

# Visualisation de traces réseaux

**Thibault LENGAGNE et Nicolas NGÔ-MAÏ**

Centrale Supélec - Campus de Rennes

5 février 2016

- 1 Retour sur les choix technologiques
- 2 Détails techniques
- 3 Démonstration
- 4 Travail à venir
- 5 Conclusion

- Le but est de créer un outil destiné aux pentesters, permettant d'analyser efficacement une trace réseau.
- L'attaquant dispose d'une trace réseau mais n'a aucune connaissance de ce réseau.
- L'analyse lui permet d'obtenir des mots de passe, la connaissance machines non sécurisées, du contenu non chiffré

L'attaquant espère soutirer des informations sur le réseau :

- Les adresse IP identifiées ( + géolocalisation, résolution DNS)
- Les protocoles utilisés ( en particulier non chiffrés ou mal configurés)
- Les noeuds importants du réseau (serveur de fichier, DNS, LDAP...)
- Extraire des traces réseaux filtrées, extraire les données sensibles

Cela lui permettra d'attaquer plus facilement le réseau

## Scénario d'utilisation

- Upload du fichier Pcap, affichage en continu des données analysées
- Liste des protocoles / hosts filtrables dans une sidebar.
- Plusieurs vue possibles qui permettent de caractériser les hosts
- Filtrage des hosts ou protocoles qui permet de valider les hypothèses
- L'outil permet donc de trouver des machines mal configurées et d'extraire des informations.

- Nous voulions à l'origine interfacier plusieurs outils (Ettercap, ChaosReader, tcptrace...)
- Finalement, Scapy nous permet de manipuler la trace réseau de façon satisfaisante.
- Enfin, le choix initial de PyQt a été remplacé par une interface web.

Nous avons d'ores et déjà remplis les objectifs suivants :

- Extraction de sessions, des utilisateurs, et des protocoles
- Création de statistiques, extraction des données en clair
- Visualisation en coordonnées parallèles

- 1 Retour sur les choix technologiques
- 2 Détails techniques**
- 3 Démonstration
- 4 Travail à venir
- 5 Conclusion



- Le travail s'articule principalement autour de deux technologies : Scapy et D3.js
- Scapy est une librairie Python pour faire de la manipulation de trames réseaux
- D3.js est une librairie Javascript qui permet de faire de la manipulation de SVG lié à des données.

# Nos choix de technologies

- Serveur web Flask (Python) associé à une base de données Postgres et un ORM python, SQLAlchemy
- Pour réaliser un update continu en fonction des données analysées par le serveur, nous avons installé SocketIO

- Framework JS pour faire de la manipulation d'objets du DOM en les liant à des données
- On peut ainsi faire apparaître des balises html ou SVG en fonction d'une donnée arbitraire
- Du coup, les graphes s'adaptent aux données de façon automatique  
-> La structure des données est ce qui détermine le graphe



# Démonstration D3.js

Quatres tables dans la base de données :

Packet
<input type="checkbox"/> id
<input type="checkbox"/> hostSrc
<input type="checkbox"/> hostDest
<input type="checkbox"/> portSrc
<input type="checkbox"/> portDest
<input type="checkbox"/> protocol
<input type="checkbox"/> data
<input type="checkbox"/> timestamp
<input type="checkbox"/> sessionId

Session
<input type="checkbox"/> id
<input type="checkbox"/> hostSrc
<input type="checkbox"/> hostDest
<input type="checkbox"/> portSrc
<input type="checkbox"/> portDest
<input type="checkbox"/> protocol

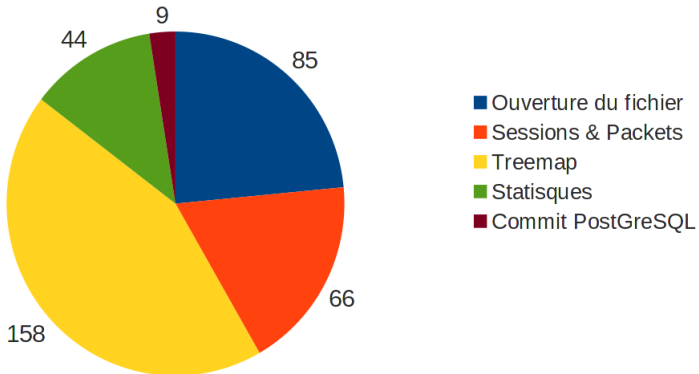
Stat
<input type="checkbox"/> id
<input type="checkbox"/> name
<input type="checkbox"/> value
<input type="checkbox"/> comment

User
<input type="checkbox"/> id
<input type="checkbox"/> address
<input type="checkbox"/> exchanged

- La librairie Scapy permet de lire nos fichiers, puis de manipuler les packets.
- Une fonction parcourt tous les paquets. Les tables User et Stats sont alimentées
- Une fonction parcourt tous les sessions. Les tables Session et Trames sont alimentées

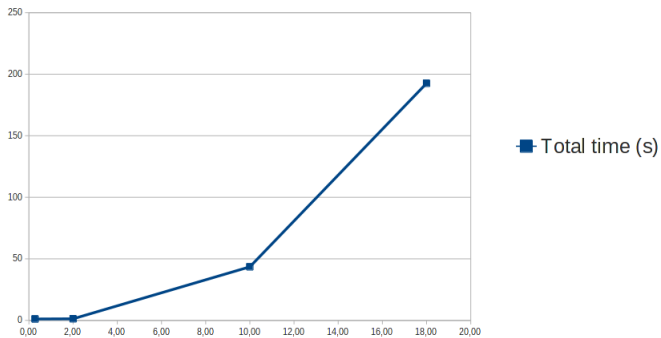
- Les fichiers pcap peuvent atteindre plusieurs Giga assez rapidement.
- L'écriture dans la base de donnée étant coûteuse, le programme écrit dans la base de donnée une seule fois, à la fin de la collecte des données
- Nous réalisons nos tests sur des traces d'entreprise anonymisées ([ftp ://ftp.bro-ids.org/enterprise-traces/hdr-traces05/](ftp://ftp.bro-ids.org/enterprise-traces/hdr-traces05/))

## Analyse d'un fichier de 18 MO





Temps de Parsing =  $f(\text{Taille})$



- 1 Retour sur les choix technologiques
- 2 Détails techniques
- 3 Démonstration**
- 4 Travail à venir
- 5 Conclusion

- 1 Retour sur les choix technologiques
- 2 Détails techniques
- 3 Démonstration
- 4 Travail à venir**
- 5 Conclusion

## Objectifs à remplir

- Ajouter les différents filtres possibles
- Extraire un .pcap a partir de trames filtrées
- Extraire les données non chiffrées des protocoles SMTP,IMAP,POP,LDAP
- Ajouter la résolution DNS
- Ajouter d'autres mode de visualisation (carte des IP,..)

## Problématiques à venir

- Technos suffisantes ? Tests de performance à venir
- Morcellement de l'analyse du Pcap
- Création de vues interactives, et donc d'une belle architecture de données

- 1 Retour sur les choix technologiques
- 2 Détails techniques
- 3 Démonstration
- 4 Travail à venir
- 5 Conclusion**

Merci de votre attention !

[https ://github.com/lechinois/Pcap-visualization-project](https://github.com/lechinois/Pcap-visualization-project)

# Détail du champs exchanged de Users

```
{
  "Volume": 28284,
  "Protocole": {
    "AUTRE": {
      "Nombrein": 291,
      "Volumeout": 12732,
      "Nombreout": 284,
      "Volumein": 13494
    },
    "POP3S": {
      "Nombrein": 9,
      "Volumeout": 386,
      "Nombreout": 7,
      "Volumein": 510
    },
    "ICMP": {
      "Nombrein": 14,
      "Volumeout": 546,
      "Nombreout": 13,
      "Volumein": 616
    }
  }
}
```