

# Supervision et audit de la sécurité système dans un réseau

Projet 2A STI

Jérémy Briffaut, Martial Szpieg

10 septembre 2014

Ce projet couvrira les modules :

- administration système
- sécurité système
- réseau
- programmation C / SHELL
- commandes système
- cryptographie
- Base de Donnée
- XML
- ...

L'évaluation de fin de projet sera basée sur :

- une soutenance de 20 minutes
- un rapport détaillant ce que vous avez fait dans ce projet
- une fiche que vous devrez remplir permettant d'évaluer les composants que vous avez abordés/implantés
- L'application que vous rendrez sous forme d'archive

- Projet ambitieux
  - Vous n'aurez pas assez des séances de projet pour finir le projet
  - Nous sommes là pour vous assister/aider, pas pour faire le projet à votre place
  - N'oubliez pas de documenter votre travail au fur et à mesure => **rapport à rendre**
  - N'oubliez pas de sauvegarder votre travail aussi ...

- Projet en **binôme** (mais vous pouvez vous entraider)
- Séance en salle SA.20X
- 2 salles réservées toutes les semaines (quand c'est possible)
- X séances de 4h encadrés en effectif complet (groupe TD)
- Y séances de 4h encadrés en demi-effectif (groupe TP)
- le reste en autonomie
- Stocker vos images dans `/usr?/NOM/projet`

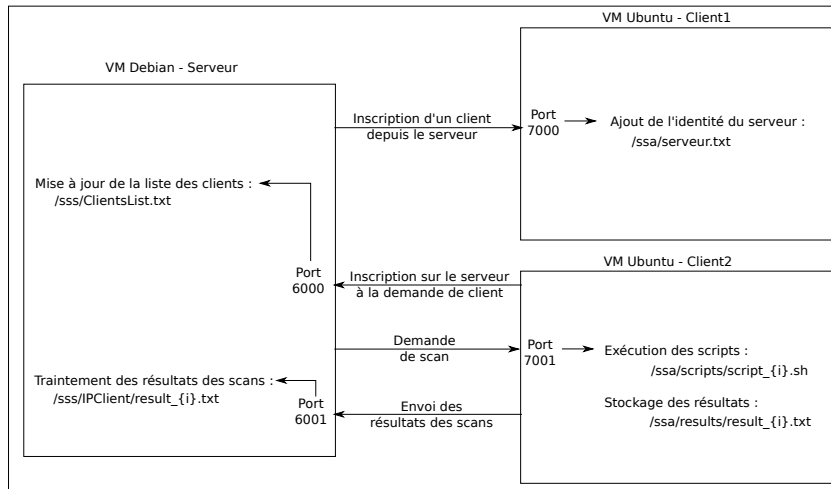
# Présentation

Créer une application client/serveur permettant de superviser/auditer la sécurité système dans un réseau. Le client, installé sur chaque hôte à superviser, pourra exécuter des scripts/scans de vérification (paquets à jour, droits sur les fichiers/répertoires, intégrité du système, informations sur l'hôte, ..). Un serveur permettra de lancer ces scans et de superviser un ensemble de clients. Le serveur aura ainsi une vision globale de l'infrastructure et de sa sécurité. Une interface, présente sur le serveur, permettra d'exécuter et d'afficher le résultat des scans.

## Objectifs

- Créer une application client/serveur en C
- Superviser/Auditer la sécurité de systèmes Linux dans un réseau

# Exemple simplifié d'architecture



# Les étapes

- ➊ Installation de machines virtuelles
- ➋ Installer un environnement de développement (GIT, IDE, ...)
- ➌ Client/serveur en C
- ➍ Inscription via le serveur d'un client
- ➎ Inscription d'un client au serveur
- ➏ Lancement d'un scan depuis le serveur
- ➐ Réception et traitement des résultats d'un scan
- ➑ Affichage des résultats d'un scan
- ➒ (++) autres surprises ensuite)



- Bon courage
- Questions ???