Suricata: détection d'intrusion réseau

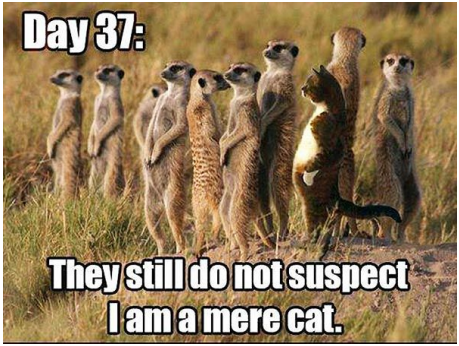
É. Leblond

Stamus Networks

15 juin 2016

STAMUS NETWORKS

Éric Leblond



- Éric Leblond aka @Regiteric
- Core développeur Suricata et Netfilter
- Co-fondateur de Stamus Networks

STAMUS NETWORKS

Stamus Networks

Sondes de détection d'intrusion réseaux

- Appliances haute performance
- Utilisant l'IDS Suricata
- Gestion centralisée par interface web

Services professionnels autour de Suricata

- Consulting
- Développements à façon
- Formations

STAMUS NETWORKS

Suricata : Intrusion Detection System

Principe

- Analyse du trafic réseaux d'une entreprise
- Recherche de motifs connus comme étant des attaques
- Émission d'alertes

Spécificités

- Logiciel libre
- Développé par l'OISF fondation à but non lucratif américaine
- Multithreadé
- Détection et analyse des protocoles

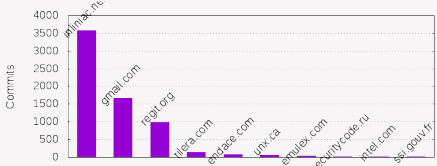
STAMUS NETWORKS

Historique de Suricata

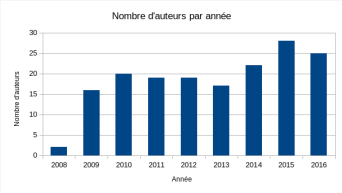
Développement de zéro

- Projet né en 2008
- Version 1.0 en 2010
- Version 3.1 en cours de préparation

Commits par domaine



Auteurs par année



Principaux commiteurs

- Victor Julien (Hollande)
- Éric Leblond (France)
- Anoop Saldanha (Inde)
- Jason Ish (Canada)

Membres du consortium OISF

Platine

proofpoint

FireEye

Or

POSITIVE TECHNOLOGIES

EVERIS
The Cornerstone of Network Security
http://www.everisinc.com

ALTERA
MEASURABLE ADVANTAGE

Bronze

PROTECTWISE
Security Enlightened

CYPHORT

Myricom

Débutants

FireDragon

BRICATA

Points clés de Suricata

Signatures

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

STAMUS NETWORKS

Signatures

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

Action : alert / drop / pass

STAMUS NETWORKS

Signatures

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

IP parameters

STAMUS NETWORKS

Signatures

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

Motif

STAMUS NETWORKS

Signatures

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

Other parameters

STAMUS NETWORKS

Signatures : sources et gestion

Sources

- Emerging Threats (Proofpoint)
- Mise à jour journalière
- ET Open : Open source
- ET Pro : Sur subscription

Gestion

- Mise à jour automatisable
- Problème des faux positifs
- Gestion régulière des alertes nécessaires
- Approche métier pour la sélection des règles

libhtp

- Un parseur HTTP orienté sécurité
- Écrit par by Ivan Ristić (ModSecurity, IronBee)
- Ajout de mots clefs
  - http_method
  - http_uri & http_raw_uri
  - http_client_body & http_server_body
  - http_header & http_raw_header
  - http_cookie
  - et quelques autres . . .
- Capable de décoder les flux compressés

STAMUS NETWORKS

Suricata : HTTP


```
{
  "timestamp": "2009-10-27T22:22:30.722151+0100",
  "flow_id": 2920622869,
  "pcap_cnt": 440389,
  "event_type": "http",
  "src_ip": "192.168.1.42",
  "src_port": 1097,
  "dest_ip": "72.32.125.159",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 6,
  "http": {
    "hostname": "www.amsterdamvisit.net",
    "url": "/images/link_div.gif",
    "http_user_agent": "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)",
    "http_content_type": "image/gif",
    "http_refer": "http://www.amsterdamvisit.net/joods_historisch_museum_amsterdam.html",
    "http_method": "GET",
    "protocol": "HTTP/1.0",
    "status": 200,
    "length": 503
  }
}
```


STAMUS NETWORKS

Utilisation des mots clefs HTTP

Exemple de signature : Chat facebook

```
alert http $HOME_NET any -> $EXTERNAL_NET any \
(
 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_hostname; \
 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; \
)
```

La signature teste :

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

STAMUS NETWORKS

Un parseur de négociation TLS

- Pas de déchiffrement
- Method
  - Analyse de la négotiation TLS
  - Parse les messages TLS
- Un parseur orienté sécurité
  - Code dédié
    - Fournit une base de code modifiable
    - Pas de dépendance externe
  - Contribué par Pierre Chifflier (ANSSI)
  - Avec un focus sécurité :
    - Resistance aux attaques (audit, fuzzing)
    - Détection d'anomalies

STAMUS NETWORKS

Un parseur de négotiations

- La syntaxe

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

- devient

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

- Interêt :
  - Pas de dépendance aux paramètres IP
  - La recherche de motif est limité aux flux du protocole
    - Moins de faux positifs
    - Plus de performance

STAMUS NETWORKS

Suricata : TLS


```
{
  "timestamp": "2009-11-24T06:45:32.794179+0100",
  "flow_id": 3317048357,
  "pcap_cnt": 1064988,
  "event_type": "tls",
  "src_ip": "192.168.1.48",
  "src_port": 1031,
  "dest_ip": "65.55.28.12",
  "dest_port": 443,
  "proto": "TCP",
  "tls": {
    "subject": "C=US, ST=Redmond, L=WA, O=Microsoft, OU=BGOS, CN=mpa.one.microsoft.com",
    "issuerdn": "C=US, ST=Washington, L=Redmond, O=Microsoft Corporation, CN=Microsoft Product Secure Server CA/emailAddress=pki@microsoft.com",
    "fingerprint": "26:ca:ea:1d:99:11:d0:14:98:ad:17:47:4a:8d:fa:94:c5:1f:53:1c",
    "version": "SSLv3"
  }
}
```

Mots clefs TLS

- *tls.version* : Match sur la version du protocole
- *tls.subject* : Match sur le sujet du certificat
- *tls.issuerdn* : Match sur le nom de la CA qui a signé la clef
- *tls.fingerprint* : Match l'empreinte du certificat
- *tls.store* : Stocke la chaine de certificats et un fichier meta sur disque

STAMUS NETWORKS

Exemple : vérification de politique de sécurité (1/2)

- Environnement :
  - Un organisme avec des serveurs
  - et une PKI officielle

STAMUS NETWORKS

Exemple : vérification de politique de sécurité (1/2)

- Environnement :
  - Un organisme avec des serveurs
  - et une PKI officielle
- L'objectif :
  - Verifier que la PKI est utilisé

STAMUS NETWORKS

Exemple : vérification de politique de sécurité (1/2)

- Environnement :
  - Un organisme avec des serveurs
  - et une PKI officielle
- L'objectif :
  - Verifier que la PKI est utilisé
  - Sans trop travailler

STAMUS NETWORKS

Exemple : vérification de politique de sécurité (2/2)

- Vérifions que les certificats utilisés quand un client négocie une connexion vers un de nos serveurs sont bien signés par notre CA.

STAMUS NETWORKS

Exemple : vérification de politique de sécurité (2/2)

- Vérifions que les certificats utilisés quand un client négocie une connexion vers un de nos serveurs sont bien signés par notre CA.
- La signature :

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

STAMUS NETWORKS

Au diable les années 90

Débarassons nous de unified2

- Standard de facto pour les alertes
- Format binaire
- Difficile à etendre
- Pas d'API

Nous avons besoin de quelque chose d'extensible

- Pour journaliser les alertes ET les événements protocolaires
- Facile à générer et à parser
- Extensible

STAMUS NETWORKS

JavaScript Object Notation

JSON

- JSON (`http://www.json.org/`) est un format léger pour l'échange de données.
- Facile à lire et écrire pour un humain.
- Facile à parser et générer pour une machine.
- Un objet est un ensemble non ordonné de clef/valeur.

Journalisation en JSON

```
{"timestamp":"2012-02-05T15:55:06.661269", "src_ip":"173.194.34.51",
 "dest_ip":"192.168.1.22",
 "alert":{"action":"allowed",rev":1,"signature":"SURICATA TLS store"}}
```

STAMUS NETWORKS

Alerte

La structure

- Les information Ip sont identiques pour tous les événements
- Suit le Common Information Model
- Permet une aggrégation simple des événements Suricata avec les sources externes

Example


```
{"timestamp":"2014-03-06T05:46:31.170567","event_type":"alert",
 "src_ip":"61.174.51.224","src_port":2555,
 "dest_ip":"192.168.1.129","dest_port":22,"proto":"TCP",
 "alert":{"action":"Pass","gid":1,"signature_id":2006435,"rev":8,
          "signature":"ET SCAN LibSSH Based SSH Connection - Often used as a BruteForce Tool"
          "category":"Misc activity","severity":3}
}
```


STAMUS NETWORKS

Network Security Monitoring

Protocoles

- HTTP
- File
- TLS
- SSH
- DNS
- SMTP

Exemple


```
{"timestamp":"2014-04-10T13:26:05.500472","event_type":"ssh",
 "src_ip":"192.168.1.129","src_port":45005,
 "dest_ip":"192.30.252.129","dest_port":22,"proto":"TCP",
 "ssh":{
  "client":{
    "proto_version":"2.0","software_version":"OpenSSH_6.6p1 Debian-2" },
  "server":{
```

Elasticsearch, Logstash et Kibana

Elasticsearch est un moteur de recherche

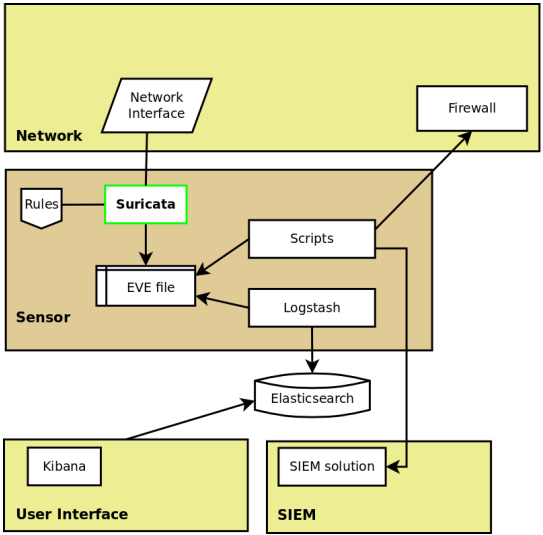
- distribué (architecture de type cloud computing)
- utilise une base de données NoSQL
- utilise la méthode REST

Une série d'outils

- Elasticsearch
- Logstash : gestion des journaux et transfert sur des sorties variées dont Elasticsearch
- Kibana : interface web de consultation des données stockées dans Elasticsearch

STAMUS NETWORKS

Un écosystème Suricata type



STAMUS NETWORKS

Amsterdam

SELKS : une ISO live et installable

- Suricata : IDS/NSM open source
  - Détection basée sur des signatures
  - Reconnaissance et analyse protocolaire
- Elasticsearch : Splunk gratuit
- Kibana : interface de tableau de bord
- Logstash : collecte, transformation, transfert
- Scirius : Gestion des jeux de signatures

Docker

- Gestion de containers sous Linux, Windows, Mac OSX
- Orchestration via docker compose

Installation d'Amsterdam

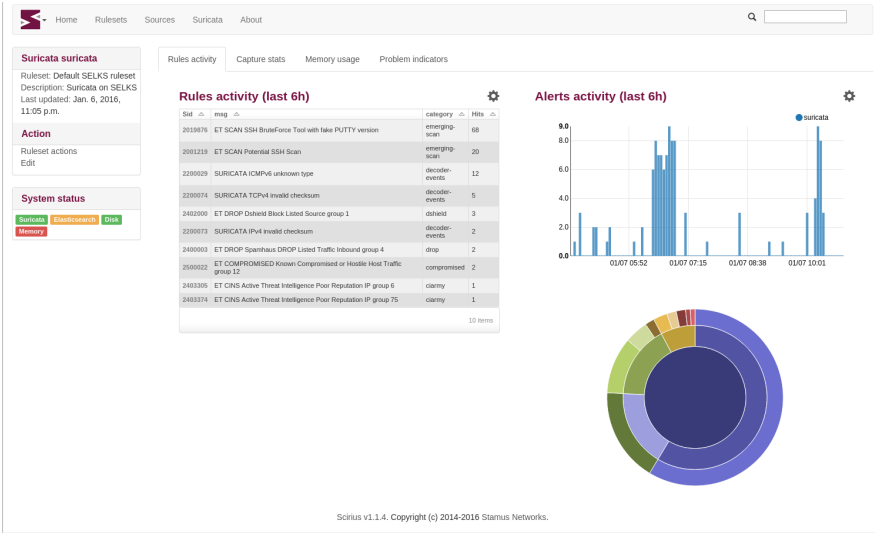
Installation

```
pip install amsterdam
# verification optionnelle de la version
pip show amsterdam
# creation de l'instance dans le repertoire ams
amsterdam -d ams -i wlan0 setup
# demarrage de l'instance
amsterdam -d ams start
```

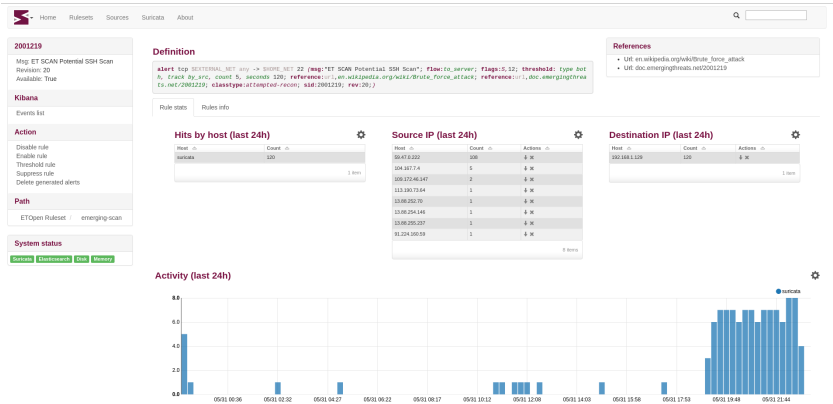
Utilisation

Pointer le navigateur sur `https://localhost/` ou sur l'IP réseau depuis une machine externe.

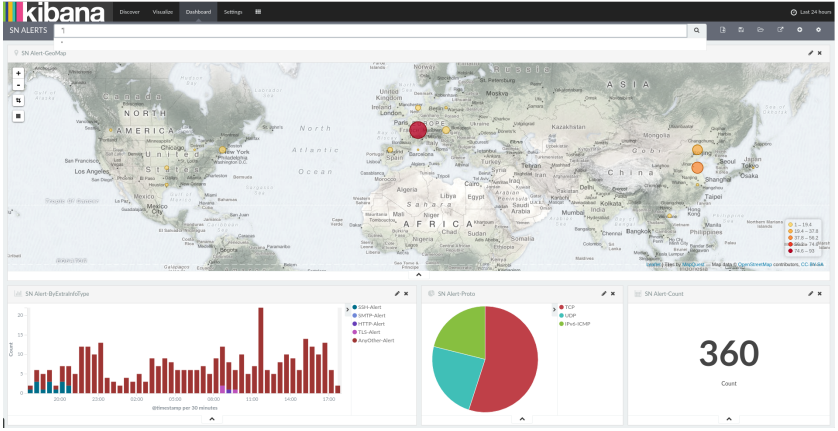
Scirius : page d'accueil (1/2)



Scirius : page d'une signature (2/2)



Kibana


kibana
Discover
Visualize
Dashboard
Settings
Last 24 hours
SN ALERTS
SN Alert-GeoMap
NORTH AMERICA
United States
Canada
Mexico
Europe
ASIA
China
India
AFRICA
Sahara
Atlantic Ocean
SN Alert-ByExtraInfoType
SSH-Alert
SMTP-Alert
HTTP-Alert
TLS-Alert
AnyOther-Alert
Count
@timestamp per 30 minutes
SN Alert-Proto
TCP
UDP
IPv6-ICMP
SN Alert-Count
360
Count

Indicateurs de compromission

IOC

- Traces permettant de trouver une compromission
- Système
  - Clef de registres
  - Fichiers
- Réseau
  - Signature IDS
  - Noms de domaine
  - Adresse IP

STAMUS NETWORKS

Indicateurs de compromission

IOC

- Traces permettant de trouver une compromission
- Système
  - Clef de registres
  - Fichiers
- Réseau
  - Signature IDS
  - Noms de domaine
  - Adresse IP

Partage d'IOC

- Sources publique et privée
- Sources communautaire (MISP)
- Sources étatique

Vérification des IOC

- Sur le traffic en temps réél avec les signatures
- A posteriori dans les logs protocolaires

Conclusion

Suricata

- Open Source
- Communautaire
- Performant
- IDS et NSM

Plus d'information

- Suricata : `http://www.suricata-ids.org/`
- Conférence utilisateurs Suricata : `http://suricon.net/`
- Formation développeur Suricata : Paris, 12-16 Septembre `https://goo.gl/9tYbWP`
- Amsterdam : `https://github.com/StamusNetworks/Amsterdam`
- Stamus Networks : `https://www.stamus-networks.com/`