Projet ISE - BadUSB

Partie 1 - Surface d'attaque et fichiers sensibles

Introduction

Ce projet a pour objectif d'explorer les concepts derrière les attaques par "*rubber ducky*" (aussi connues sur le nom d'attaques *BadUSB*) et en apprenant à utiliser un Raspberry Pico 2 pour concevoir une clé USB malveillante. Les *rubber duckies* sont des dispositifs qui émulent un clavier USB pour injecter des commandes rapidement sur une machine cible. Leur popularité provient de leur simplicité et de leur efficacité pour réaliser des attaques telles que l'exfiltration de données, l'installation de logiciels malveillants, ou la création de portes dérobées.

Ici, le projet consiste à créer une rubber ducky qui permet d'exfiltrer les mots de passes stockés dans le navigateur.

Lisez l'intégralité de l'encadré ci-dessous avant de poursivre.

> L'ensemble du projet a été testé sur les PC des salle informatiques de l'université. Il est fortement recommandé que vous fassiez également le projet sur ces ordinateurs pour plusieurs raisons :
>
> - Ça fonctionnera comme prévu.
> - Certaines fonctionnalités ne sont disponibles que sur le réseau de la fac (accès à un serveur convenablement paramétrer pour les besoins du TPs).
> - Le projet consiste à exfiltrer des données du navigateur. Le faire sur les PC de la fac limite le risque que vous envoyiez vos vrais mots de passe par inadvertance!
>
> Ce projet a été rédigé à des fins pédagogiques. Il n'a pas pour vocation à être utilisé "sur le terrain". Pour rappel l'article L.323-1 du code pénal prévoit que "*Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de trois ans d'emprisonnement et de 100 000 € d'amende.*"

Le projet est divisé en 3 parties. Cette partie est indispensable à la mise en place de l'attaque que vous allez monter durant la suite du projet : **l'étude de la surface d'attaque**. En particulier, vous allez identifier des fichiers sensibles susceptibles d'être compromis lors d'une attaque de type BadUSB.

1 Fichiers sensibles

Durant un précédent TP, nous avons vu qu'il est possible de casser des hash de mots de passe du fichier `/etc/shadow` sous certaines conditions.

▶ **Question 1.** Qui peut lire le fichier `/etc/shadow` par défaut ? Justifiez votre réponse.

▶ **Question 2.** Quel mécanisme protège ce fichier contre une lecture non autorisée ?

▶ **Question 3.** Est-ce pertinent de cibler ce fichier dans le cadre d'une attaque BadUSB ? Justifiez.

▶ **Question 4.** Identifiez au moins trois autres fichiers ou éléments sur un système Ubuntu susceptibles de contenir des informations sensibles. Expliquez pourquoi ils sont intéressants pour un attaquant.

2 Mot de passe et navigateurs

L'objectif de cet exercice est d'étudier la façon dont les navigateurs stockent les mots de passe enregistrés.

> **Note:** Ici, l'exemple de Firefox est pris, mais des conclusions similaires s'appliquent aussi à Chrome.

Avant toute chose, effectuez les opérations suivantes :

- Rendez-vous sur la page about:logins et enregistrez au moins 3 (faux) identifiants et mots de passe dans firefox.
- Assurez-vous que l'utilisation de mot de passe maître est désactivé (décochez la case correspondante sur la page about:preferences)
- Rendez-vous dans le dossier `~/.mozilla/firefox-esr/<profile_name>`. Remplacez `<profile_name>` par le nom de profile (par exemple, `46jfpiry.default-esr115`).

▶ **Question 1.** Entrez la commande suivante (qui rend le fichier plus lisible), et consultez le fichier `/tmp/login.jsons` :

```
cat logins.json| python3 -m json.tool > /tmp/login.jsons
```

Que contient ce fichier ?

▶ **Question 2.** Les noms d'utilisateurs et mot de passe sont-ils stockés directement ?

▶ **Question 3.** Toutes les informations nécessaires au déchiffrement des champs `encryptedUsername` et `encryptedPassword` sont contenu dans le fichier `key4.db` du même dossier. Quels sont les droits d'accès à ces fichiers ?

▶ **Question 4.** D'après la question précédente, que pouvez-vous conclure sur la robustesse du stockage des mots de passe du navigateur, si un attaquant à accès à l'ordinateur ?

▶ **Question 5.** Utilisez le notebook Jupyter fourni pour exécuter l'outil **firefox_decrypt** afin de récupérer les mots de passe Firefox enregistrés sur votre navigateur.

▶ **5.a.** Familiarisez-vous avec le contenu du notebook. Lisez le texte et exécutez les différentes cellules. Résumez brièvement la façon dont un attaquant peut accéder aux mots de passe.

▶ **5.b.** La dernière cellule du notebook n'est pas complète. C'est à vous de la compléter, avec les éléments qui vous sont donnés.

▶ **Question 6.** Répétez l'opération après avoir activé le mot de passe maître dans le navigateur. Concluez sur les risques encourus si aucun mot de passe maître n'est défini, et l'utilité de celui-ci.

> Dans cet exercice nous avons étudié la gestion des mots de passe, mais les navigateurs manipulent beaucoup d'autre données sensibles, comme les cookies, ou simplement l'historique de navigation. Ces données ne sont pas protégées par un éventuel mot de passe maître, et il serait parfaitement possible de récupérer avec un simple script...