Déverrouillage d'Android en simulant un clavier/souris

A. Cervoise
antoine.cervoise@gmail.com

2 Juin 2016

Résumé

Qui sommes nous ?

Antoine - @acervoise

- Test d'intrusion @ NTT Com Security FR
- Co-organisateur d'Univershell (Ateliers sécurité sur Paris)

Julien

- Amateur de cigares
- Promoteur du logiciel libre

Android

Verrouillage d'Android

- Code PIN
- Mot de passe
- Schéma a 9 points
- Reconnaissance faciale
- Reconnaissance faciale et vocale
- Empreinte digitale
- Protection spécifique du constructeur (Knock Code de LG)

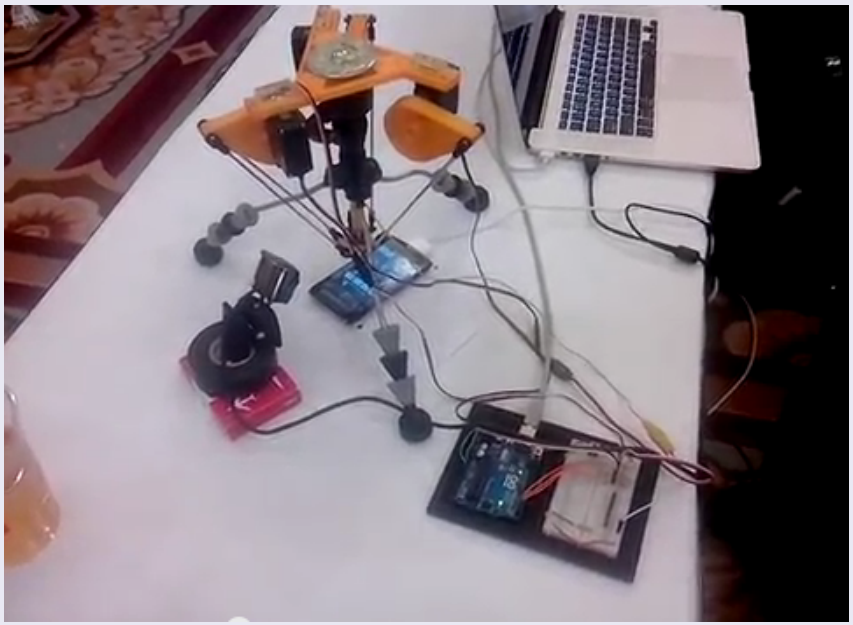
Android

Techniques d'attaque

- Mode de débogage
- ClockworkMod
- Copie de la RAM
- Analyse des traces de doigts sur l'écran
- Utilisation d'une copie (pour les verrouillages "biométriques")
- Utilisation d'un robot

Android

Utilisation d'un robot (2013)



Blackhat Arsenal USA 2013 - R2B2 PIN Cracking Robot

Notre démarche

USB OTG (*On-The-Go*)

- Permet d'utiliser Android comme hôte
- USB OTG n'est pas implémenté sur tout les téléphones

USB *Host Mode*

- Permet le support de clavier/clé USB/etc.
- Introduit dans Android version 3.1 (Honeycomb)

Notre démarche

Démarche d'attaque

- Simuler un clavier/souris avec un Arduino/Teensy
- Attaque Code PIN, mot de passe, schéma de verrouillage

Premier problème

- Maintenir la charge du téléphone

Notre démarche

Notre démarche

Détection de réussite

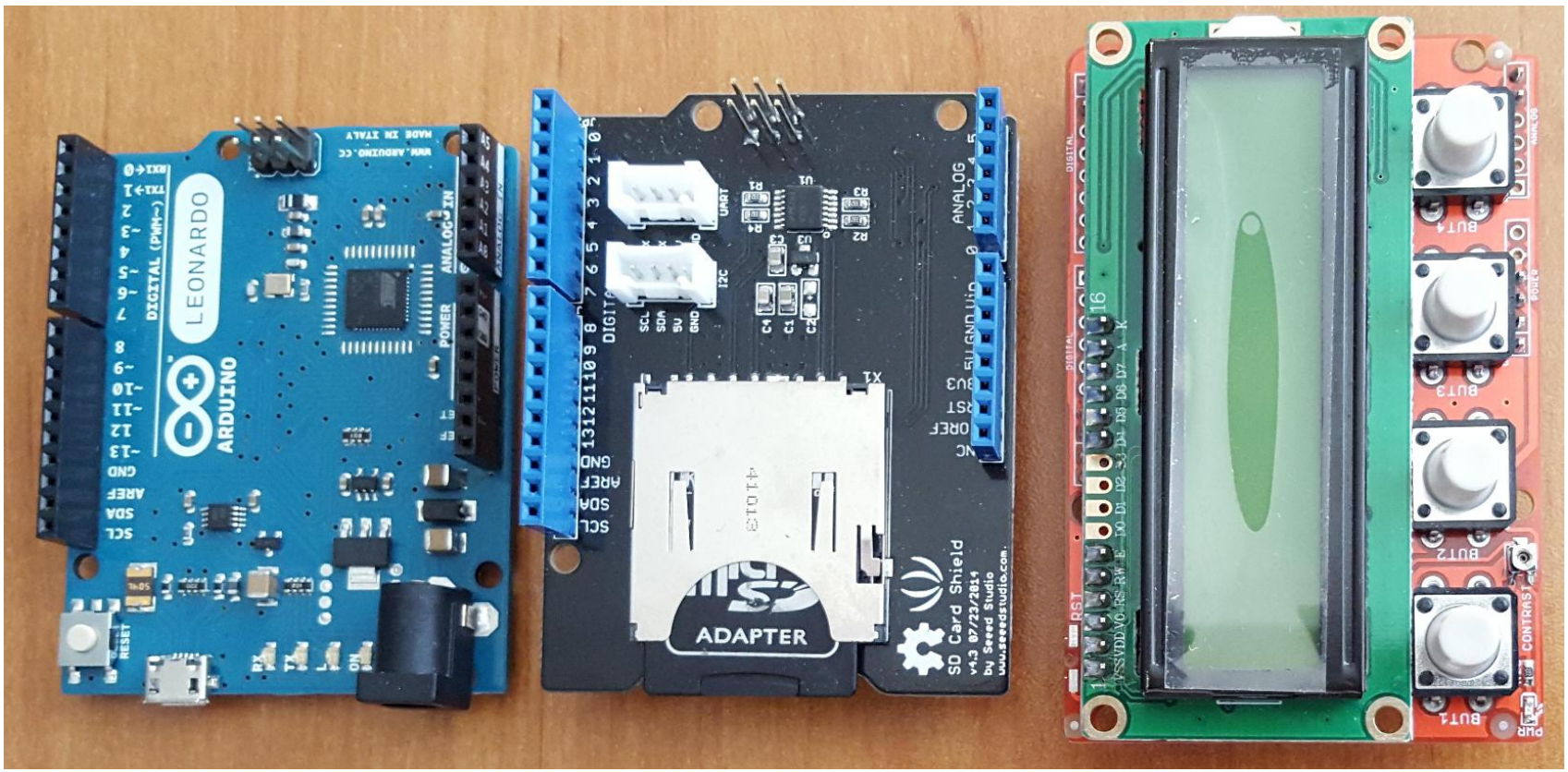
- Regarder l'état à la fin
- Regarder toute l'attaque
- Filmer l'attaque
- Prendre une photo à chaque essai et comparer les résultats
- Déterminer un comportement physique "mesurable" lors du déverrouillage du téléphone

Hardware Bruteforce Framework

Objectifs

- Moins de 100 €
- Le moins de soudures possibles
- Un maximum d'*open hardware*

Hardware Bruteforce Framework V1

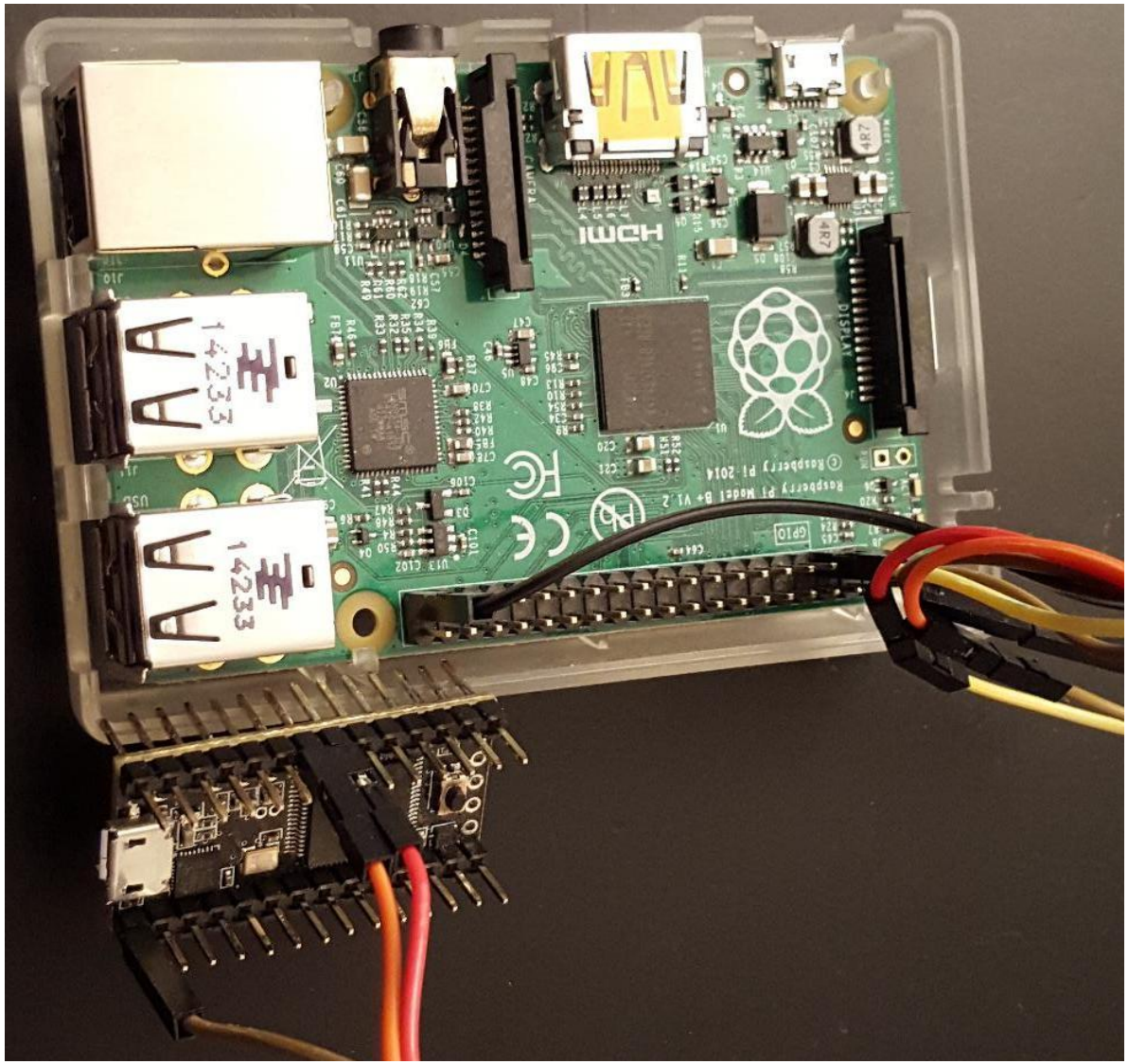


Hardware Bruteforce Framework V1

Problèmes

- Trop de code pour la mémoire d'un Arduino
- Une mise à jour Arduino a entraîné de nombreux bugs
- Pour détecter la réussite de l'attaque, il faut ajouter un Raspberry Pi et une webcam (~40 €)

Hardware Bruteforce Framework V2




04 WINS
99
SUB-ZERO
SMOKE
FIGHT !!

Résumé

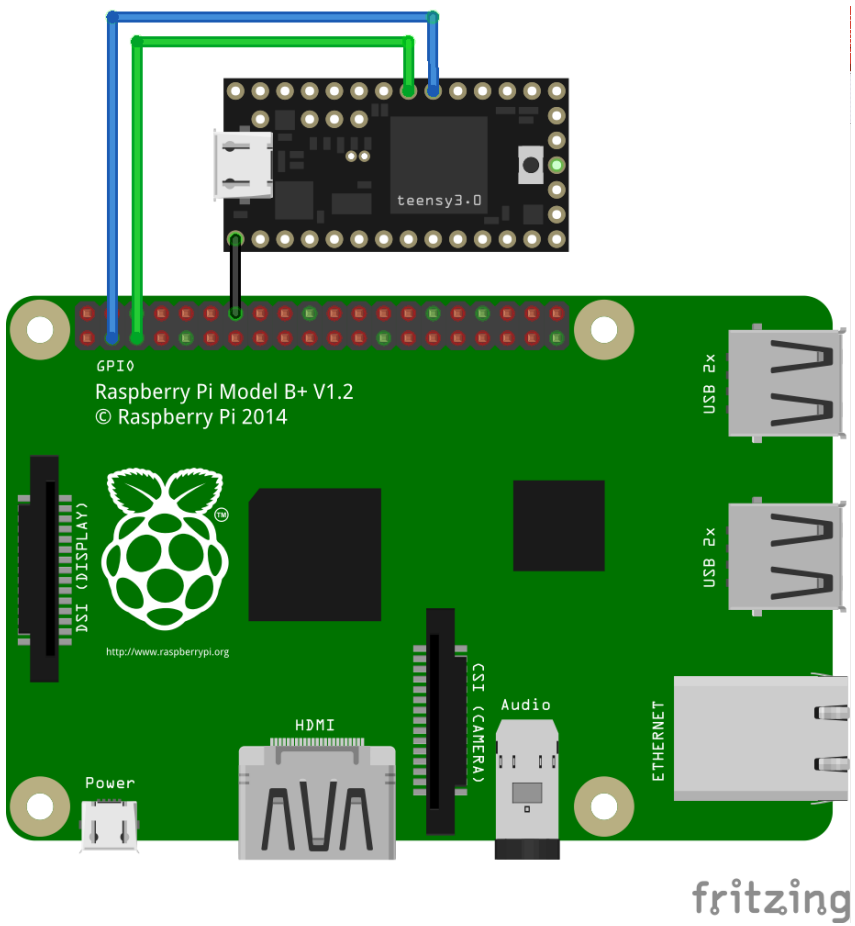
L'attaque

Récupérer le code et la doc sur
https://github.com/cervoise/Hardware-Bruteforce-Framework-2
(GPL v3)

Notre attaque

```
Arduino      Arduino/Teensy Code
Docs         Docs
Raspberry    <fruit>Pi code
```

Notre attaque



Notre attaque

Reconnaissance

- Essayer quelques codes PIN ou mots de passe manuellement (avec un clavier)
- Définir l'algorithme

Hardware Bruteforce Framework V2

```
Entrer un mot de passe
Appuyer sur Entrée
Attendre 100 ms
Réessayer
```

Notre attaque

Définir un fichier modèle

- password
- bruteforce charset size
- delay X (in ms)
- enter/backspace/delete/tabulation/F2
- screenshot
- passwordDelay
- wait a-pattern-file.txt attempt

Notre attaque

`Entrer un mot de passe`	`password`
`Appuyer sur Entrée`	`enter`
`Attendre 100 ms`	`delay 100`
`Réessayer`	

Notre attaque

```
python Raspberry/HBF.py -p password-wordlist.txt
 pattern.txt
```

Notre attaque : mot de passe

```
passwordDelay 300
password
delay 300
enter
delay 3500
screenshot
wait wait-generic.txt 5
```

100 mots de passe : 19/20 minutes

Notre attaque : mot de passe

```
enter
delay 5000
enter
delay 5000
enter
delay 5000
enter
delay 5000
enter
delay 5000
enter
delay 5000
enter
delay 6000
```

Notre attaque : code PIN

```
passwordDelay 300
bruteforce numeric 4-6
delay 300
enter
delay 3500
screenshot
delay 500
wait wait-generic.txt 5
```

Notre attaque : schéma de verrouillage

```
initMouse -240 160
delta 240
pattern
delay 200
screenshot
wait wait-generic.txt 5
```

100 schémas (taille 4) : 20/23 minutes

Notre attaque : schéma de verrouillage

Vidéo

Détection

Webcam VS Raspberry Cam

- La Webcam USB Webcam est plus rapide et moins chère
- La Raspberry Cam n'est pas utilisable sur toutes les "tartes aux fruits"

Détection

Fonctionnement

- Prendre des captures témoins
- Prendre une capture à chaque essai
- Comparer l'ensemble

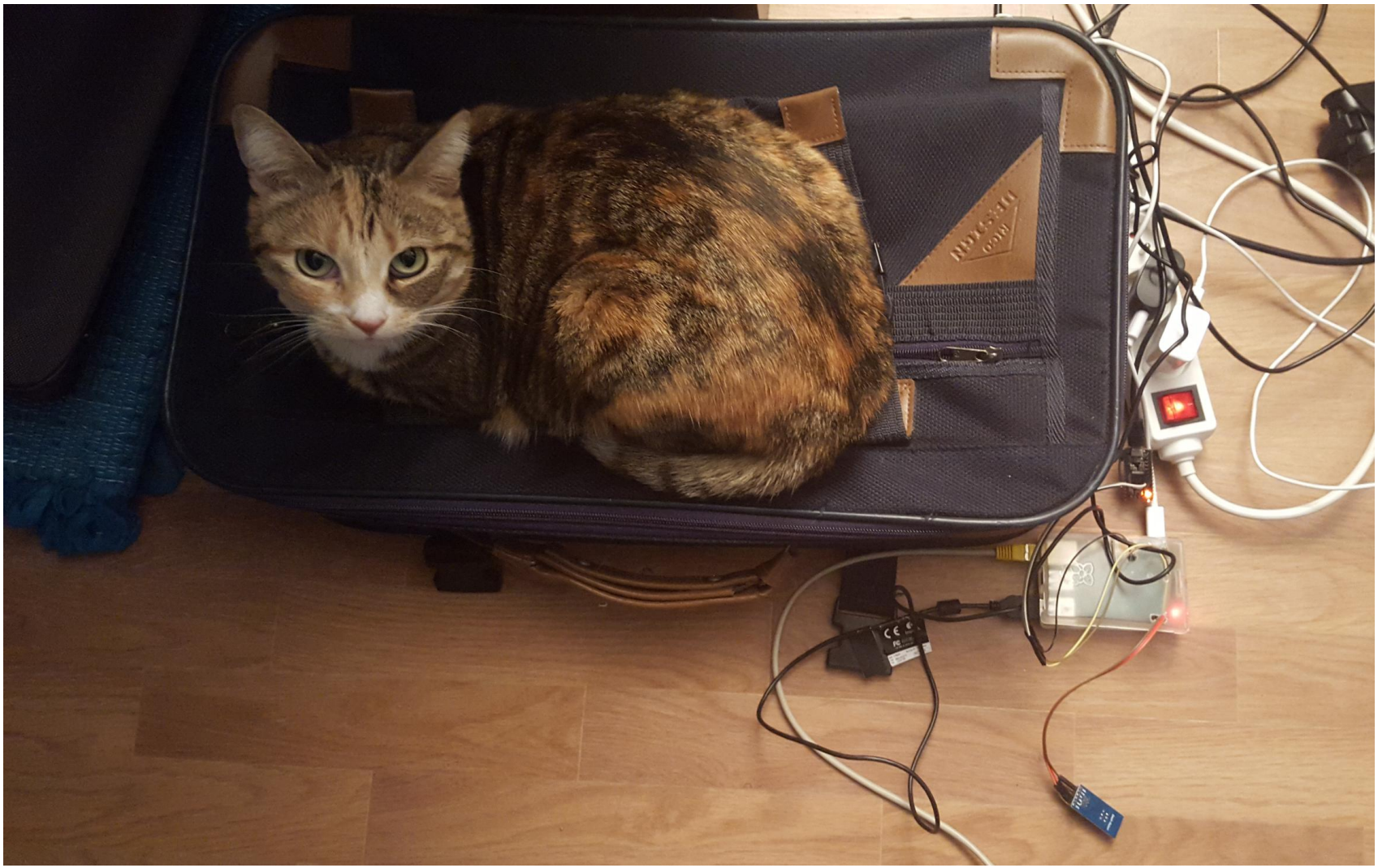
ImageMagick

```
compare -metric RMSE image1 image2 out
```

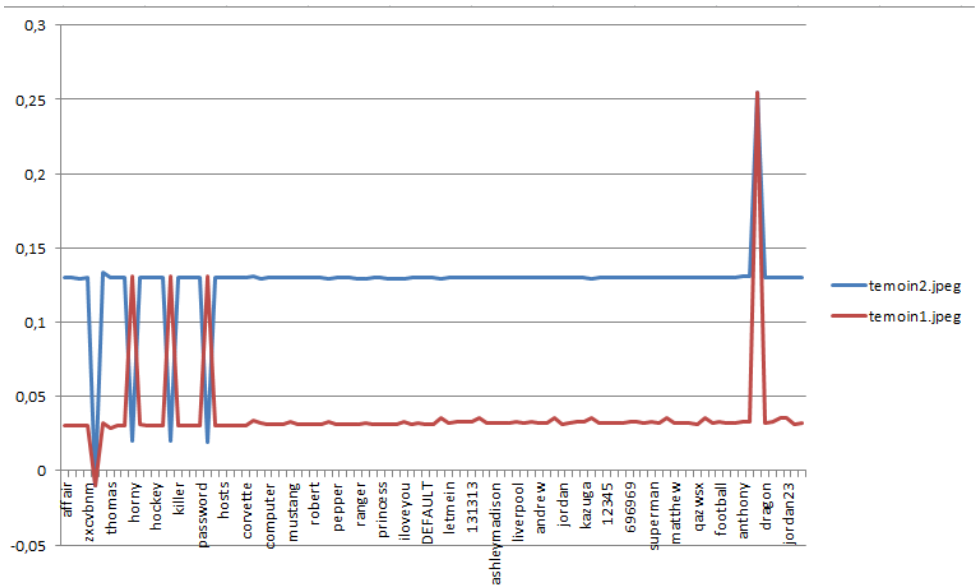
Prendre en compte la luminosité

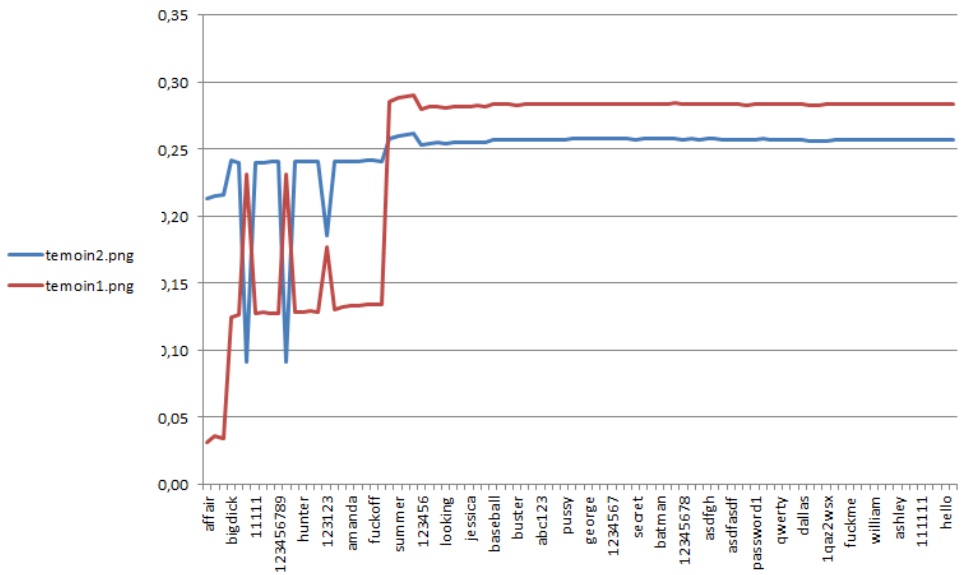
Luminosité

Luminosité



Exemple sur une attaque de mots de passe





Détection

Limites

- Chaque comportement inattendu entraîne un faux positif
- De la latence peut apparaître
- Si la luminosité change, c'est foutu

Résumé

Conclusion

Limites

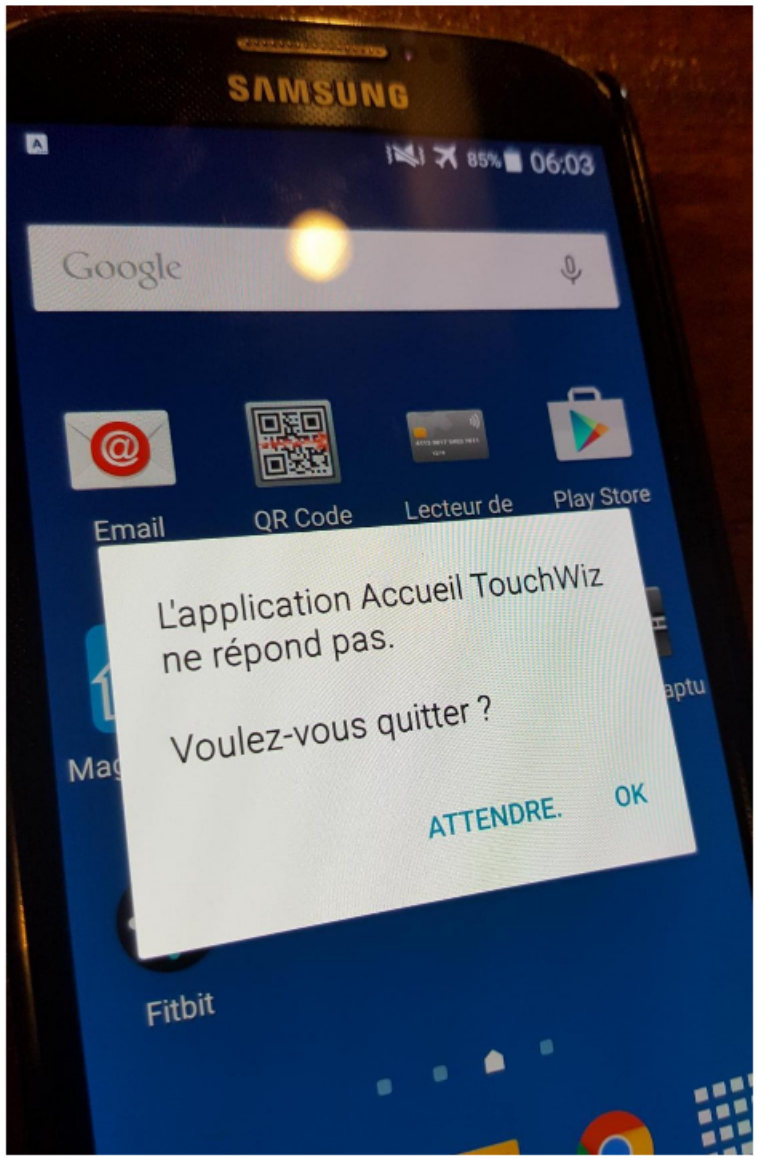
- La V.2 ne fonctionne qu'avec quelques codes PIN ou mots de passe (100/200 ; sur le Samsung Galaxy S4)
- On peut envisager de faire du *fuzzing* à la place d'une attaque par force brute.

Améliorations

- Ajouter le support de Banana/Orange Pi
- Utiliser le port OTG de l'Orange Pi ou du Raspberry Pi A/Zero

Conclusion


SAMSUNG
Erreur réseau. Veuillez vérifier vos connexions réseau, puis réessayer.


SAMSUNG
06:03
Google
Email
QR Code
Lecteur de
Play Store
L'application Accueil TouchWiz ne répond pas.
Voulez-vous quitter ?
ATTENDRE.
OK
Fitbit

Conclusion

Protection

- Effacer le téléphone après un certains nombre de tentatives
- Empêcher l'utilisation de schémas de verrouillage, de mots de passe ou de codes PIN faibles

L'échec Samsung

- Samsung Galaxy Notes 10.1 (Android 4.1.2) après 5 échecs sur le schéma de verrouillage, il est possible d'attaquer le code PIN par force brute sans restriction
- Samsung Galaxy Tab A (Android 5.0.2), il n'est plus possible d'utiliser le clavier physique sur ce code PIN

Des questions ?