Dissecting a Cloud-Connected E-Scooter

JD-HITBSecConf 2018
November 2, 2018
Beijing, China

Nikias Bassen
@pimskeks
nikias@corellium.com

Outline

- Introduction
- The Target
- Smartphone App
- GSM/GPRS Connectivity
- Small Demo
- Conclusions

About me (1)

- IT Expert from Germany, Diploma in Computer Science (University of Bremen, Germany)
- Involved in RE & Security Research for > 15 years 😱
- RE of iTunes database hashing algorithm
- RE of iTunes/iOS communication protocols
- Leading Developer of libimobiledevice project

About me (2)

- 2018-now VP of Platform & Security, CORELLIUM
- 2017-2018 VP of Platform Research, ZIMPERIUM
- 2015-2017 Mobile Security Researcher, ZIMPERIUM
- 2010-2015 Self-Employed, custom IT solutions
  - RE & Research as a hobby
  - 2013 evad3rs
  - 2012 Jailbreak Dream Team
  - 2011 Chronic-Dev Team

Why this topic?

- Started to work ~6 months ago at Corellium, virtualizing iPhones (amazing stuff!)
- We have lots of work to focus on, no time to do any research (hopefully again in the near future)
- No completed research on iOS currently
- I just bought that E-Scooter, and said "why not?!"

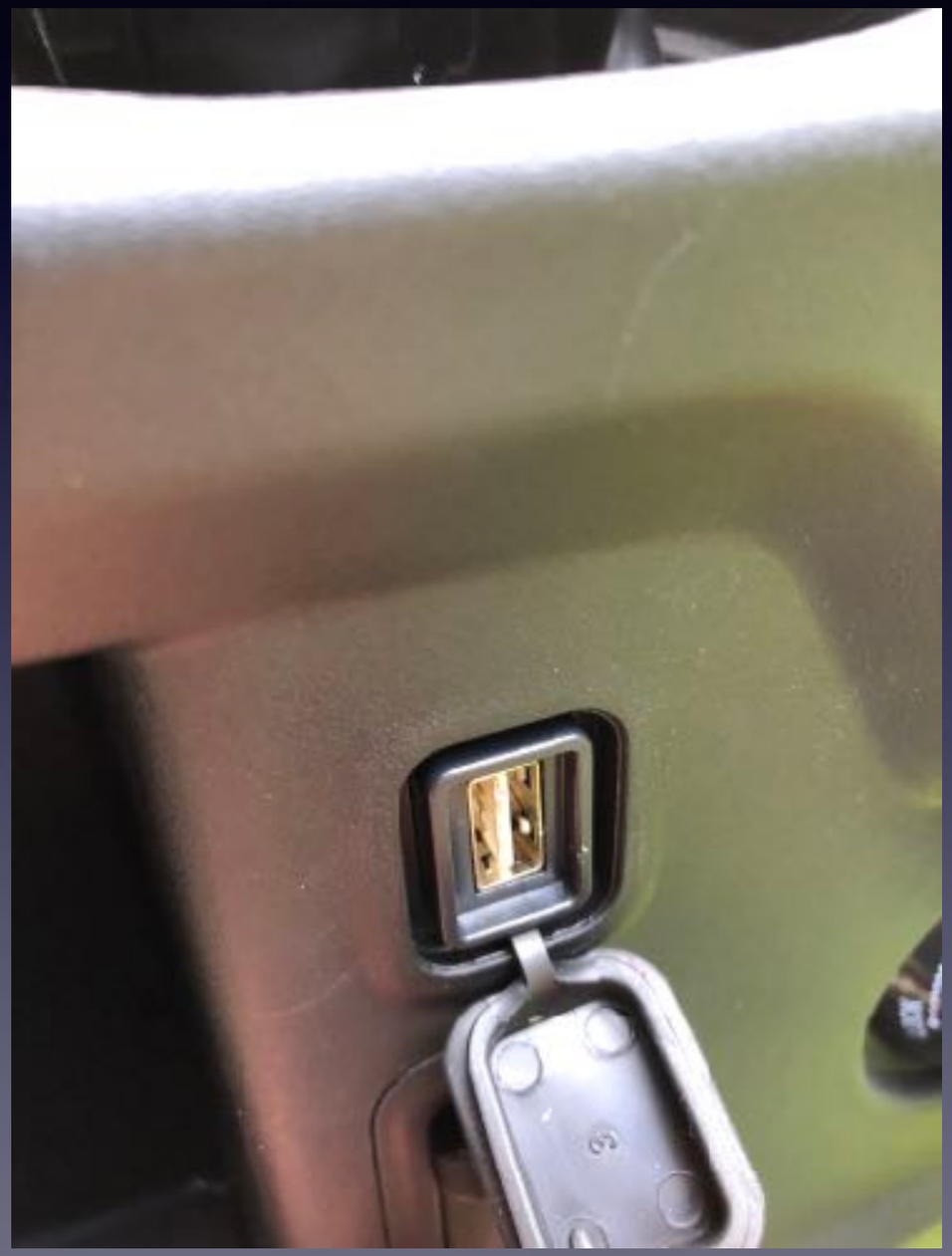
The Target

The Target

- Niu N1S E-Scooter
- Jiangsu Niu Electric Technology Co., Ltd., China
- Cloud-Connected (GSM)
- Smartphone App
- USB Port

USB Port - Diagnostics?

- Vendor: charging port for phone
- Me: maybe used for diagnostics?
- Raspberry Pi: Nah…
- Nope, doesn't detect anything



Real Diagnostics Port

- Battery charging port under the seat also used as diagnostics port
- Used by NIU dealer with dedicated diagnostics device
- Supposedly RS-485 serial communication
- Couldn’t check, lack of time and hardware

China Shopping List ++

Yes, the connector won't fit,
but you get the idea :)

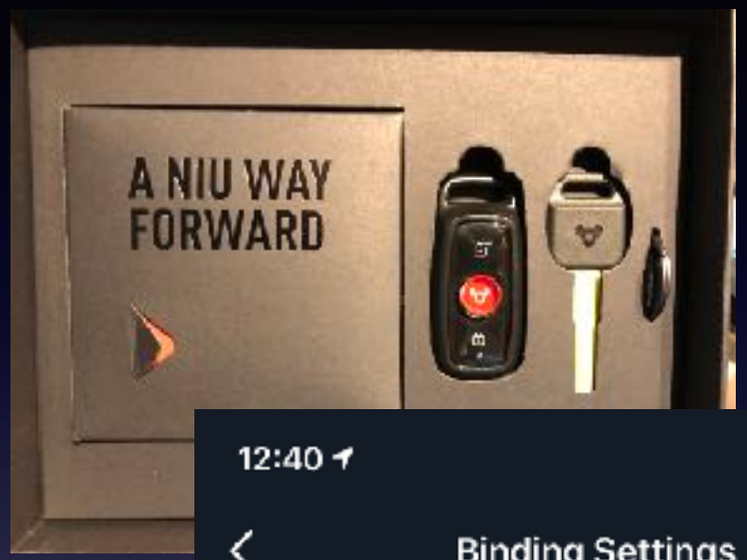
Smartphone App

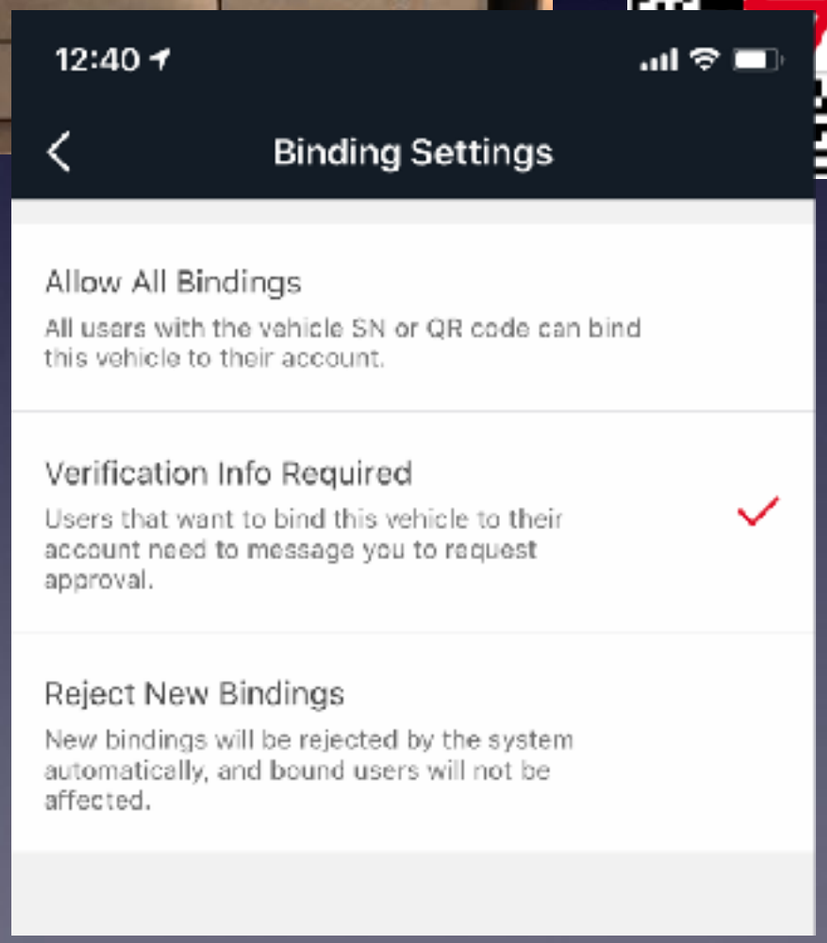
Smartphone App

- Battery level & estimated distance
- Lock status
- Current location
- Weather report
- Overview of past trips and statistics
- Smart Check (scooter self-diagnosis)
- Service information
- Push notifications about unexpected movement, battery removal, etc.

Registration + Binding

- Account registration required with Phone number or Email
- Scooter needs to be bound to account
- S/N required, printed on manual (QR code), not found on vehicle itself
- By default, adding someone's S/N requires confirmation (see screenshot on the right)
- One vehicle can be bound to 5 accounts max.





Let's dump the App

- Jailbroken iPhone + Clutch
- IDA Pro Disassembler
- ID: com.niu.xiaoniuAborad
- Lastest version: 3.4.8
  (version initially dumped 3.4.6)
- Binary: managerAborad.app/
  managerAborad
- Most likely a typo
  Aborad => Abroad

First, lets have a look

```
3_loop.json
A4_11_produce_m.bundle
A4_11_produce_n.bundle
A4_11_produce_u.bundle
A4_11_produce_um.bundle
A4_11_street.bundle
AMap.bundle
AlipaySDK.bundle
AppIcon29x29@2x.png
AppIcon29x29@2x~ipad.png
AppIcon29x29@3x.png
AppIcon40x40@2x.png
AppIcon40x40@2x~ipad.png
AppIcon40x40@3x.png
AppIcon60x60@2x.png
AppIcon60x60@3x.png
AppIcon76x76@2x~ipad.png
AppIcon76x76~ipad.png
AppIcon83.5x83.5@2x~ipad.png
Assets.car
B1_safe_open_failure.bundle
Base.lproj
DINOffcPro-Black.ttf
DINOffcPro-BlackItalic.ttf
DINOffcPro-Bold.ttf
DINOffcPro-BoldItalic.ttf
DINOffcPro-Cond.ttf
DINOffcPro-CondBlack.ttf
DINOffcPro-CondBlackItalic.ttf
DINOffcPro-CondBold.ttf
DINOffcPro-CondBoldItalic.ttf
DINOffcPro-CondExtlight.ttf
DINOffcPro-CondExtlightItalic.ttf
DINOffcPro-CondItalic.ttf
DINOffcPro-CondLight.ttf
DINOffcPro-CondLightItalic.ttf
DINOffcPro-CondMedium.ttf
DINOffcPro-CondMediumItalic.ttf
DINOffcPro-CondThin.ttf
DINOffcPro-CondThinItalic.ttf
DINOffcPro-Extlight.ttf
DINOffcPro-ExtlightItalic.ttf
DINOffcPro-Italic.ttf
DINOffcPro-Light.ttf
DINOffcPro-LightItalic.ttf
DINOffcPro-Medium.ttf
DINOffcPro-MediumItalic.ttf
DINOffcPro-Thin.ttf
DINOffcPro-ThinItalic.ttf
DINOffcPro.ttf
EXT_RELEASE.json
EXT_TEST.json
EditNickNameViewController.nib
FontAwesome.ttf
Frameworks
GoogleMaps.bundle
GooglePlaces.bundle
IQKeyboardManager.bundle
Info.plist
InitiateTransferViewController.nib
J1_10_status_ balance.bundle
J1_11_status_calibration.bundle
LaunchScreen.storyboardc
MJRefresh.bundle
MaterialIcons.ttf
NDBatteryChartTipInfo.nib
NDQRCodeSaveView.nib
NiuStatusCellConfig.plist
Pingpp.bundle
PkgInfo
ServiceRecodeHeaderView.nib
ServiceRecodeMapController.nib
ServiceSegmentView.nib
TencentOpenApi_IOS_Bundle.bundle
TransferUserHCell.nib
TransferUserHeaderView.nib
TransferUserLCell.nib
TwitterKitResources.bundle
TwitterShareExtensionUIResources.bundle
UMSocialSDKPromptResources.bundle
VeticleSelectedView.nib
WeatherCode.plist
WeiboSDK.bundle
YLStatistics.json
_CodeSignature
commonCountryCode.plist
common_loading_red.bundle
common_loading_white.bundle
common_map_skin.bundle
country_code_grouped.json
de.lproj
en.lproj
es.lproj
fr.lproj
it.lproj
language-overseas.json
managerAbcrad
mystyle.data
niu_nc_push_config.json
nl.lproj
rn.bundle
sv.lproj
zh-Hans.lproj
zh-Hant.lproj
```

First, lets have a look

```
3_loop.json
A4_11_produce_m.bundle
A4_11_produce_n.bundle
A4_11_produce_u.bundle
A4_11_produce_um.bundle
A4_11_street.bundle
AMap.bundle
AlipaySDK.bundle
AppIcon29x29@2x.png
AppIcon29x29@2x~ipad.png
AppIcon29x29@3x.png
AppIcon40x40@2x.png
AppIcon40x40@2x~ipad.png
AppIcon40x40@3x.png
AppIcon60x60@2x.png
AppIcon60x60@3x.png
AppIcon76x76@2x~ipad.png
AppIcon76x76~ipad.png
AppIcon83.5x83.5@2x~ipad.png
Assets.car
B1_safe_open_failure.bundle
Base.lproj
DINOffcPro-Black.ttf
DINOffcPro-BlackItalic.ttf
DINOffcPro-Bold.ttf
DINOffcPro-BoldItalic.ttf
DINOffcPro-Cord.ttf
DINOffcPro-CordBlack.ttf
DINOffcPro-CordBlackItalic.ttf
DINOffcPro-CordBold.ttf
DINOffcPro-CordBoldItalic.ttf
DINOffcPro-CordExtlight.ttf
DINOffcPro-CordExtlightItalic.ttf
DINOffcPro-CordItalic.ttf
DINOffcPro-CordLight.ttf
DINOffcPro-CondLightItalic.ttf
DINOffcPro-CondMedium.ttf
DINOffcPro-CondMediumItalic.ttf
DINOffcPro-CondThin.ttf
DINOffcPro-CondThinItalic.ttf
DINOffcPro-Extlight.ttf
DINOffcPro-ExtlightItalic.ttf
DINOffcPro-Italic.ttf
DINOffcPro-Light.ttf
DINOffcPro-LightItalic.ttf
DINOffcPro-Medium.ttf
DINOffcPro-MediumItalic.ttf
DINOffcPro-Thin.ttf
DINOffcPro-ThinItalic.ttf
DINOffcPro.ttf
EXT_RELEASE.json
EXT_TEST.json
EditNickNameViewController.nib
FontAwesome.ttf
Frameworks
GoogleMaps.bundle
GooglePlaces.bundle
IQKeyboardManager.bundle
Info.plist
InitiateTransferViewController.nib
J1_10_status_ balance.bundle
J1_11_status_calibration.bundle
LaunchScreen.storyboardc
MJRefresh.bundle
MaterialIcons.ttf
NDBatteryChartTipInfo.nib
NDQRCodeSaveView.nib
NiuStatusCellConfig.plist
Pingpp.bundle
PkgInfo
ServiceRecodeHeaderView.nib
ServiceRecodeMapController.nib
ServiceSegmentView.nib
TencentOpenApi_IOS_Bundle.bundle
TransferUserHCell.nib
TransferUserHeaderView.nib
TransferUserLCell.nib
TwitterKitResources.bundle
TwitterShareExtensionUIResources.bundle
UMSocialSDKPromptResources.bundle
VeticleSelectedView.nib
WeatherCode.plist
WeiboSDK.bundle
YLStatistics.json
_CodeSignature
commonCountryCode.plist
common_loading_red.bundle
common_loading_white.bundle
common_map_skin.bundle
country_code_grouped.json
de.lproj
en.lproj
es.lproj
fr.lproj
it.lproj
language-overseas.json
managerAbcrad
mystyle.data
niu_nc_push_config.json
nl.lproj
rn.bundle
sv.lproj
zh-Hans.lproj
zh-Hant.lproj
```

EXT_RELEASE.json


```
{
    "payload": {
        "USER_LOGIN": {
            "desc": "1.1. 用户名密码登陆接口",
            "url": "https://account.niu.com/appv2/login"
        },
        "USER_SENDCODE": {
            "desc": "1.2. 获取验证码接口",
            "url": "https://account.niu.com/appv2/sendcode"
        },
        "USER_RESETPASSWORD": {
            "desc": "1.3. 重置密码接口",
            "url": "https://account.niu.com/appv2/resetpassword"
        },
        "USER_SIGNUP": {
            "desc": "1.4. 用户注册接口",
            "url": "https://account.niu.com/appv2/signup"
        },
        "USER_LOGOUT": {
            "desc": "1.5. 退出登陆",
            "url": "https://account.niu.com/appv2/logout"
        },
        "USER_BASICINFO_UPDATE": {
            "desc": "1.8更新个人信息",
            "url": "https://account.niu.com/appv2/basicinfo/update"
        },
        "USER_UPDATEJPUSHID": {
            "desc": "1.10. 更新极光推送id接口",
            "url": "https://account.niu.com/appv2/updatejpushid"
        },
        "VEHICLE_SETSNNAME": {
            "desc": "3.3. 给车命名接口",
            "url": "https://app-api.niu.com/motoinfo/setsnname"
        },
        "VEHICLE_LIST": {
            "desc": "3.4获取已绑定车辆列表接口",
            "url": "https://app-api.niu.com/motoinfo/list"
        },
        "VEHICLE_SETDEAULT": {
            "desc": "3.5设置默认车辆",
            "url": "https://app-api.niu.com/userinfo/setdefault"
        },
        "VEHICLE_CURRENTPOS": {
            "desc": "3.6. 获取当前车辆坐标",
            "url": "https://app-api.niu.com/motoinfo/currentpos"
        },
        "URL_VEHICLE_BATTERYINFO": {
            "desc": "3.8. 电池信息接口",
            "url": "https://app-api.niu.com/v3/motor_data/battery_info"
        },
        "URL_VEHICLE_BATTERYINFO_2": {
            "desc": "3.23. ",
            "url": "https://app-api.niu.com/motoinfo/batteryinfo/v2"
        },
        "VEHICLE_BINDLIST": {
            "desc": "3.12. 车主查看已绑定用户列表",
            "url": "https://app-api.niu.com/userinfo/bindlist"
        },
        "VEHICLE_RENAME_BIND_USER": {
```

Web API !

- URLs for different actions:
  - User signup, login, account & permission settings
  - Vehicle position, battery and health status, smart check
  - Service status, Ownership transfer
  - Theft reports
  - Driving statistics
  - Some social media stuff

EXT_TEST.json ?

- Same API calls, just different base URL
  `account-dev.niucache.com` instead of `account.niu.com`
  `app-api-dev.niucache.com` instead of `app-api.niu.com`
- App offers test account

Let’s check how this works

```
__text:000000010012CA90    LDR     X23, [SP,#0x70+var_68]
__text:000000010012CA94    LDR     X0, [X22,#classRef_NSString@PAGEOFF] ; id
__text:000000010012CA98    MOV     X1, X27 ; SEL
__text:000000010012CA9C    MOV     X2, X20
__text:000000010012CAA0    BL      _objc_msgSend
__text:000000010012CAA4    MOV     X29, X29
__text:000000010012CAA8    BL      _objc_retainAutoreleasedReturnValue
__text:000000010012CAAC    MOV     X21, X20
__text:000000010012CAB0    MOV     X20, X0
__text:000000010012CAB4    ADRP    X3, #cfstr_Account_1@PAGE ; "account"
__text:000000010012CAB8    ADD     X3, X3, #cfstr_Account_1@PAGEOFF ; "account"
__text:000000010012CABC    MOV     X0, X25 ; id
__text:000000010012CAC0    MOV     X1, X28 ; SEL
__text:000000010012CAC4    MOV     X2, X20
__text:000000010012CAC8    BL      _objc_msgSend
__text:000000010012CACC    MOV     X0, X20 ; id
__text:000000010012CAD0    BL      _objc_release
__text:000000010012CAD4    LDR     X0, [X22,#classRef_NSString@PAGEOFF] ; id
__text:000000010012CAD8    MOV     X1, X27 ; SEL
__text:000000010012CADC    MOV     X2, X26
__text:000000010012CAE0    BL      _objc_msgSend
__text:000000010012CAE4    MOV     X29, X29
__text:000000010012CAE8    BL      _objc_retainAutoreleasedReturnValue
__text:000000010012CAEC    MOV     X20, X0
__text:000000010012CAF0    MOV     X0, X25 ; id
__text:000000010012CAF4    BL      _objc_release
__text:000000010012CAF8    ADRP    X3, #cfstr_Password_1@PAGE ; "password"
__text:000000010012CAFC    ADD     X3, X3, #cfstr_Password_1@PAGEOFF ; "password"
__text:000000010012CB00    MOV     X0, X25 ; id
__text:000000010012CB04    MOV     X1, X28 ; SEL
__text:000000010012CB08    MOV     X2, X20
__text:000000010012CB0C    BL      _objc_msgSend
__text:000000010012CB10    MOV     X0, X20 ; id
__text:000000010012CB14    BL      _objc_release
__text:000000010012CB18    ADRP    X8, #selRef_postWithUrl_parameters_success_failure_error_@PAGE
__text:000000010012CB1C    LDR     X1, [X8,#selRef_postWithUrl_parameters_success_failure_error_@PAGEOFF] ; SEL
__text:000000010012CB20    ADRP    X2, #cfstr_UserLogin@PAGE ; "USER_LOGIN"
__text:000000010012CB24    ADD     X2, X2, #cfstr_UserLogin@PAGEOFF ; "USER_LOGIN"
__text:000000010012CB28    MOV     X0, X23 ; id
__text:000000010012CB2C    MOV     X3, X25
__text:000000010012CB30    LDP     X22, X20, [SP,#0x70+var_60]
__text:000000010012CB34    MOV     X4, X20
__text:000000010012CB38    MOV     X5, X22
__text:000000010012CB3C    MOV     X6, X24
__text:000000010012CB40    BL      _objc_msgSend
```

Let’s check how this works

```
$ curl -H "Content-Type: application/json" --request POST --data '{"account":"nXXXX@YYYY.ZZ", "password":"yeah,Right"}' https://
account.niu.com/appv2/login
{"data":{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyaWQiOiI1KioqKioqKioqKioqKioqKioqKioqKioiLCJsb2dpbmlkIjoiIiwidiI6M
SwiaWF0IjoxNTQwNjc3NzMzLCJleHAiOjE1NDg0NTM3MjJ9.6xYHIyfK-RWisdwmNzp15U6ef-XnMnoWwXKbLYeX-Y7","user":{"nickname":"redacted","real
name":"Nope","headimg":"","mobile":"","uid":"5**********************","auto_id":"niu_123456789","countryCode":"49","sex":"","bi
rthdate":"","emails":[{"address":"nXXXX@YYYY.ZZ","verified":true}]}},"desc":"成功","trace":"","status":0}
```

Token!

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
  => {"typ":"JWT","alg":"HS256"}
eyJ1c2VyaWQiOiI1KioqKioqKioqKioqKioqKioqKioqKioiLCJsb2dpbmlkIjoiIiwidiI6MSwiaWF0IjoxNTQwNjc3NzMzLCJleHAiOjE1NDg0NTM3MjJ9
  => {"userid":"5**********************","loginid":"","v":1,"iat":1540677733,"exp":1548453722}
6xYHIyfK-RWisdwmNzp15U6ef-XnMnoWwXKbLYeX-Y7
  => HMACSHA256 signature
```

JSON Web Token!

We can query data!

- Vehicle(s) bound to account:

```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..."}' https://a
pp-api.niu.com/motoinfo/list
{"data":[{"sn":"NAS*************","specialEdition":"","vehicleColorImg":"https://app-api.niucache.com/static/app-api/static/prod
uct/default/engineering@2x_95b256a.png","vehicleLogoImg":"","vehicleTypeId":"N-seriesS2","indexHeaderBg":"https://app-api.niucac
he.com/static/app-api/static/pic/v3/na/headerBg/pic_background_N1s_grey@3x_d58acca.png","scootorImg":"https://app-api.niucache.c
om/static/app-api/static/pic/v3/na/productImg/pic_EU_moto_NA_grey_matte@3x_f2e4853.png","batteryInfoBg":"https://app-api.niucach
e.com/static/app-api/static/pic/v3/na/batteryBg/pic_background_N1s_grey@3x_5841b73.png","myPageHeaderBg":"https://app-api.niucac
he.com/static/app-api/static/pic/mytitlebackground/n1/bg_my_tittle_n1_@2_d8a5504.png","listScooterImg":"https://app-api.niucache
.com/static/app-api/static/pic/v3/na/listScooterImg/pic_moto_NA_grey_matte@3x_917b290.png","name":"Blitzdings","frameNo":"R1NB**
***********","engineNo":"RBNFF***********","isSelected":true,"isMaster":true,"bindNum":1,"renovated":false,"bindDate":1540649740
000,"isShow":true,"gpsTimestamp":1540688956124,"infoTimestamp":1540688956124,"productType":"native","process":"","brand":"","isD
oubleBattery":false,"features":[{"featureName":"gpsSwitch","isSupport":false,"switch_status":""}],"type":"N-series  Grey (Matte)
"}],"desc":"成功","trace":"成功","status":0}
```

- Vehicle position (requires SN):

```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...","sn":"NAS**
***********"}' https://app-api.niu.com/motoinfo/currentpos
{"data":{"lng":9.818106,"lat":53.47714,"timestamp":1541024153591,"gps":4,"gpsPrecision":0},"desc":"成功","trace":"Sucess!","stat
us":0}
```

… and some more …

- Battery information:

```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...","sn":"NAS**
**********"}' https://app-api.niu.com/motoinfo/batteryinfo/v2
{"data":{"totalPoint":504,"chargingInterval":0,"batteryCharging":80,"smallBattery":100,"isConnected":true,"status":2,"isCharging
":0,"showDetail":true,"estimatedMileage":45,"avgEnergyConsumed":11,"energyConsumedTody":0,"fullEnergeES":56,"onceMileage":0,"tem
perature":9,"chargedTimes":10,"items":[{"x":0,"y":64,"z":1},{"x":1,"y":64,"z":1},{"x":2,"y":64,"z":1},{"x":3,"y":64,"z":1},{"x":
4,"y":64,"z":1},{"x":5,"y":64,"z":1},{"x":6,"y":64,"z":1},{"x":7,"y":64,"z":1},{"x":8,"y":64,"z":1},{"x":9,"y":64,"z":1},{"x":10
,"y":64,"z":1},{"x":11,"y":64,"z":1},{"x":12,"y":64,"z":1},{"x":13,"y":64,"z":1},{"x":14,"y":64,"z":1},{"x":15,"y":64,"z":1},{"x
":16,"y":64,"z":1},{"x":17,"y":64,"z":1},{"x":18,"y":64,"z":1},{"x":19,"y":64,"z":1},{"x":20,"y":64,"z":1},{"x":21,"y":64,"z":1}
,{"x":22,"y":64,"z":1},{"x":23,"y":64,"z":1},{"x":24,"y":64,"z":1},{"x":25,"y":64,"z":1},{"x":26,"y":64,"z":1},{"x":27,"y":63,"z
":1},{"x":28,"y":63,"z":0},{"x":29,"y":63,"z":0},{"x":30,"y":63,"z":0},{"x":31,"y":63,"z":0},{"x":32,"y":63,"z":0},{"x":33,"y":6
3,"z":0},{"x":34,"y":63,"z":0},{"x":35,"y":63,"z":0},{"x":36,"y":63,"z":0},{"x":37,"y":63,"z":0},{"x":38,"y":63,"z":0},{"x":39,"
...
```

- Firmware information:

```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...","sn":"NAS**
**********"}' https://app-api.niu.com/motorota/getfirmwareversion
{"data":{"needUpdate":true,"otaDescribe":"<p class=\"p1\">A new function has been added to allow vehicle owners to change the st
atus of the GPS sensor on the scooter.</p>","nowVersion":"TRA01C07","version":"TRA01C10","hardVersion":"V2.0","ss_protocol_ver":
2,"isSupportUpdate":true,"byteSize":"42384","date":1526885222572},"desc":"成功","trace":"","status":0}
```

Let’s rename the scooter!

- Change vehicle name:

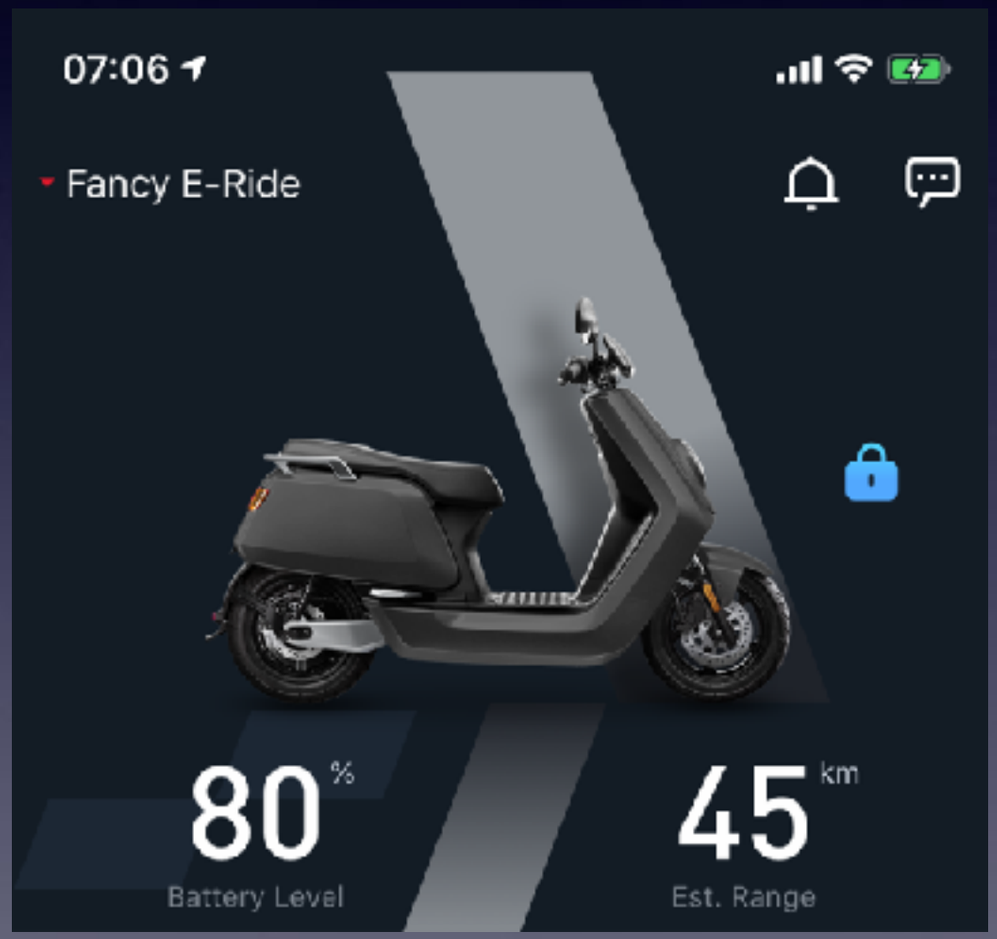
```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...","sn":"NAS**
**********"}' https://app-api.niu.com/motoinfo/setsnname
{"data":"","desc":"车辆名称不能为空","trace":"SnName cannot be empty","status":1305}
```

- Whoops! Nice, web API speaks Chinese and English!
  "车辆名称不能为空" => "Vehicle name cannot be empty"

- Let’s try again:

```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...","sn":"NAS**
**********","name":"Fancy E-Ride"}' https://app-api.niu.com/motoinfo/setsnname
{"data":"","desc":"成功","trace":"Success","status":0}
```

Let’s rename the scooter!



So what else can we do?

- Web API requires Authentication ✅
- Uses HTTPS ✅
- No certificate pinning ❌
- Vehicle S/N bound to account, can't be added by default, owner confirmation required ✅
- Some API calls even require confirmation by account owner by SMS or Email, e.g. ownership transfer ✅
- Attacker could MITM the connection, but bad stuff can't easily be done, bind permissions just require a token though 🙃

GSM/GPRS Connectivity

GSM/GPRS Connectivity

- Scooter comes with installed Prepaid SIM-Card (installed by Importer / KSR Group in Europe)
- Always connected (if there is network…)
- Scooter has a separate ECU battery, that lasts for about 3-4 days if main battery is unplugged
- Gives GPS and vehicle information without main battery

Let's hack that GSM already!

- OK What do we need?
  - Something that can modulate GSM frequencies
  - Something that acts as a GSM base station

GSM Hacking Equipment

- While certainly not the best, this equipments works:
  - bladeRF x40 + GSM Antennas
  - Raspberry Pi 3
  - YateBTS base station software
  - Power!

So how to set this up?

- My former co-worker Simone Margaritelli (@evilsocket) tried this before: https://www.evilsocket.net/2016/03/31/how-to-build-your-own-rogue-gsm-bts-for-fun-and-profit/

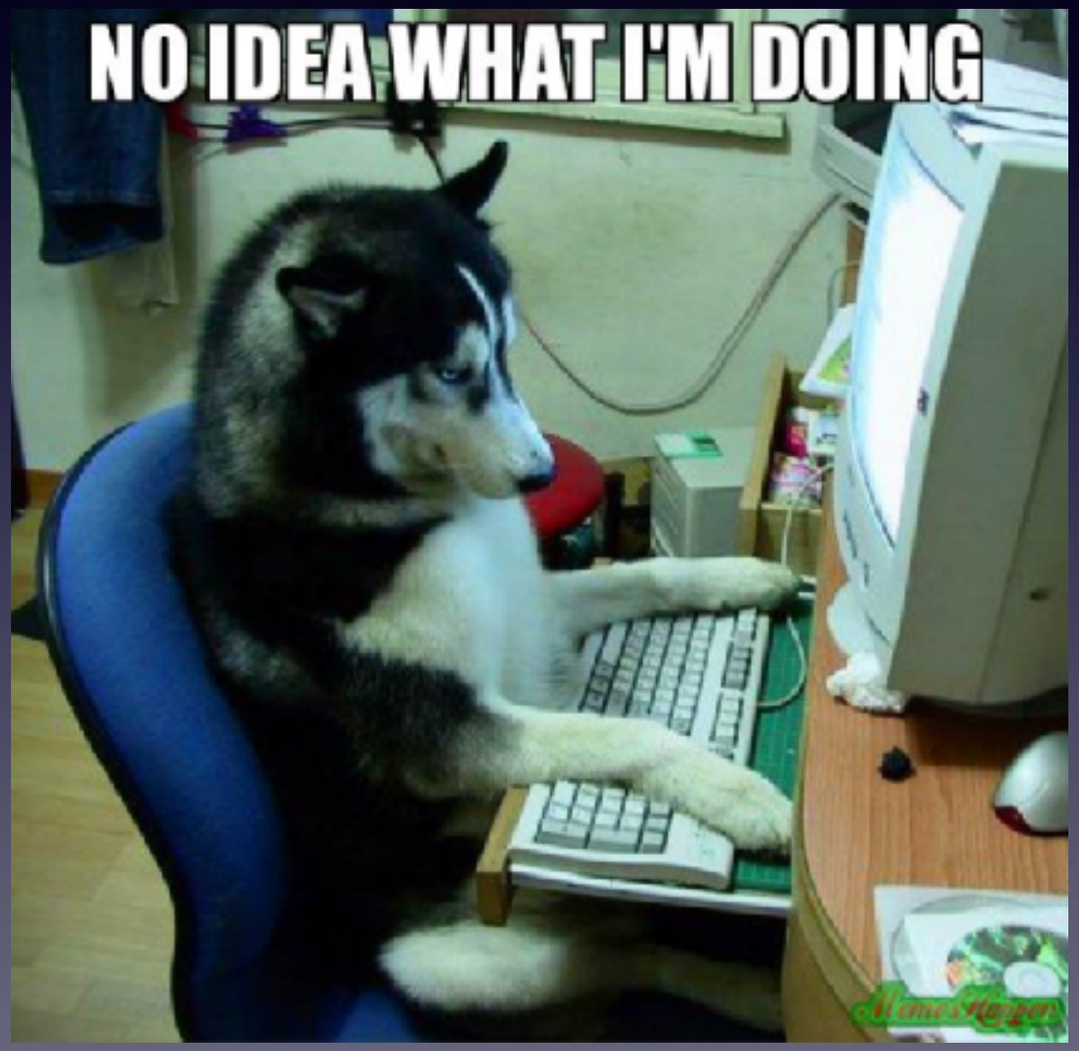
- However, he removed the version requirements which are really important for this to work.

- This blog article has all the information though: https://blog.strcpy.info/2016/04/21/building-a-portable-gsm-bts-using-bladerf-raspberry-and-yatebts-the-definitive-guide/

Let’s try it?! Not so fast.

- If you want a GSM device connect to your BTS, you need to simulate the right network

- Germany has 3 PLMNs:
  - Telekom (26201)
  - Vodafone (26202)
  - and Telefónica (26203)



Also, power…

- The integrated USB port rates 1 Amp only. This isn't enough to properly power the Raspberry Pi AND the bladeRF at the same time
- Strong battery pack or power supply via mains needed

Sounds easy, right?

- Doing a quick research, it showed that the importer said in a press release that they partnered with Vodafone
- So let's set this up to simulate Vodafone.de !
- Also, make sure to select a correct frequency in the right band (Vodafone uses GSM900 and GSM 1800)

YateBTS configuration

yate BTS NIB

Subscribers | BTS Configuration | Call Logs | Outgoing

GSM | GPRS | Control | Transceiver | Tapping | Test | YBTS

GSM | GSM Advanced

Set parameters values for section [gsm] to be written in ybts.conf file.

Radio.Band	EGSM900 ?
Radio.C0	#10: 937 MHz downlink / 89 ?
Identity.MCC	262 ?
Identity.MNC	02 ?
Identity.LAC	1000 ?
Identity.CI	10 ?
Identity.BSIC.BCC	2 ?
Identity.BSIC.NCC	0 ?
Identity.ShortName	dope ?
Radio.PowerManager.MaxAttenDB	35 ?
Radio.PowerManager.MinAttenDB	35 ?

Submit Reset

Section [gsm] controls basic GSM operation. You MUST set and review all parameters here before starting the BTS!

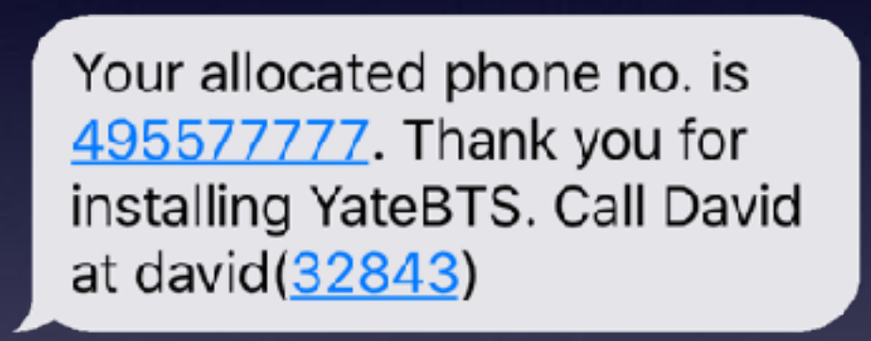
Note! To disable nib mode and enable roaming mode see Javascript Roaming

Now, wait…

- You can wait for a long time…
- Especially if you have a BTS near your home 🙃
- If a nearby BTS has a strong signal the Scooter won't connect
- But my phone always has bad network at home so this must work somehow…

Then suddenly…

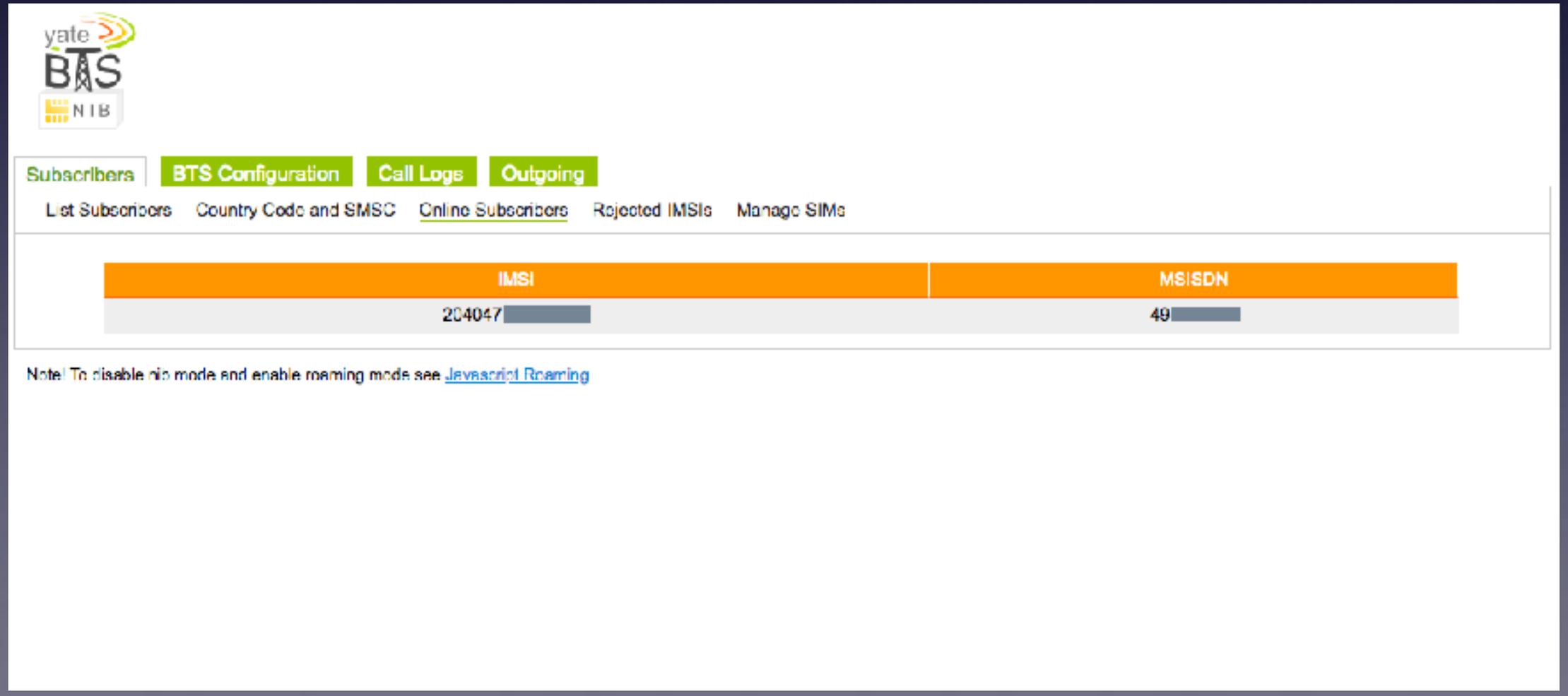
- My phone - which also uses Vodafone - receives text message



- Turns out the BTS actually works!

OK, let's wait longer…

- I was already thinking about other solutions when I suddenly realized that the BTS showed a new subscriber!



Gotcha!

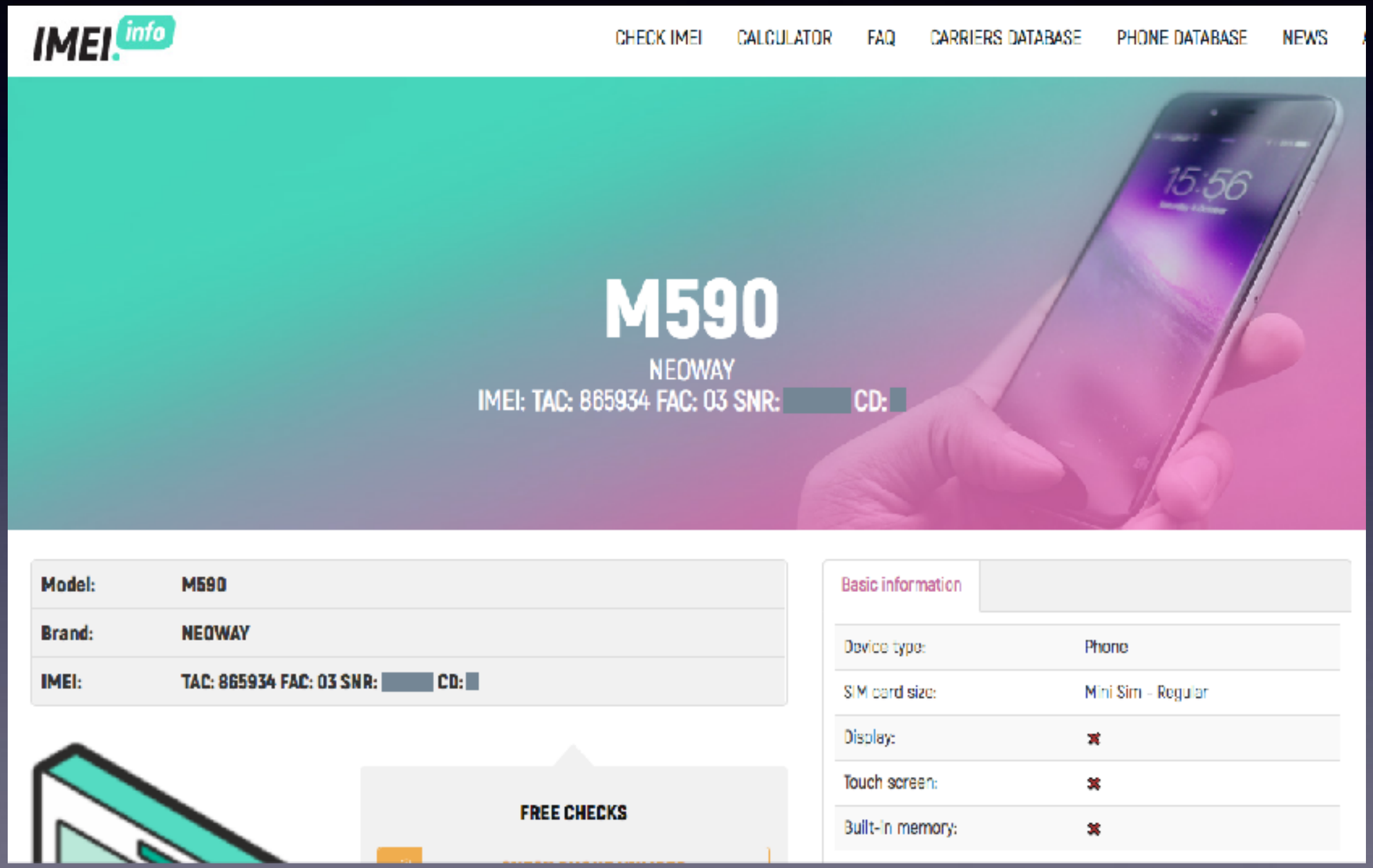
- IMSI shows prefix of 20404 - Vodafone Netherlands
- This SIM is actually Roaming! 🙃
- Let's see what else we can find out?

YateBTS verbose log output:

```
2018-10-28_22:44:11.965374 <ybts-signalling:INFO> Received [0x54c0f8]
-----
Primitive: L3Message
Info: 0
Connection: 1

<MM>
  <SkipIndicator>0</SkipIndicator>
  <NSD>1</NSD>
  <Message type="IdentityResponse">
    <MobileIdentity>
      <IMEI>86593403[redacted]</IMEI>
    </MobileIdentity>
  </Message>
</MM>
-----
2018-10-28_22:44:11.966334 <nib:INFO> Got user.register for imsi='204047[redacted]', tmsi=''
2018-10-28_22:44:11.967185 <nib:INFO> Allocated random number
2018-10-28_22:44:11.973973 <nib:INFO> Registered imsi 204047[redacted] with number 49[redacted]
2018-10-28_22:44:11.974673 <ybts-signalling:INFO> Sending [0x54c0f8]
```

Let’s check out the IMEI



OK so what next?

- We want to MITM the connection between Scooter and remote server
- YateBTS supports GPRS routing
  => Remember to enable IP forwarding and IP masquerading on the Raspberry Pi!
- Let's ask YateBTS' SGSN (Serving GPRS Support Node)

```
raspi3~ $ telnet localhost 5038
YATE 5.5.1-devel1 r (http://YATE.null.ro) ready on raspi3.
mbts sgsn list
 GMM Context: imsi=204047[redacted] ptmsi=0xc5001 tlli=0xc00c5001 state=GmmRegistrationPending age=139 idle=106 MS#1,TLLI=c00c500
1,990af0f6 IPs=none
```

- It doesn't want to connect through GPRS 😭

Then, I lost the connection…

- The Scooter disconnected. I waited and waited, but it didn't want to reconnect anymore…
- I had to come up with an idea to make it connect just to my BTS
- I tried setting the MCC and MNC to 20404, but it didn't want to connect
- I tried restarting YateBTS, but nothing worked

Ideas, I need ideas…

- Maybe it connects via 3G or even LTE? I was skeptical but then also I didn't know…
- Too bad I didn't by that frequency jammer last time I was in Shenzhen, I knew I would need it!
- Let's build a faraday cage?

Come on…

- Need to find a way it can't find a real BTS to connect to
- At my son's school there's really bad reception, let's go there…
- Still there seemed to be too much signal strength 😤
- Also, the battery pack I had, and also my MacBook couldn't properly power the bladeRF…

Then, I had an idea

- I remembered there's a parking garage nearby,
  A DARK AND SHADY PLACE !

- This must work! If there is no BTS it *JUST HAS* to connect to mine, right?

- Only problem was power…
  But I have a power converter in my car so that should do it

Into the Darkness…

- So I entered the garage and the scooter actually lost signal — PERFECT!
- I set up the BTS and everything, and waited…
- … and waited …
- I couldn’t believe it. It didn’t want to connect even though I am the only reachable BTS
- But somehow my phone also didn’t want to connect, not sure what was wrong… maybe interference? maybe the smell?

…there was another problem

- Even if it would connect to the BTS, it wouldn't be able to connect to the internet (via YateBTS' SGSN)
- Even my phone didn't have a signal so I couldn't use my hotspot
- I was disappointed and out of ideas, and went home
- I was about to give on on this, actually 😭

Let’s give it another try

- I set up my BTS at home again, because I said, hey it connected once maybe it connects again, what do I have to lose?
- But it didn’t want to connect. For an entire day, nothing happened. The real BTS was still too strong…
- I unscrewed the front panel of my scooter to check where the GSM module sits. It is in the upper front.
- But it has a sticker WARRANTY VOID IF BROKEN so I didn’t really want to mess around with that…

Making the signal weaker?

- Aluminum foil!
  - Didn't help, GPS signal lost a few bars though
- I re-parked my scooter so that my car would be between it and the BTS
  - Still no real change…


TWO HOURS LATER

Then…

- Suddenly, activity in the console where YateBTS was running 😳
- First I thought it's probably my phone again but…
- IT ACTUALLY CONNECTED

Wait, let's check the SGSN

```
raspi3~ $ telnet localhost 5038
YATE 5.5.1-devel1 r (http://YATE.null.ro) ready on raspi3.
mbts sgsn list
 GMM Context: imsi=204047[redacted] ptmsi=0xc5001 tlli=0xc00c5001 state=GmmRegisteredNormal age=539 idle=326 MS#1,TLLI=c00c5001,9
90af0f6 IPs=192.168.99.1
```

- It was connected through the SGSN!
- Let's dump some packets!
- Uh wait. How do we even do that? Did I enable GSM/GPRS tapping in YateBTS?
- I didn't but…

Phew…

- Luckily, YateBTS creates a TUN device "sgsntun"
- So on the Raspberry Pi I can now do:
  `tcpdump -i sgsntun -n -v -w dump.pcap`
- Packet counter increased slowly, every few minutes
- With ignition on, it sends packets every few seconds
- I copied the dump.pcap to my computer and ran it through Wireshark

The vehicle gateway!

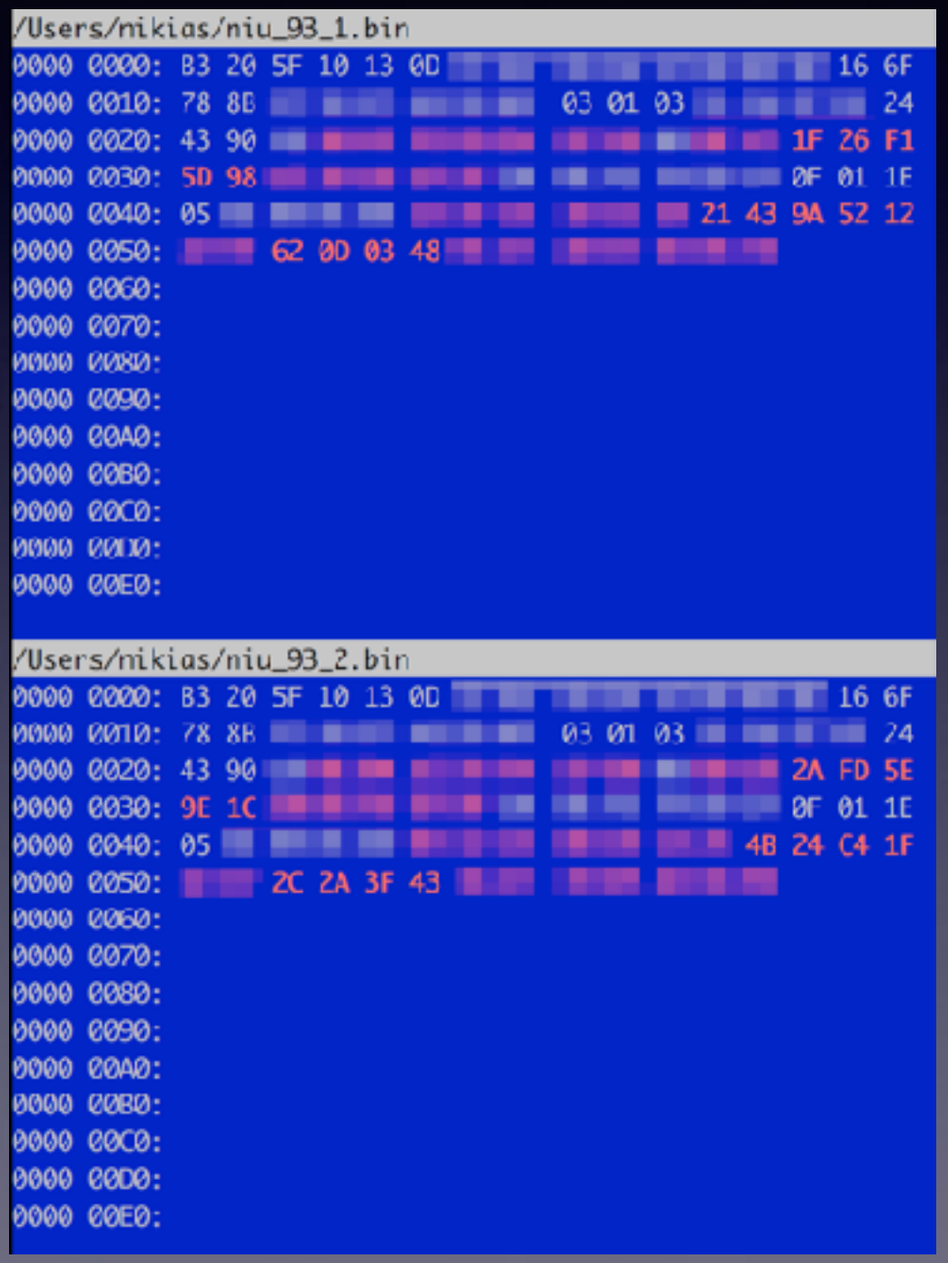
Let's have a look at what we captured:

Source	Destination	Protocol	Length	Info
192.168.99.1	1.1.1.1	DNS	57	Standard query 0x0000 A ecu.niu.com
1.1.1.1	192.168.99.1	DNS	73	Standard query response 0x0000 A ecu.niu.com A 52.58.219.193
192.168.99.1	52.58.219.193	UDP	138	57991 → 8888 Len=110
192.168.178.49	192.168.99.1	ICMP	84	Echo (ping) request id=0x0e6b, seq=1/256, ttl=64 (no respons
192.168.178.49	192.168.99.1	ICMP	84	Echo (ping) request id=0x0e6b, seq=2/512, ttl=64 (no respons
192.168.99.1	52.58.219.193	UDP	121	57991 → 8888 Len=93
192.168.99.1	52.58.219.193	UDP	138	57991 → 8888 Len=110
192.168.99.1	52.58.219.193	UDP	121	57991 → 8888 Len=93

- Resolves ecu.niu.com via DNS
- Sends UDP packets to ecu.niu.com on port 8888
- (That ICMP is my attempt to ping the scooter)

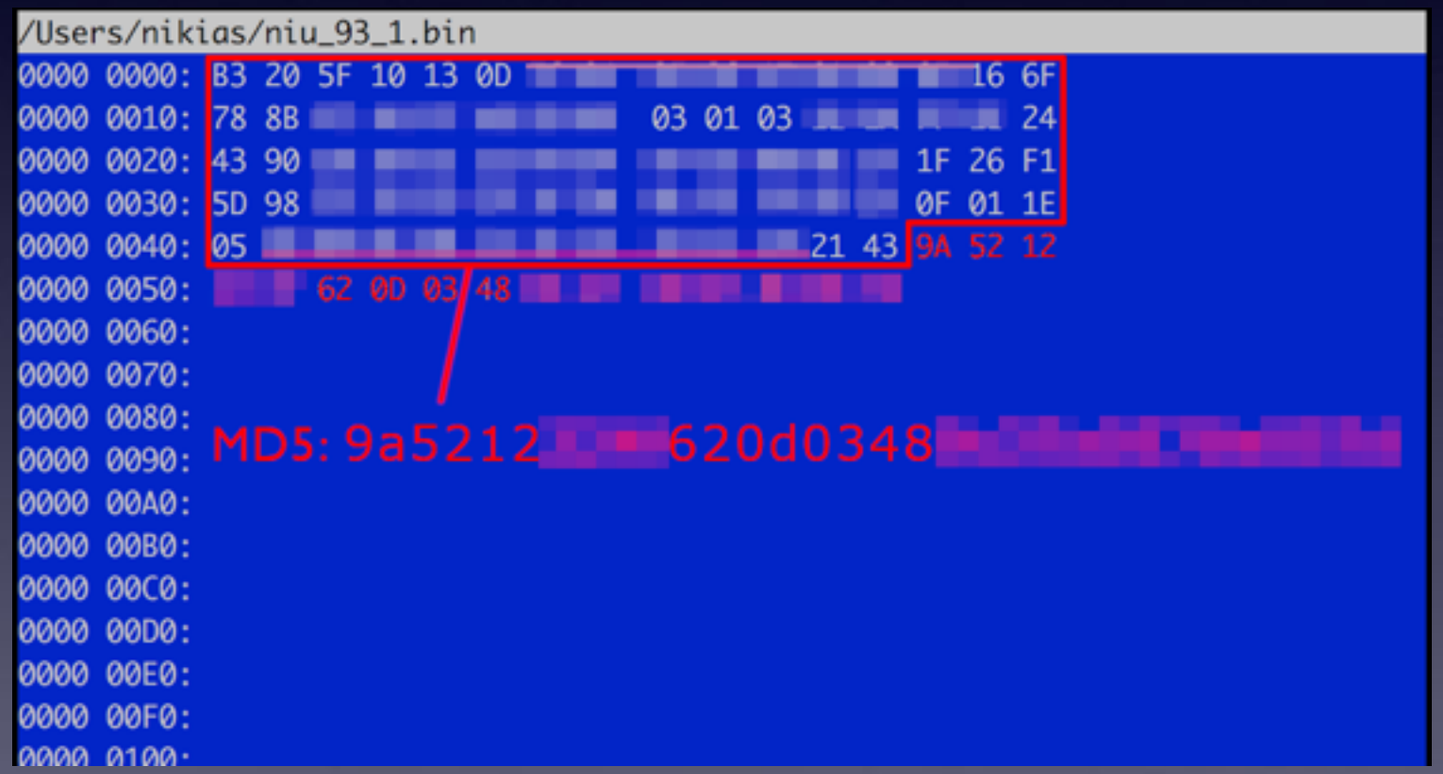
The packets

- Binary packet format
  Seriously, I was expecting JSON!
- Let's try to figure something out by looking at consecutive packets
- Shows some common patterns but also large parts that change
- Especially last few ~20 bytes
- Checksum? SHA1?



Packet Checksum

- Turns out to be MD5!



Packet format?

- First two (?) bytes seem to define the type of the packet
- Can’t really figure out a length field or anything obvious
- It needs to contain vehicle identification and GPS coordinates

Packet format?

- Seems somehow encoded. None of the Vehicle SN, or frame number or engine number seem to match in any way.
- Still it must have some kind of identification, otherwise it wouldn't know which scooter sent the data.
- Even though we don't understand the packet format completely we know that it has a checksum

What can we do with this?

- We can modify a packet, and apply the correct checksum and send it to ecu.niu.com 8888
- In the hopes of supplying different GPS coordinates I tried, but no reaction in the app…
- Research continues… (happy if someone has ideas!)

Can’t we do something?

- Maybe we can replay packets?
- Let’s use a simple python script that just reads a file and sends it to ecu.niu.com port 8888
- I could submit a slightly different position from a few minutes ago and it showed up in the app 🤘
- Let’s think about this. Meanwhile, let’s look at something else…

OTA Firmware update?

- Yes, the Niu can be updated over the air! Isn't that awesome?
- Since we can now dump the traffic, let's do this. What could possibly go wrong when it goes through our BTS?

Triggering the update

- To trigger an update, the Web API has this: POST to https://app-api.niu.com/motorota/updatemotor with SN (and token of course)

- To make the scooter start the update you have to turn the ignition off and on again, and then it shows progress:

Now be patient…

- The app says it will take about 10 minutes
- From the API we actually know the update size:

```
$ curl -H "Content-Type: application/json" --request POST --data '{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...","sn":"NAS**
**********"}' https://app-api.niu.com/motorota/getfirmwareversion
{"data":{"needUpdate":true,"otaDescribe":"<p class=\"p1\">A new function has been added to allow vehicle owners to change the st
atus of the GPS sensor on the scooter.</p>","nowVersion":"TRA01C07","version":"TRA01C10","hardVersion":"V2.0","ss_protocol_ver":
2,"isSupportUpdate":true,"byteSize":"42384","date":1526885222572},"desc":"成功","trace":"","status":0}
```

- So while we wait, let's take a look at the traffic…

Start of OTA traffic

Source	Destination	Protocol	Length	Info
192.168.99.1	52.58.219.193	UDP	121	57991 → 8888 Len=93
52.58.219.193	192.168.99.1	UDP	134	8888 → 57991 Len=106
52.58.219.193	192.168.99.1	UDP	134	8888 → 57991 Len=106
52.58.219.193	192.168.99.1	UDP	134	8888 → 57991 Len=106
192.168.99.1	1.1.1.1	DNS	63	Standard query 0x0001 A erom.niucache.com
1.1.1.1	192.168.99.1	DNS	79	Standard query response 0x0001 A erom.niucache.com A 60.…
192.168.99.1	60.205.12.173	TCP	64	58304 → 80 [SYN] Seq=0 Win=13600 Len=0 MSS=1360 WS=1 SAC…
60.205.12.173	192.168.99.1	TCP	52	80 → 58304 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=14…
60.205.12.173	192.168.99.1	TCP	52	[TCP Retransmission] 80 → 58304 [SYN, ACK] Seq=0 Ack=1 W…
192.168.99.1	60.205.12.173	TCP	64	[TCP Spurious Retransmission] 58304 → 80 [SYN] Seq=0 Win…
60.205.12.173	192.168.99.1	TCP	52	[TCP Previous segment not captured] [TCP Port numbers re…
60.205.12.173	192.168.99.1	TCP	52	[TCP Retransmission] [TCP Port numbers reused] 80 → 5830…
192.168.99.1	1.1.1.1	DNS	63	Standard query 0x0002 A erom.niucache.com
1.1.1.1	192.168.99.1	DNS	79	Standard query response 0x0002 A erom.niucache.com A 60.…
192.168.99.1	60.205.12.173	TCP	40	58304 → 80 [RST] Seq=1 Win=0 Len=0
192.168.99.1	60.205.12.173	TCP	40	58304 → 80 [RST] Seq=1 Win=0 Len=0
192.168.99.1	60.205.12.173	TCP	40	58304 → 80 [RST] Seq=1 Win=0 Len=0
192.168.99.1	60.205.12.173	TCP	40	58304 → 80 [RST] Seq=1 Win=0 Len=0
192.168.99.1	60.205.12.173	TCP	64	64200 → 80 [SYN] Seq=0 Win=13600 Len=0 MSS=1360 WS=1 SAC…
60.205.12.173	192.168.99.1	TCP	52	80 → 64200 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=14…
192.168.99.1	60.205.12.173	TCP	40	64200 → 80 [ACK] Seq=1 Ack=1 Win=13600 Len=0
192.168.99.1	60.205.12.173	HTTP	154	GET /rom/N1SP/V1.0/TRA01C10ECP001.bin?sn=NAS
60.205.12.173	192.168.99.1	TCP	40	80 → 64200 [ACK] Seq=1 Ack=115 Win=14720 Len=0
60.205.12.173	192.168.99.1	HTTP	1216	HTTP/1.1 200 OK (application/octet-stream)
60.205.12.173	192.168.99.1	TCP	40	80 → 64200 [FIN, ACK] Seq=1177 Ack=115 Win=14720 Len=0

OTA traffic continued

Source	Destination	Protocol	Length	Info
192.168.99.1	1.1.1.1	DNS	63	Standard query 0x0003 A erom.niucache.com
1.1.1.1	192.168.99.1	DNS	79	Standard query response 0x0003 A erom.niucache.com A 60.…
192.168.99.1	1.1.1.1	DNS	63	Standard query 0x0004 A erom.niucache.com
1.1.1.1	192.168.99.1	DNS	79	Standard query response 0x0004 A erom.niucache.com A 60.…
192.168.99.1	1.1.1.1	DNS	63	Standard query 0x0004 A erom.niucache.com
1.1.1.1	192.168.99.1	DNS	79	Standard query response 0x0004 A erom.niucache.com A 60.…
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
192.168.99.1	60.205.12.173	TCP	64	57548 → 80 [SYN] Seq=0 Win=13600 Len=0 MSS=1360 WS=1 SAC…
192.168.99.1	1.1.1.1	ICMP	64	Destination unreachable (Port unreachable)
60.205.12.173	192.168.99.1	TCP	52	80 → 57548 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=14…
192.168.99.1	60.205.12.173	TCP	40	57548 → 80 [ACK] Seq=1 Ack=1 Win=13600 Len=0
192.168.99.1	60.205.12.173	HTTP	154	GET /rom/N1SP/V1.0/TRA01C10ECP002.bin?sn=NAS
60.205.12.173	192.168.99.1	TCP	40	80 → 57548 [ACK] Seq=1 Ack=115 Win=14720 Len=0
60.205.12.173	192.168.99.1	HTTP	1216	HTTP/1.1 200 OK (application/octet-stream)
60.205.12.173	192.168.99.1	TCP	40	80 → 57548 [FIN, ACK] Seq=1177 Ack=115 Win=14720 Len=0
192.168.99.1	60.205.12.173	TCP	40	57548 → 80 [ACK] Seq=115 Ack=1178 Win=13600 Len=0
192.168.99.1	60.205.12.173	TCP	40	57548 → 80 [FIN, ACK] Seq=115 Ack=1178 Win=13600 Len=0
60.205.12.173	192.168.99.1	TCP	40	80 → 57548 [ACK] Seq=1178 Ack=116 Win=14720 Len=0

OTA Download

- Download via plain HTTP in 1kB chunks from:
  `http://erom.niucache.com/rom/N1SP/V1.0/TRA01C10ECP001.bin?sn=NASxxxxxxxxxxxx`
  `http://erom.niucache.com/rom/N1SP/V1.0/TRA01C10ECP002.bin?sn=NASxxxxxxxxxxxx`
  …
- Vehicle SN as query parameter, however turns out you can pass whatever you want 😜
- New connection for every chunk
- In my dump I could see chunks being re-transferred, guess my BTS hardware isn't the most reliable 😳

OTA Download

- To download the firmware you basically need to know the size and then you can do something like (bash):

```
$ for I in {1..42}; do curl http://
erom.niucache.com/rom/N1SP/V1.0/
TRA01C10ECP`printf %03d $I`.bin?sn=blah >
TRA01C10ECP`printf %03d $I`; done

$ cat TRA01C10ECP0* > firmwareTRA01C10ECP.bin
```

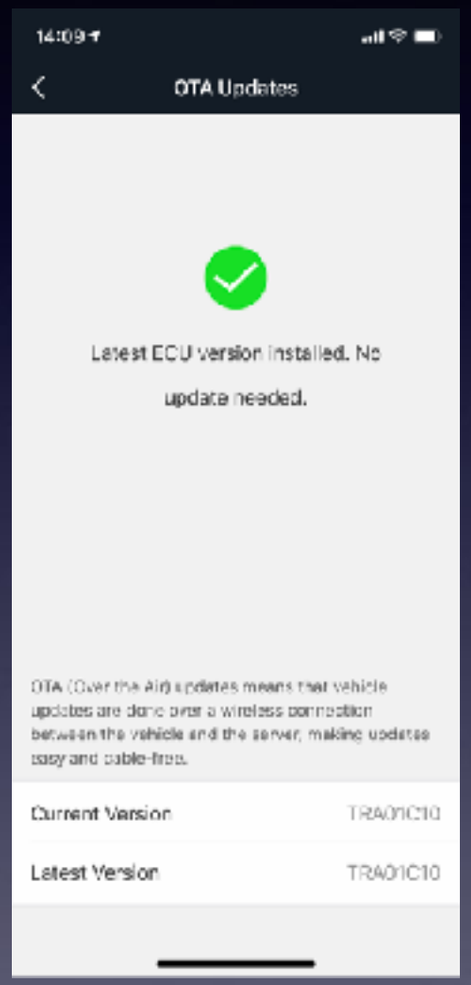
OTA Firmware

- Seems encrypted. No obvious header, high entropy, no strings… ¯_(ツ)_/¯

```
firmwareTRA01C10ECP.bin
0000 0000: 4F F2 4D 00 16 D3 40 00  D8 7D 27 25 2A 31 42 7D  O.M...@. .}'%*1B}
0000 0010: 88 D8 5D 11 9E CC 5B 2D  8E 02 C9 89 E6 54 5B 2D  ..]...[- .....T[-
0000 0020: 46 38 67 3D 33 41 83 05  03 01 03 05 0B 11 23 05  F8g=3A.. ......#.
0000 0030: 03 01 03 05 04 C3 46 85  CE 55 72 6D 93 01 03 05  ......F. .Urm....
0000 0040: D2 95 C6 7D 0E A7 DC 89  10 3B 22 0D 18 3B 32 1D  ...}.... .;"..;2.
0000 0050: 20 0B 32 1D 28 1B 42 0D  F0 BB 42 0D F8 BB 52 FD   .2.(.B. ..B...R.
0000 0060: A0 2B 32 3D 36 31 2E 55  10 FB A2 4D 18 FB B2 5D  .+2=61.U ...M...]
0000 0070: 20 0B F2 DD 28 1B 02 0D  DE DC 0B 0D F8 BB 52 FD   ...(... ......R.
0000 0080: DE FB D0 09 10 3B 22 0D  10 3B 22 0D 8E B3 38 09  .....;". .;"...8.
0000 0090: 20 0B 32 1D 28 1B 42 8D  F0 3B 42 8D F8 3B 52 7D   .2.(.B. .;B..;R}
0000 00A0: 20 2B 32 3D 28 3B 22 4D  90 FB 22 4D 98 FB 32 5D   +2=(;"M ..."M..2]
0000 00B0: A0 0B F2 5D 8A E5 36 65  B2 F0 4B 8D 8A 37 98 29  ...]..6e ..K..7.)
0000 00C0: 26 E6 4D 91 10 3B 22 0D  10 3B 22 0D 18 3B 32 1D  &.M..;". .;"..;2.
0000 00D0: 20 0B 32 1D 28 1B 42 0D  F0 BB 42 0D EA 85 26 55   .2.(.B. ..B...&U
0000 00E0: 72 0D 36 3D 8E 9C 23 4D  A0 A0 0D 11 8E D6 55 19  r.6=..#M ......U.
0000 00F0: 20 0B F2 DD E4 28 D5 DD  D3 B1 2D 43 E3 39 2B 50   ....(.. ..-C.9+P
0000 0100: 23 C5 A2 FD AB 75 27 2D  25 4F CB C8 50 57 53 60  #....u'- %O..PWS`
0000 0110: 88 1A 1C 22 38 1A 1C 1A  18 1A 1C 22 48 7A 1C 1A  ..."8... ..."Hz..
0000 0120: 18 1A 1C 22 57 E3 94 09  26 0E E5 51 5B EB EC 4F  ..."W... &..Q[..O
0000 0130: 27 6D 57 D4 D0 CA ED C2  BA 6E 76 2D 47 AF 3B 4D  'mW..... .nv-G.;M
```

Meanwhile: Update finished?

- Almost there…



- App reported an error, saying to try again

- But the scooter seems fine. After closing the app it was actually shown as being up-to date.

OTA Risks?

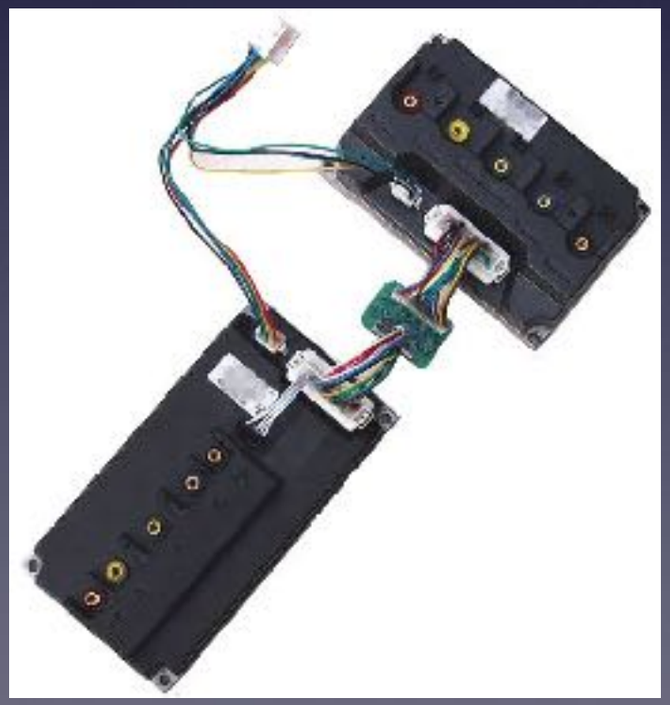
- The vehicle gateway sends update trigger packet(s) to the ECU
- In theory, the vendor could trigger an update at any time
- However if the ignition is on it won't start until you turn off the ignition and turn it back on

Firmware hackable?

- Possibly, but need to understand the firmware first
- Also the update trigger packet will probably contain information about the update package and size so the ECU knows what to download
- But… I want to make my Scooter faster!

Behold! There's a solution

- Source: `http://www.myniu.org/making-the-n1s-faster/`
- By adding a 2nd controller that drives the motor while the original controller talks to the system 😎



This is probably illegal in most countries. Don't do it.

China shopping list ++

So. Back to replaying…

- What could we actually replay to see if it works properly?
- Remember, the App has push notifications :)
- For some reason, the 'unusual movement' detection hasn’t been working for a while
- But every time you unplug the battery, the app shows a notification
- Let’s unplug the battery, dump the packet, and replay

DEMO TIME

Conclusions

- Overall, the vendor did a really good job!
- Pretty solid implementation, safety checks etc.
- It has some small issues, like missing certificate pinning, but that's minor
- (Most likely) Encrypted firmware
- Encrypted(?) packet format for GPRS connection though vulnerable to replaying

Thanks!

谢谢!