Ossir Bretagne
Conférence Inaugurale
16 octobre 2007

Netfilter : Interaction avec l'espace utilisateur Intérêt et applications

Éric Leblond, INL

Biographie

- Éric Leblond :
  - Développeur principal de NuFW
  - Contributeur Netfilter
  - Co-fondateur d'INL
- INL
  - Société de service en logiciels libres
  - Spécialisé sécurité informatique
  - Éditeur de NuFW/EdenWall

Sommaire

- Rapide état de l'art
- Limitations inhérentes au Noyau
- Émergence d'une nouvelle infrastructure
- Décider en espace utilisateur
- Manipulation du conntrack
- Perspectives d'utilisation
- Conclusion

Netfilter, État de l'art

- Suivi de connexions :
  - ipchains -> Netfilter/Iptables
  - plus sûr, plus rapide
- Modularité et extensions
- Limitations :
  - support d'IPV6
  - performances (NF Hipac)

Suivi de connexions (1/2)

- Gèrer la notion de connexion :
  - Session TCP
  - Aller Retour UDP
- Conntrack : table de suivi des connexions

```
tcp      6 431999 ESTABLISHED src=192.168.1.2 dst=19.2.3.216 sport=22
    dport=59480 packets=14 \\ bytes=1576 src=19.2.3.216 dst=192.168.1.2
    sport=59480 dport=22 packets=25  bytes=1924 \\ [ASSURED] mark=0 use=1

...
```

- Accepter les connexions établies :

```
iptables -I FORWARD -m state –state ESTABLISHED -j ACCEPT
```

Suivi de connexions (2/2)

- Intérêts
  - Sécurisation du filtrage
  - Utilisation pour le NAT
- Gestion des attentes (expectations)
  - Connexions relatives (ftp, ftp-data)
  - Nécessité d'analyse du protocole
  - Maintien d'une table des attentes
  - Filtrage par état basé sur cette notion : RELATED

Interactions nécessaires

- Utilisateur vers Noyau :
  - Configuration :
    - interaction nécessaire
    - point crucial (pb iptables)
- Noyau vers Utilisateur :
  - Journalisation :
    - interaction indispensable
    - limiter l'impact en terme de performances

Aborder la complexité

- Suivi de connexions :
  - Composante indépendante du filtrage :
    - abstraction liée à la nature des protocoles
    - persistant
  - Importance au niveau utilisateur
- Limitation du filtrage niveau 3 :
  - Protocoles non linéaires (FTP, SIP, H323)
  - Filtrage de contenu

Au delà du niveau 3

- “Connection helpers”
  - Support de nouveaux protocoles par modules noyaux
  - Protocoles de + en + complexes (H323)
- Inspection du contenu
  - IDS/IPS ou analyse protocolaire
  - base de “signatures” volumineuse

(R)évolution nécessaire

- Traitement en espace utilisateur :
  - Noyau->Utilisateur
  - Traitement par programme
  - Utilisateur->Noyau
- Gestion de :
  - Décision
  - Modifications du contenu

(R)évolution

- Interrogation du conntrack
  - Lister la table
  - Chercher dans la table
  - Récupérer les évenements
- Modification du conntrack
  - Ajout/Suppression d'entrées
  - Connexions
  - Attentes

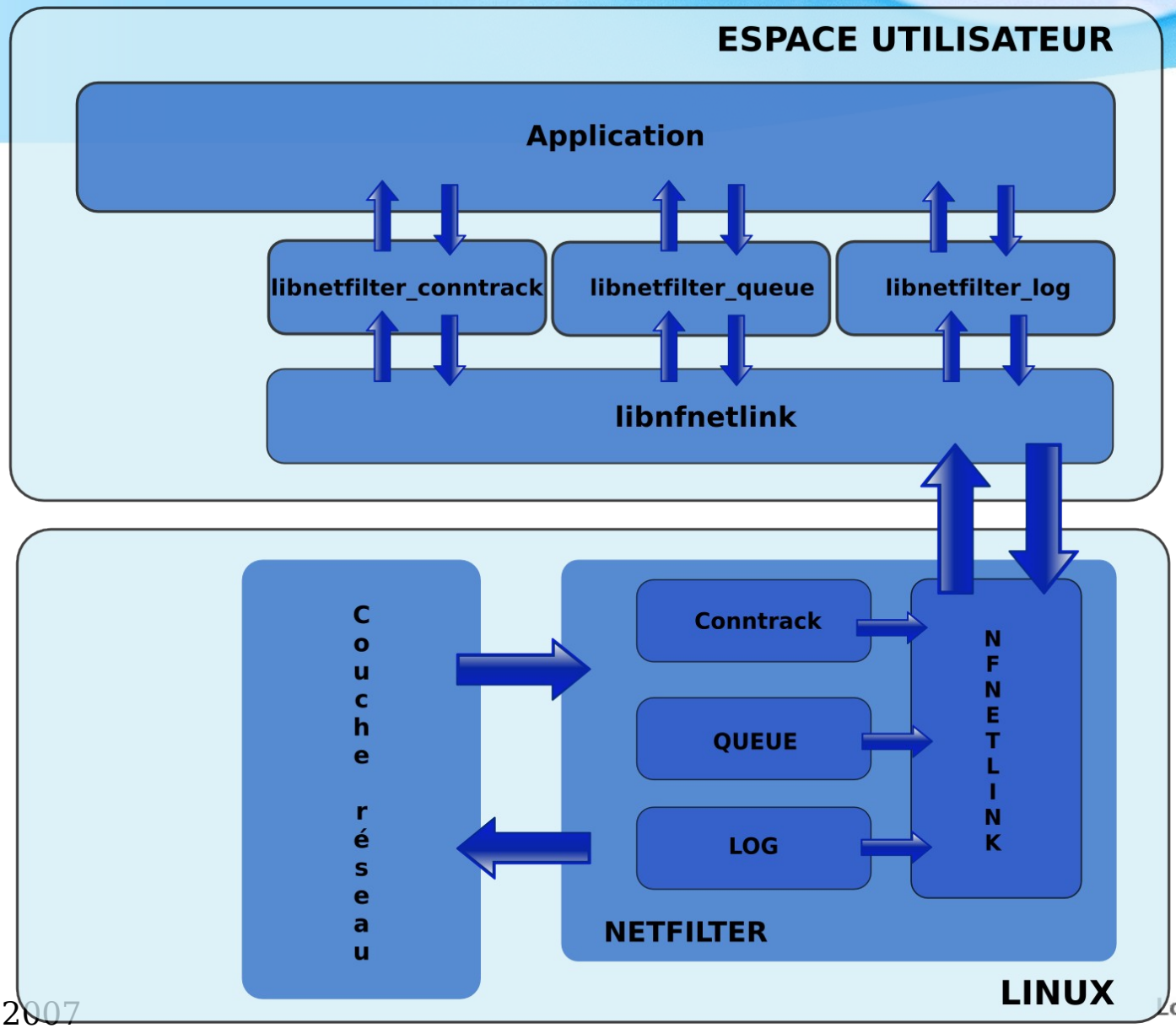
Contraintes et solutions

- Multiplications des canaux de communications Noyau<->Utilisateur.
- Netlink :
  - socket de communication noyau<->utilisateur
  - dédié à ce type de communications
  - peu de canaux, beaucoup déjà occupés
- NfNetlink :
  - “Multiplexage” sur un canal netlink dédié
  - Système de messages génériques

Avancées récentes

- nfnetlink disponible depuis 2.6.14
- Suivi de connexions :
  - Conntrack
  - libnetfilter_conntrack
- Décision en espace utilisateur :
  - libnetfilter_queue (remplace ip_queue)
- Journalisation :
  - libnetfilter_log

2.6.14 : Nouvelle architecture



Principe de NF QUEUE

- Envoi en espace utilisateur :
  - sélection des paquets à intercepter avec iptables :

    iptables -A FORWARD -p tcp –dport 21 -j NFQUEUE
  - espace utilisateur reçoit :
    - données : interfaces, hook
    - payload : entête IP, payload IP
- Traitement et réponse:
  - programme traite le message
  - renvoie : décision, payload, marque

NF QUEUE et ip_queue

- ip_queue :
  - apparu avec Netfilter (2.4)
  - limité
- NF queue :
  - 65535 queues (contre 1 pour ip_queue)
  - Marquage de connexion natif
  - Interaction facile avec la qualité de service et routage

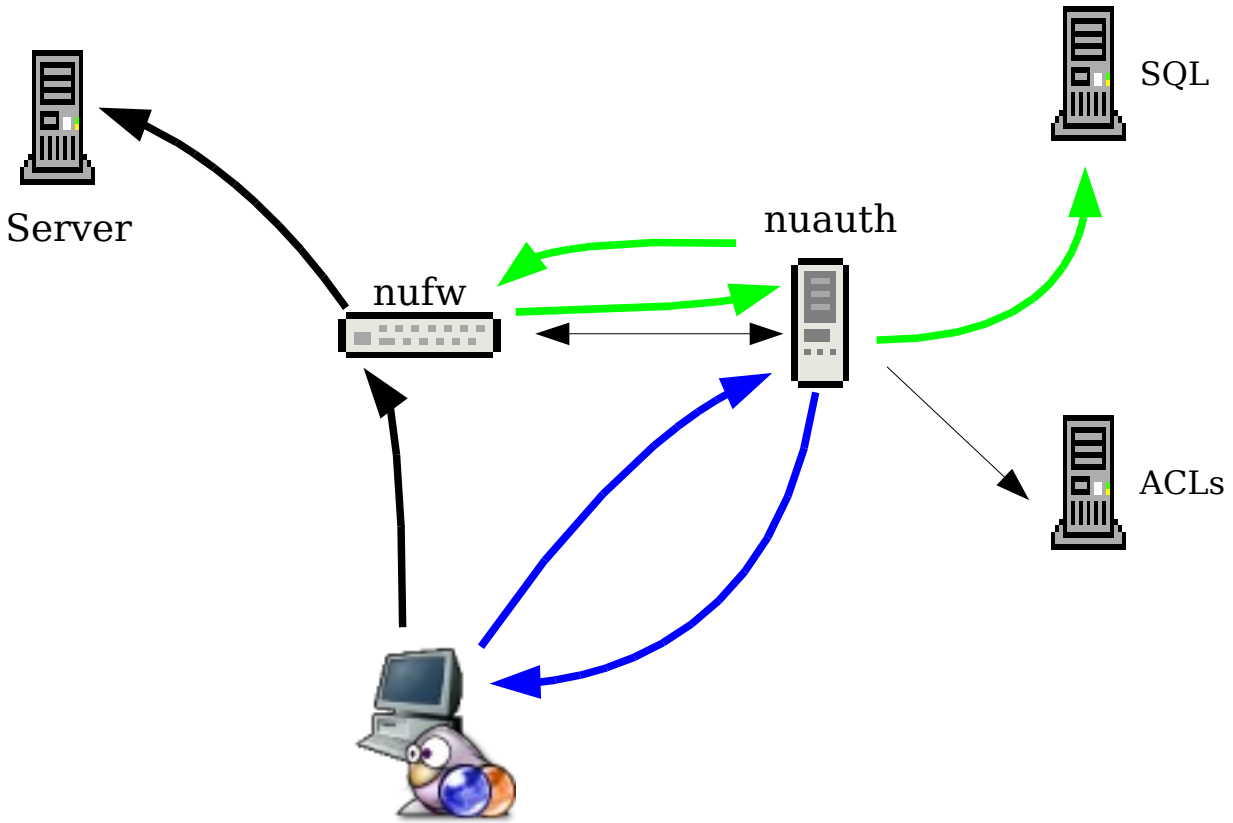
Traitement complexe

- Effectuer des tâches hors de l'espace noyau :
  - trop coûteuses
  - trop complexes
- Exemple de snort-inline (http://snort-inline.sf.net)
  - Patch de snort (IDS)
  - injecte les paquets dans le moteur de snort
  - prend la décision de bloquer si c'est une

Traitement asynchrone

- Temporiser la décision en attendant un traitement
- Exemple de NuFW :
  - Parefeu authentifiant
  - Authentification des connexions a posteriori :
    - interception du paquet d'initialisation de connexion
    - vérification de l'identité
    - prise de décision

NuFW



Fonctionnalités obtenues

- Enrichissement des critères :
  - récupération d'information externe
  - décision prenant en compte les informations systèmes
- Remontée d'informations au noyau :
  - injection d'une marque
  - QoS, routage par utilisateur

Conntrack

- Basé sur libnetfilter_conntrack
- Lister :
  - conntrack -L

```
tcp      6 431999 ESTABLISHED src=192.168.1.2 dst=19.2.3.216 sport=22
    dport=59480 packets=14 \\ bytes=1576 src=19.2.3.216 dst=192.168.1.2
    sport=59480 dport=22 packets=25 bytes=1924  [ASSURED] \\ mark=0 use=1
```

- Chercher :

```
conntrack -G -p tcp -s 192.168.1.2 -d 81.8.111.106
    --orig-port-src 33880 --orig-port-dst 993
```

```
tcp      6 431949 ESTABLISHED src=192.168.1.2 dst=81.8.111.106 sport=33880
    dport=993 packets=94 bytes=9032 src=81.8.111.106 dst=192.168.1.2 sport=993
    dport=33880 packets=80 bytes=13090 [ASSURED] mark=0 use=2
```

Conntrack (évenements)

- Conntrack -E

[NEW] tcp 6 120 SYN_SENT src=19.1.59.116 dst=192.168.1.2 sport=59485 dport=22 packets=1 bytes=60 [UNREPLIED] src=192.168.1.2 dst=19.1.59.116 sport=22 dport=59485 packets=0 bytes=0

[UPDATE] tcp 6 60 SYN_RECV src=19.1.59.116 dst=192.168.1.2 sport=59485 dport=22 packets=1 bytes=60 src=192.168.1.2 dst=19.1.59.116 sport=22 dport=59485 packets=1 bytes=60

[UPDATE] tcp 6 432000 ESTABLISHED src=19.1.59.116 dst=192.168.1.2 sport=59485 dport=22 packets=2 bytes=112 src=192.168.1.2 dst=19.1.59.116 sport=22 dport=59485 packets=1 bytes=60 [ASSURED]

[UPDATE] tcp 6 10 CLOSE src=19.1.59.116 dst=192.168.1.2 sport=59485 dport=22 packets=14 bytes=2209 src=192.168.1.2 dst=19.1.59.116 sport=22 dport=59485 packets=13 bytes=2211

[DESTROY] tcp 6 src=19.1.59.116 dst=192.168.1.2 sport=59485 dport=22 packets=14 bytes=2209 src=192.168.1.2 dst=19.1.59.116 sport=22 dport=59485 packets=13 bytes=2211

Modification du conntrack

- Effacer :

  conntrack -D -p tcp -s 192.168.1.2 -d 81.8.111.106
  --orig-port-src 33880 --orig-port-dst 993

- Créer :
  - Attentes : conntrack -I expect .....
  - Entrées : conntrack -I ....

Nouvelles possibilités

- Utilisation de libnetfilter_conntrack
- Arrêt de flux en cours :
  - indésirable
  - hors délai
- Génération d'attentes :
  - helper pour protocoles complexes
  - port knocking ?

Un manque d'outils

- Problème :
  - Mise à disposition de bibliothèque en C
  - D'un outil “primaire” : conntrack
- Solution :
  - Développement d'une bibliothèque de haut-niveau en python
  - Développement d'une interface web de gestion

pynetfilter_conntrack

- Objectif :
  - fournir une abstraction à la bibliothèque C difficile à prendre en main
  - Permettre à un administrateur d'écrire des scripts de gestion
- Méthode :
  - Utilisation de ctypes pour lier C et python
  - Création d'objets python de haut niveau

Exemple de code

- from pynetfilter_conntrack import NetfilterConntrack, CONNTRACK
- nf = NetfilterConntrack(CONNTRACK)
- table = nf.create_table()
- filter_table = table.filter(6,orig_dst_port=22)
- filter_table.display()
- for entry in filter_table:
- 	nf.delete_conntrack(entry)
-
-
-
-
-
-

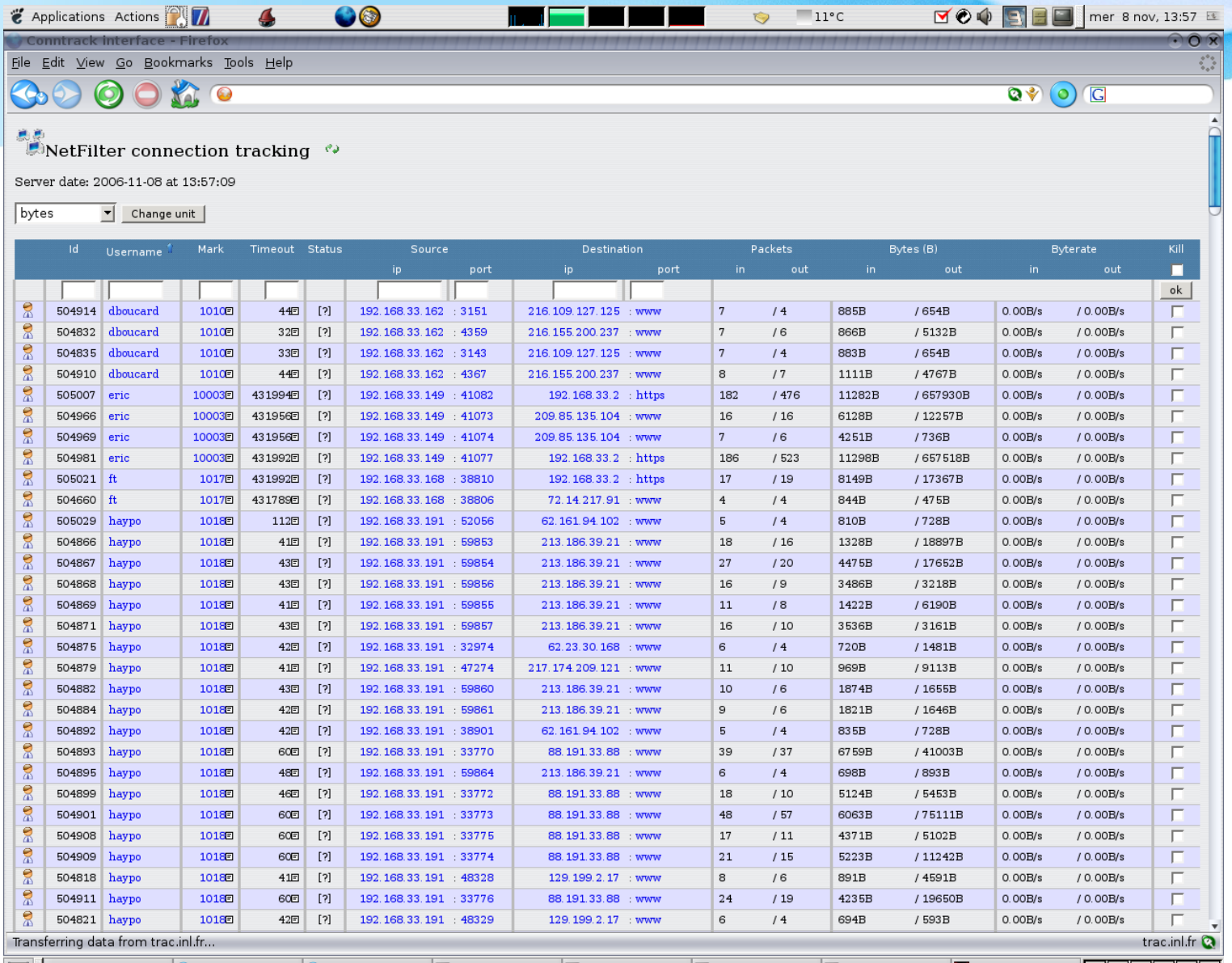
conntrack.py

- Une surcouche à pynetfilter_conntrack :
  - même niveau de fonctionnalités
  - couche d'abstraction supplémentaire (itération)
- Liste :
  - conntrack.py list
- Suppression :
  - conntrack.py delete -p tcp –orig-src-port 22

pyctd

- Serveur XML-RPC
- Mise à disposition des commandes :
  - Listing
  - Modification
  - Suppression

frontend PHP



Id	Username	Mark	Timeout	Status	Source ip	Source port	Destination ip	Destination port	Packets in	Packets out	Bytes (B) in	Bytes (B) out	Byterate in	Byterate out	Kill
															ok
504914	dboucard	1010	44	[?]	192.168.33.162	3151	216.109.127.125	www	7	/ 4	885B	/ 654B	0.00B/s	/ 0.00B/s	
504832	dboucard	1010	32	[?]	192.168.33.162	4359	216.155.200.237	www	7	/ 6	866B	/ 5132B	0.00B/s	/ 0.00B/s	
504835	dboucard	1010	33	[?]	192.168.33.162	3143	216.109.127.125	www	7	/ 4	883B	/ 654B	0.00B/s	/ 0.00B/s	
504910	dboucard	1010	44	[?]	192.168.33.162	4367	216.155.200.237	www	8	/ 7	1111B	/ 4767B	0.00B/s	/ 0.00B/s	
505007	eric	10003	431994	[?]	192.168.33.149	41082	192.168.33.2	https	182	/ 476	11282B	/ 657930B	0.00B/s	/ 0.00B/s	
504966	eric	10003	431956	[?]	192.168.33.149	41073	209.85.135.104	www	16	/ 16	6128B	/ 12257B	0.00B/s	/ 0.00B/s	
504969	eric	10003	431956	[?]	192.168.33.149	41074	209.85.135.104	www	7	/ 6	4251B	/ 736B	0.00B/s	/ 0.00B/s	
504981	eric	10003	431992	[?]	192.168.33.149	41077	192.168.33.2	https	186	/ 523	11298B	/ 657518B	0.00B/s	/ 0.00B/s	
505021	ft	1017	431992	[?]	192.168.33.168	38810	192.168.33.2	https	17	/ 19	8149B	/ 17367B	0.00B/s	/ 0.00B/s	
504660	ft	1017	431789	[?]	192.168.33.168	38806	72.14.217.91	www	4	/ 4	844B	/ 475B	0.00B/s	/ 0.00B/s	
505029	haypo	1018	112	[?]	192.168.33.191	52056	62.161.94.102	www	5	/ 4	810B	/ 728B	0.00B/s	/ 0.00B/s	
504866	haypo	1018	41	[?]	192.168.33.191	59853	213.186.39.21	www	18	/ 16	1328B	/ 18897B	0.00B/s	/ 0.00B/s	
504867	haypo	1018	43	[?]	192.168.33.191	59854	213.186.39.21	www	27	/ 20	4475B	/ 17652B	0.00B/s	/ 0.00B/s	
504868	haypo	1018	43	[?]	192.168.33.191	59856	213.186.39.21	www	16	/ 9	3486B	/ 3218B	0.00B/s	/ 0.00B/s	
504869	haypo	1018	41	[?]	192.168.33.191	59855	213.186.39.21	www	11	/ 8	1422B	/ 6190B	0.00B/s	/ 0.00B/s	
504871	haypo	1018	43	[?]	192.168.33.191	59857	213.186.39.21	www	16	/ 10	3536B	/ 3161B	0.00B/s	/ 0.00B/s	
504875	haypo	1018	42	[?]	192.168.33.191	32974	62.23.30.168	www	6	/ 4	720B	/ 1481B	0.00B/s	/ 0.00B/s	
504879	haypo	1018	41	[?]	192.168.33.191	47274	217.174.209.121	www	11	/ 10	969B	/ 9113B	0.00B/s	/ 0.00B/s	
504882	haypo	1018	43	[?]	192.168.33.191	59860	213.186.39.21	www	10	/ 6	1874B	/ 1655B	0.00B/s	/ 0.00B/s	
504884	haypo	1018	42	[?]	192.168.33.191	59861	213.186.39.21	www	9	/ 6	1821B	/ 1646B	0.00B/s	/ 0.00B/s	
504892	haypo	1018	42	[?]	192.168.33.191	38901	62.161.94.102	www	5	/ 4	835B	/ 728B	0.00B/s	/ 0.00B/s	
504893	haypo	1018	60	[?]	192.168.33.191	33770	88.191.33.88	www	39	/ 37	6759B	/ 41003B	0.00B/s	/ 0.00B/s	
504895	haypo	1018	48	[?]	192.168.33.191	59864	213.186.39.21	www	6	/ 4	698B	/ 893B	0.00B/s	/ 0.00B/s	
504899	haypo	1018	46	[?]	192.168.33.191	33772	88.191.33.88	www	18	/ 10	5124B	/ 5453B	0.00B/s	/ 0.00B/s	
504901	haypo	1018	60	[?]	192.168.33.191	33773	88.191.33.88	www	48	/ 57	6063B	/ 75111B	0.00B/s	/ 0.00B/s	
504908	haypo	1018	60	[?]	192.168.33.191	33775	88.191.33.88	www	17	/ 11	4371B	/ 5102B	0.00B/s	/ 0.00B/s	
504909	haypo	1018	60	[?]	192.168.33.191	33774	88.191.33.88	www	21	/ 15	5223B	/ 11242B	0.00B/s	/ 0.00B/s	
504818	haypo	1018	41	[?]	192.168.33.191	48328	129.199.2.17	www	8	/ 6	891B	/ 4591B	0.00B/s	/ 0.00B/s	
504911	haypo	1018	60	[?]	192.168.33.191	33776	88.191.33.88	www	24	/ 19	4235B	/ 19650B	0.00B/s	/ 0.00B/s	
504821	haypo	1018	42	[?]	192.168.33.191	48329	129.199.2.17	www	6	/ 4	694B	/ 593B	0.00B/s	/ 0.00B/s	

Réplication du conntrack

- ct-sync :
  - Réplication au niveau noyau
  - Développement plus ou moins arrété
- Conntrackd :
  - En mode utilisateur
  - Basé sur libnetfilter_conntrack
  - Actif

Conclusion

- Nouvelle infrastructure très puissante
- Voie vers des fonctionnalités avancées
  - Dynamisme
  - Modifications à la volée
- Formidable boîte à outils

Questions ?

Références :

- Netfilter : http://www.netfilter.org
- Conntrack : http://workshop.netfilter.org/2005/presentations/pab
- Snort-inline : http://snort-inline.sf.net
- NuFW : http://www.nufw.org
- INL : http://www.inl.fr