Batterie à bord : quand les jauges de carburant dépassent les limites

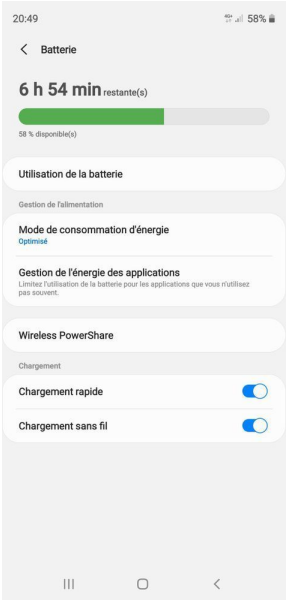
Vincent Giraud David Naccache
prénom.nom@ens.fr

École Normale Supérieure, Université PSL, CNRS
Ingenico

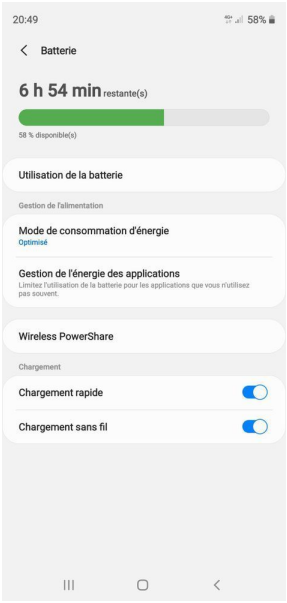
Mercredi 07 juin 2023

Ces travaux s'inscrivent dans une thèse co-encadrée par
David Naccache et Guillaume Bouffard.

L'objectif est d'enquêter sur la faisabilité d'exploitation de processus sensibles sur des environnements qui ne sont pas de confiance.

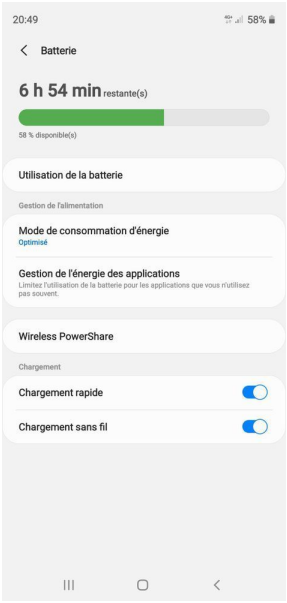


Quelle autonomie me reste-t-il sur mon téléphone ?



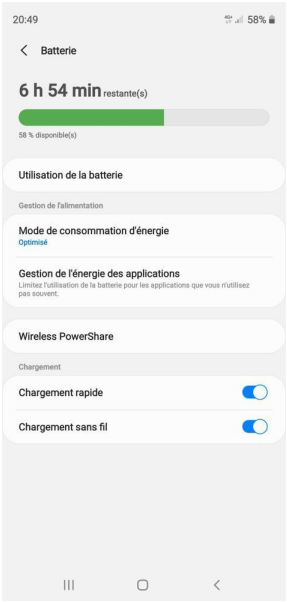
Quelle autonomie me reste-t-il sur mon téléphone ?

Difficile de la déterminer en considérant seulement la **tension aux bornes** de la batterie



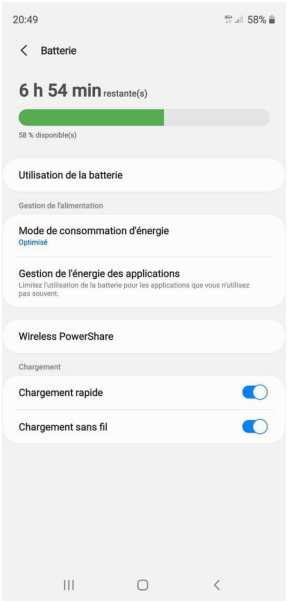
Quelle autonomie me reste-t-il sur mon téléphone ?

Difficile de la déterminer en considérant seulement la **tension aux bornes** de la batterie, mais on peut s'aider de sa **température de fonctionnement**



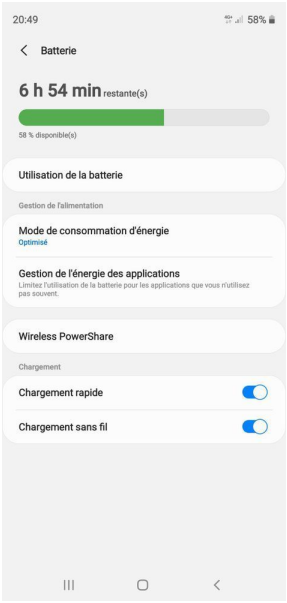
Quelle autonomie me reste-t-il sur mon téléphone ?

Difficile de la déterminer en considérant seulement la **tension aux bornes** de la batterie, mais on peut s'aider de sa **température de fonctionnement**, de son **âge**



Quelle autonomie me reste-t-il sur mon téléphone ?

Difficile de la déterminer en considérant seulement la **tension aux bornes** de la batterie, mais on peut s'aider de sa **température de fonctionnement**, de son **âge**, de la **charge extraite depuis le dernier chargement**



Quelle autonomie me reste-t-il sur mon téléphone ?

Difficile de la déterminer en considérant seulement la **tension aux bornes** de la batterie, mais on peut s'aider de sa **température de fonctionnement**, de son **âge**, de la **charge extraite depuis le dernier chargement**, de la **qualité du lithium**...

Gérer l'énergie sur une plateforme embarquée est délicat. Cela requiert notamment de :

- préserver au maximum l'état de santé de la batterie

Gérer l'énergie sur une plateforme embarquée est délicat. Cela requiert notamment de :

- préserver au maximum l'état de santé de la batterie
- optimiser la consommation

Gérer l'énergie sur une plateforme embarquée est délicat. Cela requiert notamment de :

- préserver au maximum l'état de santé de la batterie
- optimiser la consommation
- optimiser la charge

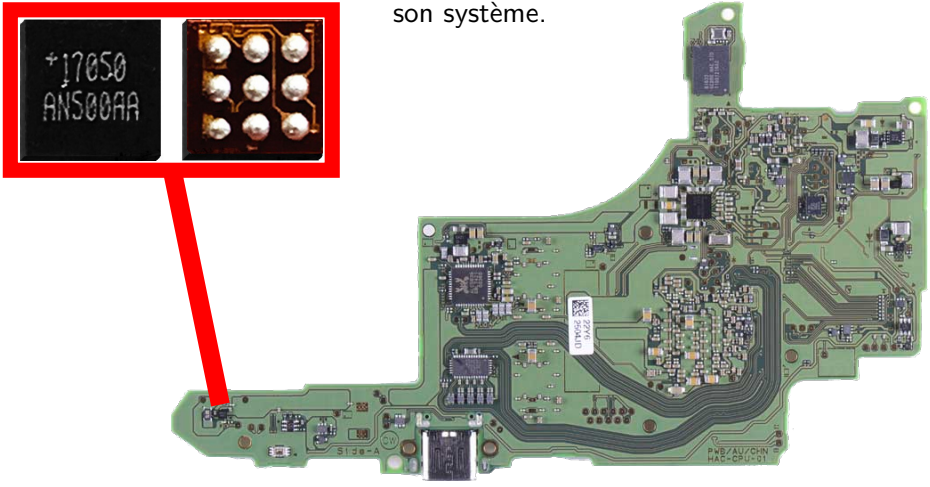
Gérer l'énergie sur une plateforme embarquée est délicat. Cela requiert notamment de :

- préserver au maximum l'état de santé de la batterie
- optimiser la consommation
- optimiser la charge
- pouvoir exploiter des batteries de qualité et d'âge différentes

Gérer l'énergie sur une plateforme embarquée est délicat. Cela requiert notamment de :

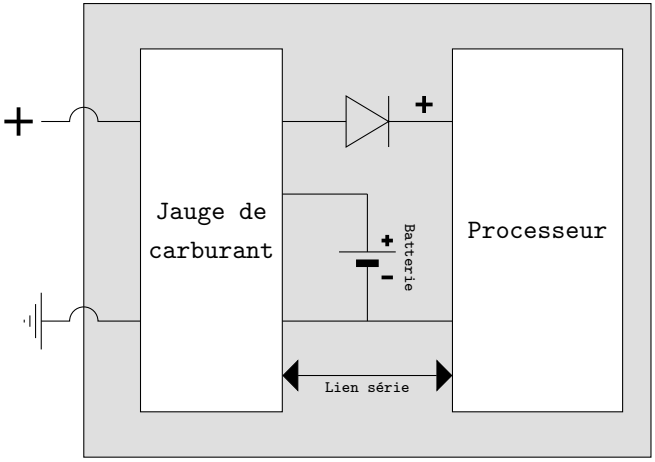
- préserver au maximum l'état de santé de la batterie
- optimiser la consommation
- optimiser la charge
- pouvoir exploiter des batteries de qualité et d'âge différentes
- gérer la sécurité du circuit de puissance

Pour faciliter cet aspect, un concepteur peut intégrer une jauge de carburant [1] dans son système.



Ce circuit intégré va réaliser des mesures en continu de plusieurs métriques autour de la batterie, puis calculer et prédire plusieurs estimations en son encontre.

1. Désignées par *fuel gauge* en anglais.



La jauge de carburant s'interpose souvent entre le système central, et les sources d'énergie.

La présence ou non de jauge à carburant n'est jamais indiquée dans les fiches techniques.

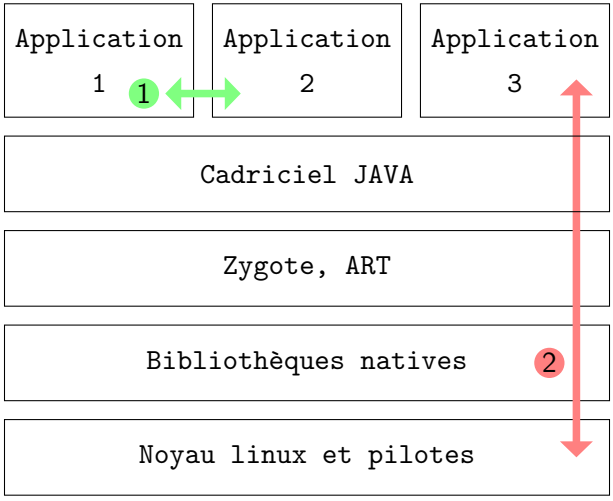
Après achat, on peut visuellement inspecter le circuit imprimé à sa recherche.
Sur Android, on peut logiciellement les détecter sans être *root* :

```
$ ls -a /sys/class/power_supply
battery
dc
gcpm
gcpm_pps
main-charger
maxfg
pca9468-mains
tcpm-source-psy-i2c-max77759tcpc
usb
wireless
```

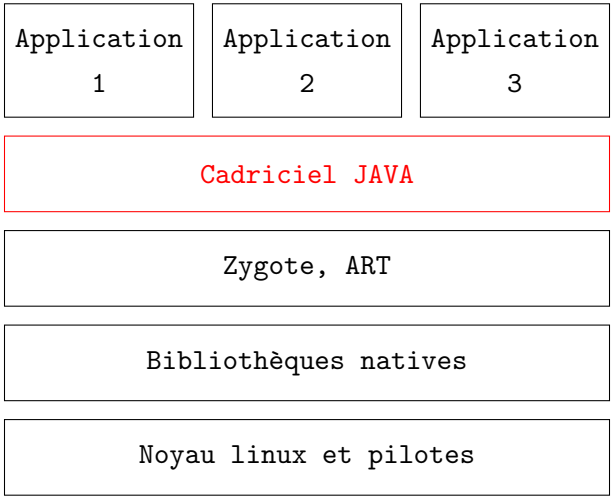
Quid de la sécurité ?

Quid de la sécurité ?

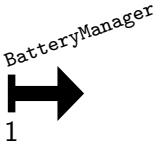
L'état de l'art est, de ce point de vue, inexistant.



Si la politique de sécurité d'Android est claire concernant les interactions horizontales 1, c'est moins le cas des interactions verticales 2.

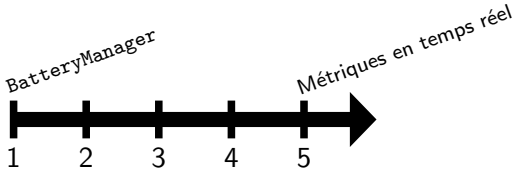


L'implémentation de la modération est essentiellement implémentée au niveau de l'environnement Java.



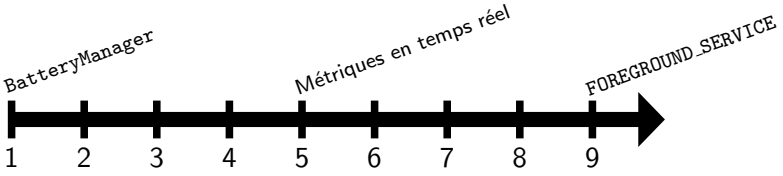
Le service système `BatteryManager` existe depuis les débuts d'Android. À l'époque il ne permettait d'obtenir que quelques rares informations sur la puissance :

- l'état de santé de la batterie
- l'état de charge de la batterie
- si des sources externes sont présentes

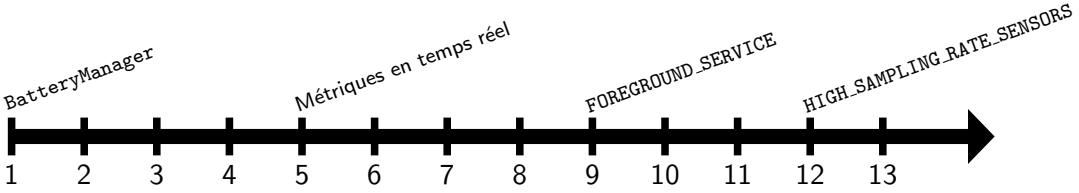


À partir de la version 5 d'Android, le service propose des métriques potentiellement sourcées à partir d'une jauge de carburant, permettant souvent un suivi en temps réel, telles que :

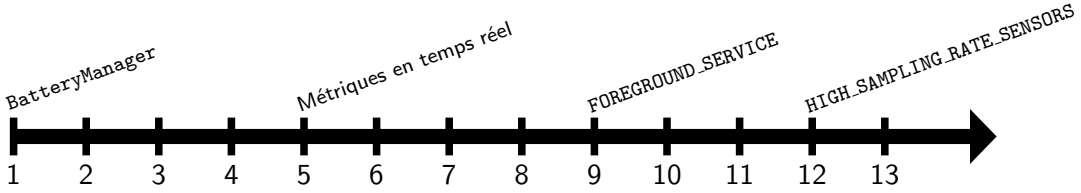
- le courant instantané consommé en microampères
- l'énergie restante en nanowattheures
- la part de charge restante en pourcentage



Avec la version 9 arrive la nécessité de déclarer une notification afin de pouvoir faire tourner un service en arrière plan.



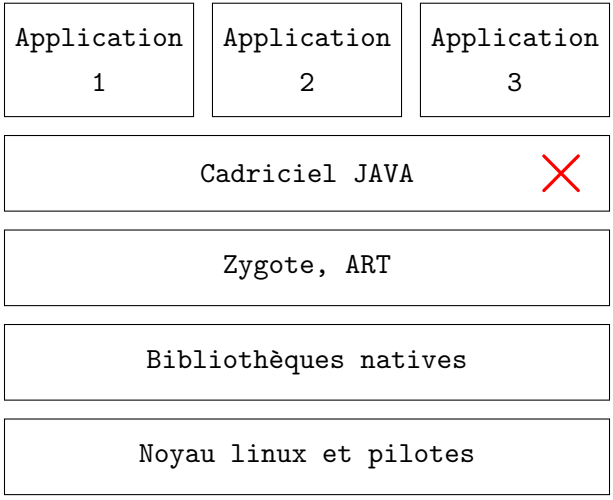
Android 12 introduit une limite de fréquence dans la scrutation des capteurs embarqués...



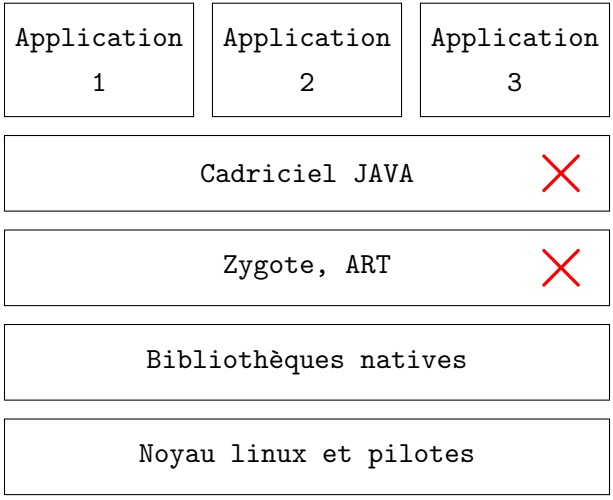
Android 12 introduit une limite de fréquence dans la scrutation des capteurs embarqués... qui ne concerne pas les jauges de carburant.

```
/**
 * Checks if a sensor should be capped according to HIGH_SAMPLING_RATE_SENSORS permission.
 * [...]
 */
private boolean isSensorInCappedSet(int sensorType) {
    return (sensorType == Sensor.TYPE_ACCELEROMETER
            || sensorType == Sensor.TYPE_ACCELEROMETER_UNCALIBRATED
            || sensorType == Sensor.TYPE_GYROSCOPE
            || sensorType == Sensor.TYPE_GYROSCOPE_UNCALIBRATED
            || sensorType == Sensor.TYPE_MAGNETIC_FIELD
            || sensorType == Sensor.TYPE_MAGNETIC_FIELD_UNCALIBRATED);
    }
```

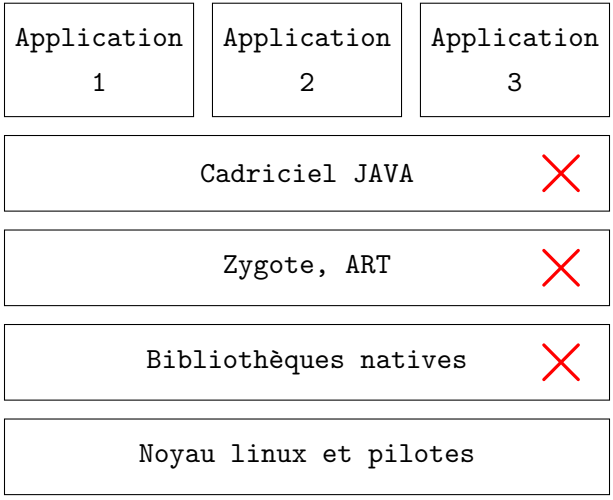
Si les accès à la jauge de carburant ne sont pas modérés au niveau de l'environnement Java, où le sont-ils ?



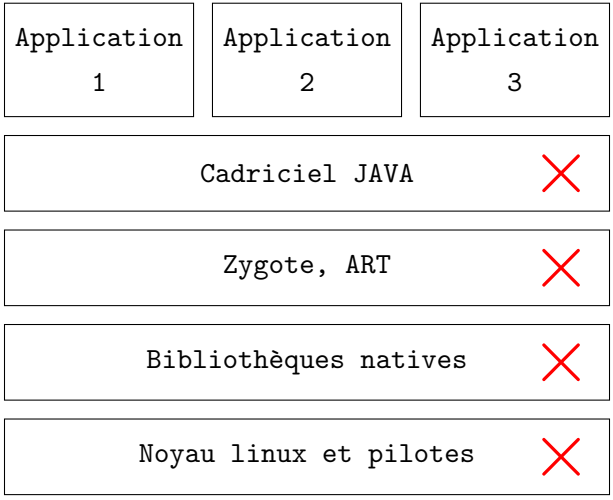
Si les accès à la jauge de carburant ne sont pas modérés au niveau de l'environnement Java, où le sont-ils ?



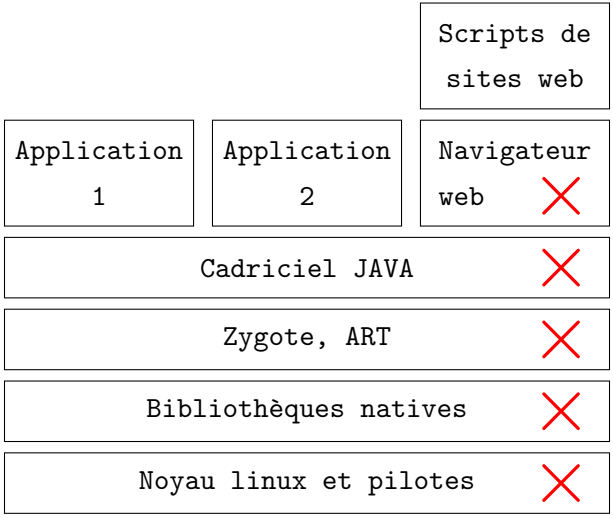
Si les accès à la jauge de carburant ne sont pas modérés au niveau de l'environnement Java, où le sont-ils ?



Si les accès à la jauge de carburant ne sont pas modérés au niveau de l'environnement Java, où le sont-ils ?



Si les accès à la jauge de carburant ne sont pas modérés au niveau de l'environnement Java, où le sont-ils ?



On identifie alors trois risques :

- celui d'un risque de journalisation de l'activité, mettant à mal la vie privée

On identifie alors trois risques :

- celui d'un risque de journalisation de l'activité, mettant à mal la vie privée
- celui d'un canal de communication caché entre deux applications

On identifie alors trois risques :

- celui d'un risque de journalisation de l'activité, mettant à mal la vie privée
- celui d'un canal de communication caché entre deux applications
- celui se basant sur l'espionnage d'un processus sécurisé

On identifie alors trois risques :

- celui d'un risque de journalisation de l'activité, mettant à mal la vie privée
- celui d'un canal de communication caché entre deux applications
- celui se basant sur l'espionnage d'un processus sécurisé

L'état de l'art regorge de techniques d'espionnage de code PIN. Celles-ci se basent souvent sur l'analyse temporelle, à partir de divers canaux auxiliaires :

- les émissions électromagnétiques

L'état de l'art regorge de techniques d'espionnage de code PIN. Celles-ci se basent souvent sur l'analyse temporelle, à partir de divers canaux auxiliaires :

- les émissions électromagnétiques
- le son

L'état de l'art regorge de techniques d'espionnage de code PIN. Celles-ci se basent souvent sur l'analyse temporelle, à partir de divers canaux auxiliaires :

- les émissions électromagnétiques
- le son
- les rotations et mouvements

L'état de l'art regorge de techniques d'espionnage de code PIN. Celles-ci se basent souvent sur l'analyse temporelle, à partir de divers canaux auxiliaires :

- les émissions électromagnétiques
- le son
- les rotations et mouvements
- le courant à travers un câble de charge

L'état de l'art regorge de techniques d'espionnage de code PIN. Celles-ci se basent souvent sur l'analyse temporelle, à partir de divers canaux auxiliaires :

- les émissions électromagnétiques
- le son
- les rotations et mouvements
- le courant à travers un câble de charge

En exploitant les jauges de carburant, on souhaite proposer une attaque non impactée par la politique de sécurité, discrète, et sans entraînement.

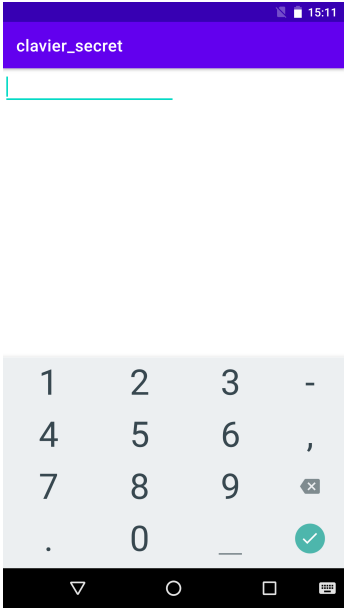
L'état de l'art regorge de techniques d'espionnage de code PIN. Celles-ci se basent souvent sur l'analyse temporelle, à partir de divers canaux auxiliaires :

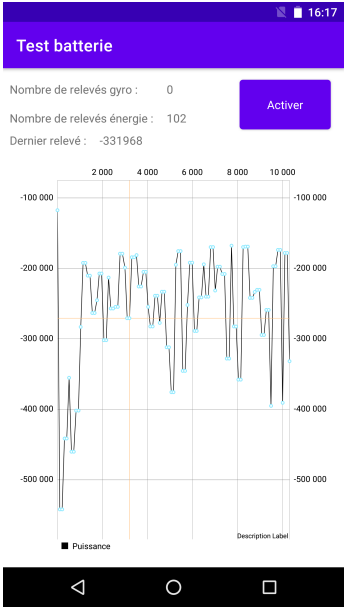
- les émissions électromagnétiques
- le son
- les rotations et mouvements
- le courant à travers un câble de charge

En exploitant les jauges de carburant, on souhaite proposer une attaque non impactée par la politique de sécurité, discrète, et sans entraînement.

Ici, on ne s'intéressera qu'à une seule métrique : le courant entrant ou sortant de la batterie en temps réel.

Pour cette preuve de concept, on considère une simple application cible.



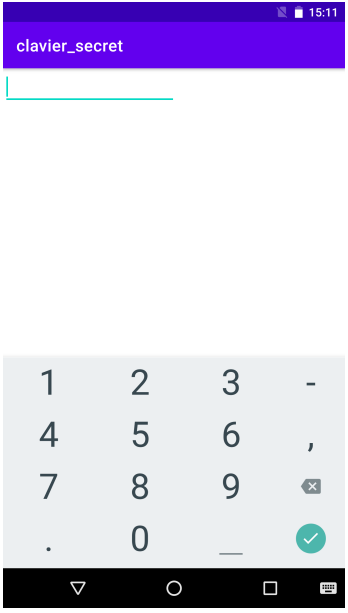


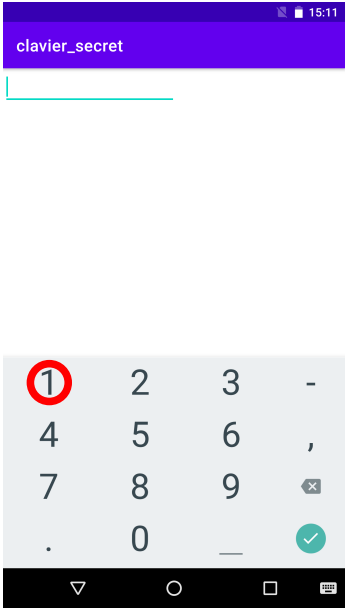
En parallèle, on a réalisé une application d'attaque, qui scrute autant que possible la consommation mesurée par la jauge de carburant du téléphone.

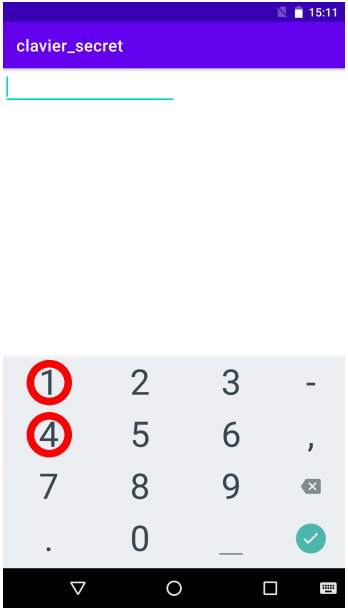
Pour éviter de perturber l'opération, on n'exporte pas les données en temps réel sur un lien filaire ou non.

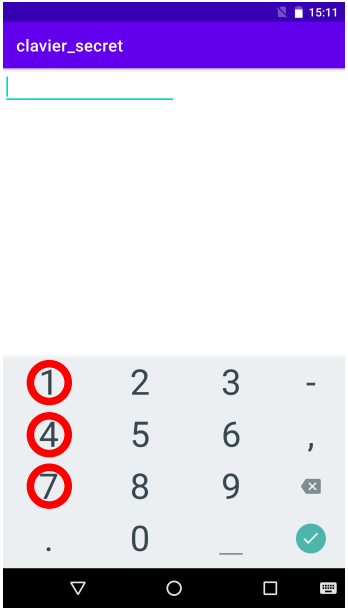
```
BatteryManager mBatteryManager =
          (BatteryManager) this.getSystemService(Context.BATTERY_SERVICE);
long courant =
          mBatteryManager.getLongProperty(BatteryManager.BATTERY_PROPERTY_CURRENT_NOW);
```

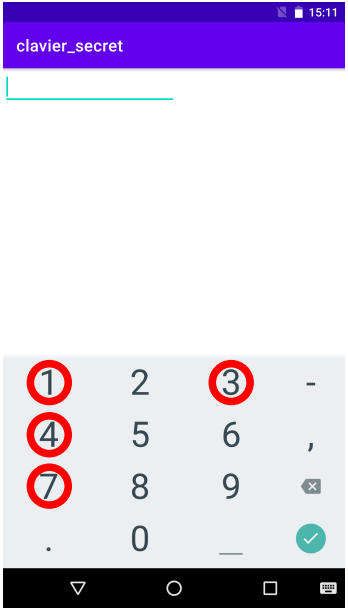
Pour que notre processus de scrutation reste actif, on le place dans un service Android.

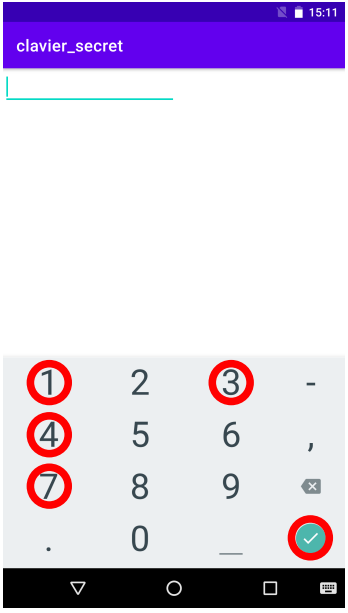

15:11
clavier_secret
1 2 3 -
4 5 6 ,
7 8 9
. 0 _

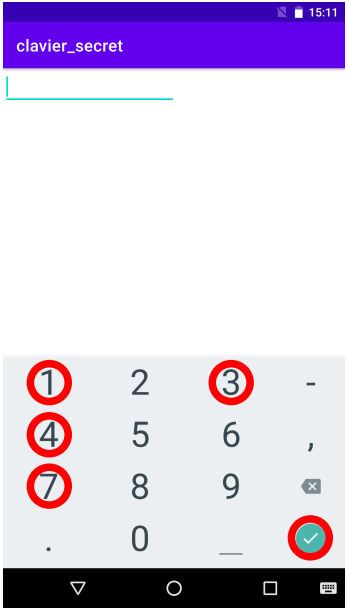

15:11
clavier_secret
1 2 3 -
4 5 6 ,
7 8 9
. 0 _

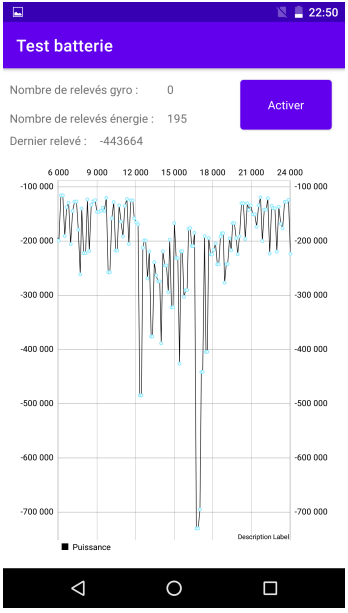

15:11
clavier_secret
1 2 3 -
4 5 6 ,
7 8 9
. 0 _

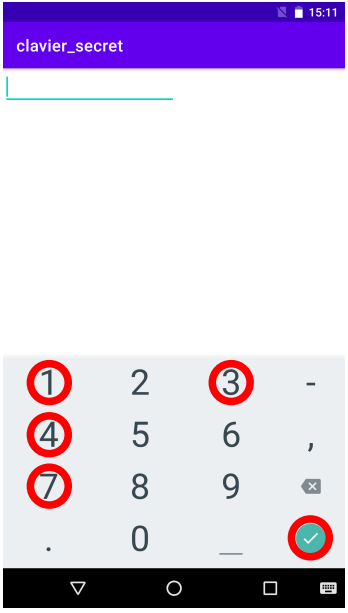

15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0

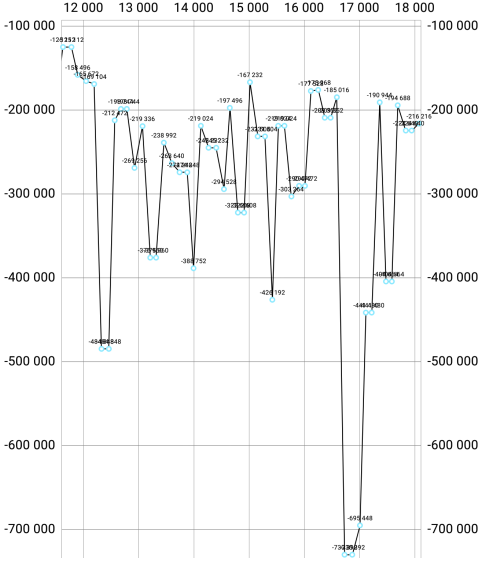

15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0

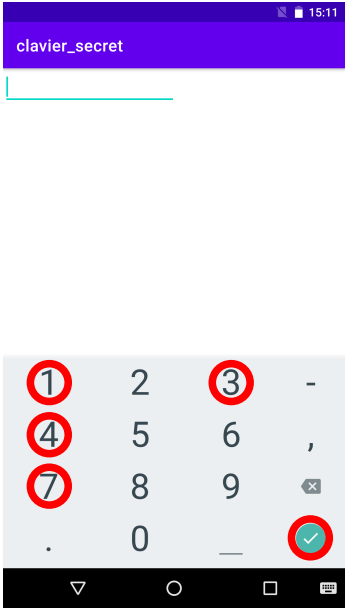

15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0
_

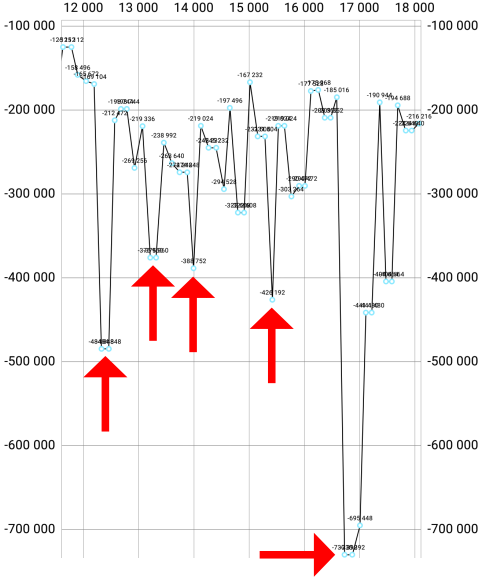

15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0

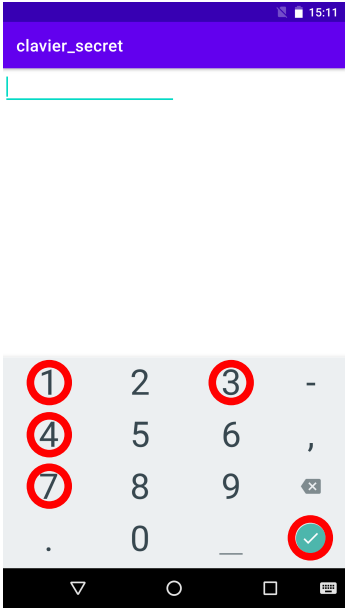

22:50
Test batterie
Nombre de relevés gyro : 0
Nombre de relevés énergie : 195
Dernier relevé : -443664
Activer
6 000
9 000
12 000
15 000
18 000
21 000
24 000
-100 000
-200 000
-300 000
-400 000
-500 000
-600 000
-700 000
Description Label
Puissance

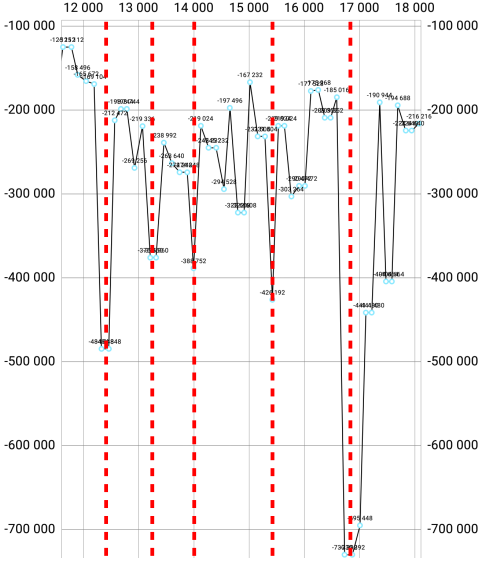

15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0
_

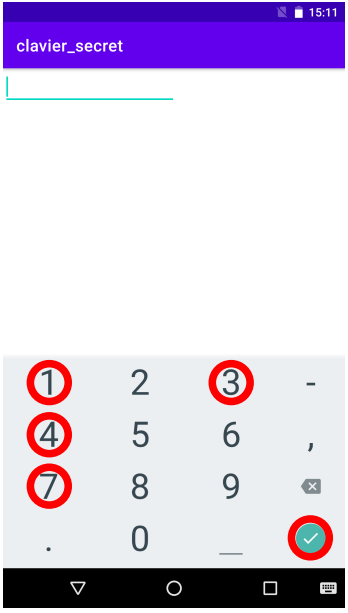

12 000
13 000
14 000
15 000
16 000
17 000
18 000
-100 000
-200 000
-300 000
-400 000
-500 000
-600 000
-700 000

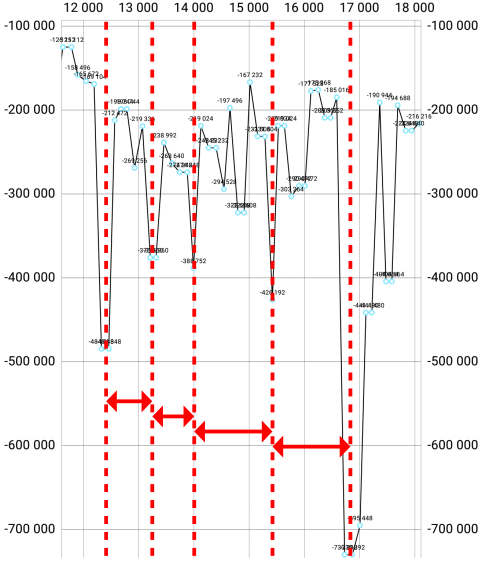

15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0
_

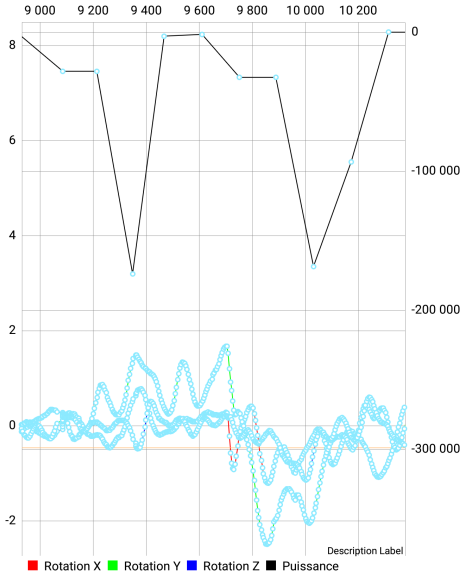

12 000
13 000
14 000
15 000
16 000
17 000
18 000
-100 000
-200 000
-300 000
-400 000
-500 000
-600 000
-700 000


15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0

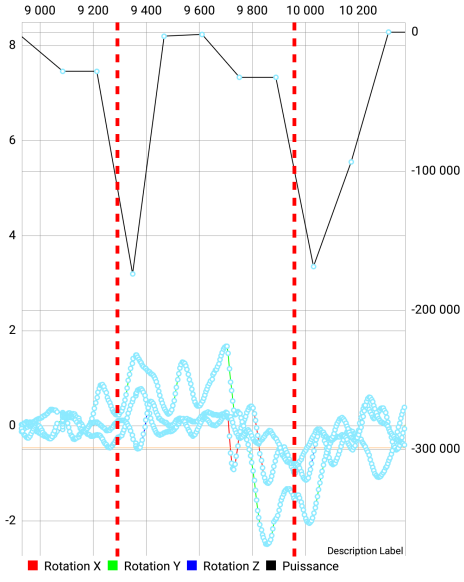

12 000
13 000
14 000
15 000
16 000
17 000
18 000
-100 000
-200 000
-300 000
-400 000
-500 000
-600 000
-700 000


15:11
clavier_secret
1
2
3
-
4
5
6
,
7
8
9
.
0

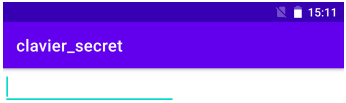

12 000
13 000
14 000
15 000
16 000
17 000
18 000
-100 000
-200 000
-300 000
-400 000
-500 000
-600 000
-700 000



On peut affiner le positionnement temporel avec les données gyroscopiques par exemple.

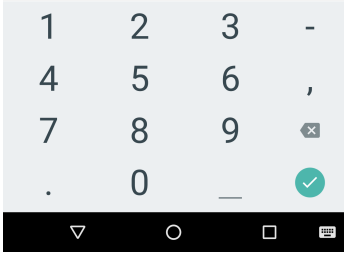


On peut affiner le positionnement temporel avec les données gyroscopiques par exemple.



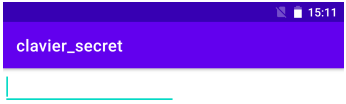
Une fois les données récoltées, on peut réappliquer les méthodes d'attaques temporelles présentes dans l'état de l'art.

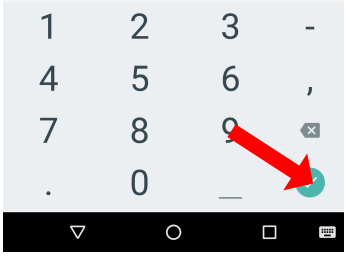
Ici, nous nous basons sur le déroulement déterministe d'un arbre des codes possibles.

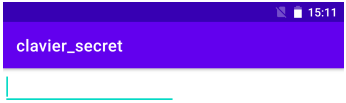


Une fois les données récoltées, on peut réappliquer les méthodes d'attaques temporelles présentes dans l'état de l'art.

Ici, nous nous basons sur le déroulement déterministe d'un arbre des codes possibles.

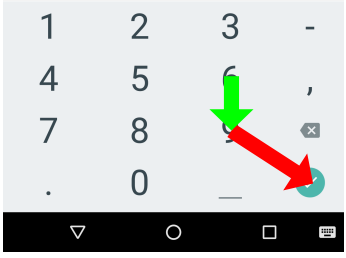


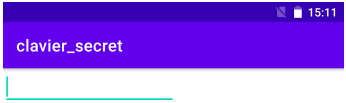




Une fois les données récoltées, on peut réappliquer les méthodes d'attaques temporelles présentes dans l'état de l'art.

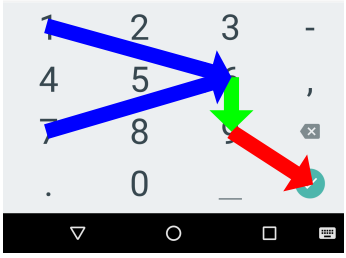
Ici, nous nous basons sur le déroulement déterministe d'un arbre des codes possibles.

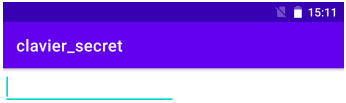




Une fois les données récoltées, on peut réappliquer les méthodes d'attaques temporelles présentes dans l'état de l'art.

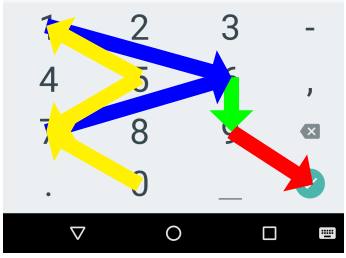
Ici, nous nous basons sur le déroulement déterministe d'un arbre des codes possibles.





Une fois les données récoltées, on peut réappliquer les méthodes d'attaques temporelles présentes dans l'état de l'art.

Ici, nous nous basons sur le déroulement déterministe d'un arbre des codes possibles.



Démonstration de la capture

Démonstration du script récursif

Les possibilités démontrées ici compromettent la confidentialité sur les plateformes Android.

Si l'on développe des solutions sensibles destinées à ces environnements, il convient de ne pas compter sur une isolation logicielle parfaite, quand bien même on serait absolument sûr que les droits administrateurs n'ont pas été débloqués.

Annihiler ce vecteur d'attaque est simple lorsqu'on contrôle le système.

Une des actions possibles est de légèrement modifier le gestionnaire de batterie au niveau du cadriciel Java.

Contrer ce genre d'attaque est plus compliqué lorsque l'on est un développeur ayant uniquement accès à la couche applicative.

Parmi les solutions actuellement en cours d'étude, on considère des techniques à base de brouillage ou de simulation de signaux sensibles.

En conclusion :

- si les jauge de carburant sont utiles pour les concepteurs de système, leur intégration requière une certaine vigilance

En conclusion :

- si les jauge de carburant sont utiles pour les concepteurs de système, leur intégration requière une certaine vigilance
- en tant que simple développeur tiers, il ne faut pas entièrement se reposer sur les garanties d'isolation d'Android

En conclusion :

- si les jauge de carburant sont utiles pour les concepteurs de système, leur intégration requière une certaine vigilance
- en tant que simple développeur tiers, il ne faut pas entièrement se reposer sur les garanties d'isolation d'Android
- les autres risques précités sont en cours d'étude

En conclusion :

- si les jauge de carburant sont utiles pour les concepteurs de système, leur intégration requière une certaine vigilance
- en tant que simple développeur tiers, il ne faut pas entièrement se reposer sur les garanties d'isolation d'Android
- les autres risques précités sont en cours d'étude
- des protections au niveau applicatif seulement le sont aussi

En conclusion :

- si les jauge de carburant sont utiles pour les concepteurs de système, leur intégration requière une certaine vigilance
- en tant que simple développeur tiers, il ne faut pas entièrement se reposer sur les garanties d'isolation d'Android
- les autres risques précités sont en cours d'étude
- des protections au niveau applicatif seulement le sont aussi

Merci à Guillaume Bouffard de l'Agence Nationale de la Sécurité des Systèmes d'Information pour son support dans ces travaux.