Histoire d'une technique pour faire des trous d'épingles ou la difficile gestion d'un problème de sécurité

Éric Leblond

OISF

Kernel Recipes 2012

Eric Leblond (Regit)

- Spécialiste de la sécurité des réseaux
- Créateur du projet NuFW, co-fondateur de la société EdenWall
- Auteur de coccigrep, un grep sémantique basé sur coccinelle
- Développeur Netfilter :
  - Ulogd2: démon de journalisation de Netfilter
  - Contributions diverses:
    - Bibliothèques NFQUEUE et dépendances.
    - Travail sur le journalisation.
- Actuellement :
  - Consultant indépendant en sécurité
  - Développeur financé de l'IDS/IPS Suricata

Une "nouvelle" technique d'attaque

Des couches OSI indépendantes

- Les couches 2 et 3 sont traités indépendamment
- Il est donc possible d'usurper une *identité* du niveau 3
- En utilisant une couche 2 valide

Une "nouvelle" technique d'attaque

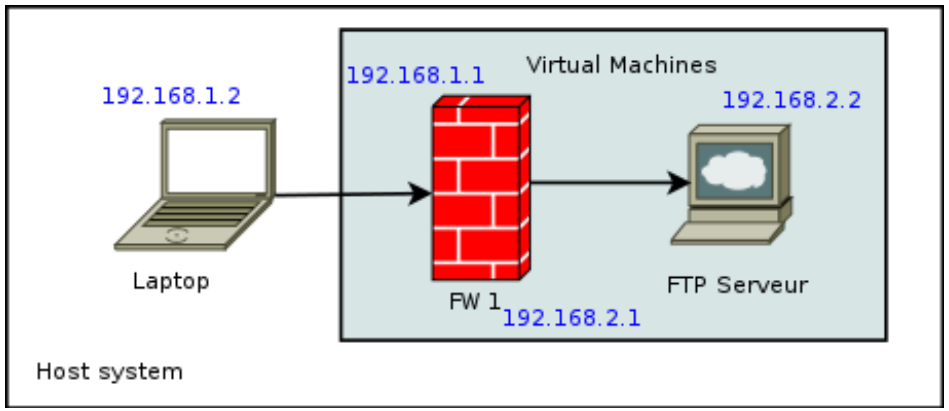
Des couches OSI indépendantes

- Les couches 2 et 3 sont traités indépendamment
- Il est donc possible d'usurper une *identité* du niveau 3
- En utilisant une couche 2 valide

Une attaque locale

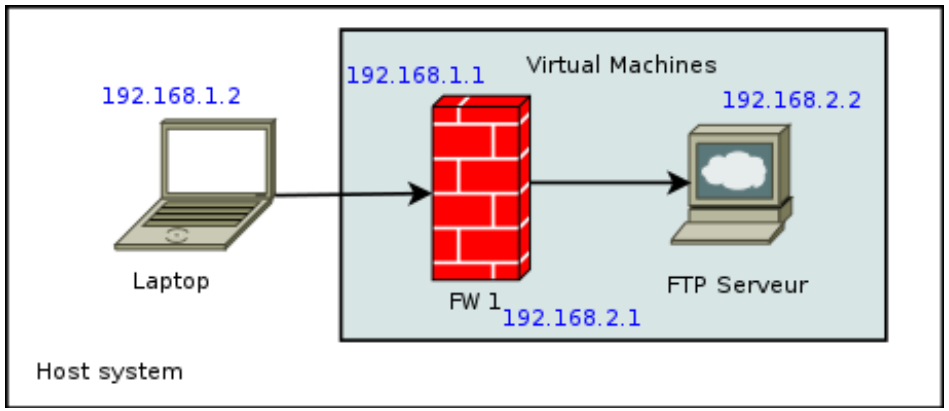
- L'attaquant doit être connecté sur le même réseau Ethernet que le pare-feu
- Pas d'ouverture distante possible

Essai sur Netfilter



Video

Essai sur Netfilter



Video

Prenons un pare-feu avec une politique de filtrage autorisant seulement le port 21 et ouvrons une connexion vers le port 22 du serveur FTP.

Violation de la politique de sécurité

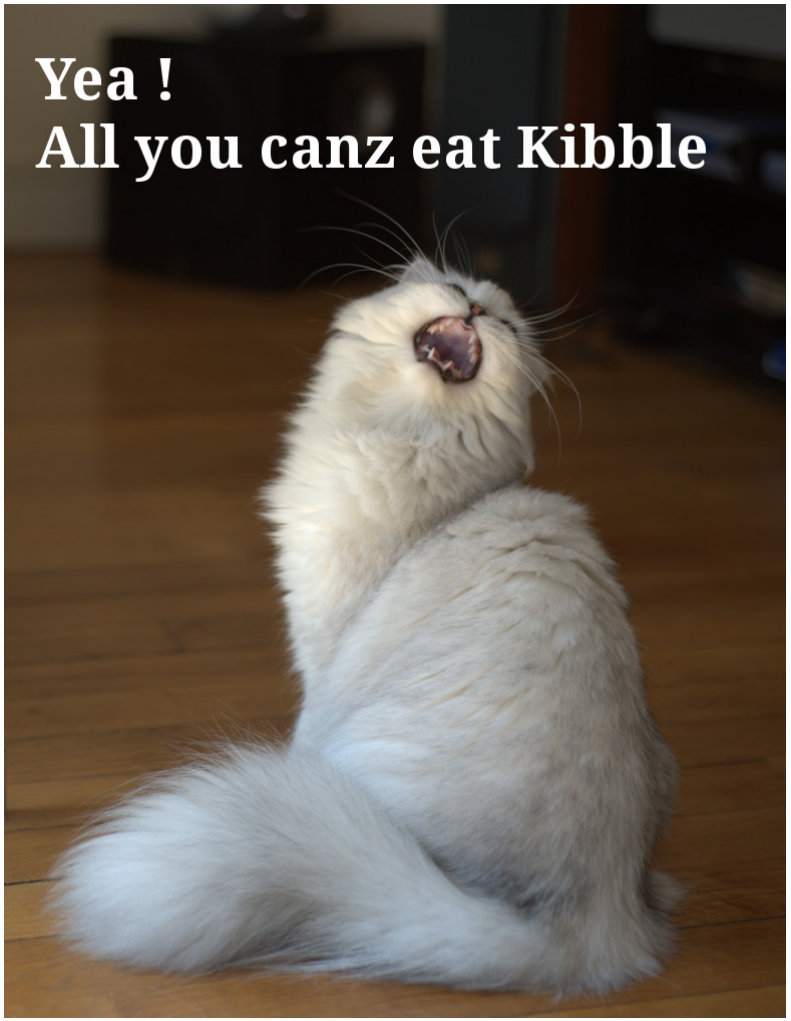
- Nous sommes parvenus à ouvrir une connexion sur le port 22.

Violation de la politique de sécurité

- Nous sommes parvenus à ouvrir une connexion sur le port 22.
- Avec une politique de filtrage ne le permettant pas.

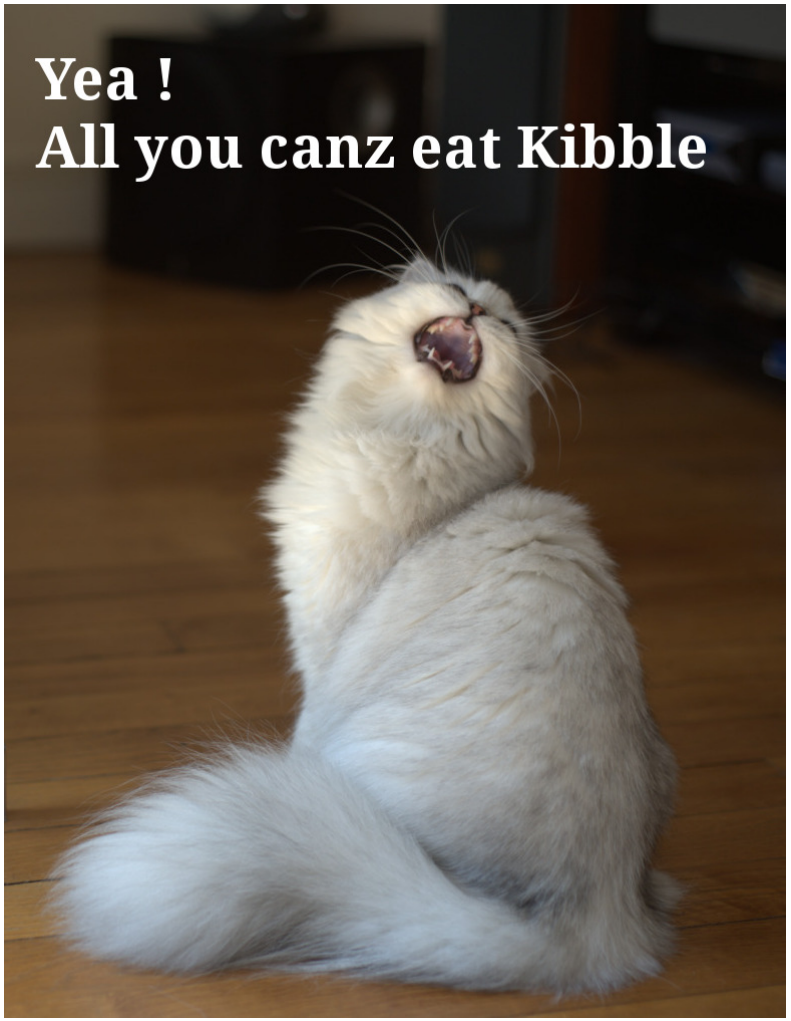
Violation de la politique de sécurité

- Nous sommes parvenus à ouvrir une connexion sur le port 22.
- Avec une politique de filtrage ne le permettant pas.



Violation de la politique de sécurité

- Nous sommes parvenus à ouvrir une connexion sur le port 22.
- Avec une politique de filtrage ne le permettant pas.
- Tout doux, chaton, tout doux!



Contre-mesures

- L'anti-spoofing est suffisant pour bloquer l'attaque.
- *Reverse path filtering* est notre ami:
  - Accepter seulement les paquet arrivant sur une interface si on a une route vers cette source passant par cette interface.
  - Le noyau ne traitera alors pas le paquet d'attaque.
- C'est si facile d'être protégé?

Contre-mesures

- L'anti-spoofing est suffisant pour bloquer l'attaque.
- *Reverse path filtering* est notre ami:
  - Accepter seulement les paquet arrivant sur une interface si on a une route vers cette source passant par cette interface.
  - Le noyau ne traitera alors pas le paquet d'attaque.
- C'est si facile d'être protégé? **Oui**

Contre-mesures

- L'anti-spoofing est suffisant pour bloquer l'attaque.
- *Reverse path filtering* est notre ami:
  - Accepter seulement les paquet arrivant sur une interface si on a une route vers cette source passant par cette interface.
  - Le noyau ne traitera alors pas le paquet d'attaque.
- C'est si facile d'être protégé? Oui
- Mais il reste quelques surprises.

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité **rp_filter**.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- **Désactivée par défaut**.

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse
  - Activée par tous les scripts de filtrage décents

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse
  - Activée par tous les scripts de filtrage décents
- Pour l'activer:

```
echo "1" > /proc/sys/net/ipv4/conf/all/rp_filter
```

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse
  - Activée par tous les scripts de filtrage décents
- Pour l'activer:

```
echo "1"> /proc/sys/net/ipv4/conf/all/rp_filter
```

- **Hummmmm**

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse
  - Activée par tous les scripts de filtrage décents
- Pour l'activer:

```
echo "1"> /proc/sys/net/ipv4/conf/all/rp_filter
```

- **Hummmmm** et pour IPv6 ?

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse
  - Activée par tous les scripts de filtrage décents
- Pour l'activer:

```
echo "1"> /proc/sys/net/ipv4/conf/all/rp_filter
```

- **Hummmmm** et pour IPv6 ?
- Trop facile, positionnons la valeur dans `/proc`:

```
echo "1"> /proc/sys/net/ipv6/conf/all/rp_filter
  /proc/sys/net/ipv6/conf/all/rp_filter: No such file or directory
```

Protection de Netfilter

- Il suffit d'utiliser la fonctionnalité rp_filter.
- Elle est disponible depuis le siècle dernier dans tout les noyaux Linux.
- Désactivée par défaut.
  - Une vérification de routage pour chaque paquet est trop couteuse
  - Activée par tous les scripts de filtrage décents
- Pour l'activer:

```
echo "1"> /proc/sys/net/ipv4/conf/all/rp_filter
```

- Hummmmm et pour IPv6 ?
- Trop facile, positionnons la valeur dans `/proc`:

```
echo "1"> /proc/sys/net/ipv6/conf/all/rp_filter
  /proc/sys/net/ipv6/conf/all/rp_filter: No such file or directory
```

Okay, Houston, we've had a problem here.

(Jack Swigert)

Phase 1 (mai 2011) : contacter les développeurs

- Envoi d'un mail à des membres de la Coreteam de Netfilter
- Ils reconnaissent le problème
- Des règles de filtrage manuelles permettent de prévenir l'attaque
- Nécessité de communiquer sur le sujet

Phase 2 (juin 2011)

Prévenir les développeurs de frontend

- Les implémentations testées sont vulnérables en IPv6
- Les développeurs sont très coopératifs

Prévenir les gens du noyau

- La solution de protection manuelle est complexe à mettre en oeuvre
- Envoi d'un mail à security@kernel.org pour évoquer la nécessité de rp_filter pour IPv6
- Réponse d'un des principaux développeurs :

```
Security issues are just bugs, and we report bugs on the public
mailing list and try to fix them.
```

Phase 3 (juin 2011) : proposer un patch

Utilisation d'un code déjà proposé

- Code déjà proposé en 2006 par Denis Semmau
- Rebasé sur la branche net-next
- Eric Dumazet me signale que la RFC sur le sujet n'est pas respectée
- Je modifie le patch en conséquence et resoumet

David Miller n'aime pas

- Mauvais pour les performances
- Ce type de sécurité doit être fait dans Netfilter
- Les détails: http://permalink.gmane.org/gmane.linux.network/197607

Phase 5 (juin 2011) : contacter les CERTs

Une attaque à prendre en compte

- Nécessité de protection sur les frontends
- Et sur les scripts maisons
- Une coordination de haut niveau semble parfaite
  - Contacter de nombreux acteurs
  - Faire des annonces suffisamment publiques pour attirer l'attention

Les CERTs refusent de traiter

- Le CERT français ne m'a même par répondu (malgré relance)
- Le CERT US a refusé de traiter
  - Alors que l'attaque est générique et
  - Que des pare-feu comme Checkpoint sont impactés en configuration par défaut

Phase 6 (aout 2011) : définition d'un plan d'action

Présentation de l'attaque au Netfilter Workshop

- Présentation de l'étude sur les helpers
- Description de l'attaque
- Discussion ouverte sur les mesures à prendre

Plan d'action

- Rédaction d'un document expliquant les contre-mesures: `https://home.regit.org/netfilter-en/secure-use-of-helpers/`
- Assistance à Florian Westphal pour son implémentation
- Ajout d'une option pour désactiver l'assignation automatique des helpers

Phase 7 : divulgation

CansecWest (mars 2012)

- Description détaillée de l'attaque
- Annonce de la mise à disposition sur demande d'un outil de tests

Présentation au SSTIC (juin 2012)

- Description détaillée de l'attaque
- Mise à disposition publique de opensvp

Questions

Avez-vous des questions ?

Merci à

- Pablo Neira, Patrick McHardy: les développeurs noyaux peuvent être sympas.
- Florian Westphal: pour son implémentation Netfilter de RP filter.

Plus d'information

- Mon blog : `https://home.regit.org`
- Secure use of Iptables and connection tracking helpers: `http://home.regit.org/netfilter-en/secure-use-of-helpers/`

Me joindre

- Courriel: eric@regit.org
- Twitter: @Regiteric