules
- IPS (mode inline) natif

Architecture globale

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

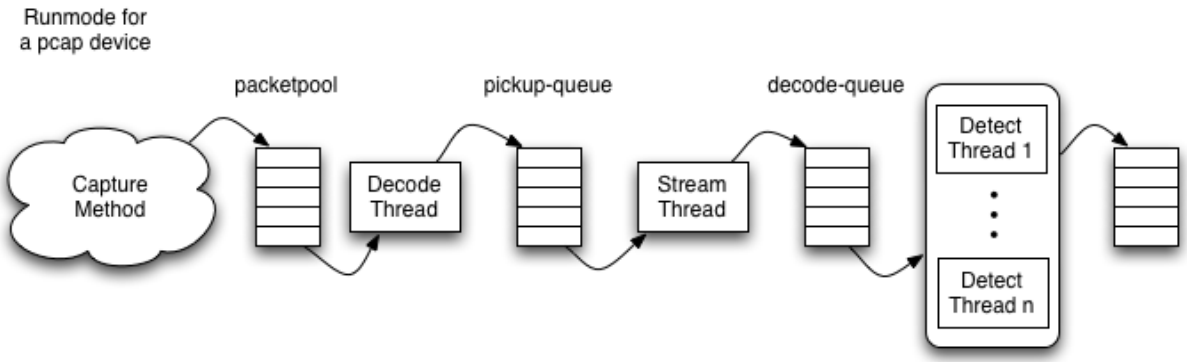
Architecture globale

- 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`

Modules d'entrées

IDS

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

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;)
```

Suricata en mode IPS

Utilisation IPS sous Linux

- Utilisation de NFQUEUE pour déléguer la décision en espace utilisateur
- Tous les paquets d'une connexion doivent être vue par Suricata
- Technique sauvage : iptables -A FORWARD -j NFQUEUE

Suricata en mode IPS

Utilisation IPS sous Linux

- Utilisation de NFQUEUE pour déléguer la décision en espace utilisateur
- Tous les paquets d'une connexion doivent être vue par Suricata
- Technique sauvage : iptables -A FORWARD -j NFQUEUE

Intéractions avec le pare-feu

- NFQUEUE est une règle terminale
  - Une décision ACCEPT courtcircuite le jeu de règle
  - C'est la seule possible ormis DROP
- La méthode décrite est donc incompatible avec l'existence d'un jeu de règles

OISF

Cohabitation de l'IPS avec le pare-feu

Solution classique

Utilisation de PREROUTING

- La règle est dans une table isolée
- Elle n'a donc pas d'intéraction avec le reste des règles

Détails : `http://home.regit.org/2011/01/building-a-suricata-compliant-ruleset/`

Cohabitation de l'IPS avec le pare-feu

Solution classique

Utilisation de PREROUTING

- La règle est dans une table isolée
- Elle n'a donc pas d'intéraction avec le reste des règles

Solution alternative

- Utilisation des fonctionnalités avancées de NFQUEUE
- Simulation de décision non terminale (© Patrick Mchardy)

Détails : `http://home.regit.org/2011/01/building-a-suricata-compliant-ruleset/`

Décision alternative et marquage des paquets

Décision alternative

- NF_REPEAT : décision non terminale
- NF_QUEUE : chainage des logiciels utilisant NFQUEUE (IPS ?)

Détails : `http://home.regit.org/2011/04/some-new-features-of-ips-mode-in-suricata-1-1beta2/`

Décision alternative et marquage des paquets

Décision alternative

- NF_REPEAT : décision non terminale
- NF_QUEUE : chainage des logiciels utilisant NFQUEUE (IPS ?)

nfq_set_mark

- Nouveau mot clé pour les signatures
- Dépôt d'une marque Netfilter sur le paquet
- Utilisation possible dans toutes les couches réseaux (QoS, routage, Netfilter)

Détails : `http://home.regit.org/2011/04/some-new-features-of-ips-mode-in-suricata-1-1beta2/`

Journalisation d'une connexion suspecte (1/2)

Objectif

- Détection d'un comportement suspecte
- Augmentation de la journalisation de la connexion

Journalisation d'une connexion suspecte (1/2)

Objectif

- Détection d'un comportement suspecte
- Augmentation de la journalisation de la connexion

Méthode

- L'alerte pose une marque Netfilter sur le paquet
- Netfilter propage la marque sur tous les paquets de la connexion
- Netfilter journalise tous les paquets marqués

Journalisation d'une connexion suspecte (2/2)

Alerte adaptée dans Suricata

```
pass tcp any any -> any any (msg:"We were expecting you"; content:"Mr Bond"; \
nfq_set_mark:0x007/0xfff;)
```

Journalisation d'une connexion suspecte (2/2)

Alerte adaptée dans Suricata

```
pass tcp any any -> any any (msg:"We were expecting you"; content:"Mr Bond"; \
nfq_set_mark:0x007/0xfff;)
```

Paramétrage de Netfilter

```
iptables -I PREROUTING -t mangle -j CONNMARK --restore-mark
iptables -A POSTROUTING -t mangle -j CONNMARK --save-mark
iptables -A POSTROUTING -t mangle -m mark --mark 0x007/0xfff -j NFLOG --nflog-prefix "Dr No log"
```

Journalisation d'une connexion suspecte (2/2)

Alerte adaptée dans Suricata

```
pass tcp any any -> any any (msg:"We were expecting you"; content:"Mr Bond"; \
nfq_set_mark:0x007/0xfff;)
```

Paramétrage de Netfilter

```
iptables -I PREROUTING -t mangle -j CONNMARK --restore-mark
iptables -A POSTROUTING -t mangle -j CONNMARK --save-mark
iptables -A POSTROUTING -t mangle -m mark --mark 0x007/0xfff -j NFLOG --nflog-prefix "Dr No log"
```

Ensuite ulogd2 envoie tout dans pcap ou SQL suivant le paramétrage

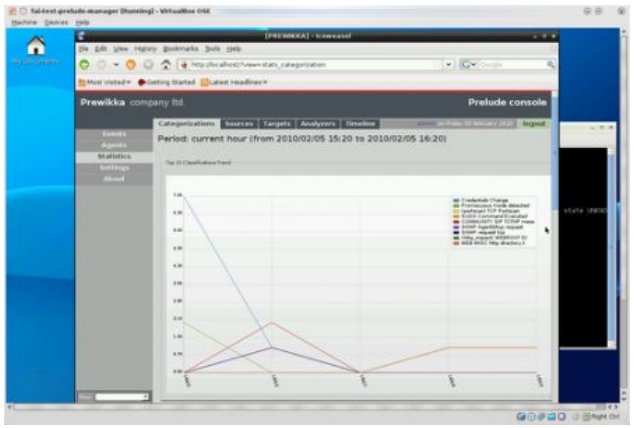
Fonctionnalités planifiées

- Finalisation de l'accélération par CUDA.
- Réputation IP et DNS
- Extraction de fichiers et leur inspection
- SCADA Preprocesseur (grâce à Digital Bond)
- Mot clé *replace*
- Mot clé *geoip*
- Rechargement du jeu de signatures sans rupture de l'analyse des flux
- Stateful Pattern Matching/Transaction-Aware Detections

Détails : `http://www.openinfosecfoundation.org/index.php/component/content/article/1-latest-news/116-oisf-state-of-the-project-report-phase-two`

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/`

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