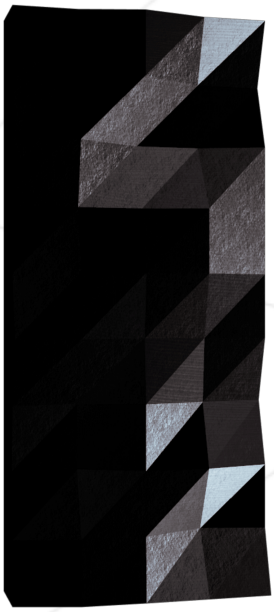

Projet UNIX

Durex

42 Staff pedago@staff.42.fr

Résumé: Ce projet consiste à coder un simple cheval de troie.

Table des matières

Chapitre I

Préambule

Chapitre II

Introduction

Un cheval de Troie (`Trojan Horse` en anglais) est un type de logiciel malveillant, souvent confondu avec les virus ou autres parasites. Le cheval de Troie est un logiciel en apparence légitime, mais qui contient une malveillance. Le rôle du cheval de Troie est de faire entrer ce parasite sur l'ordinateur et de l'installer à l'insu de l'utilisateur.

Le programme contenu est appelé la "`charge utile`". Il peut s'agir de n'importe quel type de parasite : `virus`, `keylogger`, `logiciel espion`... C'est ce parasite, et lui seul, qui va exécuter des actions au sein de l'ordinateur victime. Le cheval de Troie n'est rien d'autre que le véhicule, celui qui fait "entrer le loup dans la bergerie". Il n'est pas nuisible en lui-même car il n'exécute aucune action, si ce n'est celle de permettre l'installation du vrai parasite.

Dans le langage courant, par métonymie on nomme souvent "cheval de Troie" le parasite contenu à l'intérieur. Cette confusion est en partie alimentée par les éditeurs d'antivirus, qui utilisent "trojan" comme nom générique pour désigner différents types de programmes malveillants qui n'ont rien à voir avec des trojans.

Chapitre III

Objectifs

Ce projet est dans la suite de `Matt_daemon` ainsi que `Dr_quine` dont le but sera de créer un simple trojan !

Ce trojan ne sera pas bien compliqué à la base mais avec les bonus et un peu de motivation on va vite se rendre compte des possibilités pour améliorer notre programme.

Chapitre IV

Partie obligatoire

`Durex` est un binaire de votre propre création, qui ne se lancera uniquement qu'avec les droits `root`.

Ce programme sera purement inoffensif au premier abord (héhé...). Au lancement de `Durex`, celui ci n'affichera que votre login sur la sortie standard. **NEANMOINS**, en fond, il aura fait bieeeeen plus de choses :

- `Durex` créera un autre programme qui se nommera aussi `Durex` et qui sera situé dans le répertoire contenant tous les binaires du système d'exploitation cible et s'exécutera.
- Ce programme nouvellement crée devra d'ailleurs être exécuté au démarrage de la machine cible. La méthode de lancement est libre, soyez créatifs.
- Ce même programme sera lancé en tâche de fond à la façon d'un daemon

Bref, un trojan quoi... Mais vous allez me dire, ce trojan il va servir à quoi ? Eh ben, je vais vous le dire :

- Une seule instance du daemon doit pouvoir être lancée
- Le daemon devra écouter le port 4242
- Il doit proposer une connexion à 3 clients en simultané
- Lors de la connexion d'un client sur le daemon, un mot de passe sera demandé. Nous demandons un minimum de sécurité sur le mot de passe (un mot de passe en clair dans le code et c'est le 0 assuré)
- Lorsque la connexion est établie avec un client, le daemon doit proposer de lancer un shell avec les droits root. Ce shell sera accesible avec la commande `shell`

`Attention ! Aucune erreur ne doit être visible lors de la création et/ou l'utilisation du binaire sur la machine cible!!!`

Chapitre V

Exemple d'utilisation

Voici un exemple d'utilisation possible :

```
# ls -al /bin/Durex
ls: cannot access /bin/Durex: No such file or directory
# service --status-all | grep Durex
# ./Durex
wandre
# ls -al /bin/Durex
-rwxr-xr-x 1 root root 12384 Apr 4 14:02 /bin/Durex
# service --status-all | grep Durex
    [ + ]  Durex
# service Durex status
. Durex.service - (null)
    Loaded: loaded (/etc/init.d/Durex)
    Active: active (running) since Mon 2016-04-04 14:08:18 CEST; 1s ago
    Process: 10986 ExecStart=/etc/init.d/Durex start (code=exited, status=0/SUCCESS)
    CGroup: /system.slice/Durex.service
            -> 10988 /bin/Durex
# su wandre
$ nc localhost 4242
Keycode: 42
Keycode: 4242
$>
$> ?
?    show help
shell   Spawn remote shell on 4242
$> shell
Spawning shell on port 4242
$ nc localhost 4242
id
uid=0(root) gid=0(root) groups=0(root)
exit
$ nc localhost 4242
Keycode: ^C
$ su
# netstat -tulnp | grep 4242
tcp       0      0 0.0.0.0:4242          0.0.0.0:*               LISTEN      10988/Durex
#
```

Chapitre VI

Partie bonus

Les bonus ne seront comptabilisés que si votre partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, et qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur aussi vicieuse soit-elle, de mauvaise utilisation, etc ... Concrètement, cela signifie que si votre partie obligatoire n'est pas validée, vos bonus seront intégralement IGNORÉS.

Voici quelques idées de bonus :

- Chiffrer l'envoi et la réception des données (implique un client, logiquement).
- Ajout de fonctions pour notre petit programme (log des actions d'un utilisateur par exemple..).
- Utilisation de méthode ayant pour but de camoufler vraiment notre trojan.
- Utilisation de méthode de type `packing` sur le trojan directement dont le but sera de rendre le binaire le plus léger possible.
- Optimisation dans le but de rendre l'executable compliqué à détecter.

Chapitre VII

Rendu et peer-évaluation

- Rendez-votre travail sur votre dépôt `GiT` comme d'habitude. Seul le travail présent sur votre dépot sera évalué.

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ci-dessous.

- Vous devez coder en `C` (toutes versions confondues) ou en `ASM` et rendre un Makefile (respectant les règles habituelles).

- Vous devez gérer les erreurs de façon raisonnée. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, etc).

- Le choix du système d'exploitation est libre.

- Vous devez être sous une `VM` durant votre correction. Pour info, le barême a été fait avec `Debian 7.0 stable`.

- Vous être libre d'utiliser ce dont vous avez besoin, dans la limite des bibliothèques faisant le travail pour vous, ce qui correspondrait à un cas de triche.

- Vous pouvez poser vos questions sur le forum, Slack...