



***THOMSON***  
*images & beyond*

# SinFP, unification de la prise d'empreinte active et passive des systèmes d'exploitation

SSTIC, Rennes, 6 Juin 2008 – Patrice AUFFRET



# Plan

---

- **Introduction à la prise d'empreinte (OSFP)**
- **Limitations des outils actuels**
- **Les nouveautés apportées par SinFP**
- **Détails d'implémentation (mais pas trop)**
- **Conclusion**

# Sommaire

---

- **Introduction à la prise d'empreinte (OSFP)**
- Limitations des outils actuels
- Les nouveautés apportées par SinFP
- Détails d'implémentation (mais pas trop)
- Conclusion

# Introduction à la prise d'empreinte

---

- **Qu'est-ce que c'est exactement ?**
  - C'est l'art d'identifier la nature d'un système d'exploitation par l'analyse de la manière dont il construit ses paquets réseau
- **Deux grands modes**
  - Mode actif
    - Envoi de requêtes (tests, *probes*) afin de provoquer des réponses
    - La construction de ces requêtes est contrôlée par l'analyste
  - Mode passif
    - Ecoute/analyse des paquets réseau (*sniffing*)
    - Les requêtes qui ont provoqué l'émission de ces réponses ne sont pas contrôlée par l'analyste
- **Les empreintes obtenues de manières active sont différentes de celles obtenues de manière passive**

# Introduction à la prise d'empreinte (suite)

---

- **Mais à quoi ça sert ?**

- Tests d'intrusion : pour acquérir de l'information sur la cible et pouvoir exécuter le bon *shellcode* pour le bon système lors d'utilisation d'*exploits*
- Audits de sécurité : ajouter des lignes dans des rapports presque vides
- C'est marrant 😊

- **Pourquoi ne pas simplement se baser sur les bannières ?**

- Quand on a le choix, il vaut mieux utiliser les bannières 😊
- Ou corrélérer les bannières avec l'OSFP

# Sommaire

---

- Introduction à la prise d'empreinte (OSFP)
- **Limitations des outils actuels**
- Les nouveautés apportées par SinFP
- Détails d'implémentation (mais pas trop)
- Conclusion

# Limitation des outils actuels

---

- **Utilisation de nombreux tests utilisant des protocoles différents**
  - Ayant de grandes chances d'être filtrés
  - Très détectables par les IDS
- **Utilisation de paquets non standards**
  - Ayant de grandes chances d'être filtrés
  - Ayant des chances de faire « crasher » les cibles
  - Très détectables par les IDS
- **Utilisation de tests ciblant différents ports**
  - Ayant des chances de provoquer une réponse provenant d'une cible différente
  - Détectables par les IDS
- **Uniquement IPv4 (plus que 1000 jours)**



# Limitation des outils actuels (suite)

---

- **Leur base de signature est**
  - Soit une base de signatures actives
  - Soit une base de signatures passives
    - Nécessité de faire évoluer deux bases au lieu d'une
  - En effet, les empreintes prises de manière passive sont différentes des empreintes prises de manière active
- **Base de signatures passives**
  - Difficile à compléter
- **Base de signatures actives**
  - Nécessitant de nombreuses signatures différentes pour un unique système

# Sommaire

---

- Introduction à la prise d'empreinte (OSFP)
- Limitations des outils actuels
- **Les nouveautés apportées par SinFP**
- Détails d'implémentation (mais pas trop)
- Conclusion

# Les nouveautés apportées par SinFP

---

- **Utilise au maximum 3 paquets**
  - Tous ciblant le même port (un port TCP ouvert)
    - Détection de certaines configuration PAT/NAT (un OS par port ouvert)
  - Tous standards
  - Quasiment indétectables par les IDS
- **Une seule base de signatures**
  - Contient des signatures actives uniquement
  - L'algorithme de recherche de correspondance du mode passif utilise les signatures prises de manière active
  - L'algorithme de recherche de correspondance utilise un mode heuristique ne nécessitant que peu de signatures
  - Grâce aux masques de déformations
- **Prise d'empreinte sur IPv4 et IPv6**

# Sommaire

---

- Introduction à la prise d'empreinte (OSFP)
- Limitations des outils actuels
- Les nouveautés apportées par SinFP
- **Détails d'implémentation (mais pas trop)**
- Conclusion

# Détails d'implémentation

---

- **L'approche SinFP (concernant l'OSFP actif)**
  - Se placer dans les pires conditions réseau
    - 1 seul port TCP ouvert
    - Filtrage à inspection de l'état
    - Dispositif de normalisation de paquets (*Packet Scrubbing*)
- **Solution**
  - Obligation d'utiliser des paquets standards
  - Obligation d'utiliser des paquets TCP à destination d'un port ouvert
  - Obligation d'utiliser des paquets qui vont provoquer l'émission d'une réponse

# Détails d'implémentation (suite)

---

- **3 requêtes (tests, *probes*)**

- 1 paquet TCP SYN sans option TCP
- 1 paquet TCP SYN avec de nombreuses options TCP
- 1 paquet TCP SYN+ACK

- **Pourquoi 3 tests ?**

- Pour avoir une signature la plus discriminante possible
  - Nous aurions pu nous contenter du test TCP SYN + options TCP
- Etant donné les contraintes de départ, difficile d'avoir plus de requêtes
- Moins il y a de tests, moins on affole les IDS

# Détails d'implémentation (suite)

---

- **Création de l'empreinte de la cible**
  - Lancement des 3 requêtes
  - Obtention de tout ou partie des réponses
  - Analyse des champs pour construction de l'empreinte
    - Pour le détail, voir les actes de la conférence (ou dans le code)
- **L'empreinte possède 15 éléments (3 x 5)**

P1(R) : B11113 F0x12 W65535 00204ffff M1460

P2(R) : B11113 F0x12 W65535

00204ffff010303030402080affffff44454144 M1460

P3(R) : B11120 F0x04 W0 00 M0

# Détails d'implémentation (suite)

---

- **Une empreinte peut varier**
  - A cause des conditions réseau
  - A cause de dispositifs de filtrage/routage en coupure
  - A cause de la personnalisation de la pile TCP/IP
  
- **D'où l'introduction des masques de déformation**
  - Applicables sur une signature lors d'une recherche de correspondance dans la base
  - Personnalisables en fonction de l'expérience humaine
  - Grâce aux masques, inutile d'ajouter une signature pour chaque condition réseau existante



# Détails d'implémentation (suite)

---

- **Chaque type d'élément possède 3 masques implémentés sous forme d'expressions rationnelles**
  - Types d'éléments : B, F, W, O, M
  - Exemple de masque pour l'élément O (options TCP) :
    - **OH0** : O0204ffff0402080affffffff4445414401030306
    - **OH1** : O0204ffff(?:0402)?(?:080affffffff44454144)?(?:01)?(?:03030.)?
    - **OH2** : identique.
  - Exemple de masque pour l'élément W (*window size*):
    - **WH0** : W5792
    - **WH1** : W5[678]..
    - **WH2** : W\d+

# Détails d'implémentation (suite)

---

- **Un masque complet est une suite de masques d'éléments**
  - Exemple : BH1FH0WH1OH0MH1
- **Dans SinFP, une liste de masques est présente, classée du masque le moins déformant vers le plus déformant**
  - Moins déformant : BH0FH0WH0OH0MH0 (HEURISTIC0)
  - Plus déformant : BH2FH2WH2OH2MH2 (HEURISTIC2)
- **Ce sont ces masques qui permettent l'unification de la prise d'empreinte active avec la prise d'empreinte passive**
- **Ce sont ces masques qui permettent d'avoir une signature de référence dans la base, au lieu d'en avoir plusieurs**

# Détails d'implémentation (suite)

---

- **Déformation d'une empreinte par application successive de masques**

- Empreinte dans les conditions parfaites (HEURISTIC0) :

```
B10113 F0x12 W5840 00204ffff M1460
```

```
B10113 F0x12 W5792
```

```
00204ffff0402080affffff4445414401030306 M1460
```

```
B10120 F0x04 W0 00 M0
```

- C'est la signature de référence, telle que stockée en base

# Détails d'implémentation (suite)

---

- **Déformation d'une empreinte par application successive de masques (suite)**

- Empreinte avec le masque BH1FH0WH1OH0MH1 :

**B...13 F0x12 W5[789].. 00204ffff M1[34]..**

**B...13 F0x12 W5[678]..**

**00204ffff0402080affffff4445414401030306 M1[34]..**

**B...20 F0x04 W0 00 M0**

# Détails d'implémentation (suite)

---

- Déformation d'une empreinte par application successive de masques (suite)
  - Empreinte avec le masque BH1FH0WH2OH1MH2 :

**B...13 F0x12 W\d+ 00204ffff M1[34]..**

**B...13 F0x12 W\d+ 00204ffff(?:0402)?**

**(?:080affffff44454144)?(?:01)?(?:03030.)?**

**M1[34]..**

**B...20 F0x04 W0 00 M0**

# Détails d'implémentation (suite)

---

- **L'algorithme de recherche de correspondance**
  - Pour chaque élément de chaque réponse, nous recherchons la liste des signatures possibles (un domaine)
  - Pour chaque réponse, l'intersection des domaines de chaque élément ayant trouvé une correspondance nous donne la liste des signatures correspondantes
- **Pour chaque réponse, nous obtenons une liste de signatures (un domaine) ayant trouvé une correspondance dans la base**
- **L'intersection de ces domaines nous donne la liste des signatures correspondantes à l'empreinte**
- **Si aucun résultat n'est trouvé, une nouvelle recherche de correspondance est lancée avec le masque de déformation suivant**

# Détails d'implémentation (suite)

---

- **L'algorithme de recherche de correspondance (suite)**
  - $I(P1) = E1(P1) \cap E2(P1) \cap \dots \cap E5(P1)$
  - $I(P2) = E1(P2) \cap E2(P2) \cap \dots \cap E5(P2)$
  - $I(P3) = E1(P3) \cap E2(P3) \cap \dots \cap E5(P3)$
  - $I = I(P1) \cap I(P2) \cap I(P3)$
- **Si I est nul :**
  - $I = I(P1) \cap I(P2)$
- **Si I est encore nul :**
  - $I = I(P2)$

# Détails d'implémentation (suite)

---

- **Prise d'empreinte passive**
- **Contraintes de départ**
  - Utiliser la base de signatures existante
  - Utiliser l'algorithme de recherche existant
- **Solution**
  - Modifier les paquets capturés pour qu'ils ressemblent à des paquets de réponse du mode actif



# Détails d'implémentation (suite)

---

- **Prise d'empreinte passive (suite)**
- **Dans le mode actif**
  - Les paquets TCP SYN+ACK et RST+ACK sont traités
- **Dans le mode passif**
  - Les paquets TCP SYN+ACK et SYN sont traités
  - Mais les TCP SYN sont transformés pour ressembler à des TCP SYN+ACK lors de l'analyse
  - Certaines analyses ne sont pas possible, mais les masques de déformation permettent de contourner le problème

# Détails d'implémentation (suite)

---

- **Prise d'empreinte passive (suite)**
- **L'algorithme de recherche de correspondance devient**
  - $I = E1(P2) \cap E2(P2) \cap \dots \cap E5(P2)$
- **La recherche de correspondance ne se basant que sur un unique paquet**

# Détails d'implémentation (suite)

---

- **Pour en savoir plus sur l'outil**
  - <http://www.gomor.org/bin/view/Sinfp>
- **Une démo en ligne**
  - <http://www.gomor.org/bin/view/Sinfp/SinfpDemo>
- **La liste de diffusion**
  - <http://lists.gomor.org/mailman/listinfo/sinfp>

# Sommaire

---

- Introduction à la prise d'empreinte (OSFP)
- Limitations des outils actuels
- Les nouveautés apportées par SinFP
- Détails d'implémentation (mais pas trop)
- **Conclusion**

# Conclusion

---

- **Les fonctions suivantes n'ont pas été décrites ici**
  - Comment les paquets sont analysés
  - Prise d'empreinte sur IPv6 (actif et passif)
  - Les limitations des modes actif et passif
  - Le format de la base de signature
- **Pour ces informations, consultez les actes**

# Questions ?



# Merci de votre attention

This document is for background informational purposes only. Some points may, for example, be simplified. No guarantees, implied or otherwise, are intended

