RéSIST

I.D.S. : Snort

- Définitions
- Snort
- Implémentation
- Conclusions

SNORT™

RéSIST

Définitions

- I.D.S : Intrusion Detection System
  - Détection d'attaques
  - En coupure ou en parallèle (sniffer)
- I.P.S : Intrusion Protection System
  - Evolution de l'I.D.S. action

SNORT™

RéSIST

Snort

- http://www.snort.org
- Détecteur d'intrusion Open Source
- Unix et Windows
- Auteurs : Marty Roesch & Brian Caswell

SNORT™

RéSIST

Snort

- Placé en tant que Sniffeur
- Repère des signatures d'attaques
- Repère les scans de ports rapides (redondance avec un FW)

SNORT™

Evolution

- 1.x
  - Version initiale
- 2.0
  - Notion de suppression/seuil de règles
  - Amélioration des suivis de connexion
- 2.1
  - Apparition d'expressions régulières (PCRE) dans les règles

Points forts

- Open Source
- Large communauté d'utilisateurs
  - Beaucoup de contributions
  - Beaucoup de documentations
- Bonne base de signatures
  - Mise à jour
  - Modifiable

RéSIST

Fonctionnement

- Préprocesseurs (avec leur propre ID)
  - defragmenteurs
  - http_inspect
  - rpc_decode, etc.
- Postprocesseurs
  - syslog
  - mysql
  - tcpdump

SNORT™

RéSIST

Règles

format

- type action (alert, + configurable)
- contexte (protocole, origine, destination)
- message
- contenu, sensibilité à la casse
- type de flux
- type d'attaque
- références (SID, révision, urls, etc.)

SNORT™

RéSIST

Exemple de règle

```
alert tcp $HOME_NET 22 ->
  $EXTERNAL_NET any (msg:"ATTACK-
  RESPONSES successful gobbles ssh
  exploit(GOBBLE)";
flow:from_server,established;
content:"|2a|GOBBLE|2a|";
reference:bugtraq,5093;
classtype:successful-admin;
sid:1810; rev:3;)
```

RéSIST

Autour de Snort

- Analyse des résultats
- Transformation en I.P.S.

RéSIST

Analyse des résultats

- Acid
- Pigsentry (Temps réel)
- SnortSnarf
- Snortalog (graphiques)

SNORT™

RéSIST

Transformation en I.P.S.

- Flexresp (intégrée)
  - bloque une action par TCP/RST
- Snort-inline : modification de snort
  - http://sourceforge.net/projects/snort-inline/
- Guardian (ajout de règles iptables)

SNORT™

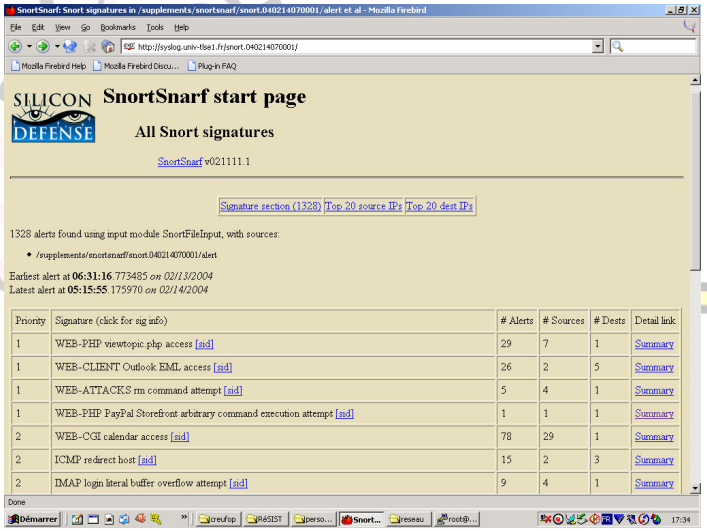
Implémentation à l'UT1

- Placé sur la patte interne du firewall
- Analyse en décalé sur une autre machine SnortSnarf
- Les difficultés rencontrées
- Repère quelques attaques fines

SnortSnarf

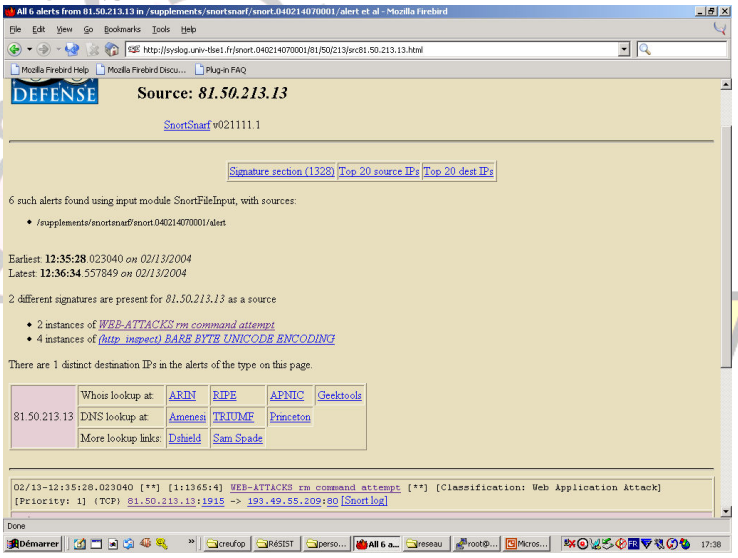
- http://www.silicondefense.com/software/snortsnarf/
- Présentation Web
- Présentation des signatures détectées
- Présentation des signatures par machine
- Visualisation des paquets fautifs

SnortSnarf

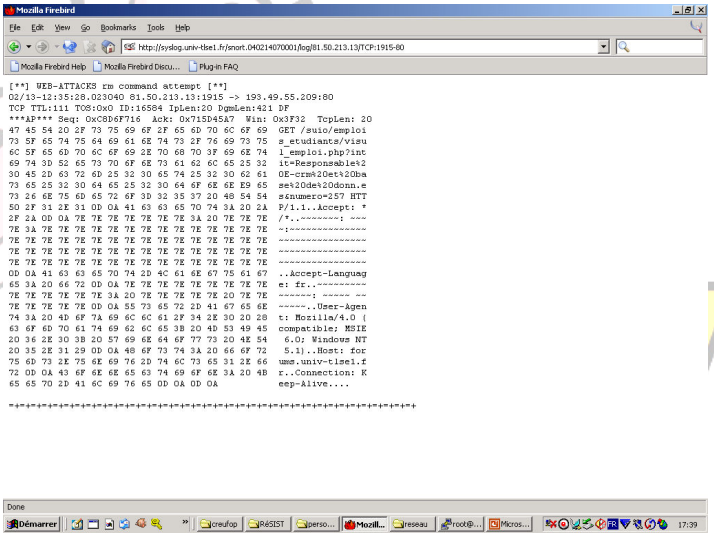


SnortSnarf

SnortSnarf



SnortSnarf



Les difficultés rencontrées

- Les faux positifs
- L'emplacement de la sonde
- La mise à jour (changement des noms de règles)
- Consommation

Les faux positifs

- Règles « inutiles » : ICMP, fingerprinting
  - Difficulté à conserver les renseignements utiles
- Très forte quantité (90%)
  - Analyse fine des applications non présentes
  - Gros travail de régulation et de nettoyage (2.x)
- Mais, les faux persistent (règles trop générales)
- Importance de la présentation des résultats

Limitations des règles I

On peut
- limiter les alertes (threshold type limit)
- détecter sur un seuil (threshold type threshold)
- annuler les règles (suppress)

On peut le faire
- de manière globale (snort.conf)
- dans la définition des règles

On peut préciser
- origine ou la destination (ip ou CIDR)

Limitations des règles II

Seuil
- threshold gen_id 1, sig_id 448, type limit, track by_src, count 1, seconds 300
- pour une même IP source ICMP Source Quench pas plus de 1 par 5 minutes

Suppression
- suppress gen_id 1, sig_id 526, track by_dst, ip 193.49.48.249
- pas d'alerte pour la signature 526 pour la destination 193.49.48.249

RéSIST

L'emplacement de la sonde

- Avant ou après le firewall ?
- Avant :
  - Prévision
  - Analyse comportementale des pirates
- Après :
  - Détection des failles utilisées

SNORT™

RéSIST

La mise à jour des règles

- 2400 règles disponibles
- Plutôt réactif
- Détection de shellcodes
- Tendance des auteurs à changer les noms des règles :-(

SNORT™

RéSIST

Consommation CPU/RAM

- Sonde
  - Pour 500 Ko/seconde
  - 4 % de CPU sur un pentium III à 933Mhz
  - 40-60 Mo de RAM
- Analyse
  - Très consommateur
  - Très fluctuant (>500 Mo de RAM possible)
  - Attention à la résolution DNS

SNORT™

RéSIST

•Consommation disque

- 30 -100 Mo de dump par jour
- 1-10 Mo d'alertes par jour
- 0.5 à 2 Mo de pages Web par jour
- Quelques pointes à 20 Mo de Web

SNORT™

Consommation RSSI

- Lancement
  - 1 journée de paramétrage
  - 30 minutes par jour pendant 15 jours
- Rythme de croisière
  - 5-15 minutes par jour (si correctif)
- Accidents de parcours
  - 3 heures (chgt de règles, attaque imprévue)

Conclusions

- Ne remplace par un firewall
- Permet de
  - Détecter les intrusions
  - Détecter les pirates
- Facile à installer
- Travail long et attentif pour ne pas noyer le RSSI