Rapport de GPE – Partie Numérique

Livraison finale

Keylogger

Comment obtenir les identifiants d'une cible facilement ? (Mais avec pas mal de sang-froid)

Nous allons étudier une technologie encore aujourd'hui utilisée par des hackers et des pentesters afin de récupérer des données confidentielles.

1 Cahier des charges

Un Keylogger est un dispositif de piratage permettant de capturer et de récupérer toutes les actions effectuées sur un clavier.
Il existe plusieurs types de keylogger :

- Les Keyloggers software qui sont des malwares qui s'exécutent en arrière-plan sur le PC de la cible.
- Les Keylogger hardware qui eux sont sous forme de clé USB avec une entrée femelle et une sortie mâle (l'entrée sera reliée au Clavier et la sortie au PC).

Aussi aujourd'hui, la majorité des claviers sont en USB, cependant, il existe encore des claviers en PS/2. Nous allons réaliser notre keylogger de manière à pirater un clavier PS/2, car les contraintes techniques liées à la vitesse de la communication ainsi que le protocole qui est utilisé sont moindres. Si le temps nous le permet, le passage à un keylogger USB pourra être envisagé.
Aussi, l'intérêt d'un keylogger étant de récupérer les données, nous ajouterons un module Bluetooth HC_05 afin d'envoyer les données vers une application Android.

Pour réaliser notre projet, il nous faut donc :

- 2 prises PS/2
- 1 module HC_05
- 1 led + résistance pour debug/état d'utilisation
- Une alimentation 5V de debug (la carte est censée s'alimenter directement avec l'alimentation du câble PS/2 par la suite)

2 Manuel d'utilisation

Ce dispositif permet le piratage d'un clavier de type PS/2, il est indétectable à l'utilisation et ne peut être découvert que s'il n'est pas bien caché. En effet, la récupération des données se faire grâce à une exploitation du matériel et non un logiciel malveillant. Il est donc impossible de détecter sa présence depuis le PC.

Afin de l'utiliser, il vous faudra tout d'abord établir une cible dont vous souhaitez récupérer des données confidentielles (mot de passe, email …).

Dès que votre cible laisse son ordinateur sans surveillance, installer le dispositif à l'arrière de son PC. Pour ce faire, branchez la prise PS/2 du clavier sur l'une des prises PS/2 du dispositif puis branchez un câble PS/2 mâle/mâle entre la deuxième prise PS/2 du dispositif et celle du PC.

Le dispositif va s'alimenter automatiquement grâce à l'alimentation du câble. Et se connecter à votre téléphone via Bluetooth.

Il ne vous reste plus qu'à dissimuler le dispositif puis vous éloigner de l'ordinateur et démarrer l'application Android.

Appairez votre téléphone avec le dispositif via Bluetooth, puis à chaque appui sur une touche, vous recevrez les caractères correspondants.

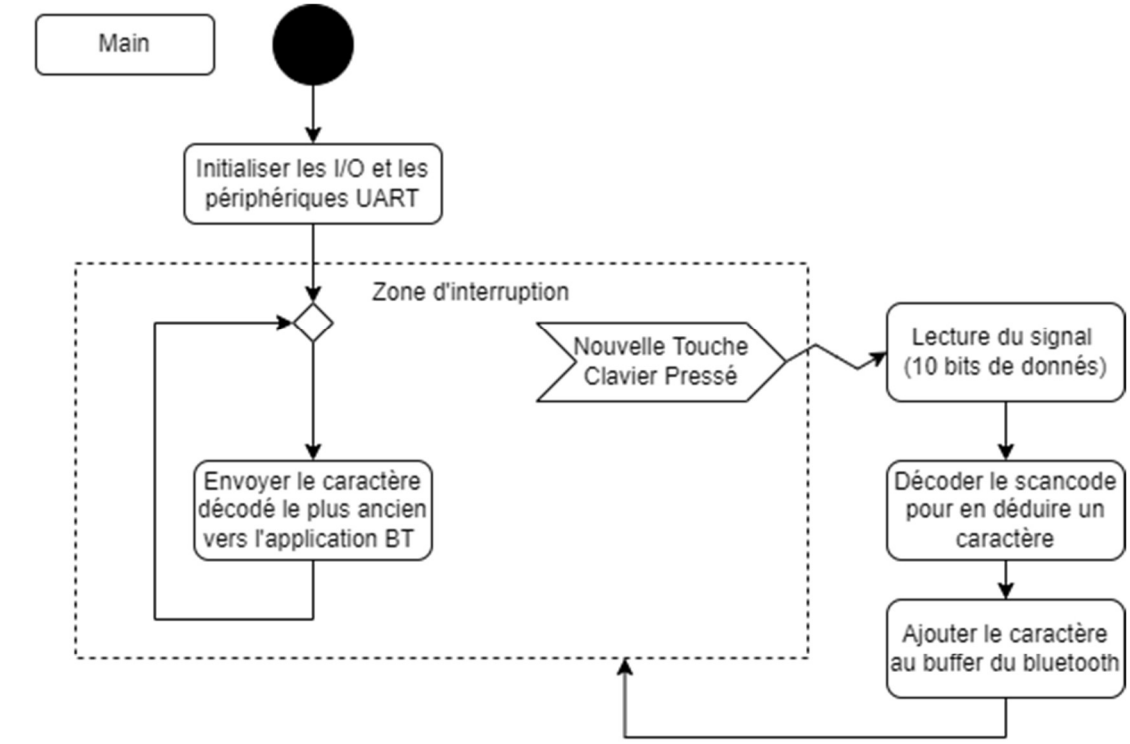
Pour information, les caractères non-imprimables (F1 … F12, la touche Backspace …) sont affichés sous cette forme : « [F1] ».

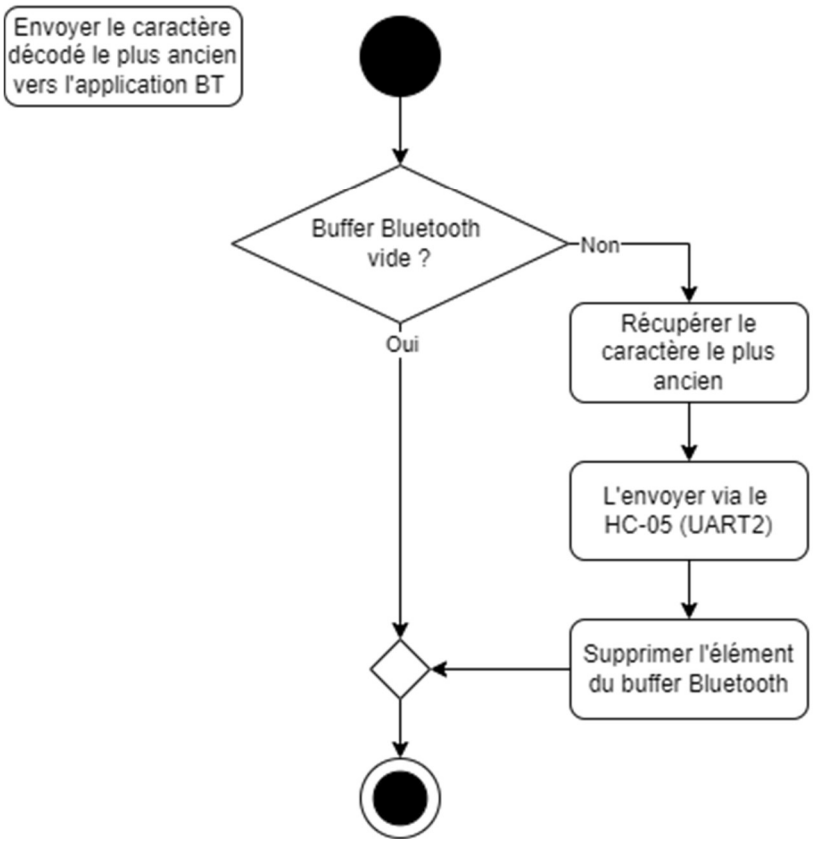
Le keylogger ne sauvegarde aucune donnée en mémoire Flash, une fois l'alimentation coupée, il est réinitialisé. Dès que l'ordinateur redémarre, le keylogger redémarre…

Si votre téléphone n'est pas appairé au Bluetooth, le keylogger a une mémoire vive de 255 caractères avant de saturer. Dès que vous vous connecter, tous les caractères qui auront été enregistrer seront directement envoyé.

3 Description d'un algorithme du programme

Ci-dessous, le diagramme d'activité global du Keylogger :





4 Structure du programme

4.1 Fichier « main.c »

Il s'agit du code principal, on y retrouve l'étape d'initialisation et la boucle principale.

Fonction	Nom du Dev	Description
`process_ms`	Clément CARDOT	Permet de mettre en place un timeout pour la lecture du PS/2
`initBluepill`	Clément CARDOT	Permet d'initialiser la Bluepill avec les bons paramètres
`main`	Clément CARDOT	Méthode principale

4.2 Fichier « Bluetooth.c »

Ce module nous permet de communiquer avec le périphérique HC-05 et donc avec un périphérique Bluetooth (ici notre application Android).

Fonction	Nom du Dev	Description
`isBluetoothPaired`	Clément CARDOT	Permet de savoir si un appareil est appairé au HC-05
`isBluetoothBufferEmpty`	Clément CARDOT	Permet de savoir s'il reste des caractères à envoyer
`getNextCharFromBluetoothBuffer`	Clément CARDOT	Permet d'obtenir le prochain caractère à envoyer
`addToBluetoothBuffer`	Clément CARDOT	Permet d'ajouter un caractère au buffer d'envoi
`sendString`	Clément CARDOT	Permet d'envoyer une chaîne de caractère
`sendChar`	Clément CARDOT	Permet d'envoyer un caractère

4.3 Fichier « PS_2.c »

Ce module permet la lecture des informations transitant entre les deux prises PS/2. Il nous permet aussi ensuite de décoder l'information lue afin d'en sortir un caractère.

Fonction	Nom du Dev	Description
Ps2Interrupt	Clément CARDOT	Permet de décoder le flux de bit passant dans le canal DATA du PS/2
scancodeToChar	Clément CARDOT	Permet de convertir un scancode en caractère imprimable

4.4 Fichier « test.c »

Ce module va nous permettre d'effectuer toute une batterie de test.

Fonction	Nom du Dev	Description
test	Clément CARDOT	Méthode exécutant tous les tests unitaires
toogleBlueLED	Clément CARDOT	Méthode de changement d'état de la Led Bleu (Built-in)
toogleRedLED	Clément CARDOT	Méthode de changement d'état de la Led Rouge (PCB)
testBluetoothSendChar	Clément CARDOT	Méthode d'envoi d'un caractère en Bluetooth
testBluetoothSendString	Clément CARDOT	Méthode d'envoi d'une chaîne de caractères en Bluetooth

5 Tests

Intitulé du test	**Description de ce qu'il faut faire pour jouer le test et de ce qu'on doit observer**	**Observation obtenue et conclusion**
Test d'alimentation du microcontrôleur :	Mesure au voltmètre ; les entrées d'alimentation doivent être à 5V	OK
Test de la LED d'interruption	Lorsqu'une interruption est détectée sur le fil de l'horloge, change l'état de la LED bleu (bluepill)	OK
Test de la LED d'acquittement d'un caractère	Lorsqu'un caractère est lu et enregistré, change l'état de la LED rouge (PCB)	OK
Test de l'envoi d'un caractère en BT	Envoi un caractère en BT via la fonction sendChar	OK
Test de l'envoi d'une chaine de caractères en BT	Envoi une chaine de caractères en BT via la fonction sendString	OK

Etant donné qu'on ne peut pas faire des tests unitaires sur la lecture des données du clavier, je n'ai pas pu développer d'autres tests. Les fonctionnalités non mentionnées ont donc été développées à tâtonnement.

6 Cahier de suivi

Date	Tâches, réalisateurs, difficultés rencontrées.	A faire la prochaine fois
9/11	Réalisation du Schéma Electrique / Routage	Vérifier cahier des charges PCB
16/11	Finalisation de la conception du PCB, Dernières vérifications avant dépôt Dépôt du PCB	Application Andoid
18/11	Création de l'appli Android APP Inventor	Perçage + soudure PCB
25/11	ABSENT	ABSENT
02/12	Réception du PCB, perçage, soudure, vérification de continuité de courant	Préparation Software
10/12	Recherche liée à la solution Software à apporter	Rédaction des diagrammes UML, début de l'écriture du code
14/12	Diagramme d'activité de la boucle principale ainsi que des deux fonctions principales Réalisations des Tests de Bluetooth	Lire le Signal du clavier sur un oscilloscope + Tester la fonction de lecture du signal
16/12	Découverte d'une erreur lors de l'implémentation des connecteurs PS/2 sur Altium (GND et 5V inversés) → Résolution du problème en isolant des pins concernés et en les reconnectant aux bons réseaux Bug sortie PA15 à 2V ➔ non définie en sortie… Première visualisation des signaux clock et Data sur l'oscilloscope !	Résoudre les problèmes découverts
30/12	L'erreur précédente due à la mauvaise documentation des ports PS/2 sur Altium a causé d'autres problèmes de connexion qui ont amené à faire cramer la bluepill.	Refaire le PCB
04/01	Une nouvelle commande de PCB a été réalisé en corrigeant cette fois-ci les erreurs liées aux ports PS/2, Aussi les Ports Clock et Data ont été relié à une barrette femelle afin d'être plus facilement sondable.	Dès réception du PCB : soudure des composants ; En attendant travail sur breadboard
06/01	Nouveau PCB reçu ! Les soudures ont été faites ce matin et les problèmes liés aux interruptions ont été levés, la fonction de test permettant de récupérer un scancode tapé sur le clavier fonctionne !	Il faut maintenant implémenter une fonction transformant le scancode en caractère sur l'application. Puis utilisé les fonctions du module Bluetooth pour les envoyer. Un système de mémoire buffer devra aussi être instauré afin de ne pas

		louper des informations en cas de coupure de la connexion.
08/01	Implémentation d'une fonction de traduction du scancode en caractère. Fonctionne pour toutes les touches « imprimables » + utilisation du shift	Il faudrait implémenter l'utilisation du AltGr et des touches non-imprimables
09/01	Ajout de l'utilisation du AltGr PS : J'ai décidé d'envoyer mes données sur une application Android du Play store plutôt que de faire la mienne car AppInventor est assez limité pour cette application. J'ai l'ambition de créer dans le futur une véritable application via Android Studio.	Gestions des caractères non imprimables en lecture et en écriture. Optimisation du code et mise en place d'un buffer en cas de déconnexion du Bluetooth
11/01	Si les caractères étaient tapés trop vite, certains pouvaient être manqués. Ce n'est plus le cas, le code a été optimisé afin de pouvoir intercepter tous les caractères. Aussi un buffer de 255 caractères a été mis en place afin de sauvegarder les données qui ne peuvent pas être envoyé. Elles sont envoyées dès que le Bluetooth est de nouveau opérationnel.	Prise en charge des caractères non-imprimables
13/01	Prise en charge du Pad Numérique + affichage du backspace Affichage des touches F1 … F12 Prise en charge du left-shift	Projet Terminé Rédaction finale du CR

7 État d'avancement et analyse du projet réalisé

Le Keylogger est terminé ! Le dispositif de piratage est opérationnel et a subi plusieurs tests prouvant son efficacité. Il est capable de décoder et transmettre presque[1] toutes les touches du clavier.

Si je devais refaire ce projet, je prendrais le temps de bien vérifier que les composants Altium sont conforment à ceux composants que j'utilise et j'essayerai de rendre compatible le dispositif avec un clavier USB.

Le PS/2 était aujourd'hui complètement obsolète, ce dispositif est donc lui aussi obsolète. Cependant, en utilisant des adaptateur PS/2 vers USB et USB vers PS/2, il est probable qu'un clavier USB soit piratable avec ce dispositif, mais des tests doivent êtes effectués pour le prouver.

[1] : Les touches non-intéressantes ne sont pas prises en compte, exemple : Echap, Flèches directionnelles, Windows, Inser, Suppr, Fin ….

8 Conclusion

Ce dispositif est certes très peu discret, car tout de même un peu gros, mais il peut être très efficace si l'ordinateur cible est sous un bureau ou peu accessible.

Une fois bien installé, il deviendra très facile de récupérer les mots de passe des utilisateurs qui se connecte au réseau (les premiers caractères tapés après la mise sous tension).

Et cela démontre qu'il est facile de se faire pirater sans s'en rendre compte. Un bureau qui reste ouvert 5min suffit !!!