

# Qu'est ce que Nessus ?

- \* Logiciel d'audit de sécurité pour systèmes libres (Linux, FreeBSD, etc...)
- \* Projet qui a publiquement débuté en Avril 1998
- \* Logiciel libre

# Pourquoi un autre scanner ?

- \* Les scanners « libres » sont dépassés ou à tendance commerciale
- \* Les scanners commerciaux sont trop chers
- \* Les autres scanners n'utilisent pas les meilleures méthodes
- \* Les autres scanners sont longs à se mettre à jour

# Architecture du produit

- \* Modèle client - serveur : le serveur est en charge des attaques, alors que le client ne sert que de simple frontend
- \* Les attaques sont regroupées sous forme de plugins
- \* Cette architecture apporte modularité, stabilité et flexibilité
- \* Il existe plusieurs clients : `nessus`, `NessusJ` (version Java), `NessusW` (version Win32)
- \* Mais qu'un seul serveur à l'heure actuelle

# Le serveur

- \* Officiellement supporté sous Linux et tous les systèmes de la famille BSD (OpenBSD, FreeBSD, NetBSD)
- \* Fonctionne sur certains Solaris et SCO avec quelques modifications
- \* Systèmes de « threads » via `fork` ou par les threads posix
- \* Devrait pouvoir se compiler sous WindowsNT
- \* Communication cryptée entre le client Unix et le serveur, et authentification de l'utilisateur par clé publique
- \* Gère plusieurs utilisateurs, avec des restrictions différentes pour chacun
- \* N'utilise pas la liste des utilisateurs Unix

## Le client Unix (nessus)

- \* Utilise the Gimp ToolKit (gtk)
- \* Supporte les toutes dernières nouveautés développées
- \* Supporte la communication sécurisée avec le serveur
- \* Supporte les options en ligne de commande, permettant de l'utiliser avec cron

## Les autres clients

- \* NessusJ : version Java, utilisant swing dans sa dernière version. Totallement identique à la version unix, et supporte la plupart des nouveautés, telles que les préférences au niveau des plugins
- \* NessusW : version Win32
- \* Mais aucun de ces deux clients ne supporte de communication chiffrée avec le serveur

# Les plugins

- \* au nombre de 208 aujourd'hui
- \* sont actuellement des librairies partagées écrite en C
- \* lancées de manière à ne pas pouvoir faire planter le serveur
- \* support d'une « base de connaissance » commune

# La conception d'un plugin

- \* La libnessus offre aux plugins des fonctions évoluées, permettant d'écrire des tests très rapidement.

- \* Exemple: la fonction

```
int ftp_log_in(int soc,  
               char * user,  
               char * pass);
```

- \* A l'avenir, un langage de script permettra un développement encore plus rapide des plugins, et moins contraignant

# Les options d'attaques

\* *Au niveau des cibles :*

- Une série de « cibles primaires » est entrée : série de machines (prof.fr.nessus.org, dorm) ou adresses IP (192.168.1.1,192.168.1.2) ou alors nom de machines plus net-mask (192.168.1.1/27,prof.fr.nessus.org/27) ou même une combinaison des deux
- Une option permet de faire des transferts de zones (AXFR) auprès des serveurs de noms, de manière récursive
- Possibilité de restreindre les tests au travers de règles, afin d'éviter les dépassements inopportuns

\* *Au niveau des plugins :*

- Les plugins peuvent être désactivées une à une ou par famille
- Les plugins peuvent avoir leurs propres préférences
- Les scanners de ports sont eux-même des plugins et peuvent donc être combinés pour obtenir plus de résultat

\* *Au niveau du serveur :*

- Choix du nombre de threads ou de childs (selon les options de compilation)
- Possibilité de déterminer si les machines sont allumées avant des les tester, via un ping TCP
- Possibilité de déclarer que le test se fait au-delà d'un firewall
- Possibilité de déclarer quel fichier distant est à obtenir (typiquement /etc/passwd)

# Les méthodes d'attaques

- \* Commencent la série des scans de port
- \* Les attaques ne se fient pas au numéro de version
- \* Les attaques ne considèrent pas qu'un service donné est en écoute sur un port donné
- \* Usage de la base de connaissances pour affiner les tests et les accélérer
- \* Effectuées dans l'ordre : information gathering, attack, denial of service

# Les rapports

- \* Sous forme d'arbre ou de fichier HTML
- \* En anglais
- \* Détaillé par machine et par port, avec couleurs pour les vulnérabilités

## Les limites actuelles

- \* Plugins écrites en C, sous formes de librairies partagées
- \* Plugins en anglais
- \* Manque de documentation
- \* Le serveur tourne en tant que root
- \* Rapports incomplets quand aux solutions

## Plus d'informations

- \* Site principal : <http://www.nessus.org>
- \* Mirroir francais : <http://www.fr.nessus.org>
- \* Site de developement : <http://cvs.nessus.org>
- \* Mailing list : <http://list.nessus.org>